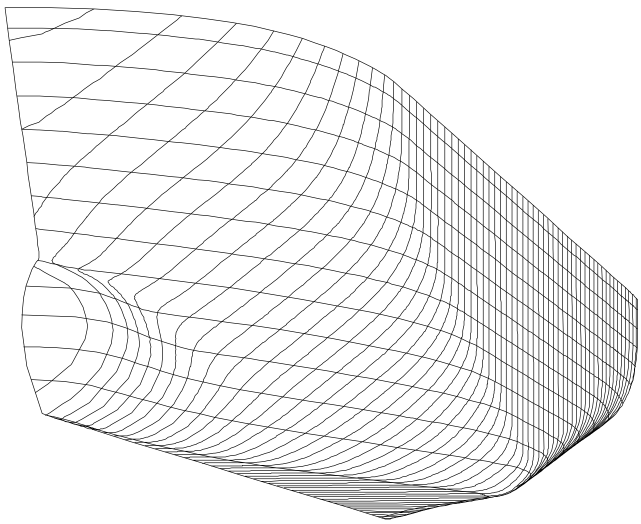
A Hull Surface Generation Technique Based on a Form Topology and Geometric Constraint Approach



A PhD Thesis by Marcus Bole

Submitted to the University of Strathclyde, July 2002.

Acknowledgements

A PhD was a challenge I had originally planned never to undertake. However, when the right idea came along I seemed like an opportunity not to miss. The following people receive my thanks and great appreciation for assisting me during this study, thanks…

The Catalysts

For the most part, these people are unaware of their contribution to the initial spark of an idea and the corresponding project. Even so, without the strange chain of events, ShipLINES would have probably been the only result. So here you are:

Prof. Dracos Vassalos	…for sending me to the Color Vision Meeting at HDW
Rolf Kjaer (Color Line)	…for allowing me to return for a second meeting at HDW
Simon Pollard (P&O Stena Line)	…Stena Invicta, a difficult upgrade project which instigated the vital inspiration
Asst. Prof. P. D. Kaklis	…for the words knowledgebase, topology and constraints, *in the right place*

The Advisor

Dr Suki Lee	It all started with YachtLINES, hopefully it is finished now

The Bosses – Safety at Sea Ltd.

Thanks for the flexibility, because it's been difficult to do a PhD with a full time job!

Prof. Dracos Vassalos
Dr Osman Turan
Dr Luca Letizia

My Parents

Thanks, it was a long time getting there!

Marcus Bole, July 2002.

Contents

Glossary of Acronyms and Terms

ASF – Aft Surface Flat

CB – Block Coefficient

CP – Prismatic Coefficient

CAD – Computer Aided Design

DWL – Design Waterline Length

FSF – Forward Surface Flat

GUI – Graphical User Interface

LCB – Longitudinal Centre of Buoyancy

LOA – Length Overall

OOP – Object Orientated Programming

PD – Parallel Deck

PDF – Parallel Deck Forward

PDA – Parallel Deck Aft

PMB – Parallel Middle Body

PMBF – Parallel Middle Body Forward

PMBA – Parallel Middle Body Aft

SIF – Surface Interface Framework

TSCHADE – Topological Shape Constrained Adaptive Hull Design Environment

1. SUMMARY

Modern computer aided hull design tools do not provide real *assistance* in the development of hull form surfaces and, as a result, there is a significant technology gap between manually manipulated definition and parametric generation approaches. This study develops a conceptual approach for a design tool that is able to address the entire hull surface development process from concept to detailed design, by adapting to the amount information that is available to the designer.

An extensive review of hull representation techniques, design methodologies and present design tools establishes that, while a great variety of techniques have been developed, the approach of defining a hull form using a surface representation driven by many control vertices has become standard. Consequently, in the manipulation of all but the most detailed features of the hull surface, the designer must manipulate many individual control vertices to effect change. In standardising to this approach, tool developers have neglected any techniques that allow rapid creation and modification at the concept design state, to establish the initial dimensions, and to be able to extend this definition on to more detailed development. Parametric hull generation techniques have been developed as the solution to this problem. However, these fixed formulations do not allow for a great range of flexibility in the number of different hull form shape that can be generated or in the variety of appendage features that can be included in the representation.

A hull form consists of many different shapes and features. By providing only parametric definition or detailed control vertices, modern tools significantly limit the number of ways a designer can control the characteristics of the surface. A tool developed to address the hull design process, from concept onward, would provide the designer with the ability to control the hull form in many different ways, each tailored to manipulating a feature in a certain way.

By using a hierarchical approach to develop an interface between the user and surface definition, a ‘space’ can be created to allow for the processing of definition data. Instead of providing the definition data in single block, several levels of definition can be used to build up the hull form shape from basic to more detailed data. The interface can selectively choose how to build up the definition by considering what information has been provided. Combining parametric and manually control geometric data is just one of the features that can be implemented in an approach that aims to control dimensions with numbers and shape with geometry.

By introducing an interface between the user and the surface representation, the definition structures controlled by the user no longer have to be limited by the characteristics of the surface technique. Consequently, a definition approach can be developed which is conducive to hull design. Furthermore, by using knowledge of the generic shape of a hull, the form topology, the interface can automatically generate and geometrically constrain the definition to ensure a valid hull form is produced, completing the design. As the design progresses, automatically generated definition can be replaced by user supplied data as the details of the surface become known. The process can just as easily be reversed to allow the hull form to be defined entirely from parametric information.

A pilot system, implemented using a single NURBS surface representation, is developed by making a detailed review of some of the techniques that can be used to control the shape of a hull form. The design of the interface and approaches for the development of compound (custom) transformations invoked by the change of parameters is discussed. Curves are identified as the most effective representation for controlling the shape of the hull form. Furthermore, a basic numerical technique for merging the shape of surfaces together, using sculpturing functions to form local appendage features is developed.

The evaluation of the concept highlights that the approach has the ability to make the hull design process easier because it speeds up the creation process, provides more clarity in the definition, allows the designer to control the surface in more ways while still being capable, if not more so, of developing the range of hull surfaces that present approaches allow. The pilot system demonstrates that even a relatively basic implementation can provide significant design advantages when compared to controlling a surface representation without the interface.

This work lays down the foundations for a more effective, functional and easy to use hull design environment that is capable of combining many different techniques and functions for creating a hull form and manipulating the surface representation.

2. INTRODUCTION

Over the last two decades, the ship design process has been required to produce better and more effective products. The demand has been driven by increasing customer specifications for more economic and viable designs and has resulted in a great deal of competitiveness between operators and designers. Furthermore, baseline design standards have been raised by society in many areas, such as safety and pollution control, in attempts to reduce the number of accidents and the burden on the world's environment.

In order to deal with these increasing demands, there has been a significant amount of development in tools, techniques and procedures that will give a particular design the "edge" over a competitor's products. The rapid development of new techniques and tools has been sustained by the ever increasing growth in the speed and capability of computing technology. Many tools have been introduced which give the designer the ability to optimise a design with respect to new rules or requirements. Techniques, such as performance simulation and CFD, allow optimisation to be carried out by taking advantage of the latest numerical approaches and product modelling tools allow the full geometric details of the design and all the related information to be stored and developed in one integrated environment. However, while design and analysis tools have been introduced into almost all areas of the discipline, the underlying approach taken by tools in aiding the design of the hull form representation has remained virtually unchanged for the last thirty years.

The hull surface represents the most complex component designed during the development of a vessel, not because it has a very intricate shape, but because it must safely support the range of systems that are required for the support and transportation of life and goods, in the hostile environment of the sea. Furthermore, many aspects of safety are dependent on hull performance factors, many of which are complex in nature. As the design of the hull surface has such far reaching consequences across the ship system, the tools used to design the component should reflect the importance of the task by allowing the designer to manipulate the form in ways that directly relate to the performance required. However, as the mathematical functions used to represent the hull surface all have practically the same approach to controlling shape, the functionality and operation of hull design tools has become very similar. Modern design tools require the user to generate and manipulate a large amount of simple data to create and develop the shape of the hull form resulting in the formation of bottlenecks in the design process.

Consequently, in trying to avoid these bottlenecks, designers are prevented from investigating solutions that may significantly improve the performance of the vessel.

As the shape of the hull surface is a principle factor in performance, the development of the hull form has always been the principle activity in the design of a vessel particularly during the early stages. By working with a representation of the hull surface shape, the designer can interactively modify and make analysis of performance until an optimal solution is found. Considering the length of time ships have existed, the ideal technique for representing a hull form has only been found relatively recently. It was not until the introduction of parametric surfaces, such as NURBS and Coon's patches in the 1970's that designers found a practical mathematical representation technique that could be modified a limitless amount of times and did not have any constraints on the variety of shapes that could be produced. As computing technology has become more affordable, these representations have found increasing use across the design and production of many products including ships. It is now taken for granted that the same hull form representation can be used for design and in all subsequent phases of the process of developing a ship through to production without requiring it to be recreated for input into different systems.

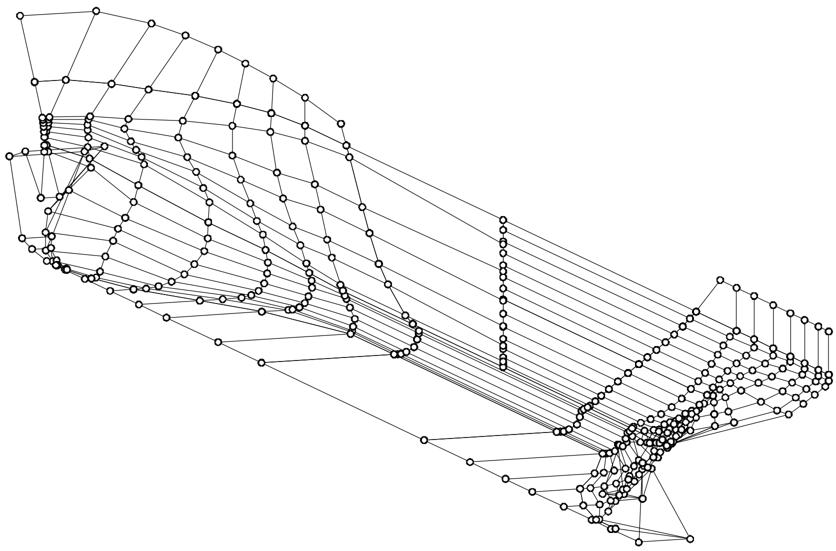


Figure 2.1, an example of a several NURBS control polygon vertex meshes combined together to form a hull surface definition.

Parametric surfaces, such as NURBS, use a regular mesh structure of control vertices, called the control polygon, to define the geometric shape, (Figure 2.1). In the case of the Coon's patch, which is defined using surface derivatives, a network of curves, again defined using control vertices, can be used to develop the shape of the hull surface using several patches combined together. The regular structure of the NURBS control polygon ensures that a significant number of control vertices are required to create an accurate representation of the surface. This can be reduced by using several NURBS surfaces of different complexity in combination to form a patch structure. In the case of Coon's patch representation, an irregular network of curves and patches can be used to handle different complexities of shape in the hull surface. Generally, around a hundred vertices are required to form a reasonably accurate and practical surface representation within the most effective hull design tools. Having this number of vertices gives the designer a high degree of flexibility to create a wide variety of shapes and all of the detailed features of a modern hull form can be modelled in the surface representation.

While these representations have improved the product development process as a whole, the activity of creating the representation and developing it into the final solution remains as difficult, if not more so, than the previous manually operated tools used to draw lines plans in two dimensions. In the initial stages of design, the level of flexibility provided by the high number of control vertices can inhibit development because the user must manipulate many vertices to accommodate the large changes in shapes that occur in this phase of the design process. However, as a surface representation of the hull form allows the designer to perform a wide range of analysis, the intricate development process has to be an acceptable part of design. Hence, the design tools enforce detailed development on the user before the initial design, perhaps even the concept, has been finalised. Moreover, changes to the design become prohibitive because so much time and labour is required to modify the definition, resulting in increased costs. Consequently, solutions that may significantly improve the performance of the ship may be missed.

The significant amount of direct manual manipulation required to develop a hull form has always been a major factor in the design of a vessel. However, ever since the first mathematical representation of a hull form has been introduced, the generation of a hull from numerical form parameters has been seen as a real possibility and has been investigated many times. Several approaches exist which are capable of developing an initial hull surface based on numerical parameters and some commercially available hull design tools offer facilities to generate a representation which can be subsequently modified using the manual approach.

While offering a potential solution for reducing the amount of manual manipulation required in developing parametric surface representations of hull forms, there is no evidence that these tools play any significant role in practical hull design in industry. The facility to change the dimensions of a hull form by modifying numeric parameters rather than manipulating tens of vertices is very attractive and this remains the motivation behind the continuing development of parametric generation techniques. However, parametric hull generation techniques are quite complex, increasingly so as more detailed features are included. The complexities of the mathematical functions within these techniques require the procedures to be hard-coded into the software, fixing flexibility. Consequently, each technique is only capable of representing a limited range of hull form shapes compared to the flexibility inherent in the manually defined surface. Some developers have tried to develop solutions which involve connecting numerical parameters with definition geometry or by including scriptable systems. However, designers rarely have the time, expertise or interest to use these features, with the chance that the system may not be able to handle future indefinable changes to the surface shape. Consequently, it is still more effective to define the hull form manually at present.

While developers have concentrated on the details of providing the best user interface to manipulate control vertices or the most flexible parametric hull generation technique, it has not been seriously challenged that neither solution adequately addresses the hull design process. Both design tools perform well in the tasks they are designed to carry out. However, neither tool allows the designer to form a basic hull surface from available information and develop the representation by performing analysis and by adding features as detailed design is arrived at, to create a shape suitable for production. In fact, present tools only address the ends of the process, and in the gap between, which represents the most important part of the development process, neither tool is very effective at responding to the type of changes the designer would be expected to make.

This study aims to address this gap by taking an open minded look, from the perspective of the designer and the developer, at the way the hull surface representation can be developed with the technology available today. As a significant amount of time has been invested in the development of tools using manual manipulation of surface definition and hull form generation from numerical form parameters over the years, it is very likely that both these techniques will feature as important approaches for an ideal hull design tool. As pointed out above, merging of both the manual manipulation and parametric approaches has been a goal for the developers of hull design tools for sometime. While this is something of a major consideration within the study, the priority is to identify a conceptual approach for a practical design tool that can adapt to the amount of available information and does not require the designer to manipulate many control vertices to achieve a simple characteristic change in the hull surface. To this end, the study aims to develop the template for a hull design tool that can maintain consistency between the characteristics of the hull form the designer is dealing with and how these characteristics are manipulated within the software environment.

In recent times, it has become unfashionable to introduce new hull form design techniques, perhaps because the modern integrated design approaches do not allow for these tools to be interfaced easily. However, the fact that there are still technological gaps in the design process illustrates that present tools are not providing the best solution. Because there has been a significant amount of research into methodologies for developing the hull form representation, this work has made an extensive review of existing and historic techniques, to identify potential solutions and approaches for a hull design tool which are not too dissimilar in operation from current design methodologies.

The thesis begins by making a brief review of the development of hull surface representation and form shape across history. The importance of the techniques and approaches used in ship design before it became dominated by the scientific approach should not be underestimated and many improvements were introduced by the innovative engineers working in the period. At the turn of the last century, the growing interest in science and the momentum of the industrial revolution gave rise to the development of numerical approaches for representing the hull form. An extensive review is performed of many of the techniques that were developed, from the early beginnings to those of the present day.

The hull design process has adapted to the capabilities of the design technology to ensure that products remain competitive. As other technology is being developed and being used in the design process, it is important to understand the changes that are taking place to ensure that hull design tools will continue to remain effective in the future. A review is made of current hull design methodologies to understand how the development process functions and how well the functionality provided by present tools compares. Subsequently, a review is made of the particular characteristic traits of commercially available design tools, identifying where these tools fail the design process and some of the approaches that can enhance the development of a hull surface.

Before embarking on the development of potential solutions, an exploration of B-Spline and NURBS technology is made. While these functions are used extensively across the CAD industry, users are largely shielded from the technology by software GUI's and implementation. Consequently, design tools may not be used very effectively, as the user is expected to provide the definition to create certain effects. This approach relies greatly on users' skill and experience of NURBS.

The study uses the many findings identified by the review to develop an approach that will combine present techniques together to produce a concept for an integrated hull surface development system that adapts to the design process. A hierarchical approach can be used to integrate presently incompatible definition and manipulation techniques into an interface to present hull representation functions. Furthermore, the formation of an interface between the user and the surface representation allows new methodologies for defining the hull form to be introduced that are not constrained by the definition structure used by the surface representation. The study develops an approach for reducing the amount of definition and detailed manipulation the designer must perform by utilising the basic topological information contained within shape of hull forms to create more effective methods for managing the definition than vertex level control alone. A pilot system is developed by considering, in detail, the structure and each of the components that go together to form an implementation of a tool using this approach.

The thesis concludes with an evaluation of the concept which compares the approach with present techniques and a discussion of the work with regards to performance, the wider implication and future development.

The appendices provide additional details of many of the areas covered in the work. There is an extensive review of individual commercially available hull development tools. Details of the parametric hull form generation tools that were developed by previous research into improving the hull surface design process are given. Extensive illustrations of the results and surface development processes produced by the pilot system are presented. The appendices also give details of some of the techniques and procedures used in the pilot system that were not directly relevant to the development of the approach.

3. AIM OF STUDY

This study aims to take a fresh look at the hull design problem and its associated tasks, to identify an approach that will make the process easier and more productive for the designer. There have been many previous attempts to produce *new* or even the *ultimate* design tool with few successes. While this study recognises this trend, it is clear that present tools are not as effective for producing the *Product* when compared to those used in other areas of engineering design. However, it is impossible to ignore the fact that many of these tools use successful processes developed over many years. Consequently, this study is focused on the search for a solution that involves the innovative integration of existing hull design techniques by maximising the use of knowledge and information inherent within the product itself, the hull form.

This aim will be achieved by considering the following objectives:

1. As the hull form representation technology forms the corner stone of any design tool, a detailed and, most importantly, practical review of the wide range of techniques that have been employed over the years is required to identify the processes involved, the advantages and fundamental limitations of each and to understand why there are only one or two representation techniques being used in modern hull design tools today.

2. To make a practical review of the methodologies being used to develop the hull form, establishing the level of functionality required from modern design tools.

3. To make a review of existing tools to identify where the designers requirements are not being met and to highlight any tools with innovative approaches which assist the designer achieve particular tasks.

4. Based on the thorough review of the processes and tools presently being used in the hull design process, a potential solution will be established by using a clear and objective approach to resolving present problems by identifying any factors that have been overlooked by present tool developers.

5. In resolving many of the individual problems faced in within hull design tools, a further step will be taken by devising the concept for an interface between the user and the hull surface representation technology that in addition to allowing the integration of present design methodologies, is capable of adapting to changing definition and information requirements as the design process progresses.

6. A pilot system will be developed to demonstrate the effectiveness of the approach.

4. EARLY DEVELOPMENT OF HULL DESIGN AND REPRESENTATION TECHNIQUES

Man's development of efficient watercraft was initially slow. It was not necessary for early shipwrights to be educated as the art could be learnt through experience and tradition. Learning from mistakes was a necessity. Hulls were sometimes sketched, but built by eye and experience. Development of designs was based around observation and improvements were made through a process of trial and error. The main construction material was wood. Wood can be a difficult material to work with and there are few ways to create a structurally sound hull. Methods of hull design and construction techniques first had to be developed together to allow durable ships to be built. Without robust ships, sea trade routes could not have been established.

Early designers tried to take as much of their inspiration from nature and vessels designed using philosophy of the "Cod's head and Mackerels Tail" became common place, Figure 4.1. Once the design and construction of hulls with wood had been mastered, the development of efficient, fast and long lasting hull forms could begin.

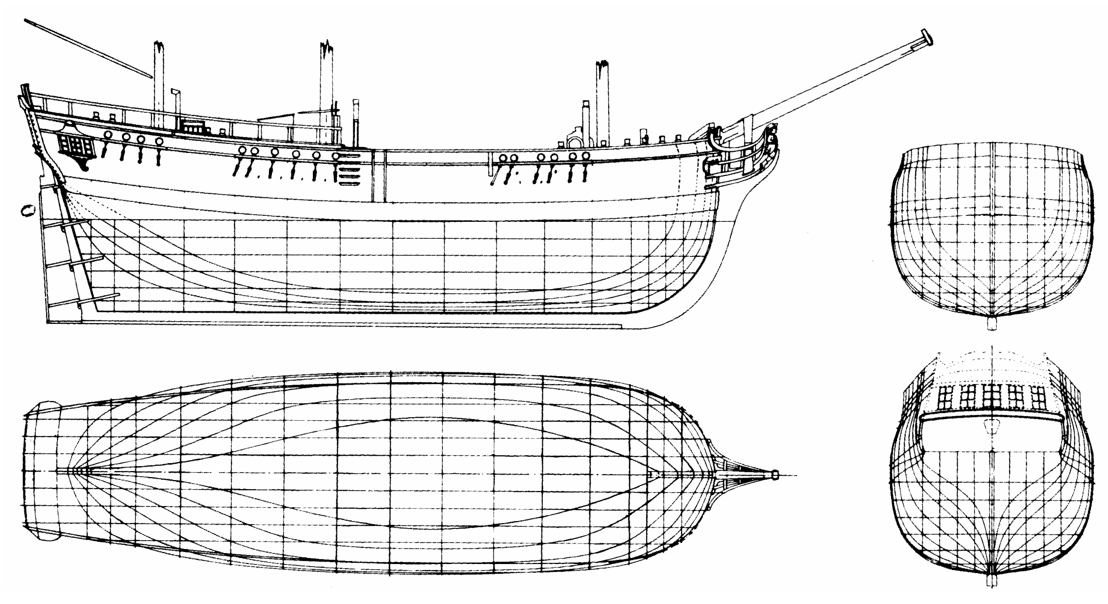


Figure 4.1, The Codrington built in 1773 was typical of early merchant ship design, based on the concept of the "Cod's head and the Mackerel's tail".

The wars of the early 19th century gave rise to significant improvements in the hull forms of ships. The British government had passed harsh laws restricting colonial trade and consequently the amount of smuggling grew, requiring fast ships. During the American War of Independence, swift Chesapeake Bay privateers, (Figure 4.2), made large profits by harassing British shipping. These vessels became known as the Baltimore Clippers. Their beginnings, however, were somewhat hesitant. In 1832 Issac McKim, a Baltimore trader, commissioned a Chesapeake shipyard to build a three-masted, square rigged ship modelled on the lines of the local clipper vessels. The *Ann McKim* was large, sleek in profile, had low freeboard and a V-shaped hull. The new vessel sailed into the wind far better and was faster than the full bodied merchant ships of the time. However, the design was not copied by other shipbuilders, as the ship could only carry half the cargo of other ships and had expensive fittings.

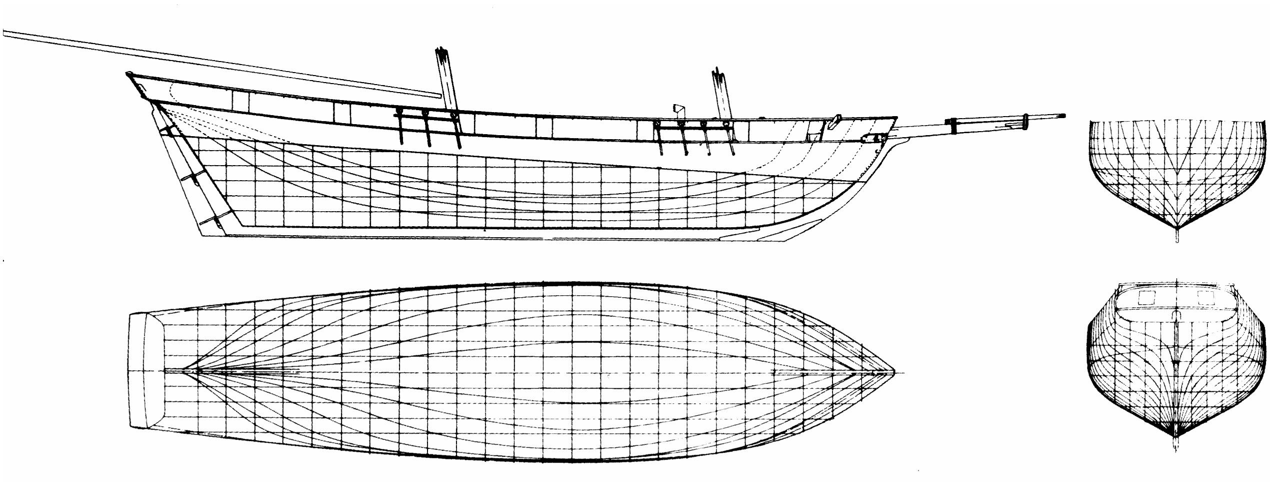


Figure 4.2, The Lynx, a Baltimore privateer lauched during the war or 1812.

With expanding world commerce, the demand for faster ships was high, especially for carrying perishable cargos. Chinese tea was one of these products and it found an increasing market in the East Coast cities of America. Merchants were eager to maximise revenues from these markets by trading the freshest cargos.

An American shipwright, John Willis Griffiths, fascinated by physical laws of resistance, performed his own tank experiments to confirm the findings of earlier research by Coronel Mark Beaufoy. Beaufoy had determined that the resistance of a body diminished with length and recommended that ship forms should be made V-shaped. Griffiths made a study of the lines of the *Ann McKim* and, with his tank experiments, deduced that the Mackerel’s tail design of contemporary ships was causing drag and that the stern should have a fuller shape so that water running past a long, thin hull would slide smoothly astern. However, the ideas were scorned by older designers and ship captains, who though that the thin shape would go through waves rather than over them. It was not until Howland and Aspinwal, New York shippers, commissioned a new fast ship from Griffiths in 1843 that the new proposals could be tried out. The *Rainbow* proved herself a good fast ship, breaking many records for fast passages. Howland and Aspinwal soon commissioned another ship from Griffiths, the *Sea Witch*, (Figure 4.3). With help from the new ships' captain, Captain Waterman, the new vessel was even faster than the *Rainbow* and was not long before other merchants were requesting similar designs. This began the period of the Clipper era, the first time that competitiveness was experienced, due to trade, at an international level. While the period of the Clipper ship was short, the characteristics features of the hull forms of these vessels remained in many ships well into the 20th century, proving the effectiveness of the design.

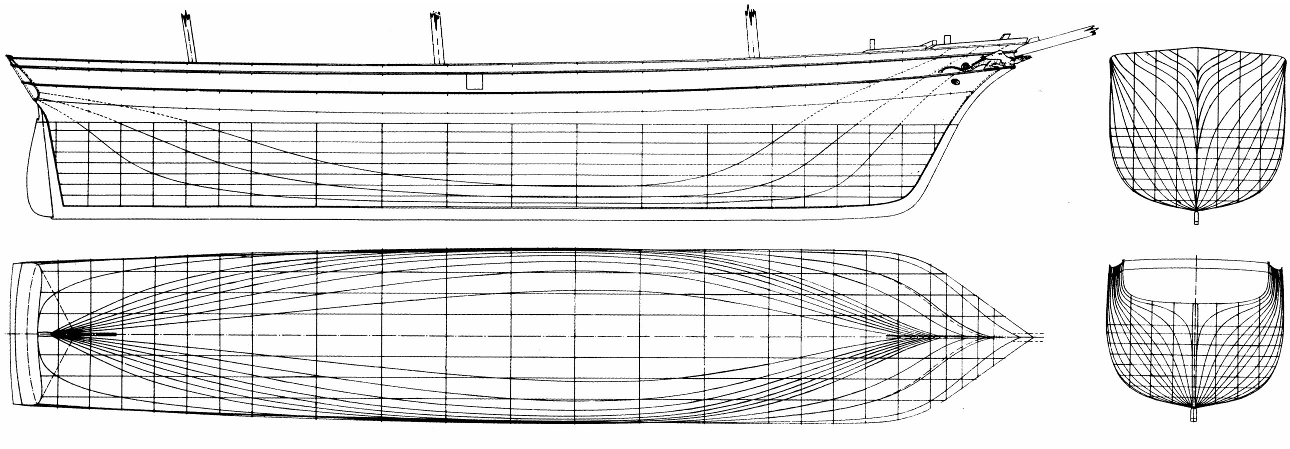


Figure 4.3, The Sea Witch, John Griffiths' record breaking clipper.

With the increasing size and number of ships being built to satisfy the needs of commerce, design techniques had to be introduced to allow ships to be built with large workforces. Ship designers could no longer oversee the complete construction process of a ship and it was necessary to develop ways of representing a design so that vital information could be communicated throughout a shipyard. The half hull or lift model, (Figure 4.4), functioned like a modern CAD database. Made of wood and about 6 feet long, the half model was used as a three dimensional sketchpad. Naval architects were normally apprenticed as shipwrights and had the skills required for accurate wood carving. Once the design was completed, the half model could be separated so that the design could be measured and the lines transferred to the Loft, where the shipwrights would follow the lines of the half model and put the design together in full scale. The half model was developed in the 18th Century but found more wide spread use during the Clipper era of the 1840's and 50's. However, by the 1870's, shipbuilders found that the half model was insufficiently accurate for the design of the new and more complex iron steam ships, and more accurate methods of hull representation were required.

Development of numerical representation in ship design is generally attributed to Fredrik Chapman [1]. A Swedish naval officer, he studied the design of ships and craft of various types, in search of improvements in speed and other desirable seekeeping qualities. Chapman's famous book 'A Treatise on Ship-Building' mentions the use of parabolas for representing waterlines and other ship's curves. However, given the calculation facilities available in the 18th Century, it is unlikely that these methods could have been used as practical tools by the ship designers of the time.

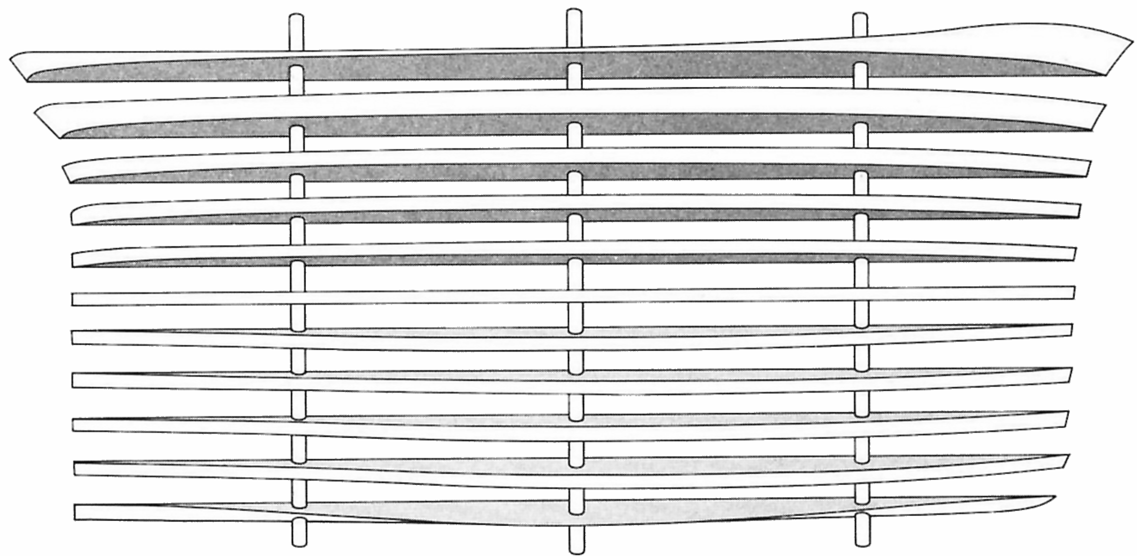


Figure 4.4, The Half or Lift model could be separated to allow the lines of the vessel to be analysed.

Chapman became a leading authority on the design of ships based on scientific theory and evolved the technique of the curve of cross sectional area, to be used as an aid to successful design. Chapman made comparisons of the curve of areas of fast frigates in an attempt to establish relationship of form with respect to speed. However, Chapman and his contemporaries had little knowledge of the laws of resistance and were unable to find a link between it and hull shape.

The 19th century saw the introduction of iron hulls and steam propulsion. These developments necessitated further evolution in the hull design of ships and an age of experimentation was born. Early resistance theories, such as Scott-Russell's "Wave Line Theory" [2], attempted to explain hull performance. However, these ideas were discredited once tank experimentation had been accepted as a practical tool for research and for predicting the performance of hull forms. Using these tools, the design of hull forms could be improved and mathematical techniques for generating known good hull shapes could be developed.

At the beginning of the 20th century, numerical hull design techniques were beginning to be used as a design tool. The improvement that numerical design techniques could bring to hull performance is illustrated in the design of H.M.S. Dreadnought in 1904-5. The waterlines of this vessel were faired using sinusoidal curves and the buttocks by using elliptic curves. Tank test results showed that the vessel could be propelled at 21 knots with 5,000 H.P. less than what R. E. Froude had estimated. Froude was so astonished at the low resistance indicated by the tank test that he ordered the model to be destroyed and a second built. The second model gave similar results to the original tests and the model followed the same fate as the first. It was not until the seventh model was run and confirmed earlier results that the E.H.P data was sent to the Admiralty. The ship was finally constructed with engines that produced 21,000 H.P. A reduction of 5,000 H.P. was a significant saving in power.

The potential benefits of introducing scientific methods into ship design had been shown, leading to a great search for the relationships between hull form and hydrodynamic performance. The search for this relationship was to be the prime factor behind the development of practical techniques for mathematical hull representation throughout the 20th century.

5. DEVELOPMENT OF MODERN HULL DESIGN SYSTEMS

Hull representation techniques took a leap forward at the start of the 20th century when D. W. Taylor [3] began a systematic study of hull resistance through model testing. Taylor developed one of the first techniques to produce a hull form using a mathematical formulation. The method was designed so that the models produced by the technique were parametric variations of a parent hull. This yielded a set of models that could be used to investigate the effect of certain hull form parameters on resistance. However, the formulation did not directly produce the hull surface shape, It produced a section area curve and, with a modification, the technique could produce waterlines of a hull. The Taylor Standard Series was the first of the methodical series of ship forms to receive wide attention and usage. The widespread use of *The Series* led others to develop the technique into something that could be used for hull surface representation, mainly for hydrodynamic analysis.

Once mathematical hull representation had been shown to be a realisable concept, hydrodynamicists and naval architects developed separate procedures based on the need of their particular field. Naval architects required a technique that would allow the development good smooth hull forms that would perform as expected and would be accurate enough to be used in the construction of a vessel. Hydrodynamicists required hull representation functions that could be used in the calculations of forces. As each field developed a new technique, the other would borrow it. Thus began a process that resulted in the progressive development of new hull representation techniques until the 1970's.

In 1940, Benson [4] developed Taylor's formulation further into a tool that could be used for hull surface design. Benson, using Froude's dictum that "resistance of a form is largely determined by the curve of transverse sectional areas, together with the extreme beam and the line of flotation of the fore-body", developed a technique that produced a sectional area curve, waterlines and sections with known properties. The method was found to be quick, practical and efficient compared to the usual methods of trial and error. Hull forms could be developed without requiring a large amount of computation allowing the technique to be used effectively in all drawing offices. However, neither Taylor's nor Benson's method produced hulls of minimum resistance. Further development in the field of hydrodynamics was necessary before hull form optimisation could be used during design.

By the 1960's, the increasing availability of computing technology prompted the development of early software for hull design and analysis. Kerwin [5] developed one of the first programs to hold a complete mathematical representation of the hull surface. Hull offsets are entered into the system and a semi-orthogonal polynomial surface is fitted to the data by using a least squares method. The representation is primarily intended for the computation of wave making resistance and other hydrodynamic properties of an oscillating thin ship. However, a major draw back is that polynomials of very high degree and a large number of input points are required. Similar polynomial techniques were developed by others for different applications. Many of these techniques are discussed by Kuo [6].

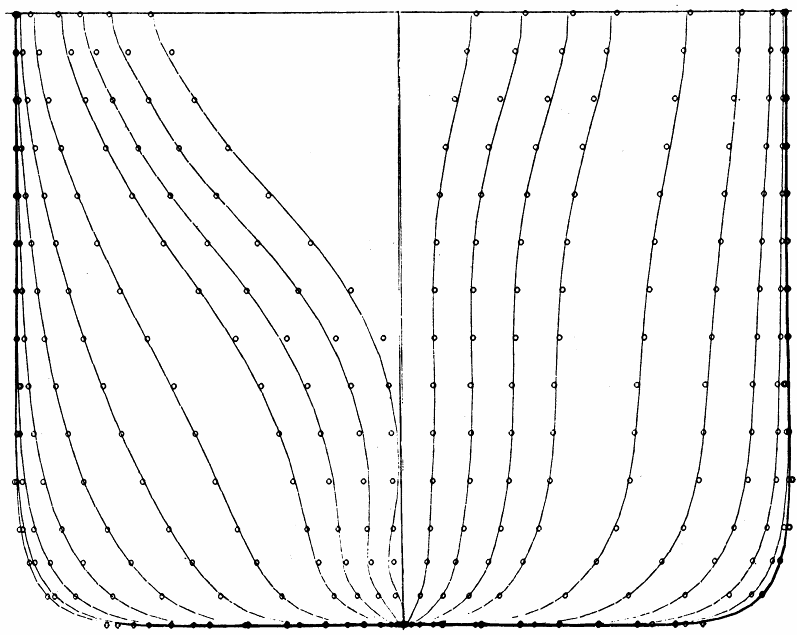


Figure 5.1, A final surface equation fit to a complete Series 60 hull form using Von Kerczek's technique [7].

While polynomial techniques could be used for hydrodynamic analysis, conformal mapping techniques represented a much more efficient method for calculating the hydrodynamic properties of a hull form. Von Kerczek [7] developed Theodorsen conformal mapping functions into a practical method for hull surface representation, (Figure 5.1). Conformal mapping is a technique that is used to transform hull sections into a circle. As the hydrodynamic properties of circles are known, the properties of the hull section can be found using the transformation. Least squares polynomials are used to control the coefficients of the mapping functions along the length of the hull thus creating a three dimensional hull surface representation. However, the conformal mapping technique has significant problems in representing certain hull sections particularly in the region of the bow bulb, (Figure 5.2). This may not be of great importance when calculating hydrodynamic properties, as only an approximation of the hull shape is required. However, this flaw precludes this technique from being used for accurate hull representation.

Although this method was primarily intended for hydrodynamic calculation, it was found that the continually evolving computer technology allowed the method to be developed into a system for hull form design and software was developed to allow the user to easily modify the hull shape interactively, using a light pen. Despite the fact that the technique is unable to represent certain hull shapes, the tool laid the foundations for future software packages.

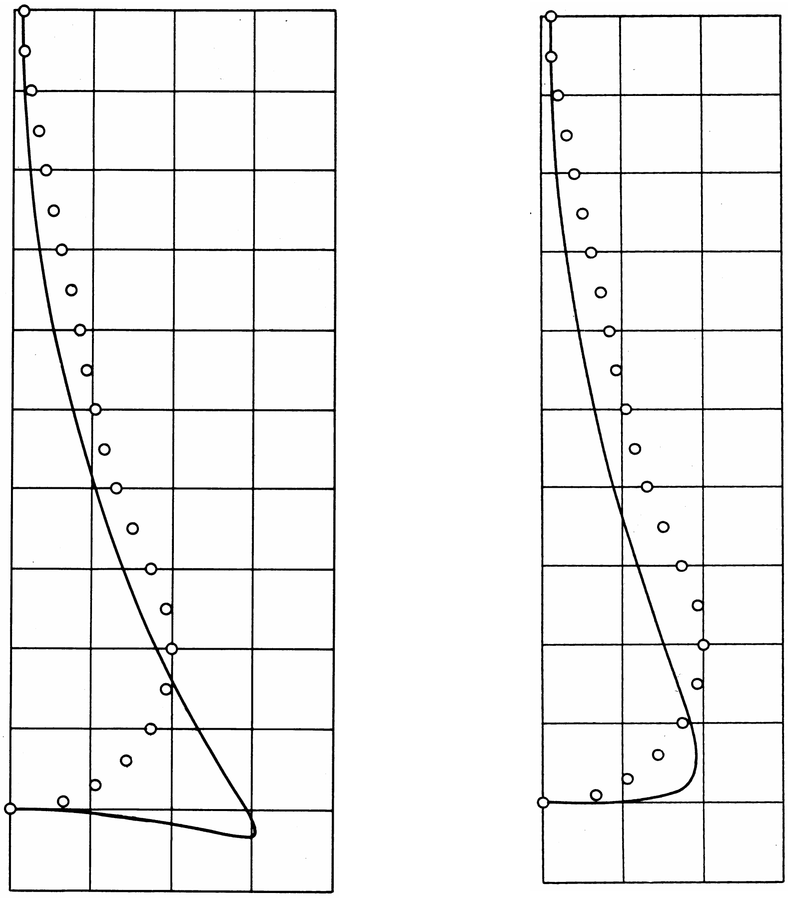


Figure 5.2, Attempts at fitting a conformal mapping representation to a bulb section by Von Kerczek [7].

The evolution of the computer allowed the characteristics of hull representation techniques to change as simpler polynomials could be handled more easily than the complex analytical formulations originally designed to restrict the amount of hand calculation. Kuiper's [8] technique, loosely based on Benson's method was developed around a new approach to hull form representation, i.e. parametric hull form generation. The hull shape is generated from design parameters rather than fitted from hull offset data. Waterline polynomials are used for hull shape representation, the coefficients of which are controlled by draught functions. The draught function polynomials are controlled by form parameters deduced from the initial design parameters. Although this method was limited in complexity of the hull form, it was the first system to demonstrate the technique of parametric hull design. A similar hull generation technique to Kuiper's is found in FORMG, the parametric hull generation tool of FORAN [9].

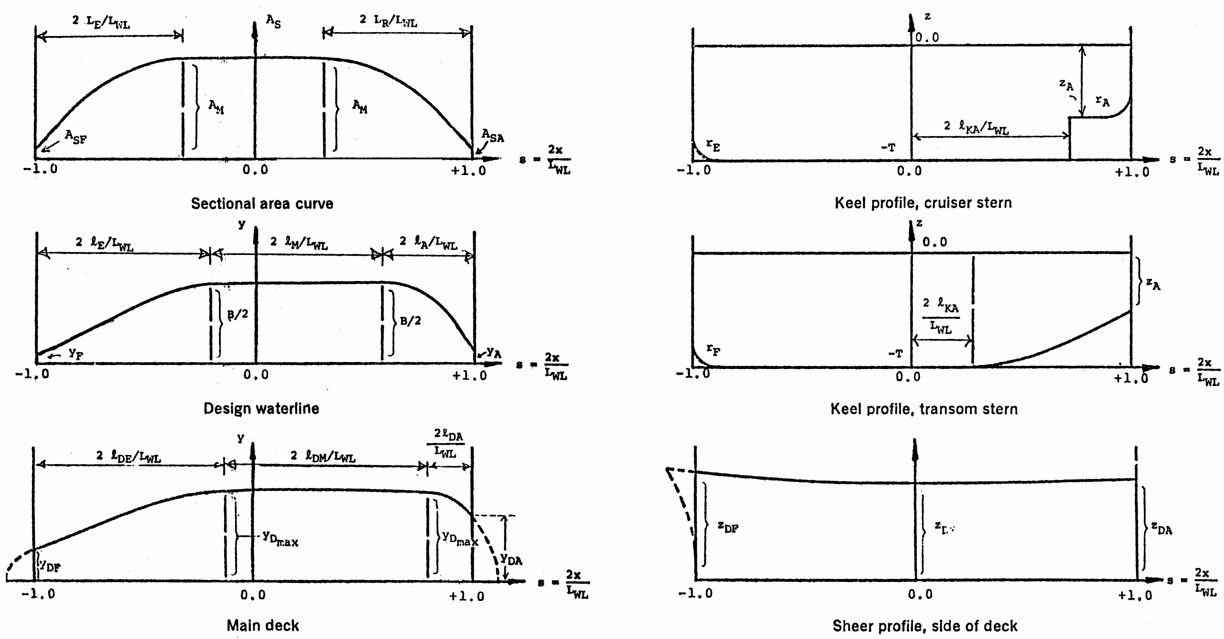


Figure 5.3, The influence functions of Reed and Nowacki's [10] technique
are constructed from simple form parameters.

Both the polynomial and conformal mapping techniques suffer from certain limitations. The major drawback of polynomial functions is that their slope remains finite for finite abscissa values, making it difficult to represent shapes with vertical tangents such as wall sided hull sections. A major drawback of conformal mapping functions is that the tangent of a section at the waterline is vertical. This makes it difficult to represent sections with flare. Reed and Nowacki [10] developed a compromise between the polynomial and conformal mapping techniques to allow a greater range of hull shapes to be represented. This technique uses conformal mapping functions for the underwater hull and polynomial functions for the hull above the waterline. The polynomial sections are blended to the conformal mapping section just below the waterline. Building on Kuiper's parametric approach to hull form generation, polynomial influence functions generated from design parameters are used to represent properties of the hull shape over the length of the vessel, (Figure 5.3). Hull sections are generated from the influence polynomials with use of the conformal mapping functions by employing various analytical procedures. Despite this hybrid approach to hull form generation, the range of hull forms that can be represented with this technique is still limited. This study demonstrated that neither polynomial nor conformal mapping techniques were entirely satisfactory hull form representation techniques for practical design purposes.

While hull form representation techniques were being developed and improved, the CAD community had also been developing techniques for representing complex shapes. Parametric functions had been found to be one of the most useful techniques for representing complex shapes, as they are axis independent, can easily represent multiple valued functions and infinite derivatives. In the early 1970's, Bézier [11] demonstrated the use of a free form curve generation technique that was appropriately named after him. The Bézier curve was initially used for numerical control and design applications at Renault. Renault's designers would program the vertex locations of the Bézier curve control polygon into a computer and the resulting curve would be drawn on a large flatbed plotter for visualization. The vertex locations would be adjusted until a pleasing shape was reached. However, the order of Bézier curve is entirely dependent on the number of vertices in the control polygon and this feature gives rise to a number of drawbacks which can be undesirable in curve generation.

Bézier's work inspired many people in the CAD community and after a visit to Bézier's laboratory in 1970, Riesenfeld [12] set about developing a curve definition technique that would produce well-controlled, smooth spline shapes of prescribed order. After considering the work of de Boor and Cox on the evaluation of splines, Riesenfeld found a technique that included the Bézier curve as a special case. These curves became known as B-spline Curves. Further work by others, to develop uses and tools for this technique resulted in the curve definition technique of Non-Uniform Rational B-Spline, normally called NURBS. NURBS are now established as one of the most important CAD entities as they can be used to represent almost any shape.

The flexibility of NURBS functions in free form design make it an ideal tool for representing hull forms. Creutz and Schubert [13] developed a set of functional requirements for curve functions used in the design of hull forms and demonstrated that the B-splines fitted all criteria. Creutz and Schubert developed a procedure for generating B-spline curves from form parameters. Further work by Munchmeyer, Schubert and Nowacki [14] resulted in the development of a full hull design system allowing hull form generation and fairing. The example output of this system, Figure 5.4, shows that the technique was quite versatile and able to create knuckle lines and bulb features.

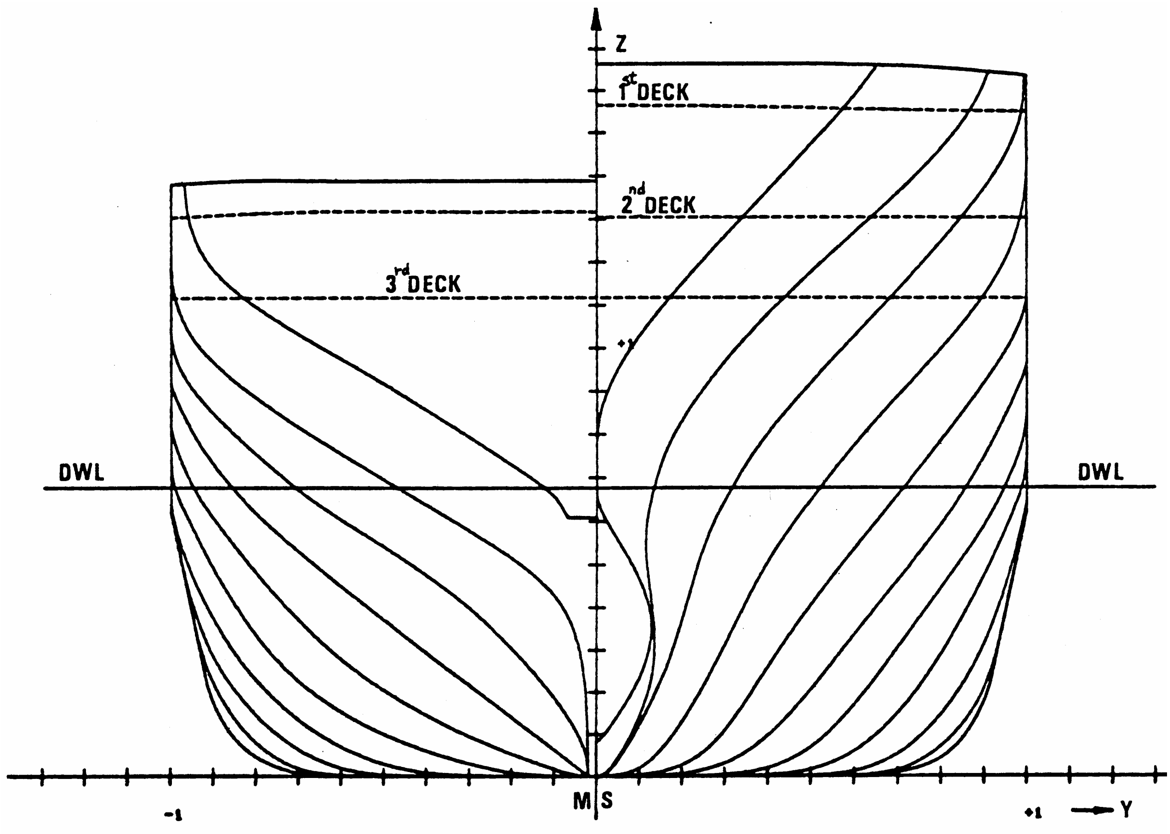


Figure 5.4, a computer generated body plan of a container ship by Munchmeyer, Schubert and Nowacki [14].

NURBS surfaces are relatively easy to implement in software and the development of the cheaper PC computer resulted in an increase of research into hull representation using these techniques. Examples from early software packages representing hull form with NURBS surfaces show that even with the relatively short development time since NURBS had been introduced, good quality hull forms could be produced, compared to previous hull representation techniques. Soon many software packages became available giving naval architects a selection of tools with regards to price and technical capability. As graphical user environments were introduced, such as Microsoft Windows, hull design packages were improved to the extent of the modern fully interactive surface design systems found today and some of these software tools are reviewed in Appendix 1.

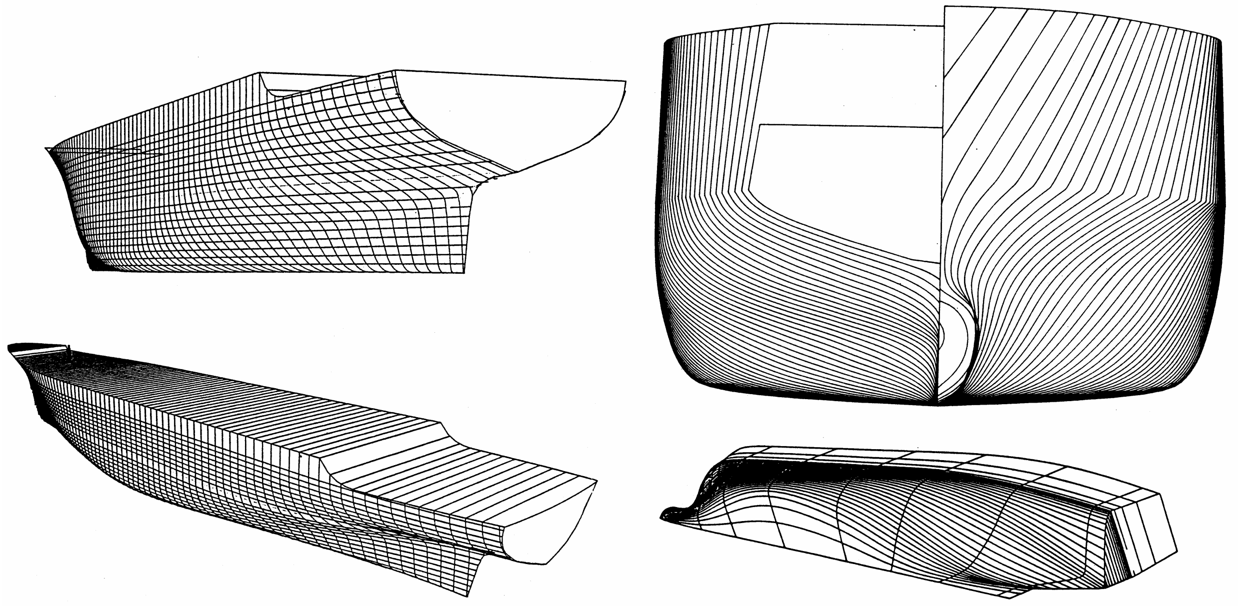


Figure 5.5, Hull forms produced by early NURBS software, Fog [15] left, Pommellet [16] right.

Once the mathematical problem of hull representation had been resolved, other tasks in the ship design process could be linked to the hull representation and completed much more efficiently than before. Computers also offered the ability to design in three dimensions and early examples of product modeling systems were developed. Yuille [17] demonstrated one of the first product modeling systems. Used in naval ship concept design, the Forward Design System used Coons patches to represent the hull form. Although the system was not capable of the wonderful rendered views of the vessel internals produced by many of today's modern packages, the system was capable of showing the inside of compartments in wire frame.

Those working in ship construction also recognized the advantages of accurate topological representations of the hull surface. The production process from the evaluation of hull surface fairness at the initial design stage through to shell expansion and seam landing at the production design phase could be greatly improved. Many shipyards instigated studies to find out which surface function yielded the most advantageous representation with regards to their production procedures.

The introduction of NURBS surfaces to hull form representation was a great improvement over all other previous representation methods. The ease of implementation and interactive nature of this technique resulted in a large reduction in research being performed in the field of hull form representation. The search for a practical hull representation technique was all but over.

The lack of research into the improvement of hull representation techniques and design tools has probably been a disservice to the ship design industry, as many other engineering fields have design tools that could be considered, “space age” when compared to naval architectural hull design packages. Despite the lack of continued research into hull form representation, there have been a few interesting developments which are worth noting.

The ability to parametrically specify a hull shape still appears to be a feature that many hull designers would like, incorporated into their favourite design package. There are a few hull generation tools of this kind in some of the modern packages. However, there are no effective implementations that work well with NURBS hull representations throughout the design process. In concept design, a parametric hull generator can be useful as it allows many hulls to be systematically generated and the feasibility of each hull evaluated with respect to other features. This is a process particularly suited to the design of new naval vessels.

In 1987, Keane [18] developed a hull form concept design system that allowed the stability of a parametrically generated hull form to be analysed. Turning back to conformal mapping techniques, Keane developed a variation of these by allowing for rise of floor and flare, Figure 5.6. The above waterline portion of the hull is added as a quadratic polynomial using a technique similar to Reed and Nowacki [10].

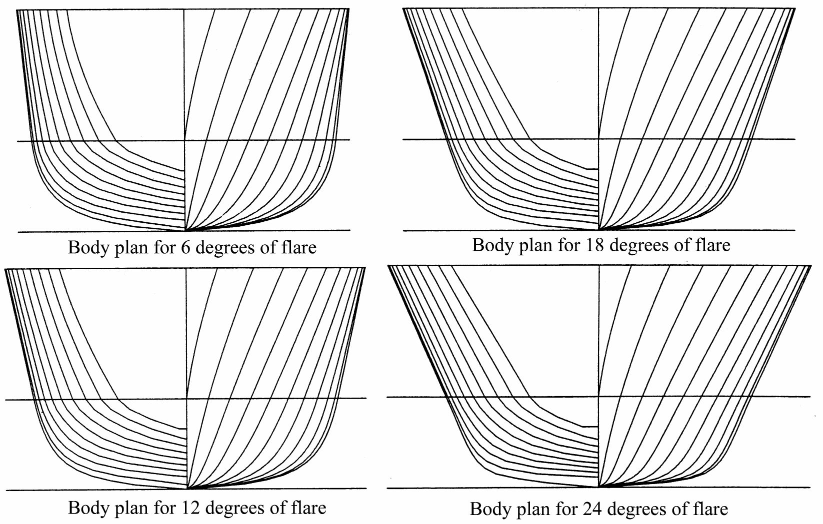
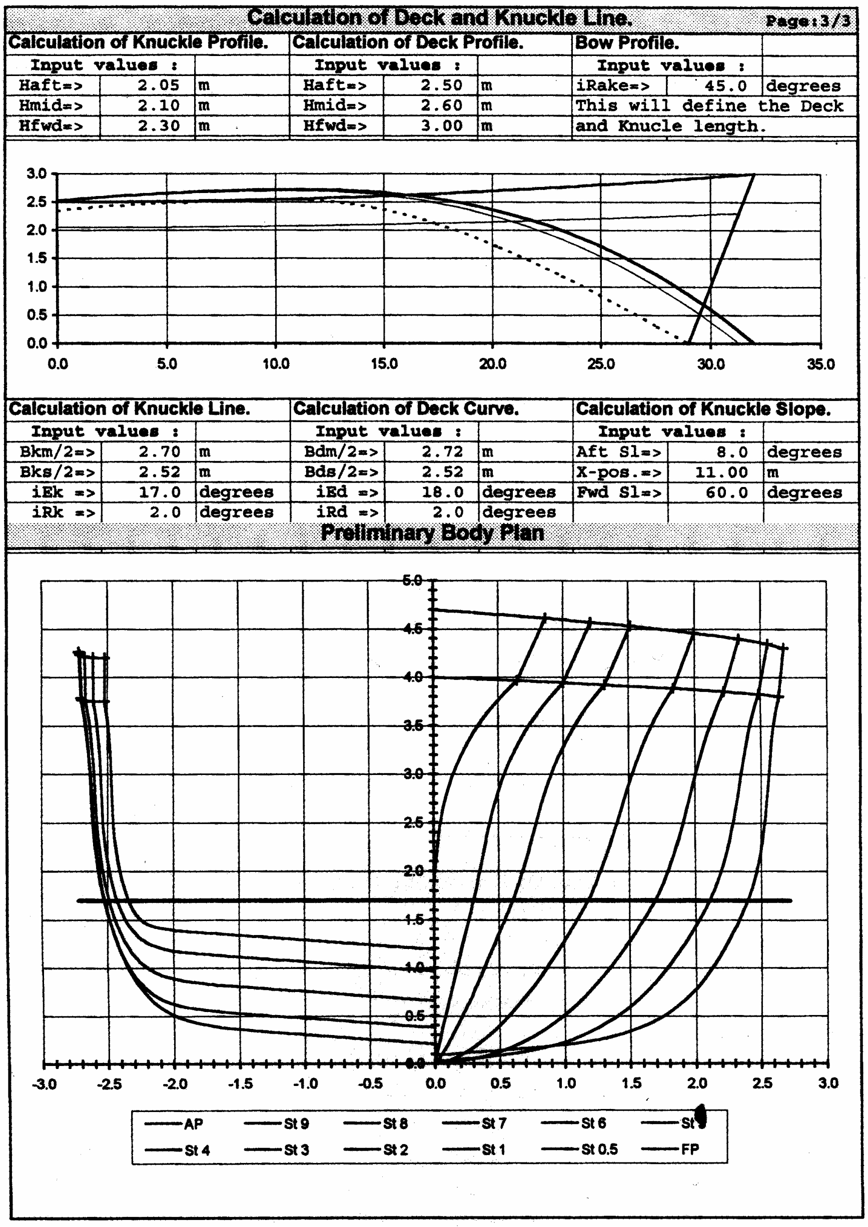


Figure 5.6, A demonstration of flare using Keane's [18] conformal mapping technique.

The method uses up to 19 parameters to define the shape of the hull and it would appear that most of the parameters control obvious geometric aspects of the vessel and can be varied by intuition. However, there does not appear to be a great variation in the types of hull that can be generated, with the shape generally resembling frigate type forms. Although the hulls are simple in shape, the system allows complex effects to be studied such as the effect of geometric parameters on the shape of the GZ curve. Despite the fact that the hulls produced by this technique are mainly relevant to frigate hull design, this program could probably be used effectively for teaching students about the effect of hull changes on stability.

In 1990, Bloor and Wilson [19] extended a PDE (Partial Differential Equation) method for blend design into the area of free form design. In blend design, it is necessary for surface continuity to be maintained over the boundaries between the parent surfaces and the blend surface. In free form design, the derivative conditions no longer need to maintain continuity across boundaries and can therefore, be used to control the shape of the surface. It has been demonstrated that this method can be used effectively to design both ship and yacht hull forms. However, the surface technique is derived from Laplace's equation, requiring a good mathematical background to feel comfortable about using and understanding such a method. Surface shape is controlled using derivatives at the boundaries of the surfaces. Consequently, as with the Coon's patch, using derivatives to control the surface shape can be difficult as the link between a derivative parameter and the resultant effect on the surface shape is not always perceptible. As a result, control over surface properties, such as flats, knuckle lines and bulbs, is likely to be very difficult. Most naval architects prefer to interact with the hull surface directly and this method does not allow them to do so.

Many of today's hull design packages are complex systems requiring high specification PC's and a good deal of training to operate. In 1997, Jorde [20] showed that it was possible to develop a hull design tool using a standard spreadsheet tool. Using simple cubic polynomials, Jorde was able to develop a spreadsheet that created a body plan, Figure 5.7, from control curves such as section area, waterline breadth and the keel profile. The section area and the waterline breadth curves are generated from numerical parameters using cubic polynomials, while the keel profile curve is created from offset data. Using the control curves, the below-waterline sections are generated. Separate section shapes for the above-waterline portion of the hull are connected to the underwater sections by maintaining the slope of the sections across the waterplane.



Calculation of Deck and Knuckle Line.								Page:3/3
Calculation of Knuckle Profile.			Calculation of Deck Profile.			Bow Profile.		
Input values :			Input values :			Input values :		
Haft=>	2.05	m	Haft=>	2.50	m	iRake=>	45.0	degrees
Hmid=>	2.10	m	Hmid=>	2.60	m	This will define the Deck		
Hfwd=>	2.30	m	Hfwd=>	3.00	m	and Knucle length.		

Calculation of Knuckle Line.			Calculation of Deck Curve.			Calculation of Knuckle Slope.		
Input values :			Input values :			Input values :		
Bkm/2=>	2.70	m	Bdm/2=>	2.72	m	Aft Sl=>	8.0	degrees
Bks/2=>	2.52	m	Bds/2=>	2.52	m	X-pos.=>	11.00	m
iEk =>	17.0	degrees	iEd =>	18.0	degrees	Fwd Sl=>	60.0	degrees
iRk =>	2.0	degrees	iRd =>	2.0	degrees			

Figure 5.7, An example of a body plan produced by Jorde's [20] spreadsheet system.

It is evidently possible to use modern spreadsheet programs for hull design. However, this procedure is not particularly effective for practical hull design. It can take a long time to set up a spreadsheet for a design and Jorde has used some of the advanced macro features to achieve this example. Spreadsheets are not designed for creating hulls and the development of the design can suffer because of this. Programs developed in spreadsheet are not very user friendly and it can be difficult to recover from mistakes. However, it would be possible to design more appropriate software which would allow the construction of hull forms using the methods described here and benefit from many of the functions found in modern hull design packages.

The introduction of the NURBS curves and surface definition technique to naval architecture provides an unusually complete solution to the problems of hull form *representation*. However, the search for an efficient hull *design* tool continues.

6. MODERN HULL FORM DESIGN

6.1. Modern Ship Design

Ships are designed for many reasons. In the majority of cases, a ship is designed for commercial gain. Naval architects must continually improve the design of ships so that each new one is more viable than the existing ships designed for the same trade. While satisfying the owners requirements, the designer must also make sure that the ship is a safe and reliable product. This is achieved by establishing an efficient design process.

A commercial ship can be considered a self contained business. Consequently, the design process should start with business investigation or a feasibility study. These studies are used to review a prospective route with respect to current transport links, route limitations, the application of new technology and environmental impact. These studies will establish whether there is a need for new ship and the required performance. If the need for a ship can be ascertained, then, if an owner is prepared for the investment, the design will progress to the initial design phases. These types of feasibility studies are provided by ship design consultants rather than the shipyard design office. Few consultancies offer these types of services, however, the information can be of great benefit to an owner. Consequently, the consultancies that can provide business feasibility studies in addition to ship design services have established a solid place in the market.

The development of the design of ships has always been a process of trial and error. However, this process is not very compatible with the requirements of business, which wants the best product for the smallest cost. As the design of the ship is dependent on the needs of an owner, several design strategies have been established which conform closely to the needs of the client.

It is not always necessary to design every ship from first principles. If an owner requires a vessel that is similar to an existing design, utilising the existing design can be the best solution, as many of the performance characteristics are already known. This ‘production’ like technique of design is well suited to the shipyard design office, as the detailed information on previous designs is normally readily available. The new design can be created by simply resizing the design of the existing vessel until the needs of the owner are met, Figure 6.1 and Figure 6.2.

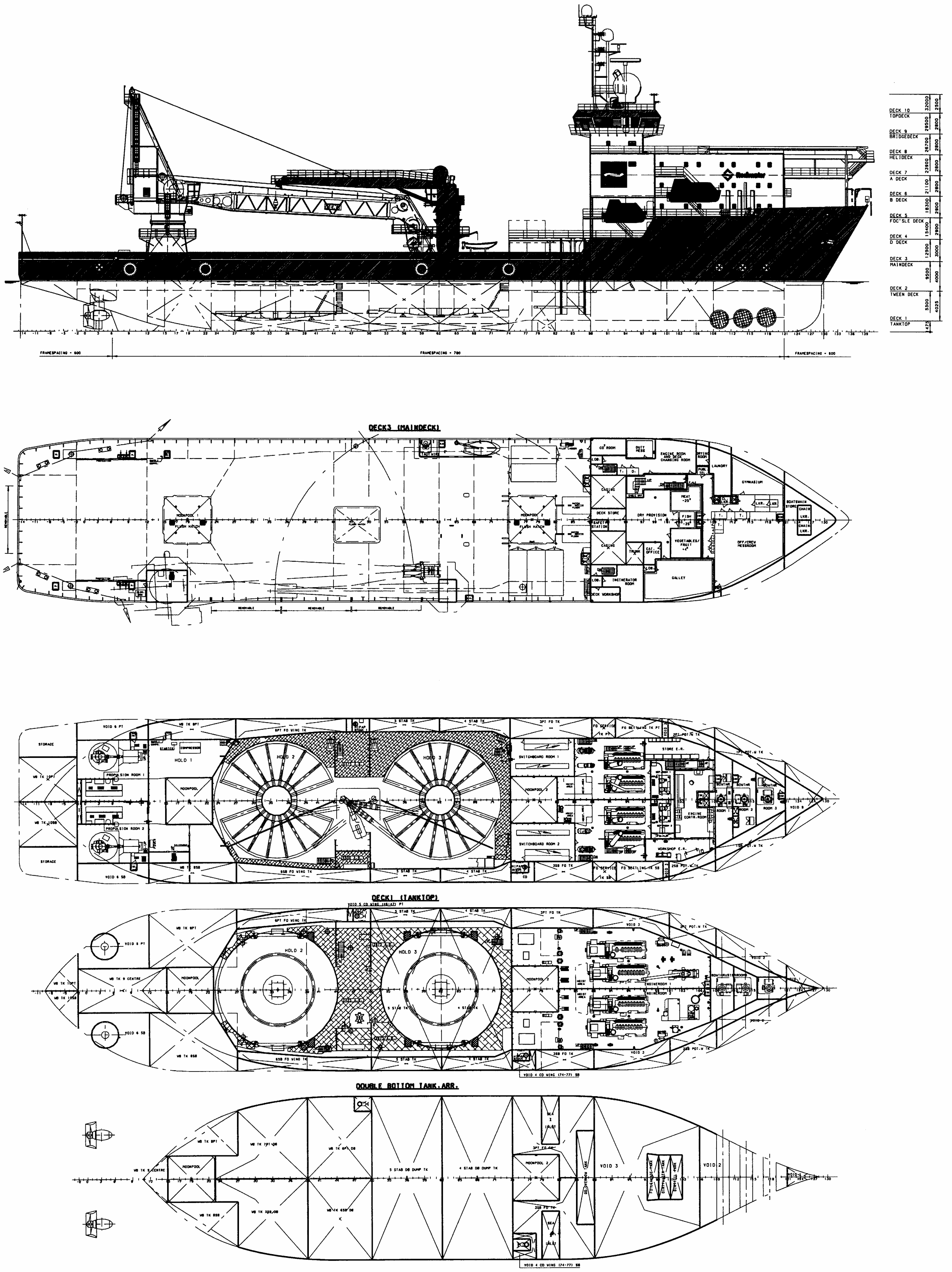


Figure 6.1, *Toisa Perseus* [21], a multi purpose offshore support vessel, parent design for *Bold Endeavour*, Figure 6.2.

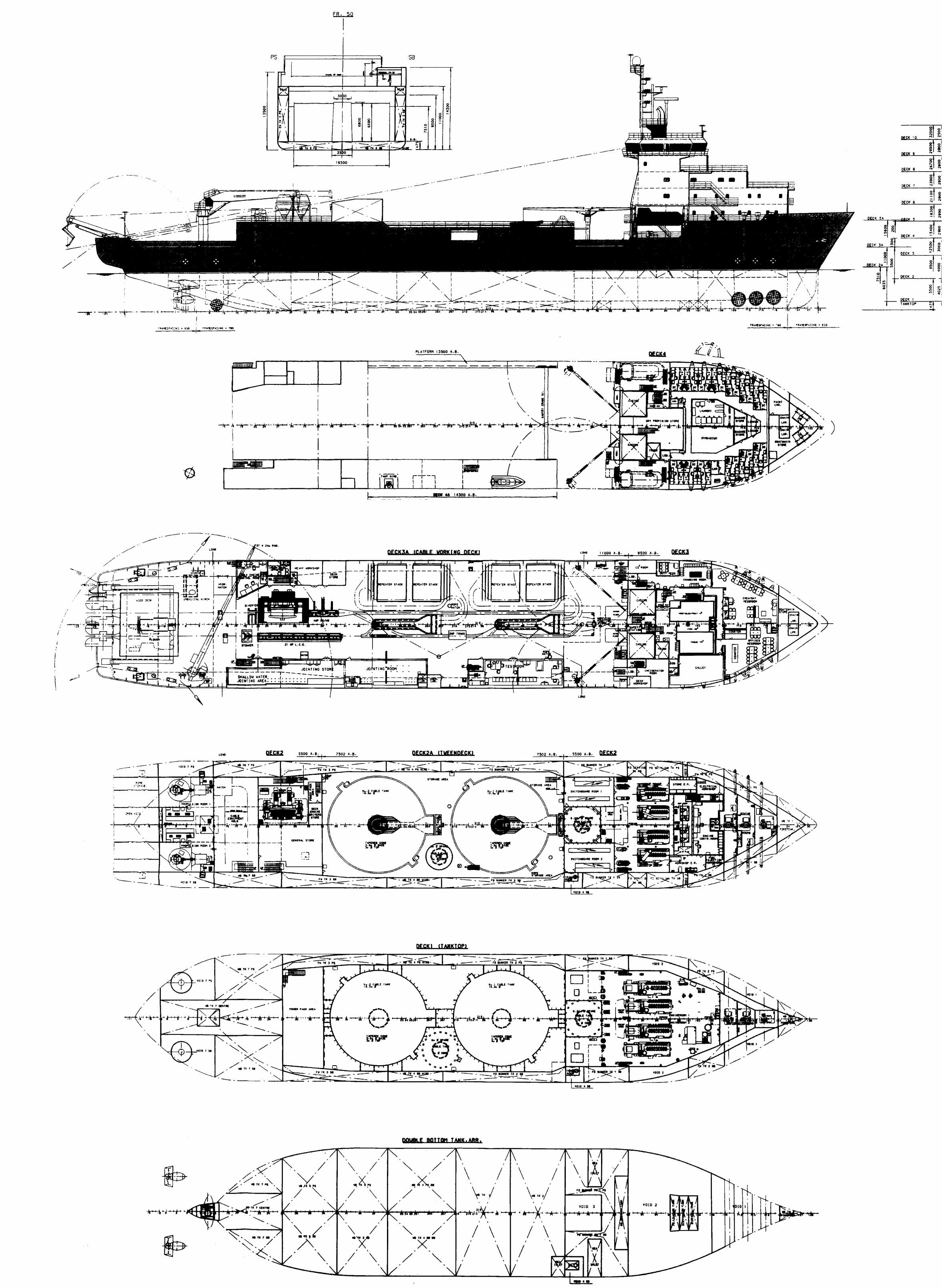


Figure 6.2, *Bold Endeavour* [22], cable laying ship, created using *Toisa Perseus* as a basis design, Figure 6.1, by extending the parallel middle body of the ship.

As long as the new design does not greatly differ from the existing design, lengthy analysis will not be required and the costs can be kept low. Reuse of existing designs is particularly cost effective for cargo ship design where the owner performance criteria are simple, only concerning the quantity of cargo and the service speed. As ship operation tends to have small profit margins compared to the initial cost of the vessel, the majority of the worlds tonnage will have been produced using this technique.

While cost effective, 'production' design is unlikely to produce a ship with any great improvements over the previous design. As soon as the designer improves the design with innovation or new technology, lengthy analysis will be required to confirm that the ship will perform to expectations. This will correspondingly increase the cost of the vessel and the risk to the owner investment. However, this 'development' is the only technique that can generate an improved optimised design. As the owner's requirements grow, it becomes increasingly difficult for existing designs to be matched well to the owner's criteria. This scenario occurs regularly for vessels being designed for evolving markets. A current, high profile example is the fast ferry and cruise ship market, particularly as passengers are a very demanding cargo. The cruise ship market is very competitive and designers must take every opportunity to include innovations or new technology.

Figure 6.3, *Disney Magic* [23].

Figure 6.4, *Voyager of the Seas* [24].

With the current booming market in passenger ferries and cruise ships, design from first principles has become more widespread as competing companies attempt to provide the largest and most technologically advanced ships to potential customers. The trend has also helped to promote the ability of design consultancies and shipyards that have developed the designs of exceptional vessels such as such as *Disney Magic,* Figure 6.3 and *Voyager of the Seas,* Figure 6.4. *Disney Magic* and her sisters can be said to be the most novel designs developed in recent years, as the design of these vessels follows the particularly unique image of the Disney Corporation. The race to build larger vessels for this market continues and it will be interesting to see how far Naval Architects can safely push the boundaries of ship design.

Both ‘production’ and ‘development’ techniques are used to create practical ship designs. However, a third technique, ‘research’ design, should not be overlooked. ‘Development’ design is the best technique to be used to improve the design of ships in general. However, rarely do the budgets of these projects allow the investigation of a large range of solutions or parameters. ‘Research’ design can be used to investigate a concept in order to make improvements to ships during the development design phases. This is generally performed by academics or research establishments, which have the facilities and expertise to investigate new design concepts without the need to provide a viable product at the end of the process. Consequently, various aspects of ship design can be analysed in detail. However, this approach requires a delicate design process, as most research establishments rarely have the expertise to consider the effect on areas of the ship besides the part being optimised and the basis design must be functional for the results of the investigation to be relevant.

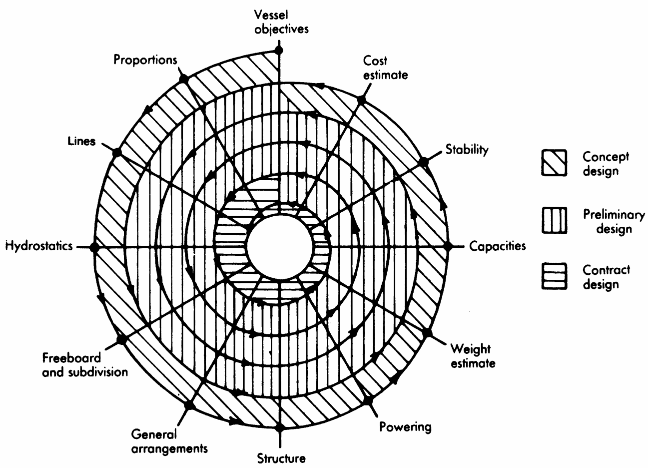


Figure 6.5, the design spiral, found in [25].

Of human engineered structures, ships and other marine based constructions must be considered some of the most complex. There are not many structures which have to supply a complete and safe life support system, as ships do, allowing people to live and work in a hostile environment such as the air-sea interface. The number of factors that a naval architect must considered is large and can at times become overwhelming. Following the design process allows each factor to be properly considered, and should result in an optimised solution. However, in practice, the limitations of time and money may compromise this process.

The classic term for this process is the Design Spiral, Figure 6.5. Each factor in the design of a ship is considered in a long sequential order taking the process from the concept stage through to the production stage. The introduction of modern technology to naval architecture means that the design process is better modelled by a network, with many tasks being performed in parallel. Better communication techniques, such as the Internet, can allow more experts to be consulted during the design process and parts of vessel can be designed in many locations of the world at once. More factors can be considered with the greatest emphasis being applied to the ships main task. In many cases, a naval architect is not required for these tasks. In a cruise ship, for example, the passenger environment is the most important part of the ship's design. An interior designer is best suited to this task. However, the interior design is likely to come into direct conflict with many of areas required for safe ship operation. In this situation, naval architect is required to facilitate compromise without reducing the safety or operability of the vessel.

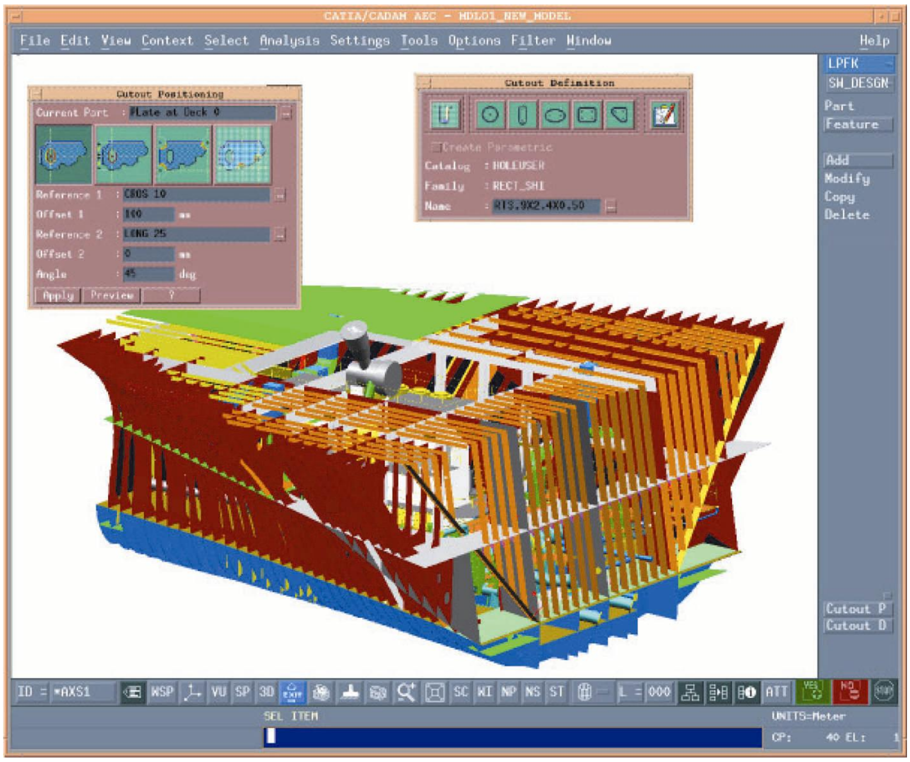


Figure 6.6, CATIA, an example of a product model database system.

The introduction of modern software design packages has greatly improved the ship design process. Integrated design tools and product model database systems, Figure 6.6, have improved ship design by providing a virtual environment in which the ship can be constructed, from the shell plating to machinery and interior decor. Any problems with the ship can be found using these systems before the design reaches the production stage.

Integrated ship design tools are becoming more common. These tools allow the designer to analyse all aspects of the ships design within same environment. A design can progress much more quickly using these tools and it is possible to go round the design spiral in a matter of hours. Integrated design systems normally function using relational techniques where the definition of the ship is built by referencing objects to others. For example, a compartment can be created by considering the hull, bulkheads and decks as boundaries. Therefore, when an item, such as a bulkhead, is moved, related objects can update correspondingly. The hull, being the foundation of the whole ship design, can be considered the most important component, especially as so many performance characteristics are dependent on it. On this basis, it should be possible to modify the hull with ease to achieve a new geometric shape. However, a review of modern hull design packages, Appendix 1, shows that hull design tools only provide the basic means to form a hull, without any techniques for easily controlling changes that may be made during the design process.

The studies performed in the initial phases of the design will be used to define the approximate size of the vessel. If there are aesthetic factors controlling the shape of the hull, the design may start with a sketch. This allows the designer to simply visualise the concept and access critical dimensions of the vessel, however, particular attention must be paid to make sure that the design is realisable in three dimensions and not an ‘impossible object’. Once the initial idea has been developed, the main particulars of the vessel can be chosen. Naval architects are great recorders of data on the many ships that have been constructed. Well established shipyards will have great databases containing detailed information on ships they and other have constructed. These databases can be accessed to allow the selection of main dimensions and form parameters for a new design based on previous trends. Using this information, an outline of the new vessel can be developed and an initial hull surface can be designed.

Depending on what type of design process is being used, a method of designing the hull form can be selected. If a new ship is being developed from an existing design, then it may only be necessary to transform an existing hull shape until the desired characteristics are met. However, a new design will require the hull to be developed from scratch. There are several ways of creating a new hull. It is possible to generate a new hull surface from form parameters or generate with respect to the optimisation of a performance characteristic. However, the most popular method is to manually define and modify the hull form shape by hand, using an appropriate surface representation technique.

6.2. Modification of a Parent Hull Form

Techniques of deriving a new hull form from a parent have been employed by designers for many centuries. In the beginning, shipwrights had their own methods for developing a new design. However, as Naval Architecture moved from an art into a science, mathematical techniques began to be applied, becoming increasingly more developed as engineers became more educated. By the early 20th century, generic techniques had developed which allowed designers to vary the shape of the hull surface by relocating hull sections. These techniques remained within the domain of the Drafting Office until they were formally documented by Lackenby [26] in 1950. These methods can be considered quite sophisticated and elegant compared to some of the crude geometric transformations used in modern Naval Architecture design software. They are designed to maintain a fair hull shape while keeping the amount of work required to implement the transformation low.

There is a standard set of transformations at the designer's disposal when creating a new ship. These can be grouped into two sets, the general coordinate transformations of scaling and translation, and volumetric transformations for changing the distribution of the hull buoyancy. However, before discussing these transformations in detail, it should be noted that there is another type of transformation frequently used to modify hull surfaces both at the design stage and in existing vessels. Hull extension is one of the most common modifications, normally implemented by adding a section of hull mid-length, increasing the parallel middle body of the ship. The modification will increase cargo capacity and in some cases, it may affect an improvement in speed due to the increased hull length. The type of modification was used to create the vessel in Figure 6.2 from the ship shown in Figure 6.1.

Despite being very simple operations, coordinate transformations can be very useful when developing a new hull from a parent design. Scaling is one of the most frequent transformations, being used to change the main dimensions of a vessel. Other transformations such as translation rotation and mirroring are used less frequently, mainly be used to transform surface definition within CAD software. Within computer software transformations can be implemented easily by using matrix functions, however, before the use of computers these transformations would have been very monotonous to apply.

Due to the simple nature of these functions, transformations can be applied indiscriminately to the surface without regards to maintaining the shape of a hull surface. Figure 6.7 shows how the shape of the bilge radius is affected by when scaled by two, breadth-wise and depth-wise independently. The resulting bilge radius is no longer a circular arc. The distortion caused by length-wise scaling of a hull surface can produce larger undesirable effects, Figure 6.8. The scaling shown in these diagrams is much larger than would be applied in reality. Nevertheless, the types of distortion demonstrated would be apparent in any hull surface that has been transformed in a scale operation and considering that the initial parent hull design may have had full analysis, such CFD on bulb and propeller appendages, for how many other designs scaled from the original will the analysis be valid?

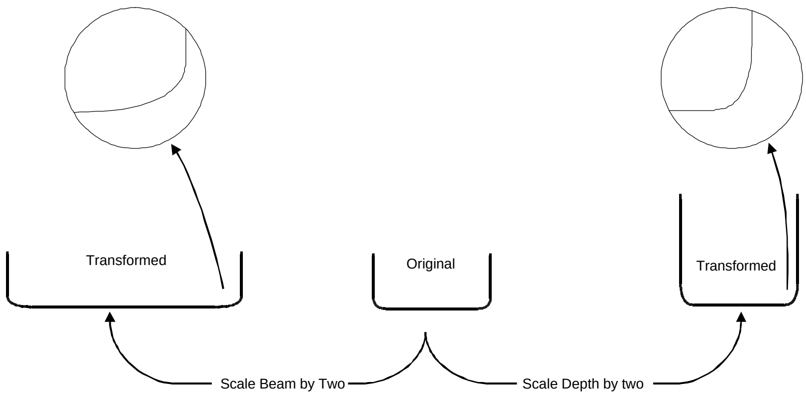


Figure 6.7, deformation in bilge radius caused by scaling the midsection of ship.

When deriving a new design from a basis ship, it may be necessary to modify the fineness of the hull to achieve a new displacement or centre of buoyancy location. Standard transformations have been derived to allow designers to achieve this while maintaining a fair hull. Theses transformations are more commonly known as the "1-C_P" technique for changing the displacement and the method of 'swinging the section area curve' for changing the location of the LCB. Despite being a well-known method, the "1-C_P" technique has particular limitations, being unable to control a number of hull parameters. Lackenby [26] details these limitations and presents a generalised method that allows further parameters to be controlled.

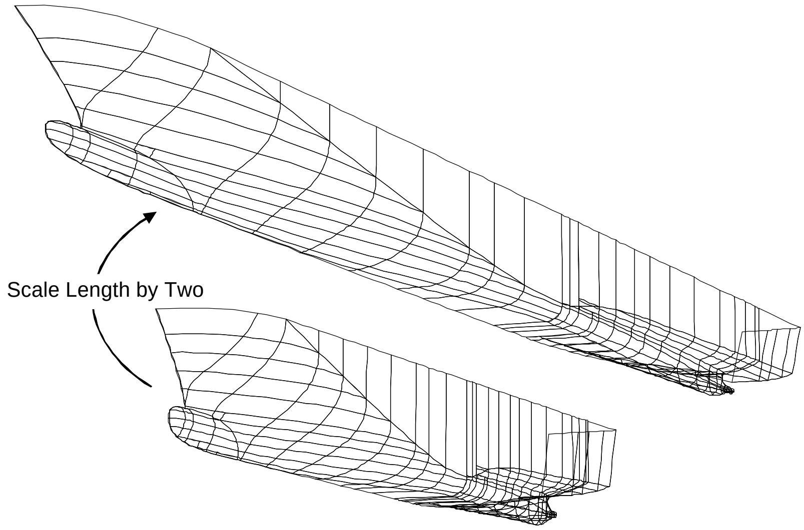


Figure 6.8, deformation in a hull surface when scaled in length.

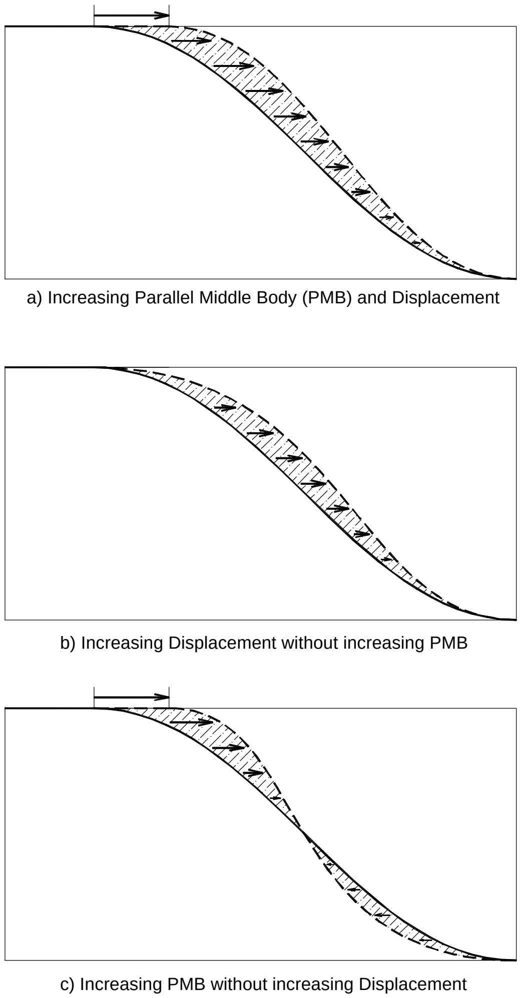


Figure 6.9, Various ways of transforming the section area curve to achieve a change in Displacement or Parallel Middle Body.

The basic concept behind the variation technique is simple. By relocating hull sections, the shape of the section area curve can be controlled to vary Displacement, LCB and Parallel Middle Body. Corresponding parameters for the Entrance and Run can also be changed. A mathematical function controls how each station is repositioned. Figure 6.9 shows how the section area curve is affected depending on what parameters are being varied. The entrance and run can be transformed independently giving good control over the hull shape.

Most hull design packages have some form of transformation tools. However, it would appear that these types of tools are not regarded as highly useful functions as they are not always located in the hull design environment. Figure 6.10 shows a well designed interface to the transformation tool in the Maxsurf [27] hull design package.

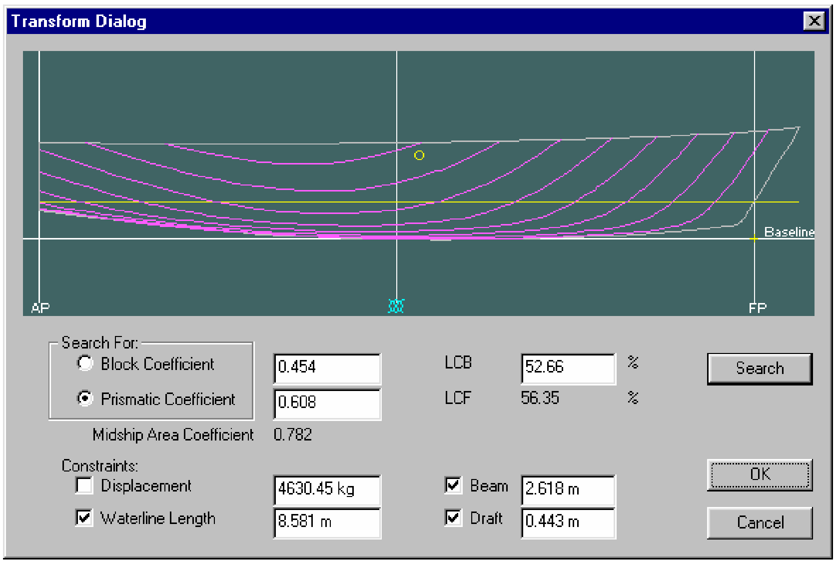


Figure 6.10, the hull transformation tools provided by Maxsurf.

As transformation techniques maintain the geometric characteristics of a hull form, the approach is not particularly effective for optimising the performance of a design. Most optimisation procedures rely on more subtle changes. This design methodology is particularly suited to vessels carrying non-perishable goods, where the performance of the hull is not a critical factor in maintaining commercial viability. Furthermore, design development costs are minimised. In cases where the ship itself is part of the commercial product, such as Cruise Liners, there is a great rivalry between competing companies to attract more customers. This can only be achieved by having the “best” product. The only effective way of continually producing the best product is carry to out the full design procedure. This will always ensure that the resulting vessel is at the cutting edge.

6.3. Manual Creation of a New Hull Surface

Modern hull design tools have been developed in a way that it is not necessary for the user to understand the mathematics behind the hull definition technique. Techniques such as NURBS are intuitive to the extent that it only requires a few minutes for a user to understand how the curve or surface will react to certain input. Using this feature, software manufactures provide a simple interface to allow users to interact with the surface. Hull surfaces are usually controlled by the position of definition vertices. In the case of NURBS, the vertices define a structure called the control polygon, a set of points, from which a curve or surface can be generated. The positions of these points are manipulated to modify the hull surface in an iterative operation very similar to manipulating a spline batten on a lines plan.

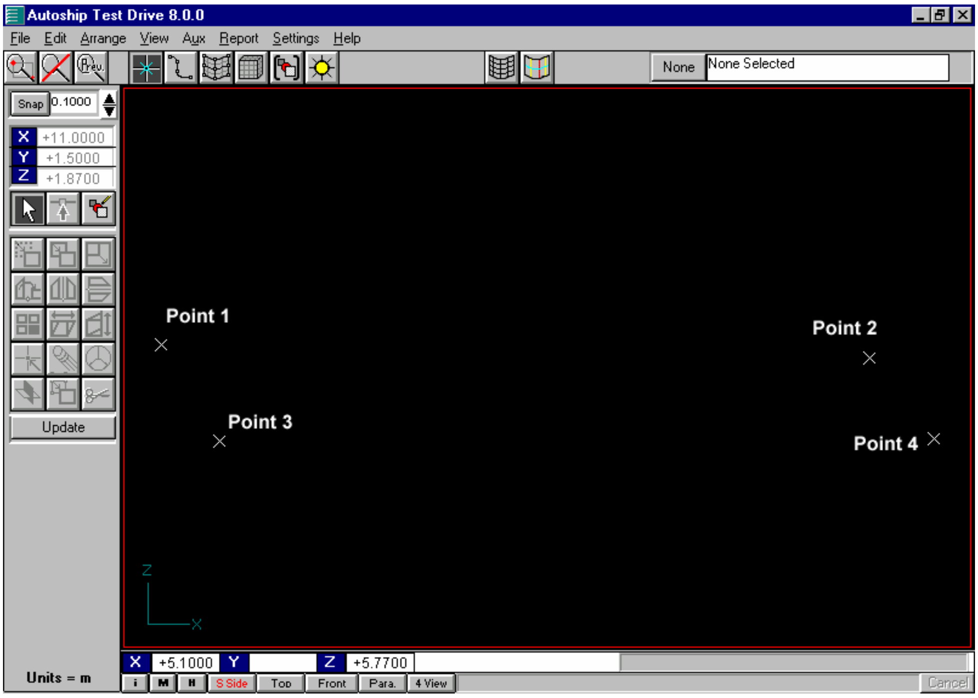
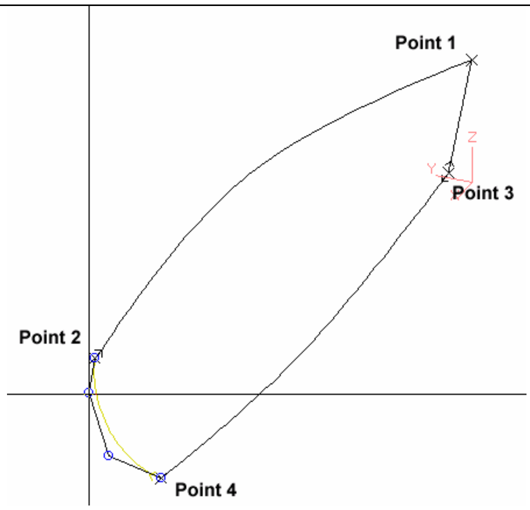
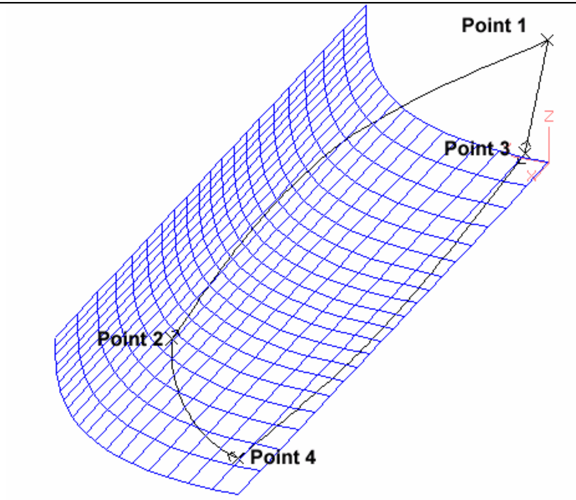


Figure 6.11, the initial definition points for a yacht hull surface using Autoship.

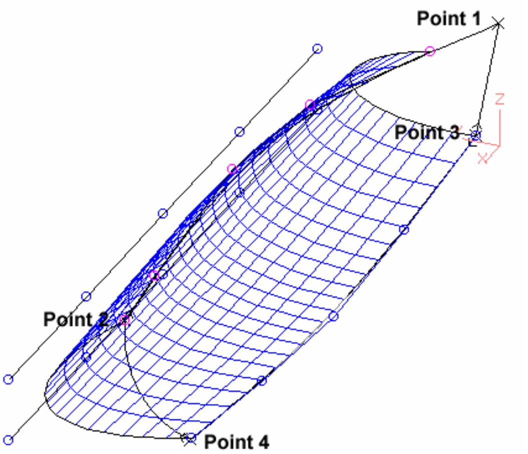
In Autoship [28], for example, as with other software packages, the design of a new hull surface starts with a blank screen, giving no feedback of the scale of the design space. The first steps are to define the limits using point entities at the extremities of the vessel, Figure 6.11. Using relational geometry, curves can be attached to the points and their shape adjusted by manipulating the control polygon vertices, Figure 6.12a. A dialogue box is used to create a curved surface of similar size to the curve structure. The surface boundaries are attached to the curves in Figure 6.12c and Figure 6.12d. The shape of the surface can now be controlled using the surface control mesh, Figure 6.12f. The completed hull form, with sections is shown in Figure 6.13.



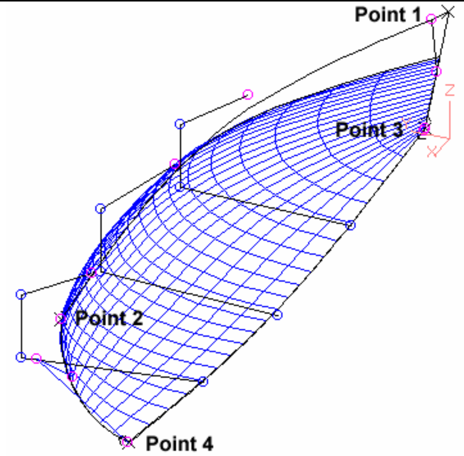
a) Curves attached to definition points.



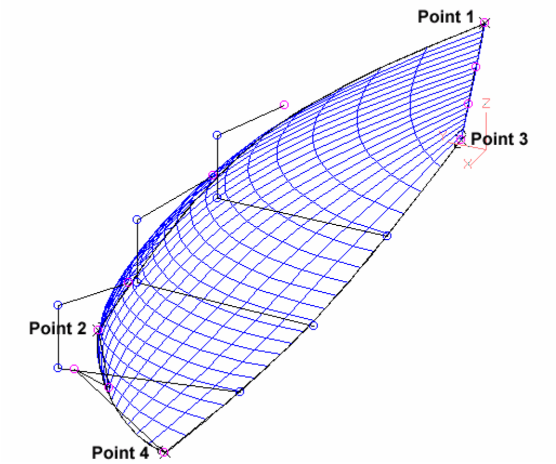
b) A new surface is added to the definition.



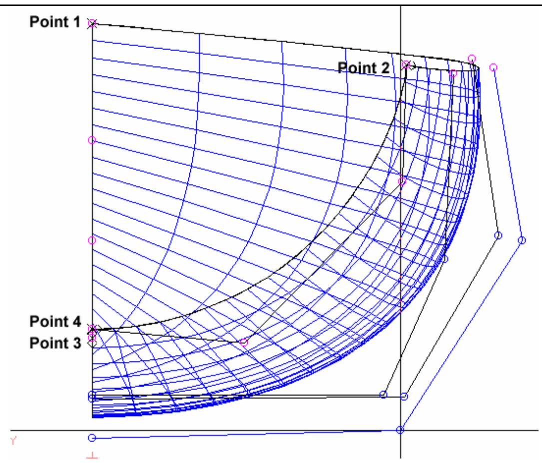
c) Upper and lower surface boundaries attached to curves.



d) Forward and Aft boundaries attached to curves. Note that upper boundary is now unattached from the deck curve.



e) Bow control vertices are adjusted to reattach the upper boundary to deck curve



f) Control vertices are adjusted to form the hull shape

Figure 6.12, the stages of attaching and manipulating curves and a surface to form a basic yacht hull.

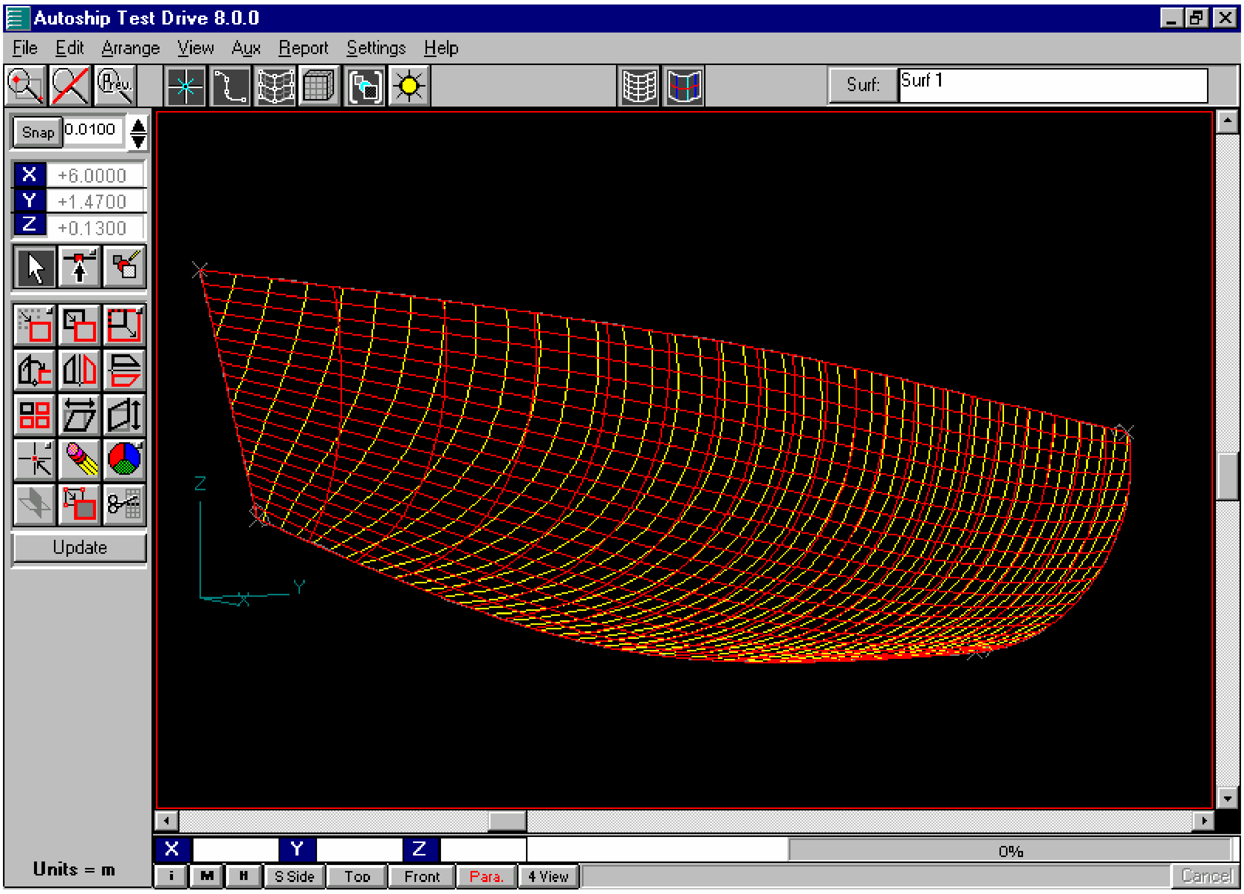


Figure 6.13, the sections and parameter lines of the yacht hull created in Figure 6.12.

The example given in Figure 6.12 is very simple, when more complex ship hulls are designed, the process becomes much more difficult. Many points may be required to generate the desired shape and the designer will require a high mental capacity to keep track of the points structure and to be able to develop strategies to implement certain modifications. Modifying the hull can be extremely time consuming and it may be necessary to revert to the modifications of individual points to achieve the required change. This may lead to resistance to decisions that require the extensive changes to the hull shape during the design process. Thus, not all avenues of the design may be investigated.

Today, naval architects have accepted the new technology that has been provided by CAD developers. However the process of designing a hull surface, from the main dimensions up, remains difficult. Naval Architecture design packages have yet to fully embrace the methods and concepts designers use when creating a hull surface.

6.4. Parametric Hull Generation

The concepts behind parametric hull design techniques were being developed just as modern freeform surface representation techniques were being discovered. The beginnings of parametric hull design can be traced back to Kuiper [8] in 1970. Kuiper extended the development of hull representation techniques, which were then based on explicit polynomials or Lewis transforms, to allow a hull form surface to be created from form parameters. Despite an early start, the hull representation techniques of the time were unable to represent hull shapes in a convenient and accurate format. As hull definition began using NURBS and other curve generation systems, the benefit of these systems to Naval Architecture was so significant that there was no immediate need to develop parametric hull generation techniques. However, as hull design packages have developed, examples of basic parametric hull generation tools have appeared. This can only demonstrate that the designer has a need for such a technique.

There are various methods that parametric hull generation systems use to generate the hull surface. Some systems generate the hull from numeric form parameters while others change the shape of the surface to match specified parameters. The hull may be created using a direct or iterative procedure.

Despite being a little used and unfamiliar technique for developing a new shape, procedures for generating hull surfaces from form parameters are implemented in some modern design packages. The best-implemented systems can be found in packages such as FORAN [9] and ShipGEN [29]. FORAN has a whole module dedicated to the parametric generation of hull surfaces. It allows quite extensive editing of form parameters and of the generation control functions themselves. ShipGEN on the other hand, allows different hull types to be parametrically modified, based on a template of the surface. Unfortunately, these templates cannot be defined by the user and it is necessary to contact the software manufactures if new templates have to be defined.

Many other hull design packages have implemented a parametric hull surface generation system. However, these are normally used to generate an initial mesh that produces a shape close to that desired. These tools are only theoretically useful, as developing an initial mesh that gives adequate control over the hull surface can be quite time consuming, especially when using a NURBS surface. However, in the long run, these tools may not save the designer time, as the distribution of control points may not be conducive to the desired surface and an even greater amount of time may have to spent adjusting the surface until an appropriate mesh of points is achieved.

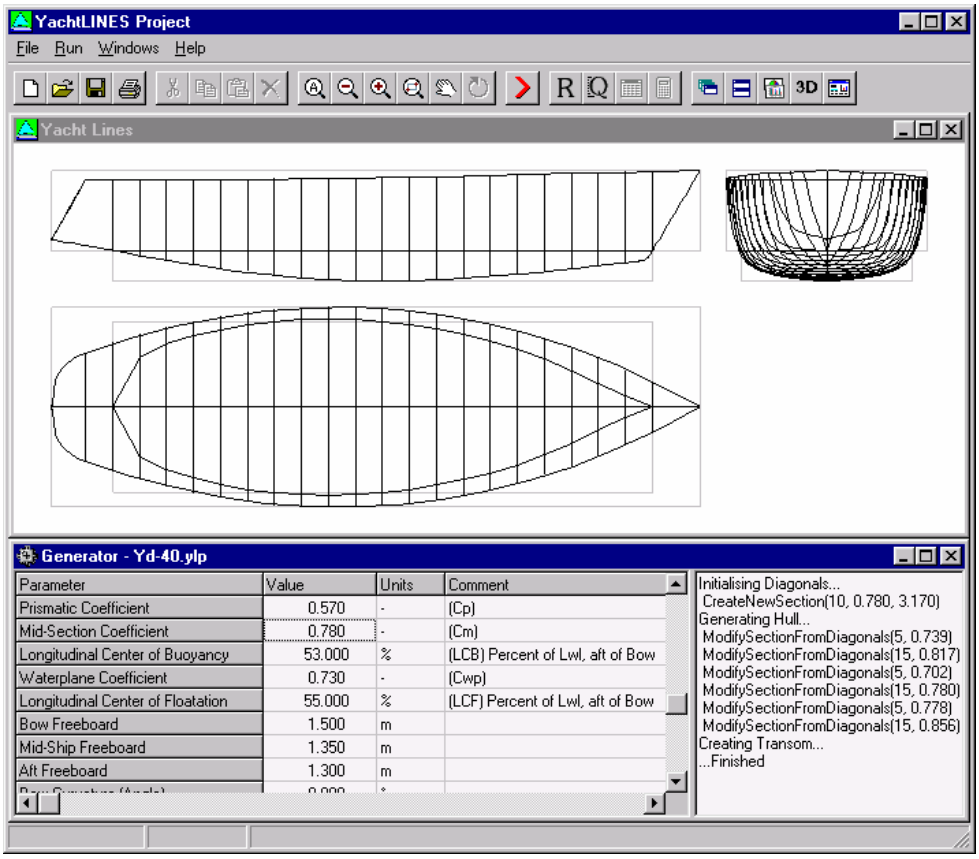


Figure 6.14, the YachtLINES interface showing the hull and the table used to enter the numerical parameters.

Judging by the number of implementations, parametric hull design is an under utilised technique for developing a hull form. However, given the fact that the development of these systems continues, it goes to show that the need for an efficient parametric hull design system still exists. In order to understand the problems associated with parametric hull generation techniques, two case studies were developed to investigate parametric hull design with respect to two different types of hull form, implemented in a modern computer software environment.

- ShipLINES: An investigation into the parametric generation of ship type hull surfaces. Surfaces, which have well defined shapes, such as flats and radii.
- YachtLINES: An investigation into the parametric generation of yacht type hull surfaces. Surfaces, which have a curved shape with no particularly special features.

YachtLINES, Figure 6.14, was developed to see if yacht hull surfaces could be generated using uniform B-spline curves. The study was used to identify how B-spline curves would behave when controlled by numerical techniques rather than manual and how a hull behaves when controlled by iterative procedures used to embody the correct hydrostatic properties. B-spline curves are used to represent the hull sections, with each point of the control polygon lying on a hull diagonal. The midship and quarter sections are used to control the shape of the hull surface. These sections are generated using an iterative procedure that varies the shape until the desired properties are met.

YachtLINES is able to successfully generate a range of hull forms. The technique is discussed in more detail in Appendix 2. Furthermore, a number of issues were also highlighted by the study: -

(a) YachtLINES can form most of the shapes applicable for the modern yacht hull. However, as modern yacht hull forms have very similar shape characteristics, this may not be a great test of the flexibility of parametric hull generation tools.

(b) Yacht hull forms are fairly easy to generate. However, it can be difficult to include accurate details of local features if the surface does not have a dense definition.

(c) Although it can be easier to generate hulls using sections, there is a tendency for control over longitudinal shape to be lost, resulting in undesirable hollows.

(d) Iterative techniques must be introduced very carefully. Badly designed iteration procedures can limit the range of shapes that can be produced in addition to the limitations that are imposed by the definition algorithms.

(e) As a GUI implementation issue, it is necessary to provide names for all parameters and it can be difficult to choose the appropriate name for parameters controlling local geometry. It is unfortunate that many local parts of the hull have different names depending on the area of Naval Architecture most familiar to you.

Using the lessons learned with YachtLINES, ShipLINES, Figure 6.15 and Figure 6.16, was developed to generate ship type surfaces using the properties associated with NURBS surface representations. By knowing how the surface will react to a certain structure in a control polygon, a mesh can be generated to obtain the desired hull shape. The properties of NURBS can be used to develop a structured control polygon directly, without the need for an iterative analytical or mathematical procedure. Difficulties arise when local shapes are put into the surface such as bulb and shaft appendages, as the transition to which must remain fair. ShipLINES also attempts to improve the technique used to enter parametric information. It was felt that a table of parameters gave no indication of what was being controlled, as small text descriptions are sometimes inadequate for communicating the exact message. In ShipLINES, diagrams are used to illustrate the part of the surface each parameter controls.

ShipLINES can successfully generate the types of ship hulls within the control of the generation technique and is detailed in Appendix 3. Furthermore, this study highlighted further issues: -

(a) Ship hull forms have very many characteristic shapes. In a parametric hull generation tool, these shapes must be controlled by parameters. The procedures used to generate these shapes can be quite complex. Even so, these may not be capable of addressing the full range of shapes that could be created if the shape was being formed manually. As shape has very many degrees of freedom, it cannot be adequately addressed by the parametric approach.

(b) It is possible to develop hull surfaces including local appendage features. However, the inclusion of these features imposes restrictions on the way the surface definition needs to be structured. Consequently, the range of flexibility in the generated hull form is reduced because it is not possible to incorporate flexibility in the shape of the main hull surface and include the local features at the same time. Increasing the number of features reduces the variety available in the basic form shape and the surface produced by the procedure converges to one characteristic type only.

(c) Procedures that can generate hull surfaces with many detailed features require a large number of parametric controls. Each must be specified, regardless of the level of detail the parameter controls, for the generation procedure to produce a surface.

(d) Simple techniques can be used in combination to develop the definition of a much more complex hull surface.

(e) The properties of NURBS surfaces can be successfully used to generate the desired hull shape.

(f) As the control polygon mesh becomes deformed, it becomes increasing difficult to control the shape of the surface. If the mesh can be maintain in a uniform arrangement, it is more obvious to see how changes effect the surface.

(g) As a GUI implementation issue, providing diagrams that indicate the area of influence of each parameter does not appear to be a successful technique. A large amount of screen space is required to implement this feature and this can obscure the view of the model.

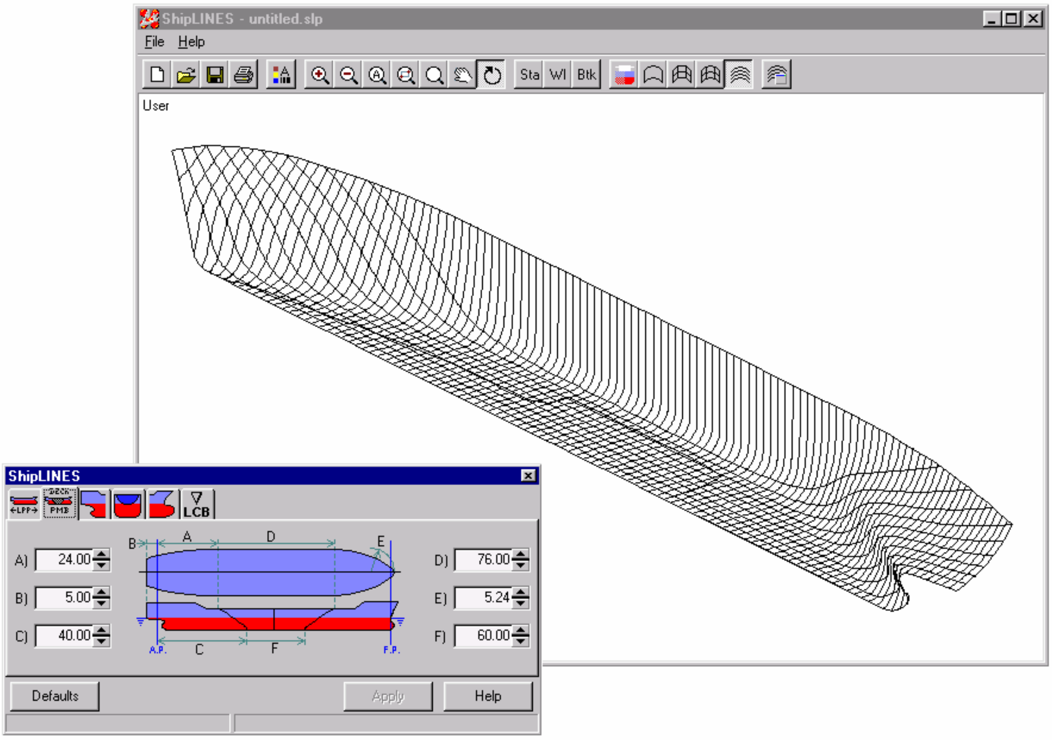


Figure 6.15, the interface to ShipLINES showing the basic hull form and numerical parameters, left.

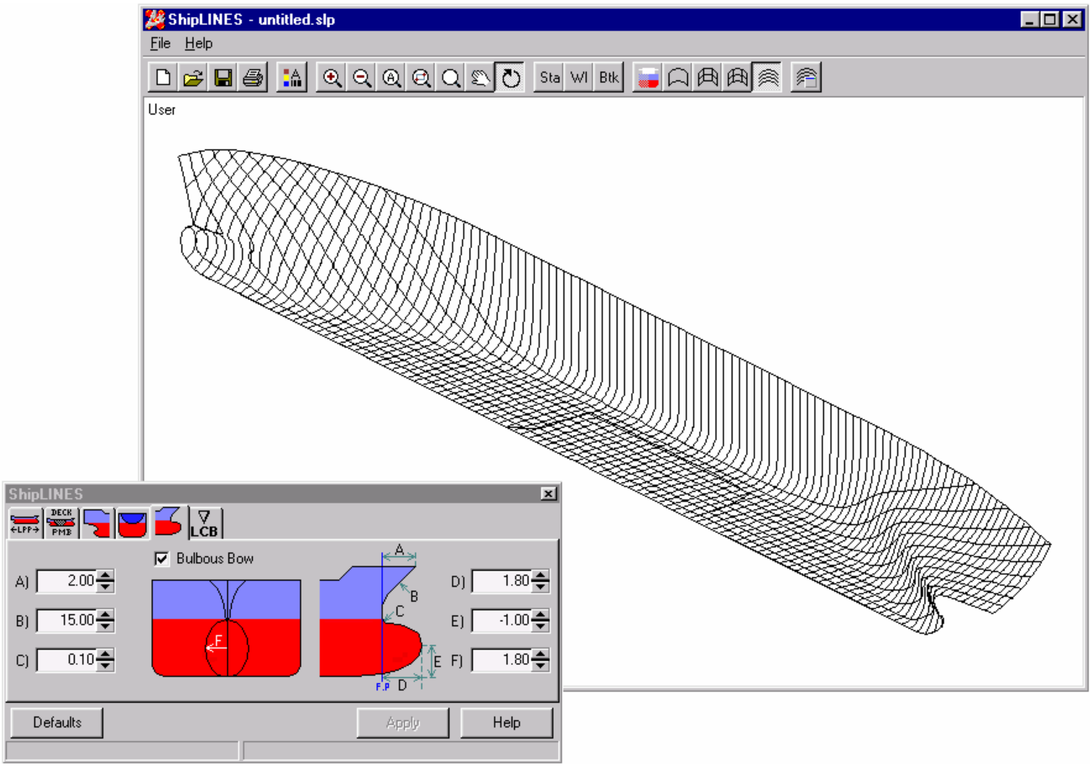


Figure 6.16, Additional features such as bulbs can be added as part of the generation procedure after the main procedure has executed.

ShipLINES and YachtLINES highlight some of the limitations of parametric hull generation techniques. One of the biggest disadvantages for these systems is that there is a complete reliance on numerical parameters to define the complete shape of the hull surface. Many different surface shapes could be produced. However, to cover all possibilities would require a large number of parameters making it difficult for a user to understand and manage the design process. TRIBON FORM [30] is an example of a tool that attempts to provide the complete parametric solution. It provides the user with the ability to change a large number of numeric parameters and curves controlling the final surface shape. However, due to the large number of options and the fact that there is no structure to the features make this package extremely complex, almost impossible to use.

An alternative solution is to keep the number of parameters low and give the designer the ability to only form the basic hull shape. Additional details can be added later using a non-parametric design tool. However, this is impractical as minor hull changes can still require a lot of work and it becomes more practical to design the whole surface using the manual process.

6.5. Performance Based Hull Generation

Performance based hull generation is a relatively new type of hull design. It is almost an extension of parametric hull design. The basic approach is that a hull is iteratively generated through the analysis of particular performance characteristics of the surface. A system would generally consist of three separate modules, Figure 6.17. The first module creates the hull surface based on some parametric definition. The second model analyses the hull characteristic of interest. The final module reviews the analysis data with respect to some criteria. If the criteria are met then the system will terminate, else the module must change the hull definition parameters until the solution is reached.

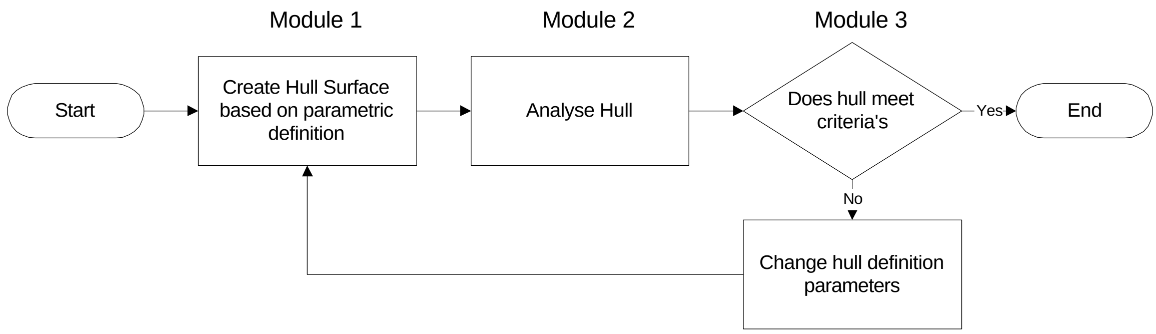


Figure 6.17, the process used to generate a hull form in
a performance based hull generation system.

As the relationship between hull definition parameters may be quite complex, it has become increasingly common to use a modern search algorithm, such as Genetic Algorithms [31]. Genetic Algorithms represent a powerful technique for finding a solution where there are a great number of parameters and many sub-optimal solutions. These techniques are an emerging technology and more details can be found in the reference.

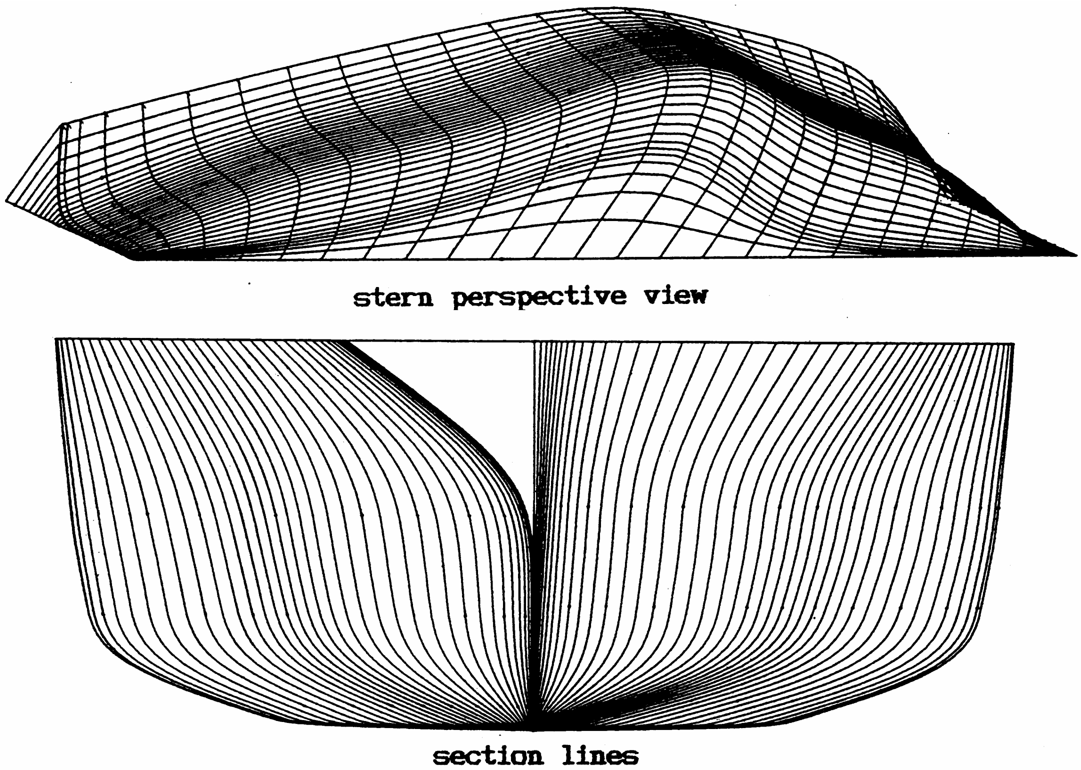


Figure 6.18, an example of a hull produced by the hull generation technique developed by Standerski [32].

Performance based design systems are best suited to specialist problems where there is a particular task to be solved. As a result, there are not many examples of this technique in Naval Architecture. Standerski [32], developed a method for deriving a single B-spline surface from numerical parameters pertaining to the geometry of the hull plus stability characteristics. Each vertex of the defining control polygon can be independently controlled. A number of constraints are applied, particularly to the boundary surface, to reduce the number of free parameters to a manageable quantity. Integral constraints are used to optimise the surface using the Lagrangian free variational form. An example hull produced by the method is shown in Figure 6.18. Given the general approach taken to develop this technique, adapting it to produce practical ship type forms would require more constraints to be added. It may be particularly difficult to add constraints to produce bulb or propeller shaft appendages. As the designer has no direct control over the shape of the surface, it is unlikely that this technique could be used for practical ship design.

Day and Doctors [33] produced a technique that used detailed analysis of resistance characteristics to generate low wash river boat hull forms given parameters and some constraints. It can be seen from the results Figure 6.19, that the hull produced is not very practical. However, given the functionality of the system, it is not surprising that results like these are produced as the surface has been analysed the basis of resistance information alone. Perhaps in the future, other modules could be added to the system to take account of other issues in the design of a hull form.

Performance based hull design systems remain a futuristic design tool. However, investigations using this technique have proved that the concept can be realised. It is particularly suited to specialist design tasks and requires a large computational capacity, something which most design offices do not have available. The performance of these systems is highly related to specification of the search goals. If it is not specified correctly the desired results may not be produced. Therefore, it is necessary to have a skilled technician or experienced operator to develop and control the generation process. Consequently, it remains in the realms of academia, where there is better availability of computing facilities and related technical personnel.

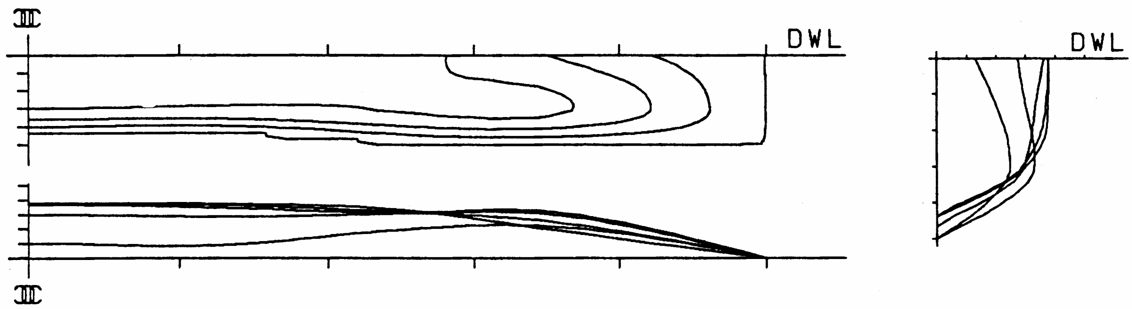


Figure 6.19, an example of a hull produced with the resistance optimisation technique of Day and Doctors. The figure shows the bow portion of a hull (the resistance technique assumes that the ship is symmetrical about midship) optimised for a Froude number of 0.6 and constrained to give a cargo space of volume 0.6L x 0.6B x 0.6T.

There are many futuristic ship design tools current under development, design tools that consider more complex issues than those found in the modules of the standard Naval Architectural design packages. However, without further improvements in the techniques used to develop the hull surface there are likely to be large bottlenecks in the efficiencies of these systems.

6.6. Discussion

It is surprising that even with modern integrated ship design packages that each hull design technique can be discussed separately when it is feasible to use the four methods together. When compared with the other facilities ship design packages provide, there has not been much development of hull design tools and the technology has largely remained the same since surface techniques, such as NURBS, had been discovered.

Generating new hull surfaces from parent hull designs is the most important technique to ship design. However, despite being an effective design philosophy, there has not been any technical development within this type of design area. The hull transformations employed today remain as they were developed, many centuries ago. With today's computing technology it is conceivable that a set of transformations could be developed that could modify the hull surface taking into account the shapes found in different locations on the hull surface without causing undesirable distortions.

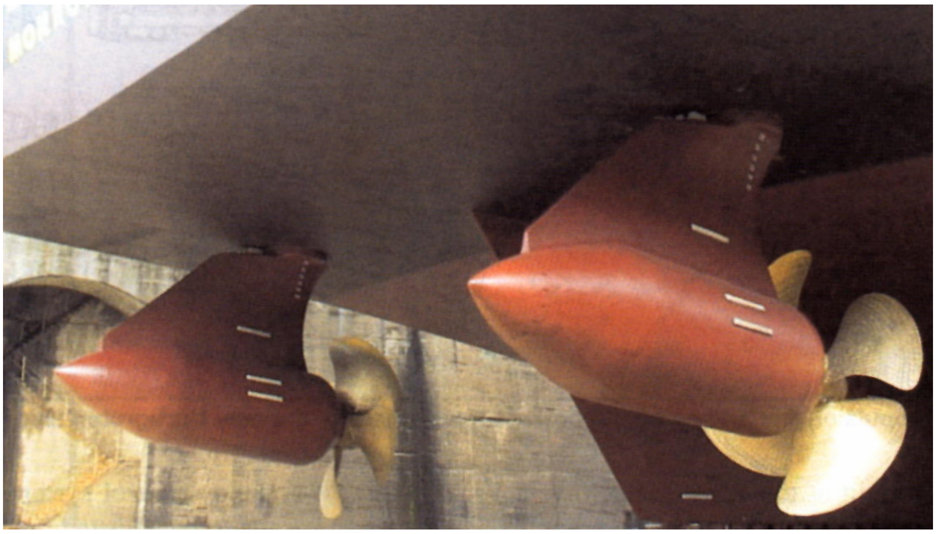


Figure 6.20, podded propulsion is becoming more common, especially on the high performance hull forms of fast ferries.

Manual creation of a new hull surface is still the only practical method of creating a new design. It requires a detailed understanding of the design problem. The designer should seek to understand the critical factors that exist in the design of a certain type of ship and create a solution that allows the design criteria to be safely achieved. The hull requires meticulous design. It is the component with control over the largest number of critical factors and the correct consideration of these factors can results in vast improvements in the ships performance. The manual hull design process gives the designer detailed and direct control over the surface, a level of control, which the other techniques cannot provide. Consequently, this technique is the only method that is practically benefiting hull design today.

The hulls of Fast ferry are an example where the manual design of the hull form is producing improvements in performance. There is a great need for rapid transportation links, especially within Europe. Resulting from detailed CFD analysis, new trends are beginning to emerge in hull form shape, with one of the key technical advances being podded propulsion. The pods, (Figure 6.20), locate the propeller in less disturbed water and, as these appendages can azimuth, give the ship much greater manoeuvrability especially during docking. Correspondingly, the shape of the hull has changed, (Figure 6.21), both from a structural point of view and to further improve the resistance.

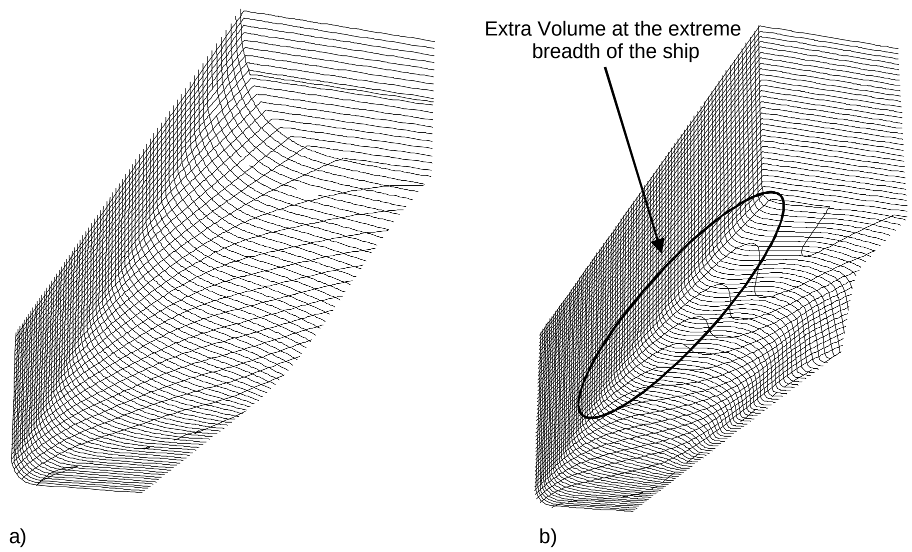


Figure 6.21, The introduction of podded propulsion has changed the development of hull forms. Traditional hull shapes (a) have U shaped sections at the stern. Modern hull forms (b) are more complex with wide transoms becoming more common, in a feature known as a 'Trim Wedge'.

Designers have always taken pleasure in a 'hands on' approach to design. However, with modern design tools, the feedback from the hull surface is more limited than when using techniques such as the half model or lines plan. Changing to a computer based design system from manual techniques must have raised difficulties for hull designers. CAD tools are developed by mathematicians and programmers who do not always appreciate the problems that hull designers face when trying to create complex shapes. The half model and Lines Plan representation techniques are both conducive to hull design. The scale can be appreciated and feedback from the shape of the model or stress in a curved spline batten can be directly felt. However, with modern computer systems, feedback can only be provided through visual senses. Even today, the only feed back methods that give a representative review of surface shape are techniques, such as Gaussian curvature. The output of these techniques is qualitative. The surface is shaded using colours to represent the different levels of curvature or by using lines drawn normal to the curve or surface, the length of which represents the amount of curvature. Translating this information into further hull modifications is difficult, as the correct amount of change must be judged by the designer through a 'trial and error' process.

Parametric hull generation techniques cannot provide the complete solution to the hull creation process. However, it is surprising that these methods do not play a greater part in hull surface design, given the amount of modification a hull may require that can be described easily with numeric parameters. However, the fact that these systems have been unable to find acceptance in naval architecture has resulted in a decline in the development of these techniques, to the extent that some parametric design modules are now being removed from naval architecture design packages. While current implementations of this technique may not be appropriate for modern hull design procedures, the lack of development means that naval architecture is loosing a tool that could make a great contribution to hull design.

To improve parametric hull generation so that it provides a useful and natural tool for the surface designer, the technique has to evolve. These techniques have often been developed in the domain of academia, where the technical merits of a system are more important than usability. It has always been up to the CAD developer to improve usability after the method has been developed. Only by considering the user in the initial design phases of a system can these methods be improved.

Performance based hull design tools are an extension of parametric hull design techniques. They allow a hull to be generated directly by optimising performance characteristics. While still in the development stage, the merits of such techniques can be appreciated. As the shipping industry becomes more competitive and profit margins get smaller, the ability to optimise the performance of the ships using business considerations, for example, will become more important.

Great advances are being made to ship design techniques and analysis tools in general. The advances are designed to improve the performance of ships and make them safer. However, this process is not complete without improvements to the methods used to design the hull surface. Modern hull form design packages present each of the tools discussed here separately. An integrated approach is required to allow designer to achieve efficient designs in a shorter time. Other industries have improved their design tools. It is about time that naval architecture improved its most important tool.

7. MODERN COMMERCIAL SOFTWARE TOOLS

Naval architecture design packages have become the tool of the modern naval architect. These tools are designed to provide an integrated environment covering all the important tasks that the naval architect commonly uses, tasks such as hull definition, compartmentation, hydrostatics, stability, weight management, powering and structural calculations. There is a wide range of packages designed to suit a range of concerns from the amateur designer to the shipyard technical office. Consequently, the procurement of a package will be based on price, features and technical support. Each suite provides design services in an individual way.

For many years, the need for mathematically accurate and flexible hull representation techniques had held back the development of effective and efficient design tools. Without an efficient method of hull definition, the development of other integrated calculation tasks could not proceed. Once freeform surface representation techniques, such as NURBS surfaces and Coons patches, had been introduced as design tools, incorporating the other tasks was simple. It was only necessary to computerise the tasks that naval architects have been doing manually for years.

Despite a range of individual solutions, the design packages are very similar in operation particularly for hull surface design. There are two key reasons for this:

a. All practical hull definition techniques use points or vertices similar to those employed for batten-and-weights. Pointing devices such as a mouse can be used to manipulate the definition points.

b. There is an element of standardisation across software packages that function within the same graphical user environment. Therefore, the use of the mouse to manipulate individual definition points becomes the accepted approach.

Previous hull representation techniques such as the half model and the lines plan are particularly beneficial for the hull designer. Both methods allow the scale of the design to be appreciated, as even at the design stage something physical is being created, i.e. a solid wooden model or lines on a plan. The designer receives physical feedback from the model shape or feels the stress in a spline batten as sharper curves are constructed. With modern CAD systems, the hull is designed in a virtual environment, with only visual feedback. User friendliness and making software packages easier to use have been one of the main problems software designers, in general, have tried to conquer. Packages that are shown to be simple and easy to use attract more customers. These improvements do not seem to have propagated into hull design packages, possibly because there is not a great level of competition between developers. On this basis, it would appear that no one has looked at how the computer interface to a modern hull development tool should operate.

Before the introduction of modern CAD packages, Kuo [6] reviewed the hull definition and design techniques of the era. The review highlighted that, even though there was a variety of different methods, none provided a practical solution to the problem. Given that a practical hull representation technique had yet to be discovered, Kuo devised the following criteria for an appropriate hull design technique:

> *A computer method can only be regarded as truly successful if it would reduce the manual effort required in applying it to a minimum. Thus, it would be highly undesirable to have a method, which divides the ship into a large number of portions and the human judgement is required to ensure successful matching of the local surfaces at all the junctions. Likewise it would also be unacceptable if the working procedure is no more than a 'computerisation' of the graphical method that inherits the drawbacks of this approach and this is particularly true when such a method needs constant human interfacing before a set of faired data are derived. To satisfy this requirement the method should preferably offer the possibility of becoming an integral part of overall design procedure so that the input information needed to generate the ship's surface may for example be derived directly from the results of analysis carried out on good hull forms based on hydrodynamic, economic and design criteria.*

While this statement refers to the methods of hull definition, it can be equally applied to the definition facilities provided by the design packages. In essence, the criteria states that a good hull design system will reduce the amount of time required to develop the hull surface and it should no longer be necessary for the designer to actively maintain the fairness of the hull. Thirty years later, do modern naval architecture packages conform to Kuo’s view of the future, by allowing the designer to concentrate on the fundamentals of hull design, rather than maintaining a fair hull shape?

	Hull Representation	Relational Geometry (RG)	Ease of Surface Creation	Ease of Surface Modification	Parametric Hull Generation	Hull Transformations	Scripting Facilities	Curvature Display	Rendering Display	Integration to other tasks	File Interfaces	Design of Interface	Overall Impression as a Hull Design Tool
Prolines 98	NURB Surfaces	No	Parametric hull Generation	Relatively Good	Initial Hull		None	No	Yes	Hydrostatics Resistance	IMS offsets	Basic but Effective on simple hulls	OK for Amateurs
ProSurf	NURB Surfaces	No	Very Difficult (Individual Surfaces)	Usable	No		None	No	No	Basic Hydrostatics		on menus	Not very usable
FastShip	NURB Surfaces	No	Very Difficult (Individual Surfaces)	Usable	Initial Hull	Affine and Hydrostatics	Difficult (PERL)	Yes (Porcupine)	Basic Shading	Hydrostatics	Standards	Too many buttons, no control over surface editing	Difficult
MultiSurf	types	Yes	(Only through RG)	Difficult	No	None	None	No	Yes	No	DXF, IGES	Too complicated, Surface editing through RG only	Difficult
MaxSurf	NURB Surfaces	Surface boundaries only	Usable	Very Good	No	Affine and Hydrostatics	None	Yes (Rendered)	Yes	Basic Hydrostatics	DXF, IGES	Basic, Very Efficient	Good
Autoship	NURB Surfaces	Yes	Usable	Good	No	Affine	None	Yes (Rendered)	Yes	Basic Hydrostatics	DXF, IGES	Not well designed	Usable
Defcar	Bézier Surfaces	Surface boundaries	Very Difficult (Individual Surfaces)	Very Good	Yes (ShipGEN)	Affine	None	Yes (Porcupine)	No	Basic Hydrostatics	DXF, IGES	Basic, Efficient	Not good for new hull designs
Napa	Cubic Patches	Yes	(Requires a hull mesh)	OK	Basic	Affine and Hydrostatics	Good	Yes (Sections)	as option	All Tasks Integrated	Standards	Old Command Line style, but very efficient.	design
Tribon	Unknown	Unknown	Parametric hull Generation	Very Difficult	Yes		None	Yes (Porcupine)	No	No		Almost impossible to use this system	Very unusable
Paramarine	NURB Surfaces	-	From File Only	-	Yes	Hydrostatics	Yes	Unknown	Yes	Hydrostatics Stability	IGES	Interface is not orientated to ship design.	Ok for concept level design of frigate hulls
Foran	Waterlines and Sections	Unknown	Offsets and Parametric	Unknown	Yes	Affine and Hydrostatics	Unknow n	Unknown	Unknown	Hydrostatics	Unknown	Appears to be reasonable efficient	available)

Table 7.1, summary of hull design packages with reference to key categories.

In order to access the state of modern hull design software, key areas will be reviewed to establish whether Kuo's criteria has been met:

1. The creation and modification of hull surfaces.
2. Hull transformation tools.
3. The user interface.
4. Parametric design tools.
5. Integration to other tasks.

Table 7.1 shows the selection of packages used to develop this review and summarises the capabilities of the packages in key performance categories. A complete review of individual packages can be found in Appendix 1.

The study shows that NURBS surfaces are the predominant representation technique for hull forms. NURBS offer efficient and flexible representation of almost any type of surface, including accurate representations of spheres, cylinders, planes and other important CAD entities. The control polygon, the structure that defines a NURBS surface, is one of the techniques most attractive features. It consists of a set of points linked together by lines or a mesh in the case of a surface. As it is part of the definition of a NURBS, it is not necessary for a software developer to provide additional techniques to allow the users to modify the entity. To provide almost complete NURBS functionality for freeform design, the software only has to display the surface and control polygon and provide a facility for the user to manipulate the control vertices.

It is possible to identify three levels of geometry for surfaces defining a ship hull shaped surface, Figure 7.1. The highest level defines the expanse of the hull surface, the main dimensions. Beneath this, there are the features of the surface, such as flats and bulb appendages. These features may be made up of simply describable shapes such as planes or cylinders. In the lowest level, the geometry of the individual vertex can be found. These elements must be manipulated to control the surface shape in both the medium and global levels.

The global level can be manipulated to control the surface shape by using geometric transformations. However, these can have an adverse effect on the medium level geometry causing undesirable deformations. Consequently, the designer must resort to modifying individual vertices to maintain accurate control over the surface. The modification of any vertex will require the surrounding vertices to be adjusted to maintain the fairness in the hull. This process results in large numbers of vertices requiring continuous modification. This can be very time consuming, as the designer must make sure that the surface continues to be fair.

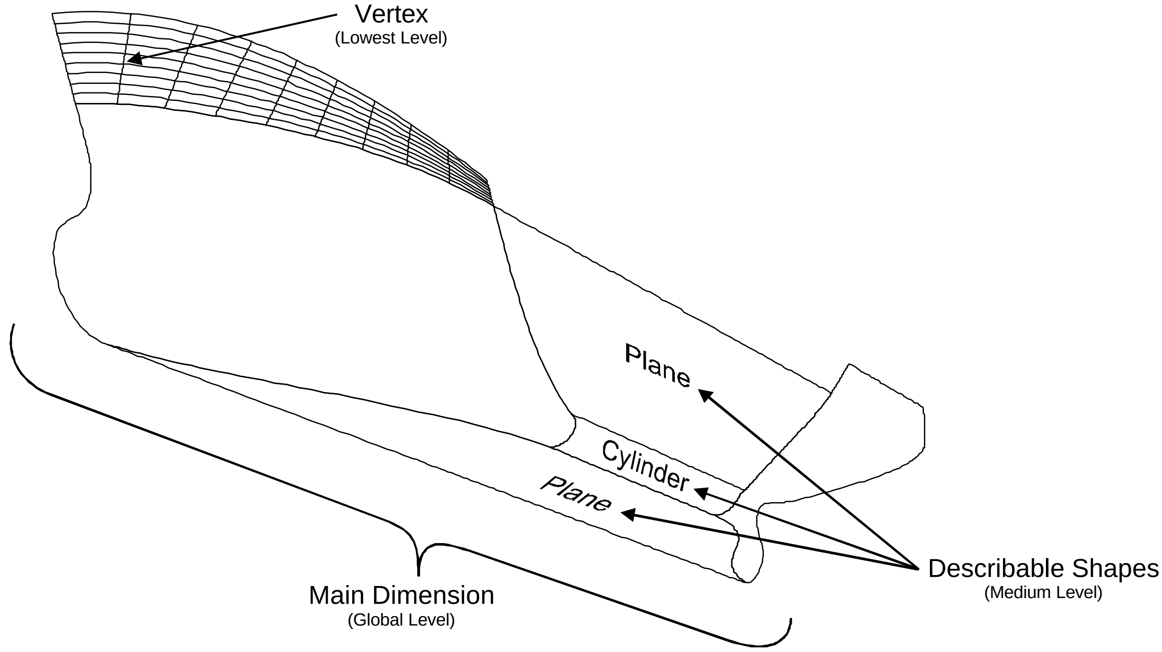


Figure 7.1, the three levels of geometric that can be found in a ship hull surface.

Another feature of NURBS surface control polygons, when used to define ship hull-type surfaces, is that they can develop into a large complex structure. Figure 7.2 is a view of DFform showing a surface and control polygon defining the bulb appendage in the hull. This surface is one of many NURBS surfaces used to define the hull shape using a multiple patch approach. When the hull surface was created, each patch had to be manually placed in position. This is achieved by either entering the numeric values describing the coordinate location of the each vertex defining the surface, or the software will provide an initially shaped surface, possibly in the form of a quarter cylinder. The user must manually drag the surface into place using each vertex until the right shape is reached. In each case, a considerable amount of skill and time is required to complete the process. New users will face frustration, as complex surfaces, where a regular control polygon mesh shape is not maintained, do not always perform how the user expects.

It is difficult for the software to provide a ready made surface shape, particularly when the hull surface is represented using a multiple patch arrangement. However, it would be possible to develop tools which aid the designer manipulate each surface. Using current software implementations, the only way to increase hull definition performance is to gain more experience using NURBS and by learning the properties which describe the behaviour of these surfaces. The current state of hull definition using NURBS surfaces cannot be considered an improvement over the more traditional techniques. Although NURBS surfaces may improve procedures at the construction stage, it would appear that the current software packages have made the design process more difficult than traditional hull development techniques. The modern design technique could be considered analogous to trying to shape the shell panels of hull while trying to fix them to the structure of the ship.

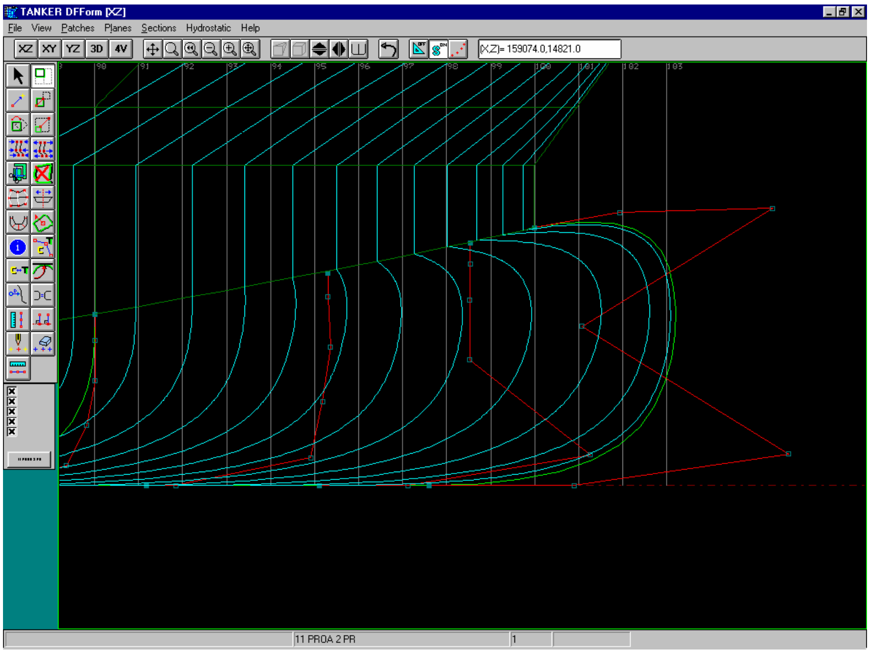


Figure 7.2, DFform [29] showing the control polygon defining the Bulb patch. The squares are the handles of each vertex. These are manipulated with the mouse to change the shape of the surface.

While the control polygon has become the standard technique for controlling the shape of the surface, many developers have a variety of methods of providing the structure to the user. It is common to be able to hide most of the control polygon so that the user can manipulate the surface using vertices from a single control column or row. Some packages such as Maxsurf [27] have taken management of the control polygon further by supplying a plan view of the mesh, allowing the user to select a section of the control polygon while editing the body plan. This is in contrast to packages such as FastShip [34] where the control polygons of all surfaces can be edited at once. The user, when attempting to select a vertex for modification, may accidentally select the vertex of another surface. Users of this package are likely to become very frustrated.

There are many ways that the surface design process can be improved. Both the Autoship [28] and NAPA [35] packages specify a procedure to be used when creating new hull surfaces. This is very helpful considering that most CAD packages start with a blank screen and wait for the user to issue first command. A procedure provides a map to indicate the first place to start.

In Autoship, relational geometry is used to begin the creation of a hull surface, (Figure 6.12). The user begins by defining points on the extents of the hull surface boundaries. The ends of curves are attached to these points and the shape adjusted until the desired boundaries of the new surface are formed. Then a surface can be attached between the curves. At this point relational geometry is no longer of use. It can only be used to control the boundaries of the surface, where there are distinct shapes. The user must resort to manipulating the internal vertices of the control polygon manually. Despite providing an incomplete solution to improving NURBS surface modification, the Autoship procedure places the surface in the correct location with a suitable initial shape. This will save the user a great deal of time.

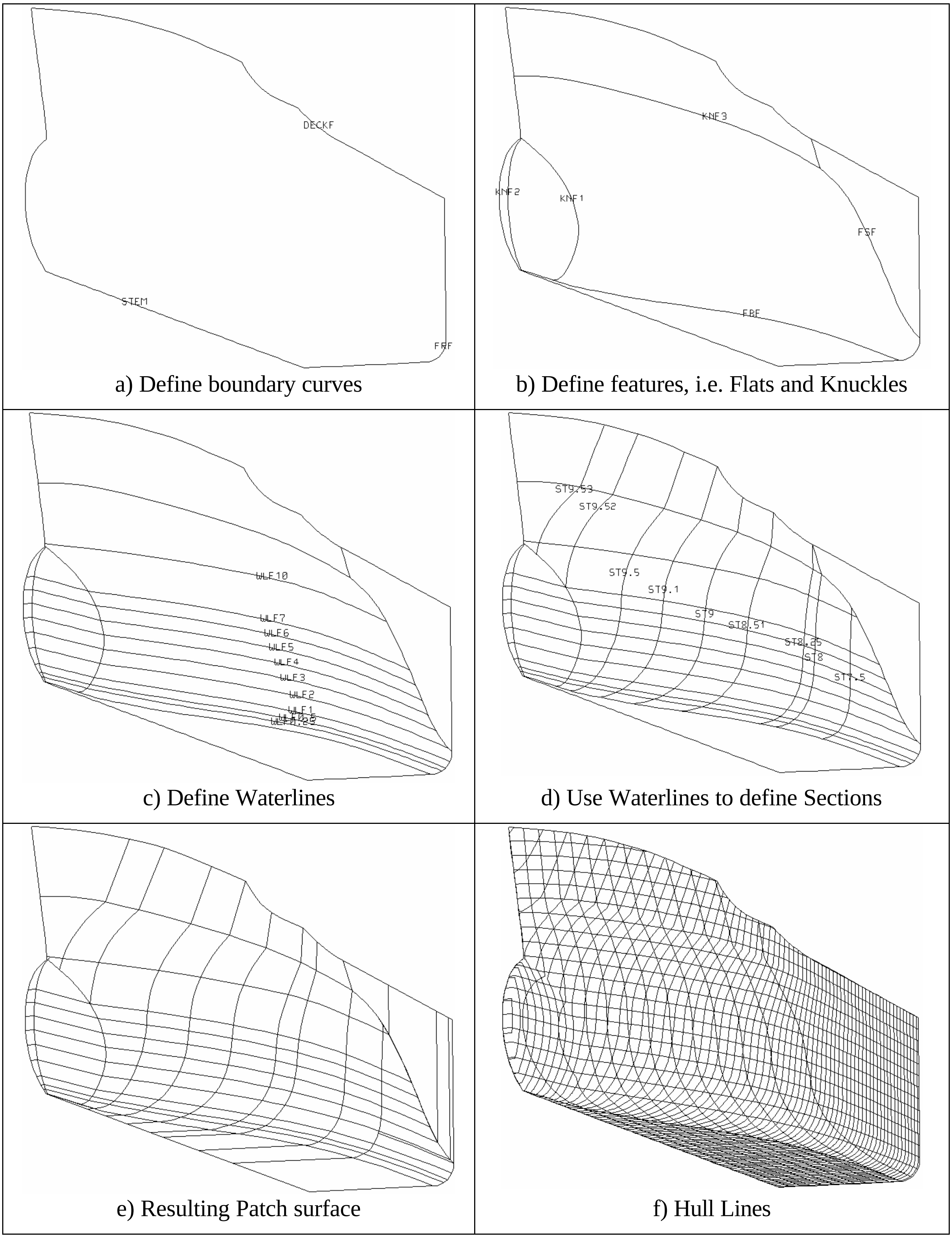
NAPA on the other hand uses cubic tensor product surfaces or Coons patches to define the hull surface. Coons patches do not have a control polygon, but are defined using the four corner vertices and derivatives. Consequently, it is not appropriate for the user to define the surface directly. The user must create a structure, similar to a control polygon, defining a freeform mesh from cubic curves, Figure 7.3. It would appear that creating a hull surface from using curves is easier than defining a NURBS surface control polygon. This would be true were it not for the three following limitation in the NAPA implementation:

1. Surfaces can only be generated if the mesh structure is fastidiously maintained. Larger changes in the hull shape normally require the mesh to be reconstructed. This can be time consuming and results in the loss of the previous hull surface. An “Undo” feature cannot be used as changing the mesh is quite an intensive process.

2. Cubic curves are not so intuitive to modify as NURBS. The technique is based on interpolation, and consequently does not have the same controllability.

3. Napa is based on a command line system. The use of a device, such as a mouse, was never planned for in the original design of the system. The mouse manipulation tools are a recent development and are still hard to use.

However, it is conceivable that a mesh system developed from curves, similar to the NAPA method, could be a more powerful design technique than the control polygon of NURBS. Despite being difficult to use for design purposes, the interpolation technique makes NAPA one of the most accurate systems for redefining existing hull forms from offset data. NURBS entities do not interpolate definition data making the redefinition of existing hull forms difficult.



a) Define boundary curves

b) Define features, i.e. Flats and Knuckles

c) Define Waterlines

d) Use Waterlines to define Sections

e) Resulting Patch surface

f) Hull Lines

Figure 7.3, the procedure for creating a hull surface from curves in NAPA.

Developing a new hull form from an existing hull surface has been shown to be an economic design technique. Surprisingly, however, the tools required to implement this process are not included in more hull design packages. Complete transformation tools that can modify hulls using both affine and hydrostatic transformations can only be found in four of the eleven reviewed packages. These functions appear to be supplied as after thoughts rather than as front line design tools. Consequently, they are found hidden away in a different part of the software. It would be more appropriate for these tools to be located in the same environment as used to create the hull surface. Then, the hull designer could benefit from these functions without having to leave the editing environment. Transformation tools were found mainly in the software packages designed for shipyard and ship design consultancies rather than those suites aimed at small craft market.

A well designed user interface is a key feature of any modern software tool. While access to the hull definition structure may be the most important feature a hull design tool must provide, good access to other features is also vitally important. The hull surface requires many other tools for managing the hull database and for controlling how the surfaces are edited. This review of these hull design tools illustrates that the style of the user interface has greater influence on usability than the method of hull definition system itself. Packages with simple, though carefully designed, user interfaces, which give the user a direct but controlled access to the surface are the easiest to use and produce the best results. Software packages, such as FastShip and Tribon [30], which provide the user with rows of buttons, overwhelming the user with information, are difficult to use. The user will spend a lot of time searching to find the right command or correct button. Tribon LINES is a particularly bad example of this.

LINES was developed from an earlier hull design system, B-LINES, which was based on a command line interface. However, when it was modernised, it acquired a graphical user interface with a large button-filled toolbar. It appears that each command was directly associated with a button on the toolbar, a standard practice when modernising a command line based system. However, each button gives no indication of whether a command is currently available. All buttons can be pressed at any time. If the user issues an invalid command, an error message is produced. However, all invalid buttons produce the same error message. This is extremely frustrating for the new user, as no information is given as to why the command cannot be used. There is a standard technique, used in user interfaces with large numbers of buttons, called Context Sensitivity. A command or button can only used when it is valid. At all other times the button would be displayed disabled or ‘greyed out’. This gives the user direct feedback and allows for an understanding of the state of the system. This technique is highly appropriate for most hull design tools where a large number of commands are required.

Parametric hull design tools have been implemented in some systems. However, most implementations are used for creating an initial hull surface only. The Defcar hull design suite has a tool, ShipGEN [29], which allows the key features of the hull surface to be modified parametrically. It appears to be quite a powerful application. However, as the basis hull template cannot be user generated, this does not allow for a great level of flexibility.

The parametric tools in NAPA allow the development of geometry under the direct control of numerical parameters. Using these tools, the user can construct their own parametric hull generation system without being restricted to routines hard coded into the software. However, the process of constructing a parametrically controlled hull definition is quite complex and given the lack of flexibility in NAPA's hull definition system, the technique is unlikely to provide satisfactory results. Moreover, the hull cannot be modified to achieve hydrostatic parameters, although, this parametric tool is still under development and this feature may be available in the future.

In contrast to all the other hull design systems, Paramarine [36], develops a ship design using numerically controlled parameters only. It is a great departure from the strategies used by other design packages. Its underlying parametric structure may be quite powerful, but the user interface is likely to slow down the development of any design. All the parameters are listed in a tree structure more commonly associated with the displaying of the "folder" structure in file managers such as the Explorer in Windows 95, (Figure 7.4). Why the developers have chosen this method is unclear, as the tree structure component was certainly not designed for this purpose and is unlikely to provide a good performance. The software has been primarily designed to develop frigate type hulls. Consequently, the complex parametric structure is unlikely to be optimal for designing normal ship hulls with great flexibility and it is unlikely to be used by any commercial ship designer.

Despite many of the hull design applications being part of a full ship design suite, many do not have direct integration to the complete range of tasks supplied by the package. In most cases, it is necessary to save the hull definition in a compatible file format before progressing to another part of the design suite. NAPA is one of the few packages that provide a completely integrated solution. The user is able to swap between any of the tasks at any time. The software uses a command line interface allowing the user to build up scripts of commands that can be used to execute routines from any part of the system. For example, it is very simple to write a script that produces a full report on the hull surface after every modification. If NAPA's hull development routines gave more freedom to the designer, it would be the most powerful ship design tool on the market.

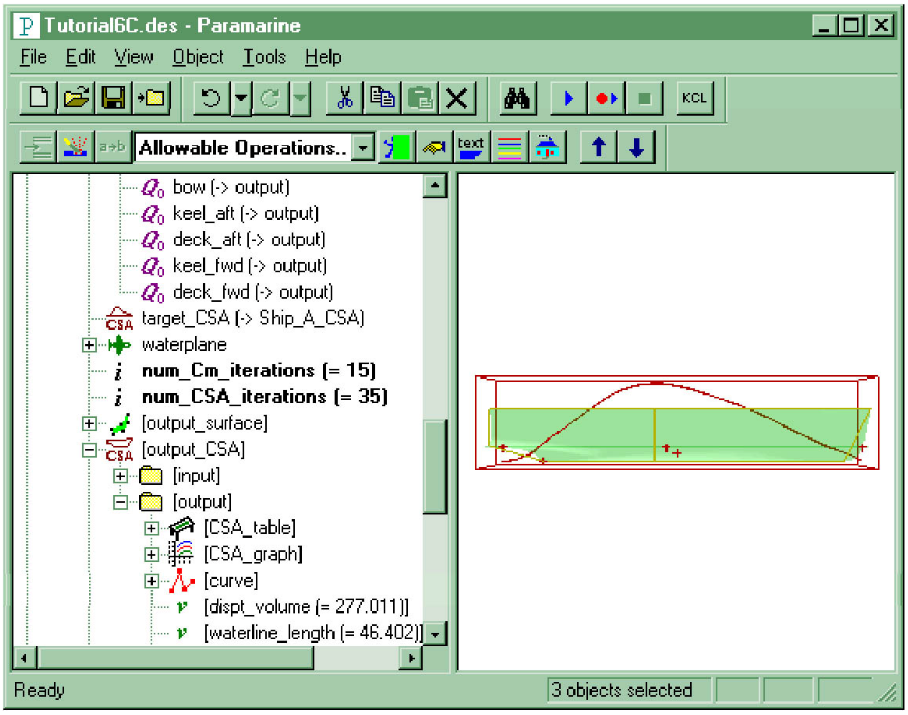


Figure 7.4, Paramarine. The unusual feature of the tree structure used to edit parameters is shown on the left.

While modern naval architecture design packages provide an adequate solution to the hull surface design problem, the approach cannot be described as efficient. Other types of software have been improved to provide tools that are more efficient for users. There may be many reasons why hull design tools do not improve the hull design process. Developers always appear to be looking to improve the technical features rather than the usability features of their packages. However, when compared to other types of design, the creation of the hull is a complex and unique task. The hull is a physically large object with intricate local details and no repeatable patterns, which can be used to simplify the definition. It is difficult to identify any other strains of engineering with similarly large design task.

Ship design has become about providing an optimal solution to the requirements of the owner while maintaining a high level safety in the design. Software packages should be fashioned to allow the efficient execution of the design process. The hull has been identified as key component, the foundation of the whole ship. It must be designed with the highest considerations for all the factors that influence its design. Factors such as: resistance, stability, structural strength, safety and efficiency. The ideal design tool would allow the hull to be created with consideration for these factors without requiring the designer to continually manage every detail of the hull shape.

8. B-SPLINES AND NURBS

When NURBS (**N**on **U**niform **R**ational **B** **S**pline) are first introduced to people, the initial reaction is one of disbelief of why this curve and surface generation technique is so popular, especially when the shape does not go through the definition vertices. This factor deters many people from learning more about NURBS. The feature is also one of the technique's more valuable assets allowing a great variety of shapes to be created using a relatively low number of parameters. The slow take up of NURBS in the early stages of development lead researchers in the field to coin an alternative acronym, **N**obody **U**nderstands **R**ational **B** **S**plines.

NURBS grew from the development of the Bézier curve. NURBS have had an interesting history, beginning as a curve generation technique known only to a select few in the academic CAD community. Developers such as Riesenfeld, Cohen, Lyche and Versprille developed the B-spline into NURBS by extending the initial formulation of the technique and by developing algorithms for changing the curve or surface definition. However, this technique would never have received the popular usage without the likes of Rogers and Adams [37] who initially presented it to the hull design community, and whose book 'Mathematical Elements of Computer Graphics' inspired a generation of graphics software developers.

NURBS are unusual functions. The implementation algorithms are very simple. However, they can be studied at a variety of levels from the knowledge required to use it, to the deeper mathematical fundamentals of the internal mechanism of the technique. Its power comes from the fact that it is able to represent many complicated shapes and simple primitives with just one formulation. This is a great advantage for CAD developers as it reduces the amount of code required to implement a full range of entities in a system. The other benefit of NURBS is in its ability in free form design. The technique has an inbuilt feature called the control polygon which provides the user with an interface to the change the shape of a curve or surface intuitively. Most CAD implementations provide the user with at least the control polygon as the primary method of modifying the shape, usually with the mouse.

It appears that NURBS are widely known and used system, however, the understanding of the details and properties of these functions is not so great. A brief description of NURBS follows to illustrate the features and properties with regards to features important to hull designers and naval architects. However, the references [37], [38], [39], [40], [41] should be reviewed for a complete appreciation of these powerful functions.

NURBS are based on the B-spline. The shape of the curve or surface is controlled by a set of definition points in a structure called the control polygon. Each point is associated with a weighting or Basis function that, in effect, attracts the shape of the curve or surface towards it Figure 8.1.

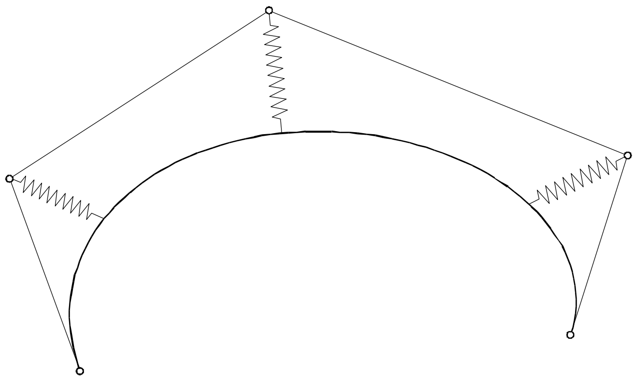


Figure 8.1, a simplistic analogy of a BSpline curve. The curve is attracted towards each vertex in almost a spring like fashion.

The B-spline is defined as follows:

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \tag{1}$$

Where: -

n is the number of vertices in the control polygon.

t is the parameter such that: $t_{min} \le t < t_{max}$

B_i is the position vectors of the control polygon.

$N_{i,k}$ represents the normalised basis functions associated with each vertex of the control polygon.

For the i^{th} normalised basis function of order k (degree k-1) the basis functions are defined by the Cox - de Boor recursive algorithm:

$$N_{i,k}(t) = \frac{(t - x_i)N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}} \tag{2}$$

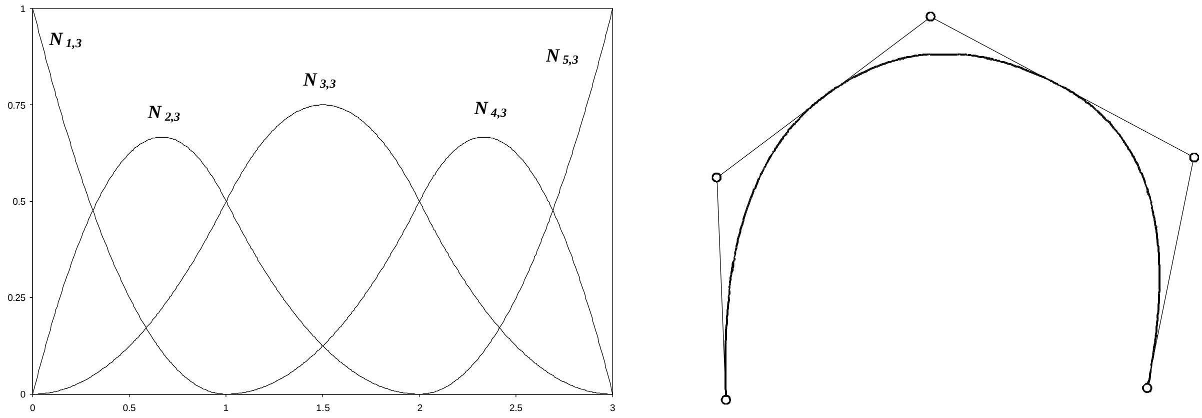
The values of x_i are elements of a knot vector satisfying the relationship $x_i \le x_{i+1}$. The convention that 0/0 = 0 is also adopted. From equation (2), it can be seen that the curve shape can be greatly affected by the knot vector. There are two types of knot vector Open and Periodic. The former can be further split into two groups known as Uniform and Non Uniform. A Uniform knot vector has evenly spaced values, a non-uniform knot vector does not. Periodic knot vectors create a periodic basis functions for which.

$$N_{i,k}(t) = N_{i-1,k}(t) = N_{i+1,k}(t) \tag{3}$$

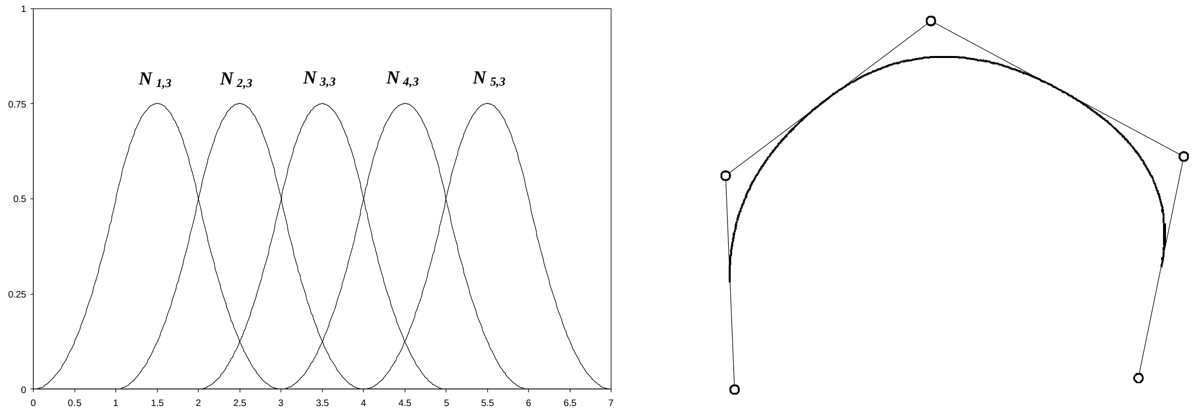
Thus, each basis function is similar with an incremental translation in t.

Open uniform knot vectors have a multiplicity of knot values at the ends of the knot vector, equal to the order of the B-spline basis functions. Knot vectors are not generated within the B-spline functions and it is normally necessary to provide an appropriate knot vector along with the control polygon when using a NURBS software library. Most knot vectors are open uniform, using integer increments starting at zero. Non-uniform vectors can be generated considering the chord distances between the vertices of the control polygon. Other non-uniform knot vectors will have to be generated using custom techniques particular to the software application.

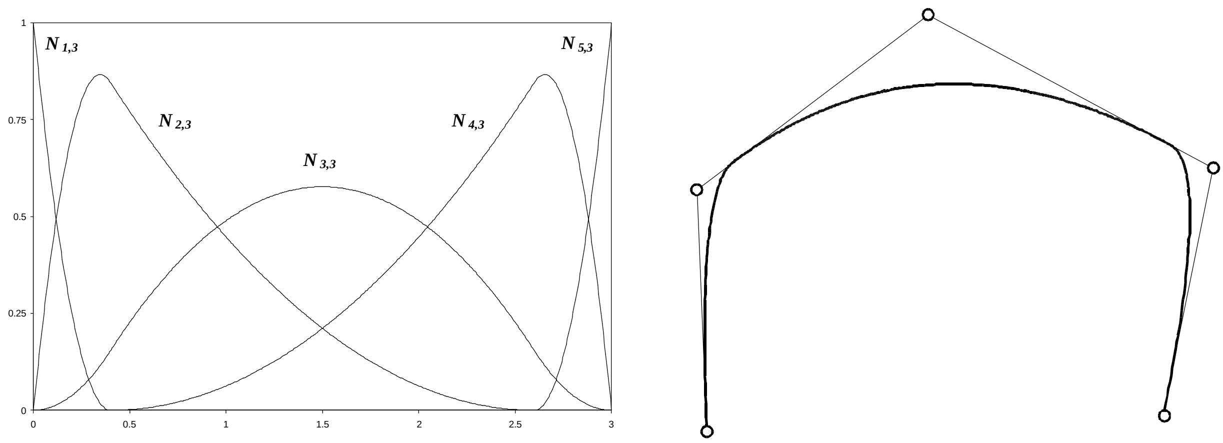
Figure 8.2 shows the effect of the knot vector on the basis functions and on a quadratic B-spline curve. Figure 8.2a shows an open uniform knot vector, the curve starts and finishes at the terminator vertices of the control polygon. A feature of quadratic B-splines is that the curve touches the each line segment of the control polygon. In contrast, Figure 8.2b shows a periodic knot vector. The curve does not connect to the terminal vertices of the control polygon. The curve has a very even shape compared to the open uniform knot vector, which is stretched so that the curve attaches to the end vertices of the control polygon. Notice that each basis function is similar, with an equal translation in t. Figure 8.2c is an example of an open non-uniform knot vector. There is a larger interval in the centre of the knot vector, which has the effect of forcing the curve away from the third vertex. In Figure 8.2d, the centre interval is smaller, attracting the curve towards the third vertex. Figure 8.2e has multiple knot values creating a cusp in the basis functions. The curve intersects with the third vertex creating a knuckle point. The knot vector in Figure 8.2f produces exactly the same curve shape. However, the cusp is located at a different value of t, resulting in a different parameterisation of the curve in Figure 8.2e.



a) Open Uniform – [0 0 0 1 2 3 3 3]

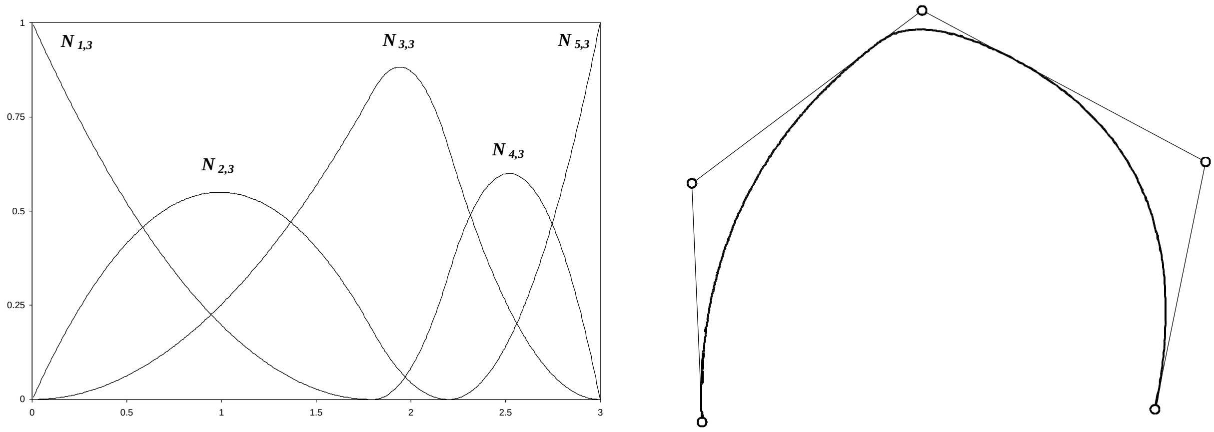


b) Periodic – [0 1 2 3 4 5 6 7]

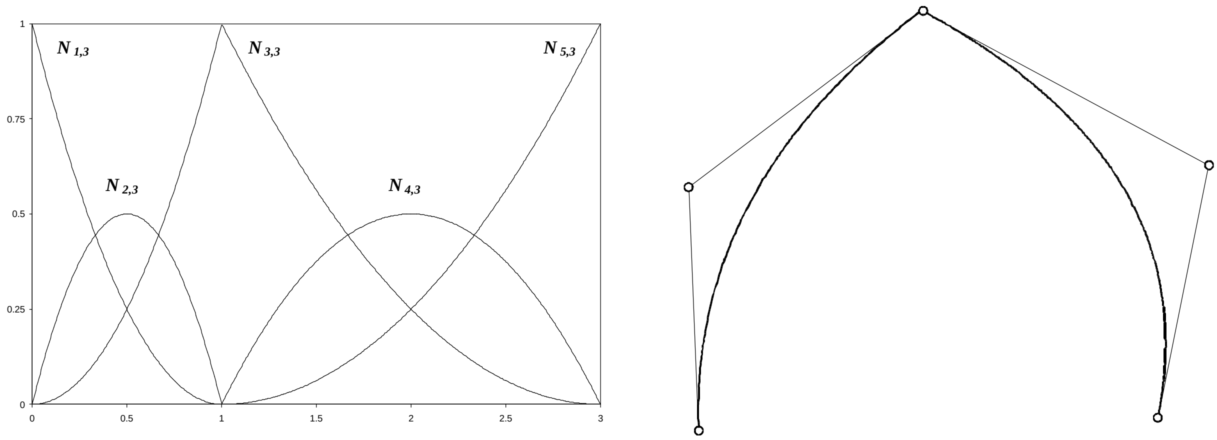


c) Non Uniform – [0 0 0 0.4 2.6 3 3 3]

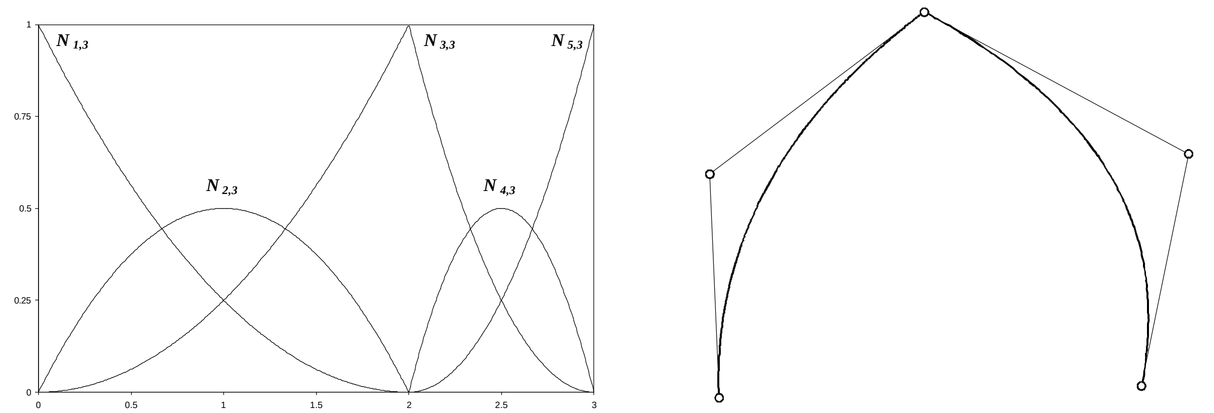
Figure 8.2, a-c, examples of Knot Vectors, resulting Basis Functions and the effect on a quadratic B-spline curve.



d) Non Uniform – [0 0 0 1.8 2.2 3 3 3]



e) Non Uniform – [0 0 0 1 1 3 3 3]



f) Non Uniform – [0 0 0 2 2 3 3 3]

Figure 8.2, d-f, examples of Knot Vectors, resulting Basis Functions
and the effect on a quadratic B-spline curve.

Rational B-splines provide a single precise form capable of representing the common analytical shapes such as lines, planes, conics, including circles, freeform curves and sculptured surfaces that are used in Computer Graphics and Computer Aided Design, Figure 8.3. Versprille was one of the first people to discuss Rational B-splines. A rational B-spline curve is the projection of a non-rational polynomial B-spline curve defined in four-dimensional homogenous coordinates, back into three-dimensional physical space. The projection back into physical space is achieved by dividing through by the homogenous coordinate, as is standard for projective transformations, yielding the rational B-spline curve, similarly for a surface. This is achieved through minor modifications to the Basis function definition to form the rational Basis functions, equation (4), which are replaced into the main evaluation function, equation (5). A more complete discussion of homogenous coordinates can be found in [37].

$$\mathrm{R}_{\mathrm{i,k}}(\mathrm{t}) = \frac{\mathrm{h_i N_{i,k}(t)}}{\sum_{\mathrm{i=1}}^{\mathrm{n+1}} \mathrm{h_i N_{i,k}(t)}} \tag{4}$$

Where h_i is the homogeneous coordinate of control point i.

$$P(t) = \sum_{i=1}^{n+1} B_i R_{i,k}(t) \tag{5}$$

NURBS curves are naturally extended to create surfaces using a simple and widely used method called the tensor product scheme. More information on tensor product surfaces can be found in [39]. The resulting surface equation becomes.

$$\mathrm{P}(\mathrm{t}) = \sum_{\mathrm{i=1}}^{\mathrm{n+1}} \sum_{\mathrm{j+1}}^{\mathrm{m+1}} \mathrm{B_{i,j}\, N_{i,k}(u)\, M_{j,l}(v)} \tag{6}$$

Where N and M are the B-spline basis functions, in the bi-parametric u and v directions respectively. The surface has two separate knot vectors and the order of the surface can be controlled on the u and v directions independently. Using rational B-splines and surfaces, shapes such as spheres can be represented, Figure 8.4.

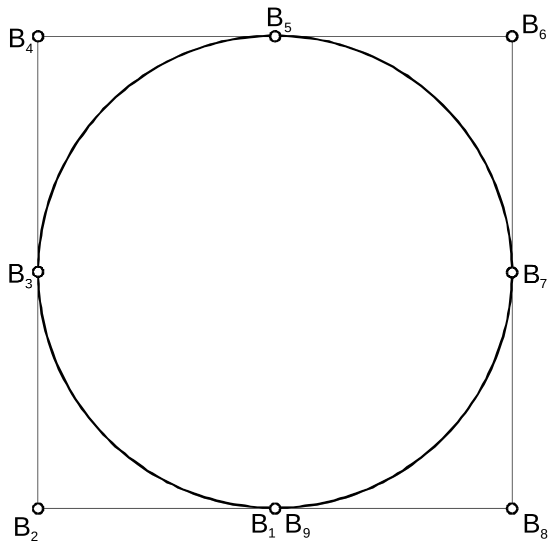


Figure 8.3, A circle can be created with the control polygon as shown,
the non uniform knot vector [0 0 0 1 1 2 2 3 3 3 4 4 4],
and homogenous coordinates [1 v2/2 1 v2/2 1 v2/2 1 v2/2 1] .

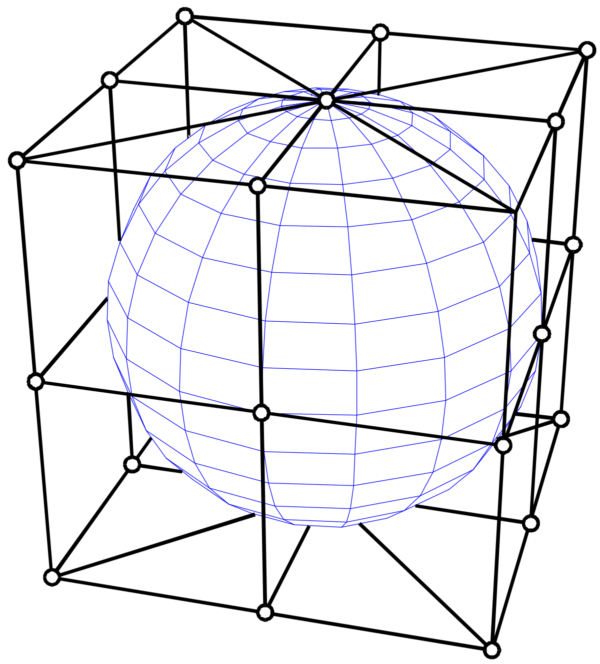


Figure 8.4, by extending the control polygon in

Figure 8.3 into
a cube structure, a spherical surface can be generated.

NURBs are associated with a number of properties that can be derived from the B-spline basis functions. Many of these properties relate to purely mathematical features, however, the final three properties define NURBS as very stable flexible freeform design tools. These properties lay the foundation for other more apparent features of NURBS such as the property of local modification, Figure 8.5. By having knowledge of these properties, a user will be able to accurately specify the control polygon to create certain shapes without the need for large adjustments. Moreover, this will result in an overall improvement in the design process.

The NURBS properties are listed below for both rational and non rational curves and can be directly applied to surfaces unless stated.

- Each basis function is positive or zero for all parameter values, i.e. $N_{i,k} = 0$.
- The sum of the basis functions for any parameter value t is one, i.e.

$$\sum_{i=1}^{n+1} N_{i,k}(t) \equiv 1 \qquad (6)$$

- Except for first order basis functions, i.e., k = 1, each basis function has precisely one maximum.
- A B-spline curve of order k (degree k - 1) is C^{k-2} continuous everywhere.
- The maximum order of the B-spline curve is equal to the number of control polygon vertices.
- A B-spline curve exhibits the variation-diminishing property. The variation-diminishing property for surfaces is currently not known.
- A B-spline curve generally follows the shape of the control polygon curve.
- A B-spline curve lies within the union of convex hulls formed by k successive control polygon vertices, except for rational B-spline curves with $h_i \leq 0$.
- A rational B-spline curve is invariant to both *projective* and *affine* transformations. A non-rational B-spline is only invariant with respect to *affine* transformations.

The control polygon and the knot vector allow a user to easily manipulate a curve or surface into the correct shape. However, hull designers frequently start with a simple shape and gradually refine it into a more complex entity. To achieve this, it is necessary to allow the designer to add or remove vertices from the control polygon and change the degree without affecting the shape of the NURBS in the process. These operations are implemented using by Oslo Algorithms.

The number of vertices in a control polygon can be changed by using Oslo Algorithms to perform knot insertion or removal. To achieve knot modification, an appropriate new knot vector is supplied with the existing NURBS definition. The algorithm will then modify the control polygon to produce the same shape using the new knot vector. Knot removal raises certain problems as the reduction in definition makes it more difficult maintain the original shape.

There are two methods of knot vector modification: A Subdivision operation doubles the values of the knot vector and the inserts further entries to maintain integral intervals. The number of new vertices in the control polygon is dependant on the original number of vertices in the control polygon and the order of the NURBS. The change in the number of vertices in the control polygon can be controlled by using Knot Insertion. However, this operation does not maintain uniformity in the knot vector. This may be a large issue for some applications using NURBS surface.

The degree of a NURBS directly controls how many basis functions affect the shape of a curve at any one point. In Figure 8.2, it can be seen that for quadratic B-spline curves, the number of nonzero basis function for any value of t is three. As the degree is increased, the number of nonzero basis functions for a value of t increases accordingly. A low degree B-spline curve closely follows the shape of the control polygon. Increasing the degree of a B-spline curve makes the shape tighter. A designer may want to change the degree of a NURBS entity without affecting the shape. Oslo Algorithms facilitate this operation also.

NURBs have several features that are of particular benefit to the hull designer. If understood, the designer can use a mental procedure developed from experience to plan the surface definition before starting, reducing the amount of time necessary to complete a design. One of the most important features to a designer is the property of local modification. A vertex only has control over the NURBS shape while its basis function remains nonzero. Therefore, the modification of one vertex does not affect the whole shape of the curve. A designer directly befits from this feature by being able to concentrate on the modification of the shape local to the vertex without having to consider any changes that would be happening elsewhere were it not for this property. The property of local modification is illustrated in Figure 8.5a.

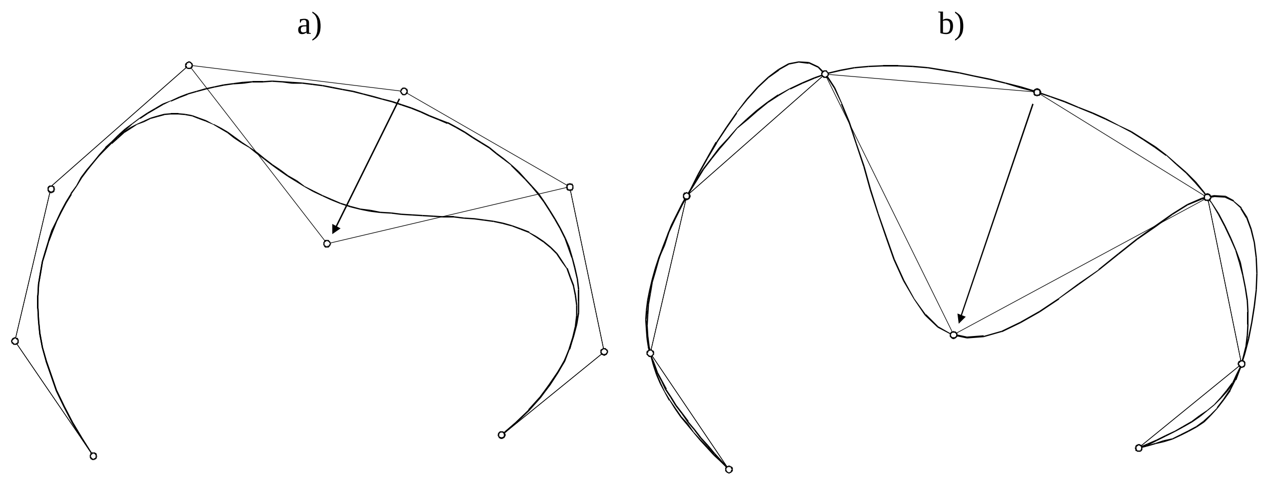
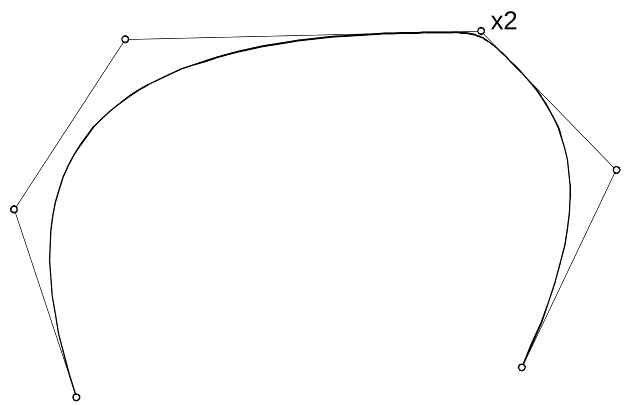
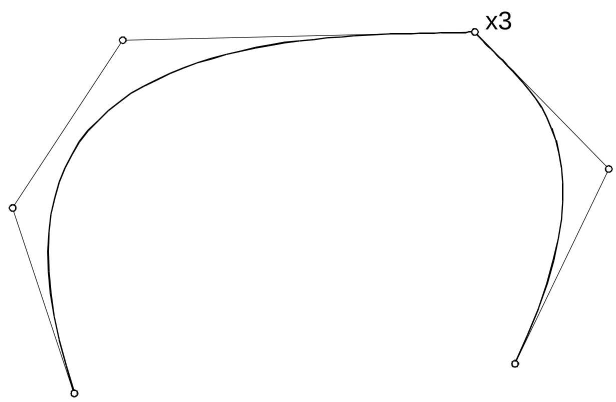


Figure 8.5, the property of local modification. When a point is moved in a B-spline curve (a), the change in geometry is only effective for a limited length of the curve. However, for a Cubic Spline, (b), which does not have this property, the relocation of one vertex changes the shape of the curve throughout the length.

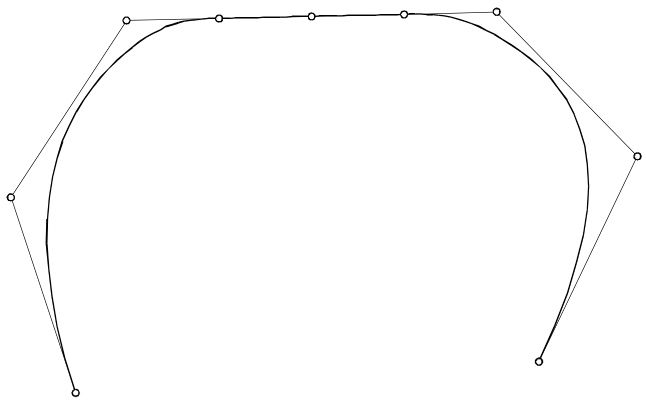
NURBs have C^{k-2} continuity, which means that the shape of the curve remains smooth up to the second derivative. However, the hull surface designer may want to introduce a sharp curve or corner into the shape. This can be achieved by duplicating values in the knot vector, Figure 8.2e for example. However, as the knot vector has a semi-global affect on shape of a NURBS this is undesirable. A more appropriate technique is to duplicate vertices in the control polygon. k-1 vertices are required to create a second order discontinuity in the curve which results in a sharp corner or knuckle point, Figure 8.6b. This feature can be used more subtlety to create second order discontinuities, tighter curves without sharp corners, Figure 8.6a. The parametric nature of NURBS allows them to function independently in the three axes of physical space. This can be used as an advantage to create discontinuities that are apparent from only certain directions, Figure 8.6d and Figure 8.6e. This can be used to generate straight line segments in free form curves. Figure 8.6 shows some examples of discontinuities in B-spline curves.



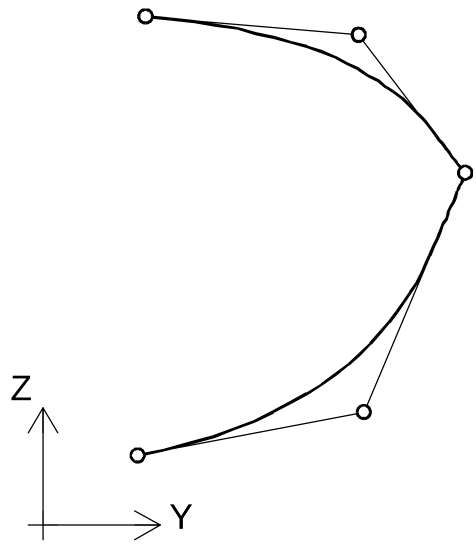
a) A double vertex in a cubic B-spline, creating a tight curve.



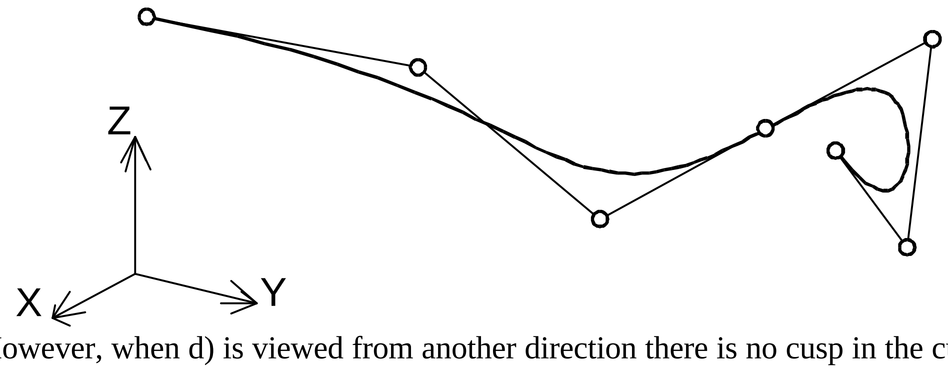
b) A triple vertex in a cubic B-spline, creating a cusp or knuckle point.



c) By aligning vertices, accurate straight segments can be created in curve.



d) A cubic B-spline with what appears to be a cusp at the centre vertex.



e) However, when d) is viewed from another direction there is no cusp in the curve.

Figure 8.6, Examples of using the polygon vertices to create discontinuities in NURBS curves.

The intuitive nature of NURBS has been shown to be the major advantage. However, there are a few important issues that the hull surface designer should be aware of when developing a new hull form.

Hull surfaces can be very complex. While it is possible for a NURBS surface to represent almost any shape, in the case of hull surface design, the shape is generally entered manually. If a designer tries to model every hull feature and appendage, difficulties are likely to arise because a large amount of data will be required and in certain areas, the definition data may be very dense. The problems associated with hull definition with NURBS were illustrated during the development of ShipLINES. It was found that NURBS surfaces give the best performance when the control polygon is maintained in a regular mesh. When local appendage shapes are modelled within the larger hull surface, the mesh can become deformed. Once the regularity of the mesh is lost, it becomes more difficult to control the surface. Figure 8.7 shows the aft end of a single screw hull generated by ShipLINES. To include the propeller bossing, the control polygon must be stretched. The mesh degenerates into diamond and triangle shapes and there are no control vertices between the bottom of the shaft appendage and amidships. Furthermore, Intuitive control of the surface around the propeller is no longer possible.

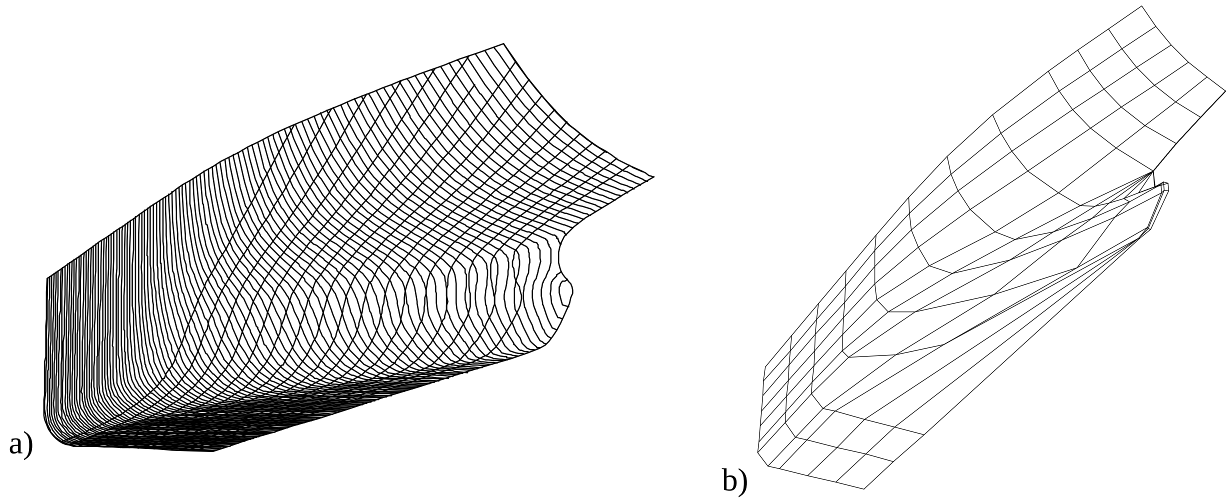


Figure 8.7, While complex shapes can be represented in hull surfaces using NURBS a), the control polygon can become deformed b)

NURBS surfaces can be used to great advantage to represent accurate ship components. However, when used for hull form design, a crucial factor is usually overlooked. NURBS are efficient freeform shape design tools. However, care must be taken to reduce the amount of double curvature prevalent in the surface. Increasing amounts of double curvature makes a hull constructed from sheet materials more difficult to form and increases the expense. It has been shown that if double curvature is actively reduced then significant savings can be made in the final cost of the ship and in construction time [42]. Most hull design packages include tools that indicate the type and amount of curvature in the hull surface. Gaussian Curvature, for example, can be presented by shading the hull surface using colour to represent the degrees of curvature. However, these can be difficult to use, as it is necessary to translate the coloured diagram into a modification of the surface geometry. This usually requires a large amount of trial and error. A better solution would be a tool that modifies the surface to reduce the double curvature, perhaps as part of a fairing process. Alternatively, a feature that gives active feedback to the designer as the surfaces is modified could be developed. Active feedback is now a feature of many computer games.

While NURBS surfaces can be efficiently used for design, it is much more difficult to construct a NURBS representation of an existing object. As NURBS do not interpolate the definition data, it is necessary to develop an analytical technique or algorithm to perform this task. Over the years, many techniques [43] have been developed that produce NURBS representations of existing shapes. However, in most cases, the control polygons generated by these techniques are not very regular. Consequently, the hull representation cannot be modified with any ease.

Despite being powerful representation techniques, software developers have had to limit NURBS functionality in hull design packages. Unless the user of a software package knows the mathematics behind NURBS, they would not notice that there are no controls for the knot vector and limited control over the homogenous coordinates. This simplification allows many of the operations that work with the surfaces to function more efficiently. Consequently, in many software packages, open uniform knot vectors are used as standard. Software developers use many other simplifications many of which have been detailed by Hollister [44].

Many applications give the user the ability to control the rational weighting factors or homogeneous coordinates of NURBS. These parameters are normally used to allow NURBs surfaces to represent primitives such as spheres and ellipsoids. However, these parameters can be used to give additional control over the shape of NURBS. To edit the weighting factors, the user must specify the values numerically. Compared to the intuitive methods used to edit the control polygon, the use of a numerical parameter requires a trial and error approach. It would appear that software developers have not considered other, more interactive, means of controlling this data.

Today, NURBS have found wide acceptance in the CAD community. Especially as they can be used to represent a large variety of shapes with one single algorithm. Consequently, NURBS are now the most popular technique for representing complex shapes when transferring information between different CAD systems or machining facilities. As a result, NURBs have been incorporated into several International and American national standards, examples of which are IGES [45] and STEP [46].

In Naval Architecture, the introduction of NURBS surfaces to hull representation formed the solution to the long search for an accurate and well behaved technique. As the hull forms the foundation of the whole ship, it is important that the design of this component is executed flexibly and efficiently. The review of modern hull design packages highlighted that most systems provide a basic implementation of NURBS functions. However, given the shapes that hull designers create, it can be said that this approach provides the designer with too much control over the surface. Creating a hull surface has now become a large and complex task with the designer forced to modify the surface at vertex level.

Most hull surfaces follow a pattern of shape dictated by the environment in which they function and the method in which they are constructed. Using the known patterns and structure of the shapes, hull design software could be improved so that the user can modify the surface using features associated with the product rather than the basic units of the definition technique. This cannot be achieved using NURBS alone. It will be necessary to develop the functionality of the software to use NURBS properties as part of the hull surface design process without requiring the designer to implement the properties manually.

9. A HIERARCHICAL APPROACH TO HULL SURFACE DEVELOPMENT

9.1. Two Different Processes of Designing the Modern Hull Form

Over the previous chapters, many different tools and methodologies used to develop the hull form surface have been considered and reviewed. The techniques have been chosen from a wide ranging period in history. However, despite the variety of different approaches that have been applied over the years, the modern tool uses two different and, presently, incompatible techniques.

The direct manipulation of a hull surface representation is the primary approach adopted by designers, with the parametric hull generation being used in special circumstances such optimisation and when there is a significant amount of concept design work. Because there is a reasonable amount of concept development in all cases of hull design, parametric hull generation tools could be expected to play a larger role at the beginning of the design. However, the fact that these tools do not feature more widely in the design process is not a result of limited capabilities. Packages such as Tribon [30] and Foran [9] are capable of generating a wide range of different hull shapes.

The development stage that occurs after the hull form has been generated by the parametric design tool is where problems begin. Once the designer has found a set of parameters that satisfy the requirements, the hull form definition must be transferred to a detailed hull surface design environment. In doing so, parametric control over the hull form is lost. Parametric design tools offer no practical intermediate design environment. Furthermore, as the hull surface representation has been generated by an internal mathematical procedure, the arrangement of the definition geometry may not be very suitable for further modification by hand. Consequently, it may be necessary to destroy much of the initial definition to incorporate the local features which the parametric hull design tool could not consider. This may involve significant changes to the generated hull form requiring many hours of rework and, in that case, the time spent using the parametric design tools cannot be justified. Consequently, designers do not take advantage of any of the benefits that parametric design tools offer and start by developing the surface in the detailed design environment. In the cases where parametric design tools are used, the best practice is to redefine the hull surface so that it best suits the detailed design environment.

The gap between parametric and detailed hull design tools is growing increasingly wider. Designers using concept design tools are aware of these problems and are unhappy about the way all of the design information is lost as the hull transfers from one stage to the next. As new approaches like product modelling tools are being introduced, it appears that there is now a greater emphasis on modelling detailed information rather than providing effective tools to improve the way the designer develops a vessel.

The design process of a ship has changed very little over the years and it is only civilisation rather than changes in the sea environment that has created the greater demand for information. If this trend continues without any changes in the tools designers use, the direct manual manipulation of definition data is going to become an increasing bottleneck in the design process. Modern computer systems can handle the processing of data and information very easily. It is only necessary to provide a procedure and the structure under which the information must be handled. This approach could be applied to hull design to reduce the amount of unnecessary manual manipulation. Hull forms have a very distinct shape structure that is understood very well. If this structure can be understood by a computer it can aid the user in developing the surface by creating the definition information where it is required. Consequently, hull design tools can become more process orientated rather than information orientated.

This change in development cannot be made without understanding the fundamental hull design process. It is an iterative procedure that gradually builds up information from a basic concept to a detailed design. Present design tools do not reflect this process. Hull surface generation tools are restricted to a fixed set of numerical parameters and manually controlled hull surface design tools require the complete definition to be provided at a level of high accuracy to produce a surface of basic quality. While these tools do not suit the design process very closely, they fulfil the tasks that they were designed to perform very well. Consequently, these tools cannot be thrown away as so many systems, particularly production, rely on the representation developed by these detailed design tools. In progressing to process orientated design tools, the approach taken by the two techniques must be reviewed and adapted to allow the gap in functionality to be bridged.

As the hull surface representations used by these tools are so effective and important to other parts of the ship design process, the problem becomes one of developing better interfaces to these representations. The design of the interfaces must be based around the way that the hull design process progresses. Consequently, the design process has to be the primary consideration in the development of a conceptual approach for an effective hull form design interface between the user and the surface representation.

9.2. Identifying the Foundation of a Tool that can adapt to the Design Process

In the development of an interface to aid the designer's effort, the amount and type of information that is available at each stage needs to be an important consideration. This factor has been the major cause of the gap between concept and detailed design tools. Concept design tools have a limit to the amount of information that can be used to control the hull form, dictated by the maximum number of parameters. Detailed design tools require a considerable amount of information to be supplied before a suitable hull form can be developed. Hence the need for a significant amount of iteration using this information until the right solution is found.

The design tools presently used are only valid at two particular points in the design process. The parametric hull design tool will only function effectively once the number of known design parameters is equal to the number provided by the technique. The detailed hull surface design tool will only function effectively once the detailed form of the hull surface shape is known to a sufficient level. However, the design process is one that starts with no information and gradually increases knowledge until a solution to the requirements is reached. In modern hull design, where only the detailed hull design tools are considered practical, it is the designer's responsibility to manage the shortfall of knowledge between what is known and what is required by the tools. Developers of these tools have made no attempt to aid the designer in this process.

While both groups of tools do not provide an effective solution throughout the design process, they each provide a solution to two specific problems. In the early stages of design the main task is to identify the principle dimensions of the hull form. As a result, design is more parametrically orientated. Shape and local features of the hull form are less important. However, there is a great need to be able to perform very important analysis to determine the viability of the vessel. Many of these calculations can only be performed with a hull form, hence the present desire for parametric hull form generation.

As design progresses, the principle dimensions become of less concern. The design process is no longer orientated to determining these parameters once they have been fixed. The consideration of detailed features becomes more important. The shape of various proportions and local features affect the resistance characteristics or the look of the vessel. Here the detailed design tools are better suited for subtle modification to the shape of the hull form as it is impractical to make these changes parametrically. Even so, modern tools require these changes to be implemented numerically. Each number could be considered a parameter, in geometric terms. This is in contrast to the form parameters considered by the generation tools. However, to control the shape of the surface, unlike the generation tools, the value that each parameter takes is not as important as the relationship that the parameter (control vertex) has with neighbouring parameters. Based on the functionality provided by the two approaches to hull design, the following points can be highlighted:

- Parameters are best for controlling dimensions
- Direct form or geometric definition is best for controlling shape

In present hull design tool implementations, these two functional approaches are incompatible yet both approaches would benefit from using the other. Parametric hull generation tools have a great problem controlling shape. Shape has many dimensions. Consequently, to make a more flexible hull generation tool many parameters are required. Conversely, modifying dimensions in a detailed hull design tool is a major problem as it is necessary to manually modify a significant proportion of the surface definition.

In the development of a hull form surface design interface, some factors would have to be considered. The approach would have to be capable of being able to control the hull form parametrically, yet also be able to modify the shape of the form features. This is better handled by a definition approach. Furthermore, the technique would have to consider that, initially, the detailed definition of shape would be difficult and it would either provide a default shape or more parameters. Yet as the design progresses, shape will become an increasingly more predominant factor in the design. Consequently, some parameters may no longer be adequate to control the shape of the hull form and should no longer be available. However, a major change to the principle dimensions of the vessel is always a realistic possibility. For example, after a detailed iteration, it may be found that the breadth of the vessel is no longer adequate to support the requirements. Parametric facilities should be maintained, but they could be used to modify manually generated definition rather than generate the hull form. Ultimately, this tool would fit the design process very well, (Figure 9.1). Adaptation to the level of information in the design process is maintained and the tool would be able to go backwards in the design process to accommodate unforeseen changes as well as forward.

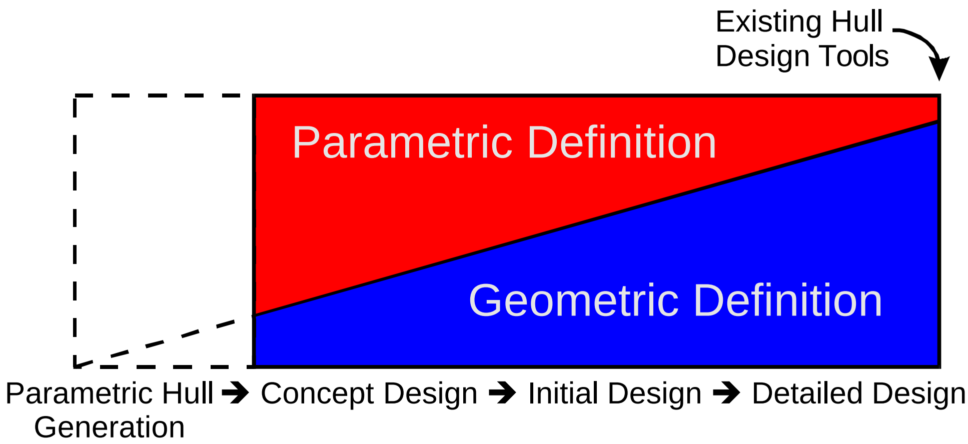


Figure 9.1, a flexible approach to hull design using more parametric definition at the concept stage progressing to more geometric definition in the detailed phases. Consequently, a hull surface development tool can adapt to the design process is developed.

9.3. Interfacing Parametric and Detailed Hull Form Surface Design Together

The identification that parametric hull generation and detailed hull surface design tools have mutually beneficial approaches is not a great leap and it raises the question as to why there has been hardly any attempt to combine these two approaches before. (It could be argued that the Form and FormG modules of Tribon and Foran allow the user to control the definition of the hull surface. However, rather than allowing the user to manipulate the detailed surface definition directly, these tools only allow the user to manipulate intermediate geometric results in the hull form generation process.) It would appear that the difficulties arise because in their present form, both methodologies cannot be interface together. Consequently, a more detailed review of each technique is required, with respect to implementing the opposing approach, to identify the difficulties and to propose some solutions.

9.3.1. Implementing parametric control in detailed hull design tools

Detailed hull design tools function by allowing the user direct access to the definition. This approach provides the maximum possible level of flexibility, allowing the user to create whatever shape is desired. This approach, as previously discussed, also restricts the design process by requiring the designer to provide so much detailed information. As detailed hull design tools must also cover the development of surfaces that may not be hull forms, implementing aspects of functionality specifically developed for hull form design would limit the range of the surface design tool. Present programmer doctrine identifies the generic approach as the best, even if this compromises the primary functionality of the tool. Consequently, in the definition of a hull form representation, present tools have no implicit knowledge that the surface represents a hull form.

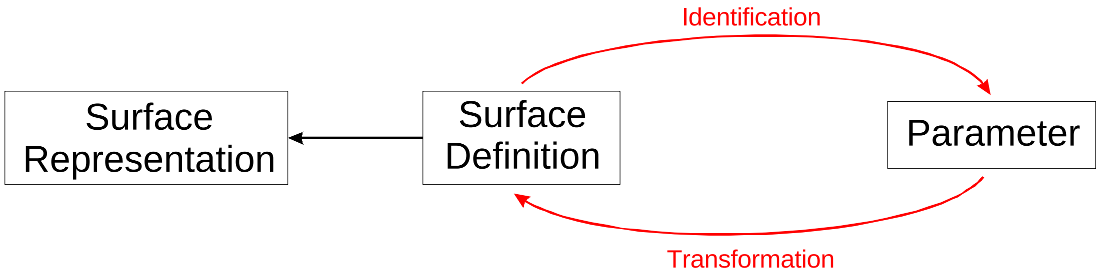


Figure 9.2, parameters can be identified from the surface definition which, with the appropriate transformation, could be used to change the dimensions of the surface

To incorporate implicit knowledge, the system would be required to analyse the shape in some way, to identify the features of the hull form. This would require some quite extensive development, particularly to cover cases where there is an unorthodox definition. An alternative approach would be to use a particular definition structure to create the hull form. Instead of directly analysing the surface representation, the tool could analyse the definition to identify the features and dimensions. Some existing tools, such as Napa [35], already use a structure to define the hull form and it would only take a small amount of additional development to obtain parametric information. Once the structure of the hull form is available, it becomes very easy to identify and measure the parameters of the hull form. The next step would be to use this information to implement transformations, thus feeding back into the definition of the hull surface, (Figure 9.2). Presently, these modifications could be implemented using the standard transformations available in modern tools. However, as previously discussed, some transformations cause undesirable deformation. Consequently, a more effective approach should be considered.

9.3.2. Implementing Detailed Definition Control in Parametric Generation Tools

Unlike the detailed approach, the parametric approach relies entirely on information to develop the shape of the hull form. This information resides entirely within the mathematical procedures that are used to develop the hull form representation. However, it is the mathematical procedures that are the major reason why present parametric hull generation technique will never achieve similar capabilities to the detailed design tools. Parametric generation tools must develop the whole hull surface definition from the relatively low number of parameters provided by the user. This information must be used to form all the dimensions, shapes and volumetric properties of the hull surface. Generating the correct dimensions and volumetric properties is relatively straight forward. However, once parameters are introduced to control shape, the tool becomes less useful because the formulation becomes fixed to one type of hull form. Furthermore, as it is difficult to control shape with numerical parameters, very little variety can be achieved. Developing mathematical procedures to cover a variety of different hull form is exceedingly difficult and it becomes necessary to consider a different formulation for each type. Further complications are introduced when it is desirable to use a subset of the parameters to define the surface.

As the major problem with the mathematical approach is that the functions are hard-coded into the software, some developers have provided facilities allowing users to develop their own generation procedure through the use of scripting or geometrical relationships between parameters and definition data. However, considering the amount of time it takes an experienced developer to create these procedures, the user defined approach cannot be considered a viable alternative to detailed definition of the hull surface. Given the difficulties in trying to develop a hull generation procedure that adapts to the varying amounts of parametric information, the development of a technique that could in addition cope with changes due to manual interaction with the hull definition is going to be impossible.

The rigorous mathematical relationship between the parameters and the generated geometry, (Figure 9.3a), is the primary factor causing the inflexibility within hull generation tools. Consequently, as a result of this limitation, parametric hull generation tools developed using the traditional approach will never yield a practical and flexible solution to the design problem. However, if the limitation were removed, by having parameters defined on the basis of the definition geometry, (Figure 9.3b), and modifications to the parameters altered the geometry by invoking a transformation function, a system could be developed that allowed the use of both manual manipulation and parametric modification to change the design of a hull form surface. The tool could make more effective use of manual manipulation to control the shape of features not suited to parametric control. This would reduce the need to have so many different parameters and make parametric control of the hull form more concise and a lot easier to understand.

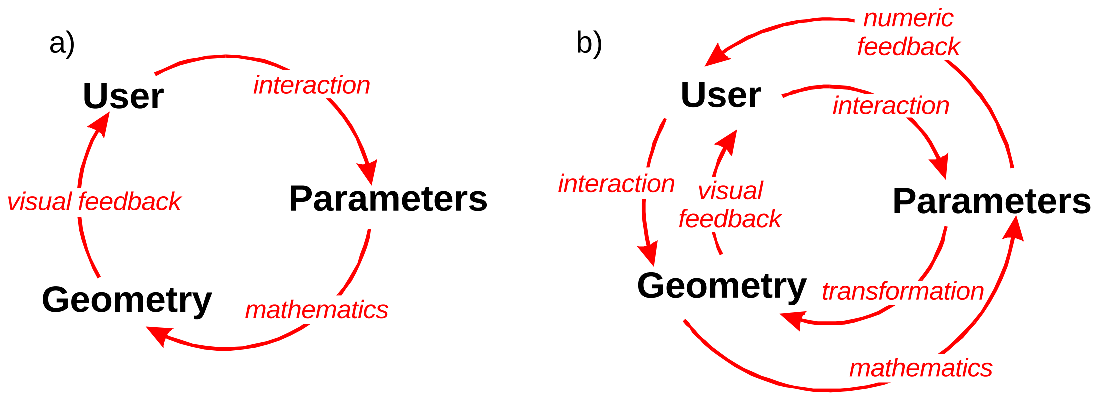


Figure 9.3, a) existing approach to parametric hull generation,
b) a desirable approach to hull surface design.

The reviews have identified that there is a plausible method that may allow the two approached to be integrated. In combination, there is a compatible solution which involves making more effective use of a structured approach in the definition of the hull form. The structure is used to identify, and hence define, numerical parameters which, under modification, can invoke a transformation of the hull form definition structure.

9.4. Building a Hierarchical Solution to Combine Parametric and Manual Manipulation Approaches

An improved structured definition approach can form the basis for a more effective solution. However, it does not yet consider the practical details of hull form definition. In its present state, the parameters and structure mainly consider the definition of the global dimensions and control of shape on a general level. However, the structure needs to adequately allow all the smaller local features of the hull form to be defined and still allow transformation modifications to be invoked by changes in parameters without causing undesirable deformation.

The consideration of local features has always been a problem in the hull design process. In the case of the manually defined surface, local features greatly increase the density of the definition, making it more difficult to implement changes to the hull surface. In the case of parametric hull generation tools, the incorporation of local features, as ShipLINES demonstrates, can have a detrimental effect on the success of the generation process altogether. However, while both cases use completely different approaches to produce the detailed definition, they use the same strategy. Both techniques develop the whole hull form definition in one step. Consequently, the dimensions, general shape and all the local features must be considered at the same time. This creates an incredibly complex design process and it can be seen why tasks such as fairing are so involved.

While this approach to the design process is perhaps a legacy of traditional techniques, modern computer technology can provide a more effective solution by allowing the hull form to be developed in several stages of simpler definition. Rather than developing the hull form using one definition cycle, a divide-and-conquer approach could be used to build up the definition for the actual surface representation in stages. It could start by considering the basic but normally the most important details such as the shape of the midship section and in successive cycles include more detailed definition. This approach would allow the tool to take control of the arrangement of the surface definition rather than rely on the management and skill of the user as present techniques require. Local features could be considered as sub-components of the main hull surface. The method could decide on how to introduce these features to the main surface, in the case of NURBS deciding when and where to perform refinement of the control polygon through knot insertion. As a result, the definition of the main hull surface would become simple and easier as it would not need to consider how the local features are going to be added.

The divide-and-conquer approach also offers other opportunities in the definition process. Because the system builds up the definition instead of processing it in one batch, it has the opportunity to identify individual features in the hull form shape. In identifying features, it is possible to allow parametric definition to be combined with manual definition. The technique could be used to form any definition that has not been already provided by using a default template controlled by parametric information and definition that has been provided by the user. In this way, the surface development process can accommodate the shift of the design process from parametric to geometric control as it moves from conceptual to detailed design.

In this act of separating the surface definition process into stages, the technique of generating and designing a practical hull form becomes more hierarchical. It now consists of components and sub-components of information and definition, (Figure 9.4). This conceptual approach alone offers an exceedingly wide range of possibilities. Instead of using a single process to create the hull representation, the technique can use a more analytical approach in the sense of being able to review, change and control parts of the definition itself. It can oversee and carry out tasks which present users must perform manually, perhaps on a repetitive basis for each design iteration. Furthermore, as the technique is managing the definition process, it develops most of the detailed definition, using the hierarchy, that has to be manually manipulated by the designer today.

Consequently, if there is a requirement for a large change in the shape of the hull surface, the designer has only to modify the definition at the relevant level in the hierarchy and all subsequent levels would update on the basis of this change. It also becomes very easy to reverse a change after it has been made. An action that may require many levels of "undo" in present hull development tools.

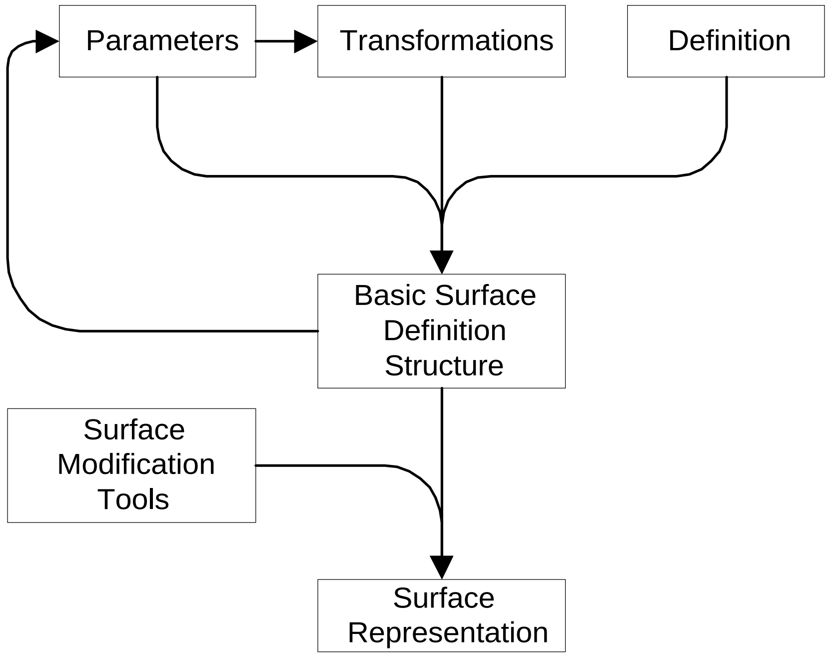


Figure 9.4, the development of hierarchical hull design process allows parameters, manual definition, transformation and surface modification tools to be combined in one integrated process.

The definition stages in the hierarchical hull development process provide an opportunity to introduce tools that would not presently be considered. Because the present definition process requires so much time and concentration, the designer is always wary of functions that may ruin it in one application. Consequently, surface modification tools such as warping and bending would not usually be considered practical or safe. Furthermore, the process can make better use of some of the representations qualities. In the case of NURBS, it would be possible to make more effective use of the properties which indicate how to define certain shapes such as planar and corner shapes. As a result, the hull definition process could become more tool based instead of manipulation based, much like the approach in the development of a half model.

Present detailed hull design tools rely on a definition process driven by the surface representation. This means that the structure the user controls to create the surface is one which is functionally required by the surface, not the designer or the design process. In the development of a new approach to a hull design tool that has layers of separation from the definition of the resulting surface representation, the definition structure can be designed to address the concepts, shapes and features incorporated in the hull form. Consequently, the amount of translation of the idea, from the shape of the hull form to implementation in the definition, could be vastly reduced. In combining the parametric approach and manual manipulated definition, the concept of using parameters to control dimensions and definition to control shape and character is a very realistic possibility if a hierarchical approach to practical hull form surface design is found to be feasible.

10. A HULL SURFACE DEFINITION SOLUTION BASED ON FORM TOPOLOGY AND GEOMETRIC CONSTRAINTS

10.1. Searching for an Appropriate Solution

The review of past and present hull design techniques has shown that a variety of solutions have been found. However, in recent times, development tools have become very similar, relying more on the capabilities of the mathematical representation techniques than providing a design solution. The reliance of standard surface representation technique requires the designer to specify a great deal of definition information to create an adequate representation. As a result, these tools do not adapt to the ship design process. On the contrary, they positively hamper efficient design and optimisation.

A hierarchical approach to hull surface design offers a potential solution by breaking the definition up into smaller and simpler manageable pieces. The approach forms an interface to the hull surface representation. Consequently, any manual manipulation of the surface does not necessarily need to be performed directly on the hull representation definition. Furthermore, the hull definition structure used by the technique does not have to be driven by the surface representation. In forming an interface between the designer and the surface, the technique can provide a definition strategy more suited to hull form design. However, while it is quite possible to develop a very abstract definition technique in comparison to those used today, the present methodologies, of manipulating control vertex definition points, are practical and well understood by the users. In developing a new definition technique for a hierarchical design process, best solution is to identify the essence of how designers would like to create the hull form surface and implement using strategies that have been found to be well proven.

10.2. The Solution to Hull Design in Modern Tools: Constrain the Designer!

Naval architecture has produced many different methodologies and tools over the years and it has become a common feature that every so often a new technique is promoted as the answer to the hull design problem. However, these design tools rarely meet expectations. The introduction of a new method, which does not consider the functionality, particularly the advantages, of previous techniques, is not going to be very successful. The wealth of hull design techniques available has occurred because of the particularly large number of disciplines that form naval architecture. Each has tried to develop methodologies that are appropriate for the way hull design features in the discipline. This blinkered approach has led to many task specific and un-optimised generic tools. No developer has attempted to produce a tool by understanding the processes involved, particularly the design process, by utilising modern technology. In fact, tradition can be considered one of the many factors that have held back the development of hull surface design tools. The review of hull design techniques and methodologies identified that many systems incorporate particular features which are advantageous to the hull design process. While it could be possible to develop a hull design technique which could try to interface each of these features together, a more effective approach would be to identify the concepts within the present sets of tools that work efficiently in the development of the surface representation.

It is only in the last twenty to thirty years that software based hull design tools have become the primary medium for creating the hull form representation. A critical issue in the development of these software tools that must be highlighted is that the majority of developers rarely have the experience of the hull designer and vice versa. Consequently, it can be increasingly seen that the lack of proper consideration in the development of the hull design environment leads to tools based on the software developers' ideal approach to problem. This problem is clearly illustrated in Paramarine [36], where a good concept has been implemented in a tool which has been realised, in totality, utilising the generic user interface components of the Windows operating system and the resulting design environment is very alien to the practical designer.

Before software-based design tools became the primary development environment, naval architects were solely responsible for developing their own tools and, as capable engineers, these tools made very effective use of the technology that was at their disposal. In Chapter 2, the half model and, later, the spline batten were identified as the primary design tools. The half model, as a tool for developing a representation of hull form, provides a great level of tactile and visual feedback about the shape of the surface. It is a three-dimensional design system. Hull designers of the era would have been skilled craftsmen capable of developing their own personal tools to work with the medium. A designer could, as experience was gained, develop tools which could quickly form the general shape of the hull form and others that could handle detailed modelling. As a development tool, the half model must be considered one of the most effective *design* tools that have been developed for hull form creation. It was only through the increasing needs of improved accuracy for construction and emerging analysis techniques that forced this design methodology into obsolescence.

The need for accuracy in construction and a reproducible representation led to the Lines Plan becoming the primary means for developing a hull form. In modern terms, the move from a three to a two-dimensional representation would be considered backtracking. The introduction of this two dimensional approach to hull design places sole responsibility onto the designer for ensuring that the shape can be constructed three dimensionally. Consequently, the design process is prolonged due to the need for the designer to continually ensure that the surface shape matches up across all three orthographic views. While the move to a two dimensional design process was detrimental to formation of a valid three-dimensional shape, it did allow designers to develop improved means of controlling curved shapes. The spline batten relies on the bending properties of beams of homogeneous material to ensure continuous curved shapes can be formed. Furthermore, the two dimensional design medium presents an appropriate way of controlling the shape of the batten by applying constraints in the form of weights. There is no way that this technique could be implemented in a practical manner for controlling the shape of the half model.

For a long time, the Lines Plan was a sufficient technique for hull form design. However, as modern computer-based technology became important in design and construction, it was natural for the hull design process to adapt, as it had done with the half model. Initially, computer-based hull design tools were electronic replications of the Lines Plan design environment. However, as it became increasingly desirable to have the hull represented as a surface, design tools reverted back to three-dimensions and provided basic features to allow the designer to create the hull form using the mathematical surface representations. As CAD techniques developed, parametric curve and surface representations, particularly NURBS, have become the primary means of representing shapes. Standardisation throughout computer aided design has resulted in the practical requirement for all engineering and design to use these techniques when representing any freeform shape. However, mathematical techniques, such as NURBS, are practically analogous to using a Flexicurve to design a hull form, by the way that shape is directly manipulated. The use of a Flexicurve to design a hull form would be considered absolute sacrilege by any designer experienced in non-electronic forms of hull development, so why has it been accepted in the modern hull design environment?

As both the half model and spline technique were primarily developed by hull designers, features were incorporated, by design, that would only allow the designer to form shapes appropriate for the hull form. The designer was physically prevented from creating inappropriate shapes by the fact that the tools would have to be significantly misused, perhaps resulting in tool failure, i.e. a batten snapping. This fact seems to indicate that the hull design process functions more effectively if the designer is *constrained* from producing invalid hull surface shapes. The wealth of shape generation functions and the specific shape properties within these functions, such as the property of local modification within NURBS, increases the flexibility within the design environment to the extent that it works against the designer. However, as the introduction of these standardised representation techniques have brought so much benefit to naval architecture, particularly in other areas of ship design and construction, it would be inappropriate to try to introduce new representations now. Any new design methodologies must utilise existing representation techniques. However, by using constraint functions, tools which can be used to limit parts or all of a representations definition to a certain rule, it is possible to achieve a reduction in flexibility. Furthermore, in the case of NURBS, properties provide an excellent source of knowledge when devising methods to constrain the definition. Having identified the concept of aiding the designer by constraining the hull representation, a strategy is required which will allow constraints to be applied and managed in a rational manner conducive to an efficient hull design process.

10.3. A Structured Approach

A wide variety of methodologies for forming the hull surface are presented in the review of modern tools in Chapter 7. Some tools represent the hull with a single surface, some with many patches. However, the existence of structural and procedural approaches for entering definition data was found to be a critical factor affecting the efficiency of the hull design process with these tools, functioning independently of the representation technology. These factors would also have been very important in the early hull design tools. The half model, for example, requires a particularly effective procedure to ensure that mistakes and errors are minimised. A block of wood does not have the “Undo” feature present in modern design tools. It is instinctive to start by cutting the block to the main dimensions and then work down to the midship section shape. With this approach in mind, it becomes obvious where terminologies such as Block and Prismatic Coefficient originate from.

The hull surface is made of many shapes of different complexity and descriptions resulting from form shape, such as principal dimensions and local features. However, modern hull design tools have not taken advantage of the existence of form shape at different scales within the surface. The designer is provided with only one level of definition to create the entire representation. Main dimensions must be developed with the same definition as the shape of the bulb. Consequently, there is no independence between the defined shapes and the resulting fairing process, which must be performed when the design is ready for construction, remains lengthy. To define the hull form using a single level of representation data, while keeping tool implementation simple, makes the task of forming the hull very difficult and time consuming. The designer must be able to rationalise the correct shape by considering the larger trends in the form and local features within the same definition data. As a result, there are absolutely no possibilities to develop practical and efficient techniques for automatically fairing the hull form which do not require considerable assistance and management by the user. If hull surface definition was subdivided into layers, a hierarchy of surface features, enabling global shape and local features to be controlled independently, the shapes of the definition at each level would be considerably simpler. Consequently, it becomes easier for the designer to manipulate a level definition within the hierarchy because it is no longer necessary to consider how an individual definition level will directly affect the definition of other shapes in the surface. In fact, once the shape of the hull form at each level is identified, it becomes very easy to implement definition quality control features such as fairing. It follows that, if the shape of the definition geometry within a certain level remains consistent, the user can have the option to apply an automatic constraint to the definition geometry so that it will retain the shape during any transformations. Additional control for customisation could be achieved through parametric control of the constraint functions. Some tools, such as Napa [35], already allow the user to apply limited amount of constraint to the surface geometry through the structure of the surface definition. However, the potential benefits to hull form design using a structured *hierarchical* approach in the definition is something that have never been considered before.

A structured approach is always present in the more effective hull design tools. These tools have tended to construct the hull surface using definition curves and relational geometry [47]. This combination provides the designer with the ability to represent the arrangement of the structure or topology of the hull surface, an arrangement which is difficult to reproduce using surface definition techniques alone. The technique ensures that, as the designer manipulates the definition, a surface with the properties defined by the definition curves will be maintained. However, present techniques do not ensure that the surface will be a representation of a valid hull form. The topology that is represented by these definition structures is only one that relates to the formation of a surface with the desired local properties or constraints. Consequently, the task of maintaining the definition in an arrangement that forms a valid hull form remains solely with the designer. If the relational geometry technique was extended to allow the form of the hull surface to be maintained by constraining geometry across regions of the surface definition through constraints, the amount of manipulation would be considerably reduced. The approach could be further streamlined by developing a definition procedure that details how to build the topological and constraint relationships in a hierarchical structure relevant to the design and manipulation process of hull forms.

10.4. Hull Surface Design through Form Topology and the introduction of Geometric Constraints

While many present systems allow the designer to build up the structure or topology of the hull surface, there is no technique that takes advantage of the natural shape and structure of the hull surface, the *Form Topology*. One of the most difficult and time consuming procedures in developing a hull surface definition is the initial task of creating the definition structure itself. Designers using systems which provide a structured approach to hull surface definition will find that the same arrangement must be consistently reproduced. By taking advantages of the similarities within the structure of hull forms, the residual form topology, the tool could aid the designer develop the surface definition and reduce the amount of manipulation. Topology is quite a technical and abstract subject and there is ample material to go in a detailed study of the topology of hull forms. However, as ship design is, importantly, a very practical discipline, a practical approach must be taken to ensure that present users of design tools will be able to use and understand any new approaches developed as a result of its direct introduction into the hull design process. Consequently, rather than directly analysing hull surface and form shape to identify topological structures, it is more appropriate to review the actual hull surface design process, to identify the structures that are presently familiar to the designer.

A naval architect can sketch, on paper, the basic design of a hull form very easily in a matter of seconds (Figure 10.1). In terms of modern engineering design techniques, it would not be very accurate, the curves would not be mathematically fair and, if it is a quick sketch, proportions could be wrong. With a little more time and care, these curves could be entered into a good CAD system in a matter of minutes (Figure 10.2), in three dimensions. It should be clear to somebody with even a modicum of maritime experience that the sketch represents a ship hull form. The curves represent the characteristic features and form of the hull surface. Furthermore, it is possible to render the hand-drawn sketch representations of the surface shapes (Figure 10.3). Additionally, some more accurate descriptions of surface shape have been overlaid in the figure. The sketches do not take more than a couple of minutes to develop, yet to develop the same basic representation in modern hull definition tool, (Figure 10.4), will take many hours even though the shape is known. The knowledge of form shape needs to be transferred from the designer to the hull definition system in a way that will allow the designer to develop the hull surface in a similar manner to the sketch of the hull form, in three dimensions.

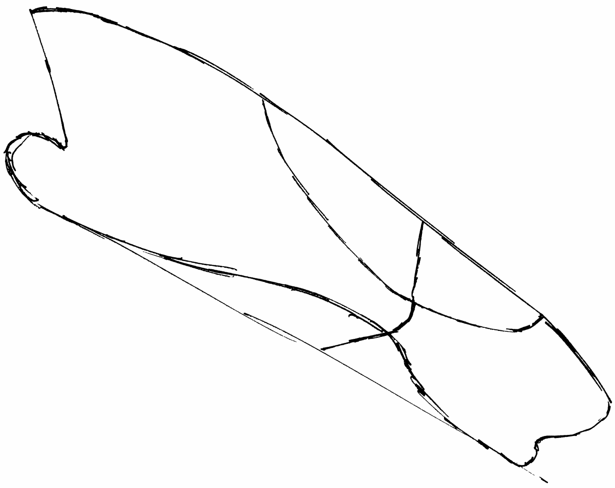


Figure 10.1, Basic hand sketch of a hull surface.

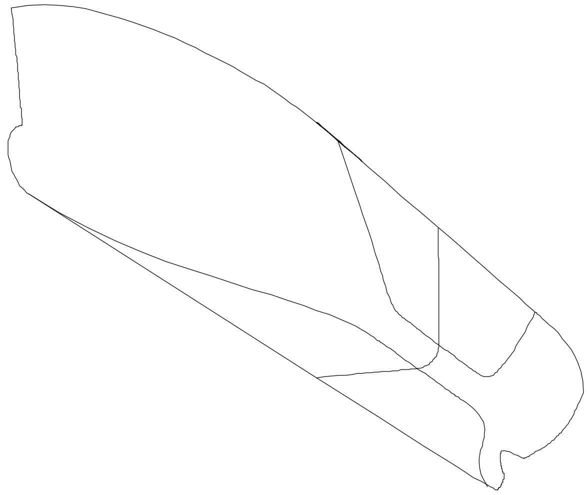


Figure 10.2, A quick three dimension CAD sketch of the major hull definition curves.

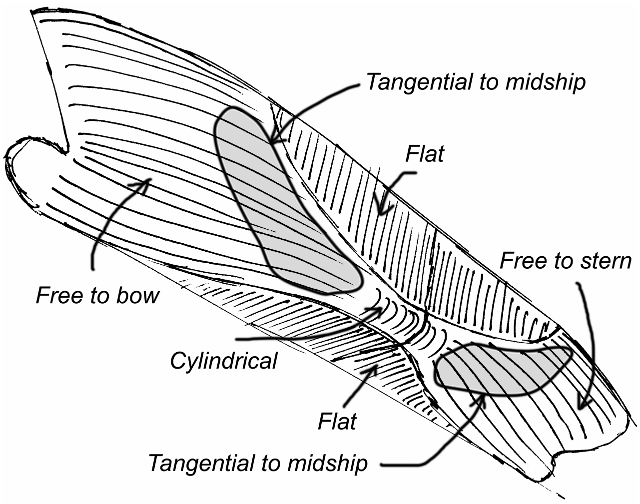


Figure 10.3, Hand sketch with rendering and text descriptions of surface shape.

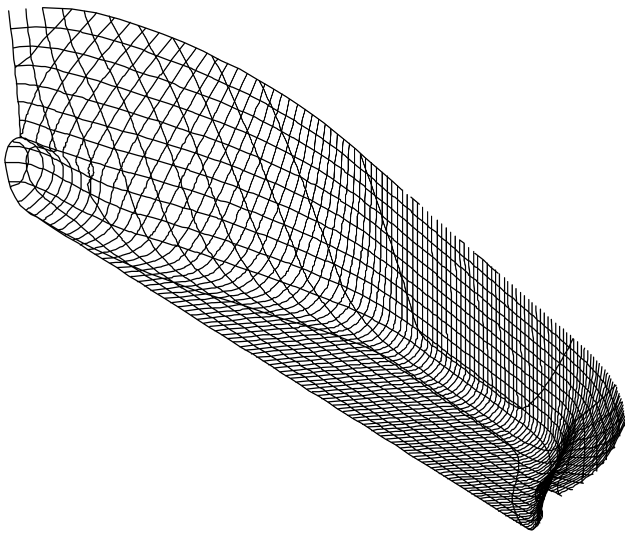


Figure 10.4, A hull surface developed through the basic sketched curves.

Present design systems, using structured definition techniques, allow the designer to create these curves, using them to geometrically control the shape of the surface representation. As these curves can be used with additional definition information to control tangents of the surface, the structure of curves could be considered a representation of the surface topology. Furthermore, as these curves exist within a hull design environment, they also represent the key characteristics of a hull form. Together, the group of curves represent the structure of the form topology. However, present hull design tools, in providing a generic surface design solution, have not taken advantage of any of this information.

The hand sketches show that a very limited amount of information is required to represent the form topology. Furthermore, in representing topology, the structure provides a lot of additional information on the relationships that exist between the curves within the structure, information that can be used to control individual curves and the shape of surface regions between the curves. For example, by definition, curves controlling the boundaries of the flat-of-side and flat-of-bottom lie on the prismatic surface defined by the midship section curve and the hull surface between these curves also lies on the prismatic surface. Consequently, any modification to the midship section curve should propagate to the definition of the flat-of-side and flat-of-bottom curves and the surface between, without the need for any additional manipulation. Even this basic example shows that if form topology information is utilised, there is a tremendous capacity for the hull design tool to assist the designer.

Current approaches used in relational geometry could be extended to produce practical tools which create topological relationships between definition curves. The relationships can be implemented using constraints which control geometric definition using references to other definition elements and additional parameters if necessary. Most geometric constraints would be simple functions as, as Figure 10.3 illustrates, most of the shape relationships are easily described.

Giving the user the ability to define the form topology which is used to produce a hull surface representation would obviously greatly improve the hull definition process, as indicated. In fact, this could be introduced to existing hull design tools without significant development, as an extension of any relational geometry implementation. However, the user is still required to manually generate the whole surface definition and now has to apply constraints. Consequently, the formation of the hull surface still remains, largely, a definition process rather than a design process. More improvements can be made by taking advantage of the existence of the form topology definition structure, to develop a more design orientated approach to the creation of the hull form surface.

Hull design tools implementing present definition techniques may only be able to identify the surface definition as a collection of points or curves. However, with the introduction of form topology, a structure is defined which can be traversed by the design tool to identify key features of the form. Furthermore, the fact that the form shape is maintained as parts of the hull definition are manipulated presents a key opportunity to review the use of control using numerical design parameters and transformations. It can be seen that the development of a tool that provides parametric and manual manipulation of the hull form within the same design environment at the same time is within reach. However, before reviewing the applications of form topology within design tools, there is a more fundamental benefit which significantly changes the present approach to hull form development.

For a tool designed to allow the user to form a surface through the use of a generic structure of curves and simple geometrical and topological relationships, the number of different shapes and topology structures is still going to be very large. However, the number of combinations that will actually form a topology that is a valid hull form is going to be much more limited. If there are a limited number of relevant topology structures, the design tool could use a knowledgebase of hull form topology structures to develop most of the definition based on a small amount of geometrical information provided by the user. Each topology in the knowledgebase would provide an outline of how a particular format of hull form should be defined. In doing so, the characteristic features and dimensions could be automatically identified without requiring the user to physically link geometric definition to parametric definition. Moreover, any missing definition geometry could be identified and automatically generated, and form constraints could be applied automatically to complete the definition required to produce a particular hull surface. Consequently, form topology can actually become a top-level parameter, although somewhat abstract, in the development of a hull surface. With the possibility of developing a large proportion of definition structure automatically, the designer can better concentrate on developing the actual features of a design, a process that is practically lost in the extent of manual manipulation that is required in present tools.

10.5. A Constraint Approach to Hull Surface Design

While the automatic generation of a great proportion of the hull surface definition data is initially welcomed by the designer, the idea of constraints and automatically generated geometry creates an impression of a tool that actively prevents the designer from interacting with the shape of the surface. This does not have to be the case. One of the biggest problems with the manual definition is the level of flexibility present in the definition, which is far beyond what is required at the earlier stages of design. The constraint approach attempts to address this problem by providing the means to restrict the definition by allowing it to be controlling by a range of relationships. These relationships can be simple, producing a straight line in a curve, for example, or complex, such as to control the hydrostatic properties of the hull form in combination with input certain parameters. The constraint approach can be used to reduce the amount of definition the user is able to manipulate and is capable of controlling most areas of the hull form. As the sketches demonstrate, curves are very effective at representing the shapes within the hull form that the designer wants to control. However, as CAD curve representations have the flexibility to represent an unlimited range of shapes, much manipulation is still required to produce accurate shapes characteristic of hull forms. By using constraint tools on the curve definition, common shapes can be accurately reproduced and manipulated without the need for the user to tend to each individual definition vertex. Consequently, the hull surface design process is primarily achieved by assigning persistent shapes (constraints) to the definition rather than through an iterative vertex manipulation process.

Once the idea of constraints is accepted, it is very easy to see how to restrict the flexibility of control curves to make a more effective hull form definition structure. For example, as the midship section curve lies on a plane, it is very easy to restrict the representation to a plane definition. Consequently, at definition level, it is only possible to manipulate the curve in two dimensions. In this case, the flexibility of the representation has been significantly reduced by the application of a single constraining relationship. Furthermore, by assigning this constraint relationship, the designer knows that the representation will remain true and it will not be necessary to check back to see if any data has moved. Curves representing the stem and the transom can also be developed using this constraint approach. If the shape of the hull form is considered in a little more detail, all curves representing part of the parallel section of the hull will lie on the same prismatic shape defined by the midship section. Consequently, these curves can be linked to the midship section definition curve. By this relationship, these representations will not have to be manipulated to ensure that the parallel middle body keeps the prismatic shape and will update if the midship section definition curve is changed. Both these constraint relationships are defined by the form topology structure. A single ship hull form has, by definition, a stem curve on the centre plane, and a prismatic parallel middle body for example. Therefore, identifying these constraints, producing parameters and developing a corresponding hull form to match is all part of the hierarchical approach. These types of definition constraints are available in many existing tools. However, it is possible to take the concept further, to develop more complex shapes of the hull form surface.

Developing constraints to restrict curves to two dimensions and form the parallel middle body are trivial activities. In both cases the constraints function by using linear relationships. The parallel surface shape can be developed by forming a linear surface between the two curves representing the extents of this region. However, the remaining regions of the hull form, the entrance and the run, require much more information to ensure the surface has a smooth and transitional curved shape to the parallel middle body. In present systems, the designer will have to manage a considerable amount of definition to develop these parts of the hull form. The designer must consider the position of the extents of the parallel middle body, how to form the surface tangents, make arrangements to control the volumetric characteristic and consider the details of local appendages.

If the constraint approach is used, the development of these parts of the hull form is no longer a problem for the designer. Firstly, as appendages are local features, they can be handled in later stages of the hierarchy. Consequently, the problem is just one of developing a smooth shape between the surface boundary and the parallel middle body with control over volumetric properties. The extents of these regions are defined with the characteristic shape definition curves. Consequently, the problem of developing the definition for the surface representation becomes one of producing data based on blending the shape from one end of the region to the next. This scheme can be quite adequately controlled by using tangents at each region boundary with a further degree of freedom inserted midway to control the volumetric properties. Rather than handle this shape with one constraint function, several generic functions can be used together to allow the tool to have greater flexibility, with the appropriate combination being selected and automatically applied on the basis of the form topology definition. Consequently, the problem of controlling the entrance or the run is handled by constraints controlling the tangents at each boundary with a further constraint controlling the fullness of the surface midway between the boundaries. These constraints can be controlled by single parameters so that the extents of the tangents or the volumetric properties of the hull form are now related to single parameters rather than many definition vertices.

Once constraints are to be used to create particular shapes in the surface, such as the tangent features, the functions will require the appropriate information or structure to be able to control the surface representation correctly. In the case of NURBS, the constraints may use the properties to develop the desired effects, implementing the constraint on the control polygon. However, it is just as applicable to consider the use of tangent vectors or derivatives if the constraints were to be applied to cubic spline or Coons patch representations respectively. As NURBS are controlled by the geometric arrangement of vertices with the control polygon, a constraint function may be implemented by producing further geometric definition. In the case of forming the tangent to a boundary, for example, the tangent is defined by the direction of the line segment between the end vertex of the control polygon and the next internal vertex. In this case, the constraint may generate definition to locate the internal control vertex based on the location of the end vertex. This approach can be used to develop further constraints for controlling NURBS based on the properties of the representation. Constraints to develop straight segments and knuckled lines are just some of the examples of what can be achieved. The implementation of these functions with respect to a pilot system will be explored in more detail in Chapter 12.

The constraint approach is a very appealing method of reducing the amount of definition data required to produce a hull surface representation. However, a balance has to be struck between the automatic and manual application of constraints to the surface definition, to ensure that the user feels that they always have complete control over the shape of the design. The best approach to take is one that ensures the designer remains in control of the definition that has been provided manually. Consequently, constraints need only be automatically applied when missing definition needs to be generated to form the hull surface. Manually defined curves can be controlled by a full range of individual constraint tools that can optionally applied to the definition. In extending the hierarchical approach using a definition structure to represent the hull form topology and controlling the flexibility of the definition representation using constraints, the conceptual structure of a hull design system that can use parametric and geometric information is developed into a more formal process. This process is shown schematically in Figure 10.5.

The tools implementing this approach will have a user interface that has many more features than present systems. The concept defines the basis for a design environment for developing the hull surface. Subsequently, in keeping with the benefits it provides to the designer, it has been termed the Topological Shape Constrained Adaptive Hull Design Environment (TSCAHDE – pronounced T-Shad).

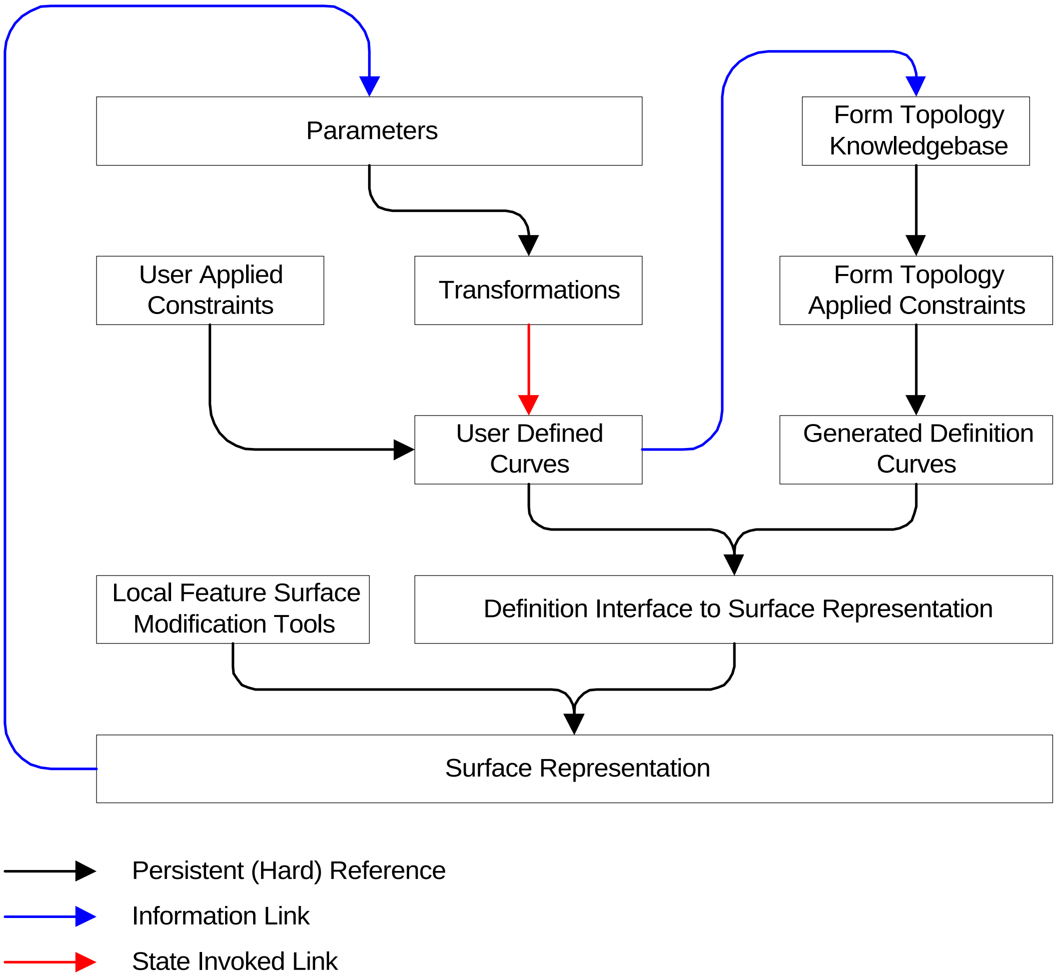


Figure 10.5, the hierarchical structure defined in chapter 9, elaborated to consider the tools developed to design the hull surface, i.e. Form Topology, Geometric Constraints.

11. A KNOWLEDGEBASE OF HULL FORM TOPOLOGY STRUCTURES

11.1. Identifying the Topology Structures

The success of a tool developed to aid a hull designer through the application of constraints and the creation of geometry will depend on the effectiveness of the knowledgebase and complexity of the form topology representation. The form topology representations must be designed to be effective enough to represent the needs of the task involved in developing the hull surface, yet simple enough that the already task-laden designer is not burdened with additional considerations during the design process. There are particular areas of computing dedicated to the development of systems for the recognition of patterns and shapes. The theories and applications of this area of technology are particularly complex, related to the number of shapes and patterns that each tool must recognise. By keeping the form topology structures as simple and generic as possible, the need to resort to such a complex technology should be prevented. Consequently, the success of this tool will be directly related to the number of form topology structures that will be required to allow the tool to design any hull form using the constraint approach. The general format and the number of shapes within the hull form surface is a direct relation to the number of different functions that must be accommodated by the hull component within the vessel system. By identifying the number of factors involved in defining shape of the hull surface combined with the approach taken by designers in considering these factors, a knowledgebase of form topology structures can be developed.

The obvious primary design characteristic of the hull surface of any waterborne vessel is that it should a) float and b) be upright. Many shapes satisfy this requirement including something as simple as a box. However, it is not until movement requirements are introduced that the particular shape characteristics of hull surfaces are introduced as a result of considering the hydrodynamic effects resulting from water flow around the vessel. By the nature of the environment, it is accepted that to move through water efficiently, the resistance of the form needs to be minimised, in balance with other design considerations of the vessel. By considering the surface geometry that best minimises resistance, an initial format of hull form topology can be developed. Furthermore, if the many details of hydrodynamics and some of the modern geometric solutions (appendages) that are used to improve performance are not considered, such as the bulbous bow, the structure of the form topology can be kept simple.

It is obvious to most that basic hydrodynamic requirements dictate that the angle at the entrance of the form to the water flow should be kept small. The larger the angle, the greater the impulse imparted to the water flow resulting in a higher force resisting movement. Consequently, bow profiles of hull forms are either a vertical boundary dividing the water to flow symmetrically down each side, or are wide, flat and angled, forcing the flow beneath the form.

It follows that, where the flow leaves the vessel, the surface should be shaped to allow the water to go back to the state that is as close to undisturbed as possible. Again, small angles between the water flow and hull surface are required to ensure that separation and turbulence in the flow, further increasing the resistance, are minimised. However, it should be noted that improved performance is gained for high Froude numbers with forms using large exit angles and by ensuring that total separation occurs, i.e. planing.

Figure 11.1, vessels moving through the water need to have a thinning at the bow and stern to minimise the resistance characteristics of the vessel

A detailed review of hull shape with respect to hydrodynamics could produce a form topology structure that is unnecessarily complex once other form shape characteristics are considered. However, the basic conceptual idea of a fluid flowing around an object naturally suggests a thinning at the entrance and the exit of the form. (Figure 11.1).

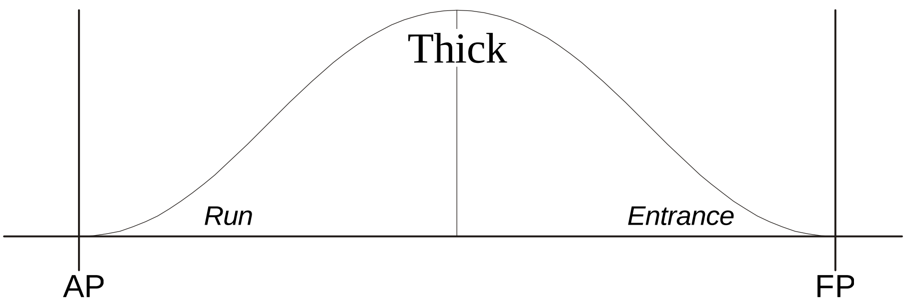


Figure 11.2, in smoothly transitioning to a greater volume amidships, the characteristic section area curve shape is produced.

To achieve the first basic requirement, to float, the vessel must have some volume. This suggests that the vessel will need to thicken to increase volume between the bow and stern. Furthermore, the flow characteristics dictate that the transition to the larger volume should be gentle. As a result of these considerations at the bow, the stern and in the midbody, for displacement mode operation, the characteristic form of the section area curve is produced, (Figure 11.2). Indeed, early engineers such as Scott-Russell [2] and Colin Archer [48] tried to relate the longitudinal form shape to resistance. To the waterline shape in the case of Scott-Russell and the section area curve in the case of Colin Archer.

The surface resulting from the definition of this form suggests a shape that is curved all over, with no clearly defined internal boundaries. The form gently transitions flow around the form, from the thin forward sectional shapes to larger midbody sections and back. The yacht hull form best illustrates this shape, because, in the design of these vessels, the minimisation of resistance is the most fundamental factor.

For non-displacement mode operation, separation of the water flow at the stern is of primary importance. Consequently, the weight of the vessel becomes more supported by hydrodynamic lift rather than hydrostatic buoyancy. To improve hydrodynamic support, the shape of the hull form needs to be changed so that lift can act directly against the weight of the vessel. The characteristic shape of these forms becomes dominated by large flat areas to maximise lift and sharp corners to induce separation. If hydrodynamic lift becomes the highest priority, the hull shape becomes very similar to a flat plate. The air boats used in the Florida Everglades are a good example of hull forms designed to maximise lift. These vessels rarely meet waves in the sheltered waters of the Everglades and consequently, good sea-keeping does not have to feature in the design considerations. In these cases, the topology structure of the hull surface tends toward the ideal flat plate solution for planing forms.

While both the yacht and the air boat designs are good examples of minimal resistance forms operating in displacement, and planing modes respectively, the design of the hull form in both cases is dominated by the need to minimise resistance. All other design considerations must be compromised and adapted around this primary factor. Consequently, the design factors of the hull form are unbalanced.

In the case of the yacht, being a craft of leisure, there is no need to introduce considerations for the maximisation of the internal volume for cargo carriage. This factor must be introduced in the design of ships, where there is the need to carry the largest amount of cargo at the most economical rate. Consequently, the hull form shape must be adapted to take account of this design factor and a compromise needs to be reached in the shape of the surface through the balance of the minimisation of resistance and maximisation of cargo capacity,

The introduction of this design parameter affects the shape of the hull form topology structure through the introduction of parallel form in the middle body, (Figure 11.3). Consequently, boundaries in surface shape occur at each end of the parallel section. Furthermore, if the hull shape becomes rectangular in section, it will better accommodate the shapes of cargo. This results in flat areas of the hull form in the vertical and horizontal planes, again creating boundaries between shapes in the hull surface.

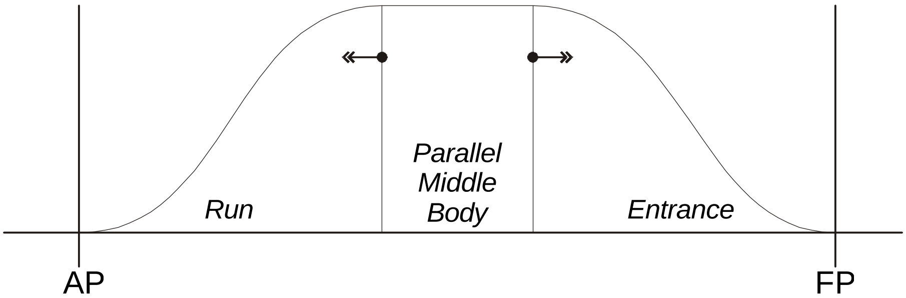


Figure 11.3, inserting parallel middle body into the hull form developed a region of constant section area. This divides the section area curve into three different characteristic parts.

While the efficient carriage of cargo is the more important factor for any trading vessel, the ease of construction has a primary effect on the initial cost of the vessel and subsequently governs when the vessel will begin to turn a profit. Consequently, the factors within the physical construction of the hull surface are based on using the cheapest materials with respect to overall manufacture and lifecycle costs such as ease of maintenance and product longevity. There are many technical materials available, such as GRP, which allow any shape to be formed with minimal changes in the cost of manufacture. However, these may not be as resistant and as easy to maintain as other materials. Consequently, for the majority of vessels a compromised is reached by choosing a material that is cheap, resilient, easy to repair and, if care is taken in the design of the hull surface shape, does not require extensive effect to form. Steel, as it fits these specifications well, is the material of choice in most cases.

Steel is produced in the form of flat plates and, as a consequence, difficulties can be experienced when manufacturing the curved shape of the hull form. Surface shape with double curvature should be avoided because, while is usually possible to work the material into these types of shapes, it increases the time of construction and hence the cost of the vessel. It has been shown that if the hull surface is designed with a minimisation of double curvature significant savings can be made [42]. The design of the hull adapts to these needs by developing more boundaries between specific, easily manufactured, shapes in the hull surface or, if cost of construction becomes the most important factor in the design, developing a hull form that uses minimal working of the construction material.

The result of the application of each of these design considerations, in balance or with priority, has a profound effect on the hull form shape topology. As one design factor becomes more dominant the hull form becomes increasingly defined by the shapes required for that particular factor. In a hull form designed to accommodate many factors, the number of shapes that are introduced is going to increasingly result in a more irregular topological structure. If the form topology knowledgebase were to be indexed around design factors, there would be an infinite number of structures. However, if the index were based around shape, the number of different classifications of topology structures is vastly reduced. Over the review of topological shape with respect to hull surface design factors, the different kinds of shape required for each factor has either been curved or flat. Consequently, it is possible to develop the knowledgebase around the need for the hull to be dominated by curved shaped areas, flat shaped areas or a balanced mixture of both (Figure 11.4):

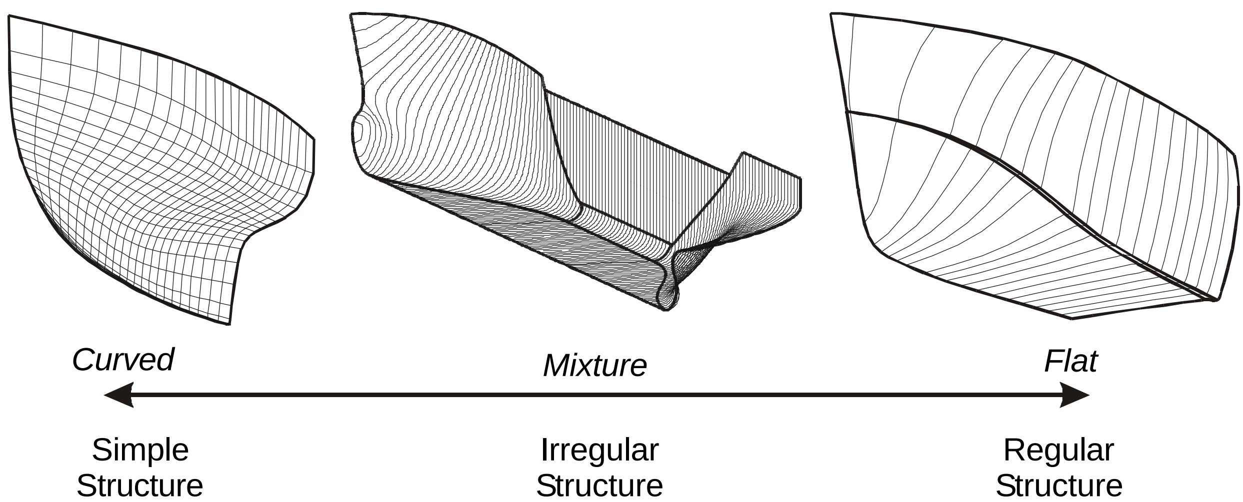


Figure 11.4, the range of topology structures governed by the form of the hull surface and the function of the vessel. The topology structures are identified by the degree of unbounded curve shape to the structure of flat shape in the hull surface.

11.2. The Topology Structures in Detail

Topological Structure	Description and Review
Simple Topology, for curved hulls	For hull forms completely dominated by curved shapes there is a great difficulty in defining a structure as the definition of curved is used qualitatively. As these hull forms are primarily designed for the minimisation of resistance, any features should be as streamlined and as smooth as possible. Consequently, the hull surface can only benefit from form topology and constraints at the boundaries. Elaborate techniques would be required to implement constraints that control the form of the hull surface within the boundaries and it is unlikely that any such technique would bring any practical benefits to the hull design process. This form topology is mainly characteristic of round bilge hull forms and is generally found in small craft vessels such as yachts and fishing boats. Present hull design system using relational geometry techniques that allow surfaces to be attached to curves provide as much assistance that could be expected to be received from a system developed using the form topology and constraints approach.
Regular Topology, for hull forms consisting mainly of flats.	Vessels that operate in non-displacement modes or hull forms that must minimise the cost of the hull manufacture must maximise the amount of flat area in the hull form surface. In the case of planing craft, the flat areas maximise the amount of the extent of the hull surface that will produce hydrodynamic lift force. In the case of cheap manufacture, the larger quantities of flats reduce the need to work the construction material into specific shapes. The result of this approach is that boundaries between the flat areas run the whole length of the hull surface producing knuckles. These knuckle lines form a hull form topology consisting of a regular arrangement of panels. Present hull design tools using relational geometry allow the form topology structure to be constructed. However, present techniques do not provide any mechanisms to control edge tangency into knuckle lines and would definitely not be able to constrain the surface in a persistently developable state. The form constraint approach has an ability to implement these features.

Irregular Form Topology Structure, for forms consisting of curved and flat areas.	For vessels that must balance many design factor together such as hydrodynamic performance with cargo capacity, and manufacturing considerations. The hull form needs to be designed to maximise the number of easily manufactured areas in the hull surface without imposing a detrimental effect on hull performance. Consequently, areas of the hull surface that have minimal effect on hull performance and more on cargo capacity are shaped more simply, i.e. flat, and are easier to construct. Areas contributing to the hydrodynamic performance cannot be simplified and the hull in these areas remains smoothly curved. As a result, the form topology structure the hull form becomes irregular. However, due to the irregular nature of the form, the number of different areas of the hull surfaces becomes larger and there is a more definitive topological structure. Consequently, form constraints are easier to implement and ultimately more beneficial to the designer and the design process.

12. DEVELOPMENT OF A PILOT SYSTEM

The concept of a hierarchical process that develops the hull surface from components of definition using form topology and geometric constraints is a radical departure from present techniques. It appears to be considerable more complicated when compared to the manipulation of individual vertices. However, it should not be seen as a new method of representing a hull form, but as an interface to existing representation techniques which provides additional tools for controlling the surface definition in structured ways. Consequently, if the designer wishes to export the surface to tool that will only allow direct manipulation, it can be achieved. However, in developing a tool that employs the form topology approach, this course of action should be unnecessary.

It is difficult to explain the full functionality of a system developed using the approach at a conceptual level. The technique attempts to solve the problems faced with present hull development tools by breaking the process up into many simpler component tasks. Consequently, as each individual component is only a detail in the structure, it is not possible to discuss the mechanics at the conceptual level. Therefore, a pilot system will be created and a detailed review will be performed in the development of each individual component.

The development of the pilot system will try to address the major concepts identified by this approach. Consequently, functions that are available in present techniques may not be covered in great detail and areas of the concept that can be developed extensively may be initially implemented in a simple fashion leaving work for future development.

Three different structures of form topology were identified in Chapter 11. The simple topology structure cannot be addressed by this approach at this early stage in development. In the future constraint relationships of a more complex nature may be developed. The regular topology hull form can be addressed fairly well by present design tools, although without the aid of the form topology process. Hull forms defined using an irregular topology structure would benefit the development of the tool because these surfaces consist of many different shapes and this allows a wide range of different constraint relationships to be developed.

In separating the definition from the surface representation, the approach develops a blank canvas to design a new hull form definition technique without having to consider many of the limitations found within present tools. However, while the pilot system will be developed within a capable CAD platform tool, it will not be possible to take an approach that will produce the most effective and practical hull design tools. As a research project, the pilot system will attempt to demonstrate the *concept* in the most effective manner possible.

The development of the pilot system will cover the concept by looking at four areas in great detail. Firstly, it is necessary to design the interface (Surface Interface Framework) that will link the design definition to the hull surface representation. This area of development is primarily concerned with the development of a framework environment in which the characteristic form definition curves and geometric constraints can be implemented and effect the surface in the correct manner.

As the concept identifies a different approach to the use of parameters, unlike previous techniques where a set has to be chosen on the basis of the minimum number that will allow the hull form to be generated, parameters in this approach can be selected on the basis of which will be the most useful to the designer.

The definition curves are the medium through which the shape of the hull form is controlled. The parameters invoke transformations of the definition curves which are, in turn, used to control the surface interface framework. Furthermore, the geometric constraints function by restricting the shape of individual definition curves or by transferring shape between curves. Consequently, a considerable amount of development involves ensuring that the definition curves can provide the designer with all the functionality required to develop a wide range of hull surface shapes.

Finally, approaches for incorporating separately defined local features are reviewed. Techniques such as surface trimming and surface sculpturing tools such as warping are considered as possible ways of incorporating local appendage features into the surface. A technique for incorporating a separate parametrically defined bulb appendage surface is presented to demonstrate that the possibility of forming the surface by combining sub-component definition geometry does exist.

13. HULL SURFACE FRAMEWORK

13.1. NURBS Surfaces and Hull Generation Techniques

NURBS surfaces have not been widely used in hull generation techniques. The representation is not defined with gradient and curvature derivatives, as in Coon's patches, and the control polygon mesh has no direct method of representing these parameters without additional calculation. Previous hull generation techniques using NURBS surface representations have either produced the hull form using interpolation techniques or by developing a system of equations and constraints for each control vertex, which are solved to achieve desirable hull qualities, as in the case of Sanderski's [32] technique. The large number of control vertices required for the definition of a hull form and the complexity of the process required to generate discreet control vertices from analytical data produced by a traditional approach to hull form generation does not make the NURBS surface the best choice for these techniques. However, as NURBS are now the industry standard for exchanging curve and surface representations and are easy to manipulate by hand when local changes need to be made. The advantages of the representation outweigh the difficulties that must be overcome to generate NURBS hull surfaces.

The control of NURBS surfaces can appear to be complex, especially if the contribution of individual vertices to surface shape is to be understood. However, as NURBS have known properties and behaviours, it is possible to take advantage of these qualities to produce the control polygon mesh of a NURBS surface with the desired hull features, without using complex vertex placement procedures. The NURBS properties that are of particular interest in the production of a hull surface representation are as follows:

- The boundaries of a NURBS surface are NURBS curves with the same knot vector and weights corresponding to the *u* or *v* parameter of the boundary.
- The end segments of the control polygon represent the tangent of NURBS at the boundary.
- Features, such as knuckle lines and planes, can be developed without the need for additional surfaces.
- The surface generally follows the shape of the control polygon. From this behaviour, it can be deduced that a fair control polygon should produce a fair surface.

Parametric hull generation techniques can be considered data expansion procedures, as a small amount of information, by comparison, is used to develop the information rich hull surface representation. A procedure is required to achieve a similar effect, to create an interface between the definition curves within the TSCAHDE and the NURBS surface control polygon structure, using the known shape properties to produce the location of the vertices. However, as the procedure is dependant on the particular way surfaces are used to represent the hull form, it is necessary to select an arrangement that will allow for the best demonstration of the TSCAHDE concept.

13.2. Hull form representation approaches using NURBS surfaces

There are three main approaches being used across different hull design tools to represent the hull form using NURBS surfaces:

1. Single surface representations

   Systems using single NURBS surfaces to represent the hull form allow the user to manipulate the hull form more easily as only one control polygon structure is present. However, the construction of surfaces with varying amounts of local complexity is more difficult as the user must manually create the right control polygon arrangement in consideration for the other shapes that are to be modelled in the surface. Moreover, the accurate construction of particular shapes in the surface, such as flat of side boundaries, is very difficult due to the nature of the regular mesh structure. Consequently, single surface hull representations are best suited to small craft forms, such as yachts where the hull shape is much more continuous.

2. Simply structured multi-patch surfaces

   This approach is very appropriate for ship hull surfaces particularly when the patch boundaries are arranged to represent the features of the hull form, such as the extent of the flat of side. With correctly located patch boundaries, it becomes easier to represent particular features such as the cylindrical nature of the bilge radius. It is also possible to adjust the density of control polygon definition so that simple areas such as the flat areas of the hull surface have few vertices, perhaps only on the surface boundary. Complex areas, such as the bulb region, are better defined with more vertices. However, in present implementations, it is the users responsibility to maintain the surface structure and while the control of boundaries is improved, control of shapes within surface patches is not improved. Examples of this approach can be found in ShipGEN [29] and FORM [30].

3. Complex structured multi-patch surfaces

   Large multi-patch hull representations are being successfully used in the NAPA [35] system. The quality of patch surface representation technique is highly dependant on the grid formed by definition curves. A large structure of curves is required to produce an accurate hull form definition. The grid of curves can be irregular, allowing areas of complex shape to be created with more dense definition than simpler regions of the hull, such as the flat of bottom. A complex algorithm is required to develop the surface patches from the curve structure, especially as curves can be used in special constructions such as "roundings". Furthermore, as the curves are entered manually by the user, the technique must deal with the additional problems caused by differing qualities of input data. Finally, it is practically impossible to manipulate the hull outside of the hull design environment, as the patch structure is too large and detailed to be maintained manually.

The second approach is the most appropriate technique for developing a ship hull form as each feature and shaped region of the hull form can be represented by a different surface with an appropriate level of definition and the transitions between surfaces can be adequately controlled as NURBS surface boundaries behave as NURBS curves. However, it is often beneficial to have an irregular patch structure to develop features such as well-defined disappearing knuckle lines. The development of an algorithm to generate appropriate multi-patch surfaces would take some time and it would be less robust that a corresponding technique for generating single surface representations. The single surface representation is capable of demonstrating that the concept is realisable and it is simpler to implement. However, the detailed control of boundary shape is more difficult. Therefore, it should be kept in mind that better performance would be achieved with a multi-patch hull surface.

13.3. Developing an Appropriate Surface Definition Tool

The interface between the curves and surface is the most important component of the whole hull design technique. It must translate the shape information contained in the definition curve data into a full NURBS surface representation of the desired hull form. The flexibility of the NURBS surface representation has been highlighted as one of the main impediments to efficient hull design.

The nature of the regular mesh of the surface control polygon often means that many more vertices may have to be manipulated in comparison to the change in shape and presently, there are not any facilities to constrain groups of definition vertices to maintain specific shapes in the surface. As the control polygon mesh is an integral part of NURBS, it cannot, as a definition technique, be modified without the chance of introducing incompatibilities to other design systems during data exchange. However, a technique can be developed to produce a control polygon mesh from a less well-defined structure using the known properties of NURBS.

The concept defines a low number of definition curves, to allow the quick development of designs and flexible control of surface shape. As NURBS surfaces require a relatively large amount of definition data, there will not be enough data in the definition curves alone to directly specify the vertex locations for the NURBS control polygon. Consequently, the interface will require a geometric framework, functioning in a similar manner to the physical frames of the ship hull, to support the surface control polygon. The simpler control-curve based definition structure can be directly tailored to the needs of the hull designer to allow the production of the common patterns of shape found in ship hull forms through the use of constraints. As the framework supports the entire definition of the surface representation, it is important to ensure that the structure is flexible enough to produce the range of shapes covered by the form topology by using a definition structure that is intuitively understood by the designer.

Traditionally, naval architects have always worked with curves. Curves are used represent the contours of the hull and the boundaries of particular shapes in the surface. However, in parametric surfaces, the generation of any curves, except the boundaries, is difficult because parametric distortions occur around the areas of dense definition required to produce particular features in the surface. These distortions can be visualised if the iso-parametric lines of the surface are displayed. Furthermore, the iso-parametric lines can deceive the user to the true nature of the hull shape, especially if the traditional contoured approach to hull design is more familiar. As contours are so important in hull design, particularly when considering usability and feedback, it is desirable to use definition curves closely representative of contour shapes to control the surface interface framework (SIF).

In the previous chapter, sketched hull forms (Figure 10.1 and Figure 10.3) were used to identify the initial shapes a designer would use capture the essence of a hull surface. The sketches illustrate that it is more natural to represent the initial hull surface using transverse rather than longitudinal depictions of shape. Longitudinal shape is very important to the performance of the vessel. However, the aspect ratio of longitudinal curves is such that is very difficult to produce the correct shapes without the aid of curve generation and analysis (fairing) tools. As longitudinal shape is more difficult to produce manually, it is more effective to produce it mathematically by interpolating the definition of the transverse curves. Moreover, if the number of transverse definition curves is low, it is easier to maintain a smoother and more mathematically fair longitudinal shape.

Separate control in transverse and longitudinal shape is a good basis for developing the means to produce the control polygon of the NURBS surface. The ideal curve representation technique, considering the surface representation, is, of course, NURBS. From the known properties of the surface, the locations of control polygon vertices at the boundaries of the surface would correspond to those within the control polygons of the definition curves to produce exactly the same shape, given the same knot vectors and weights. This approach can be used throughout the control curves to aid the development of the surface control polygon, however, with certain restrictions. For simplicity, it will be necessary for all curves and the surface in the transverse parametric direction to have the same knot vector and weights. Consequently, the number of control vertices and degree in the transverse direction will be constant. Limiting the degree of the definition curve representation will not cause any difficult for the user as it is generally standard practice to work with cubic functions. The restriction on the number of control points will require better planning of the control polygon, on the part of the user, to allow the development of desired shapes. Additional editing tools and features can be developed to aid the user by automatically managing the number of control vertices on all curves when any single curve is being refined.

As each control curve has the same number of vertices and each vertex corresponds to a longitudinal row in the surface control polygon, the shape of the rows of the surface control polygon can be developed from longitudinal curve functions defined through the corresponding vertices on each control curve. Consequently, the number of control curves can be independent of the number of columns in the surface control polygon. A low number of control curves will allow the development of a more free and fair shape and a number equal to the columns in the control polygon gives the user direct influence over the surface. It is, as if the vertices of the surface control polygon are on a *curtain rail* defined by the longitudinal curve functions through the vertices of the control curves. The process of developing the hull surface from the transverse control curve can be considered similar to the technique of Skinning.

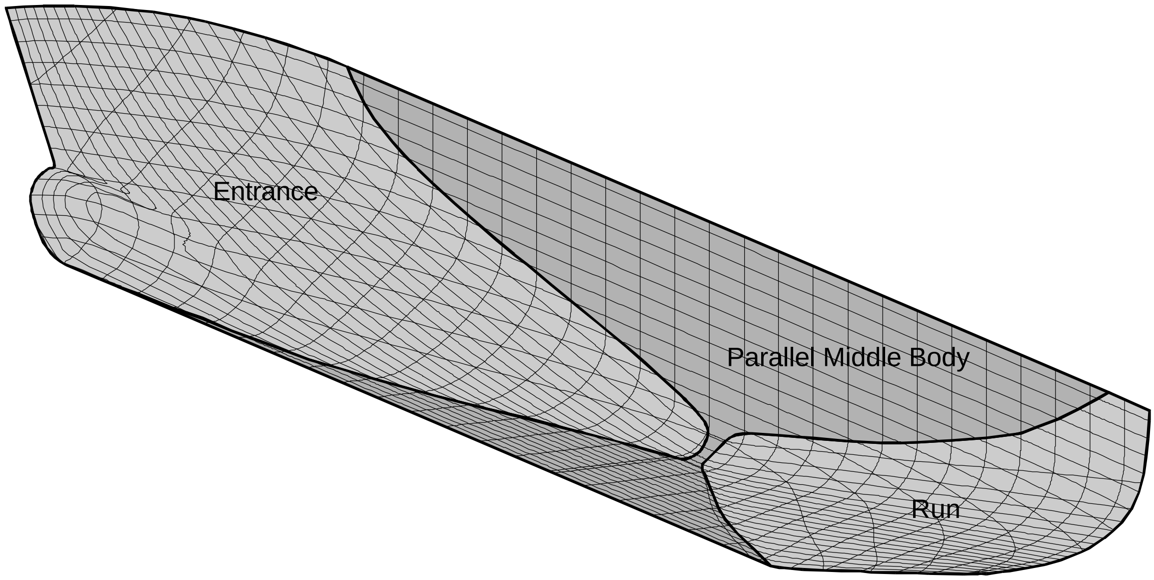


Figure 13.1, by considering the parallel part of the hull as one region, consisting of the flat of side, flat of bottom and bilge radius, three regions, although deformed, are created that fit the regular mesh arrangement of the control polygon of a single NURBS surface.

If the different regions of shape in the hull are identified, by the form topology, and the surface development process considers each region separately, then the generation of the functions to control longitudinal shape is made more manageable. A multiple patch hull representation could use individual surfaces corresponding to the forebody, the afterbody, the flat-of-side, the flat-of-bottom and the bilge radius regions. However, the regular rectangular nature of the single NURBS control polygon does not easily allow separate representations of the flat-of-side and flat-of-bottom regions of the hull surface to be made. By taking an alternative approach in the way the form topology structure is defined, control curves can be used to represent the extents of the parallel middle body instead of bounding individual regions of specific shape. This arrangement produces a regular structure that can be accurately represented in the control polygon while maintaining a definition arrangement that can still be understood by the designer. Consequently, the surface consists of the fore-body, the parallel middle body and after-body regions, (Figure 13.1). Furthermore, the process of developing the longitudinal shape control functions is simplified because the range of correct shapes that must be produced, given that the surface represents a hull form, is limited as a result of reducing the number and arrangement of the regions.

13.4. Creating a Fair NURBS Surface Control Polygon

By inspection, the functions required to develop the parallel middle body (PMB) are going to be very simple, the PMB represents an extrusion of the midship section curve. Additionally, the process to implement the extrusion in the hull surface will ensure that the midship section control curve lies on the hull surface without the need for further calculation. For hulls, without parallel shape in the hull, this will not be the case.

For the Entrance and Run of hull surface, the development of the longitudinal (control polygon rows) vertex control functions is less trivial. However, by reviewing the shape of these regions to identify what information can be extracted from the control curves and what additional information must be provided with respect to the basic range of longitudinal shapes that must be produced, a specification for the form of the longitudinal control functions can be produced. By considering the shape of the forebody, (Figure 13.2), the following statements can be used as a basis to design the longitudinal functions:

1. The forward boundary of the region is represented by the Bow curve.
2. The aft boundary is represented by the curve defining the forward extent of the parallel middle body, known as the forward surface flat (FSF) curve.
3. The surface is longitudinally tangential to the FSF (3).
4. The tangent to the Bow curve needs to be controlled to adjust the angle of entrance (4).
5. The volume of the entrance needs to be controlled (5).

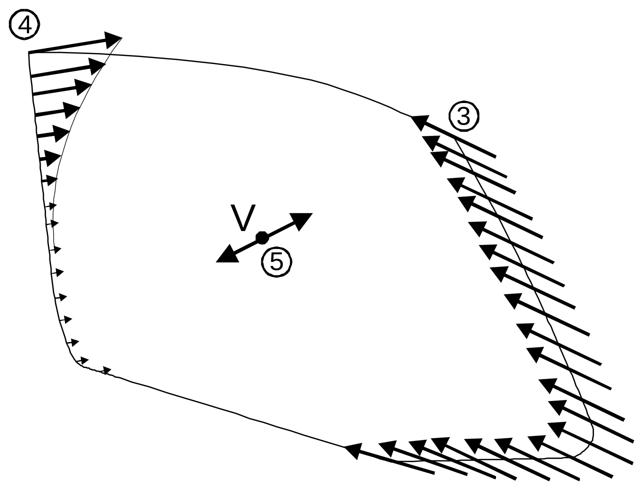


Figure 13.2, the shape each hull surface region must take account of requirements at the boundaries and volumetric considerations.

The development of each longitudinal function can be considered a blending problem, with an additional requirement to be able to control the volume of the hull. A family of blending functions needs to be generated to maintain a good transverse shape across the region. Transverse continuity can be maintained by using the form topology definition curves with additional curves to control tangency and volume of the region. In a similar way to the approach taken to develop the definition of the surface control polygon, the blending functions can be developed directly from the control curves if the blending functions are NURBS curves. Subsequently, on the basis of the previous statements made to describe the form of the blending functions, a five vertex NURBS curve representation, (Figure 13.3), is all that is required to control the surface across a region with respect to boundary locations, surface tangency and fullness corresponding to volumetric control.

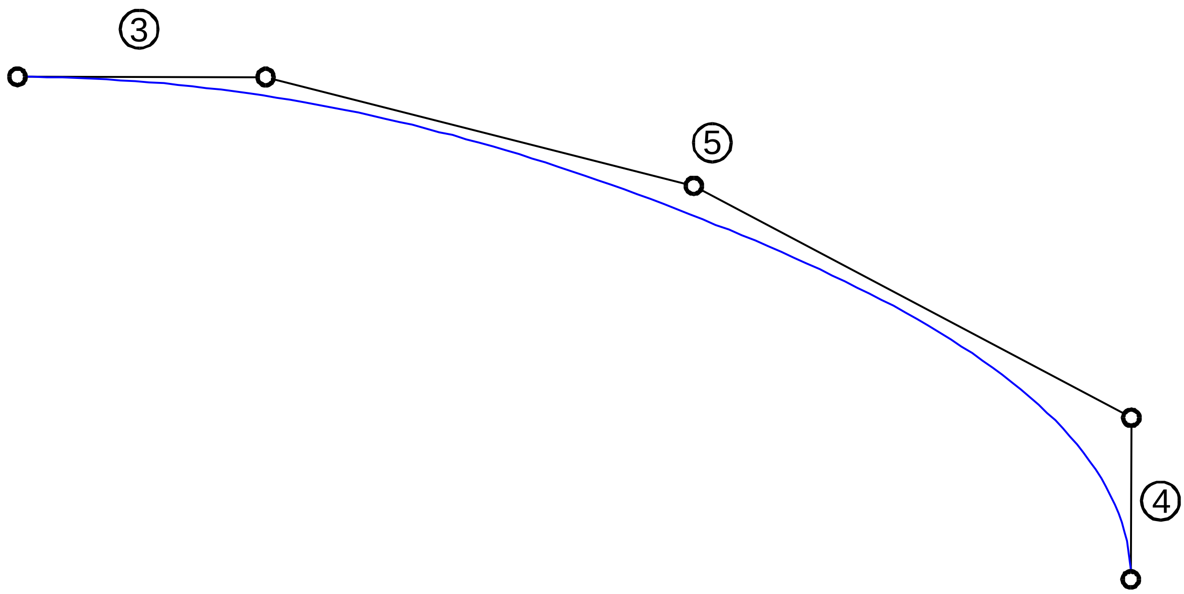


Figure 13.3, a five vertex B-spline curve can be used to represent the blending function, with respect to tangency and volumetric considerations.

When the family of NURBS blending functions are placed together (Figure 13.4), connected by five transverse control curves, one for each vertex on every curve, a region of hull surface shape can be controlled very accurately. To achieve the correct tangent shapes in the surface, further transverse control curves will be required in addition to the existing boundary control curves. Chapter 15 explains the how the Blending functions are constructed within the boundary curves using automatically generated tangent definition curves.

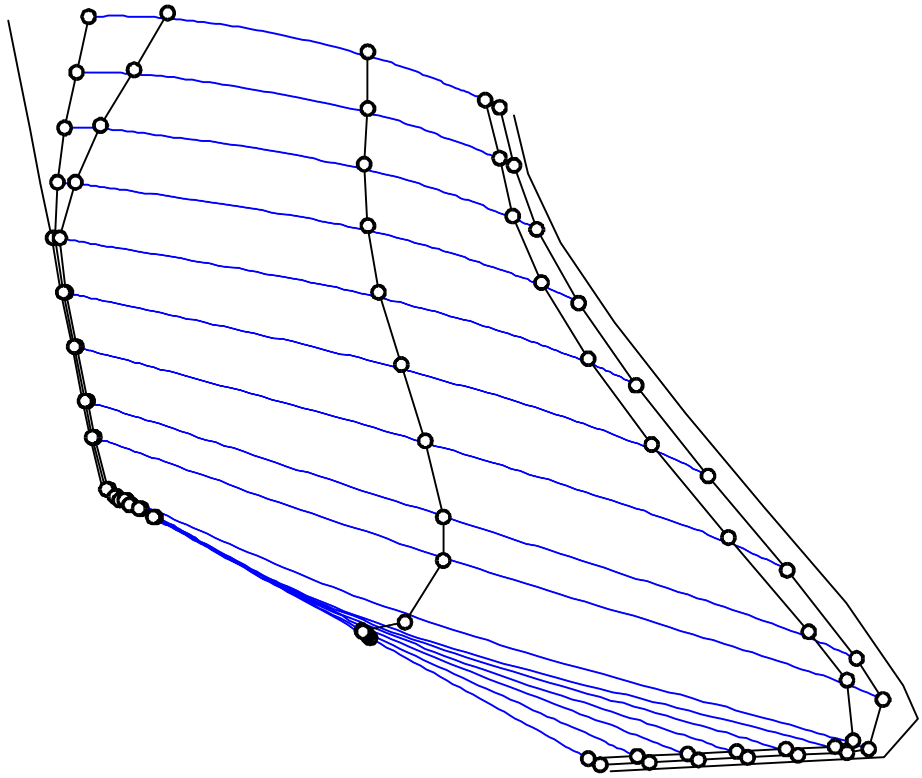


Figure 13.4, a family of Blending function representing the Entrance region of the hull surface.

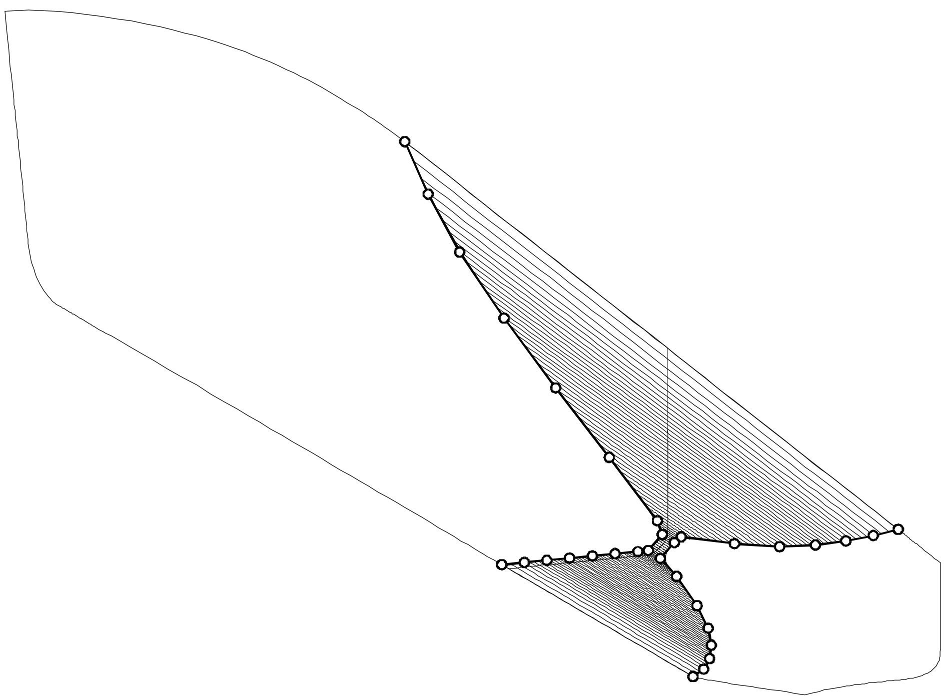


Figure 13.5, the parallel middle body is just an extrusion of the midship section curve. Consequently, it only requires two vertices within the control polygon to create linear Blending functions. As the midship curve is included because of it general importance to the hull form defintion.

While at least five control points are required on the NURBS blending curves to produce the prescribed range of shapes for an individual regions, this specification can be flexible to allow for optimisation in the definition and for further customisation by the user. For example, to develop the straight parallel middle body, (Figure 13.5), only the two end vertices of the blending function control polygons are required. Additional curves are unnecessary and lessen the conciseness of the definition. Furthermore, as more detailed design phases are approached, the user may wish to use more than five control curves to control each region of the surface. An advantage of using NURBS to represent the blending curves is that there is a great deal of flexibility in the number of parameters (control curves) used to define the shape. Simpler blending functions, specifically derived for this particular task, would not be able to accommodate flexibility of this magnitude.

By adopting a generic approach in the surface interface framework to the definition of the individual regions of surface shape, each region can be generated similarly, keeping the implementation small and concise and maintaining flexibility in the technique. As previously illustrated, the definition of a hull surface requires three shape regions, with up to five internal curves each to control definition properly. However, as there are two shared boundaries and the midship section is an extrusion, only thirteen curves are actually required to create a hull form using this approach (Figure 13.6).

Although the surface interface framework provides a structure that requires much less data and manipulation than present hull surface design techniques, if a designer were to try to manipulate it directly, the task, although improved, would still be difficult. The curves representing the boundary shapes are relatively easy to control and it is necessary to ensure that the designer can manipulate these curves directly with the full flexibility of the definition, if desired. However, the curves controlling the shape of tangents and producing the volumetric qualities of the hull have too specific a shape, due to the geometric relationships between the vertices and other curves, to make it impractical to manipulate them by hand. These curves are better controlled using geometric constraints in conjunction with automatic definition generation. For example, a constraint which introduces and controls the whole of a tangent definition curve based upon geometric relationships to elements in the form topology and parameters. If the hull generation procedure automatically adds these curves to the surface definition, the importance of these curves is, initially, not so obvious to the designer. Consequently, the hull generation process appears simpler, especially if the automatically generated curves are not directly displayed to the user.

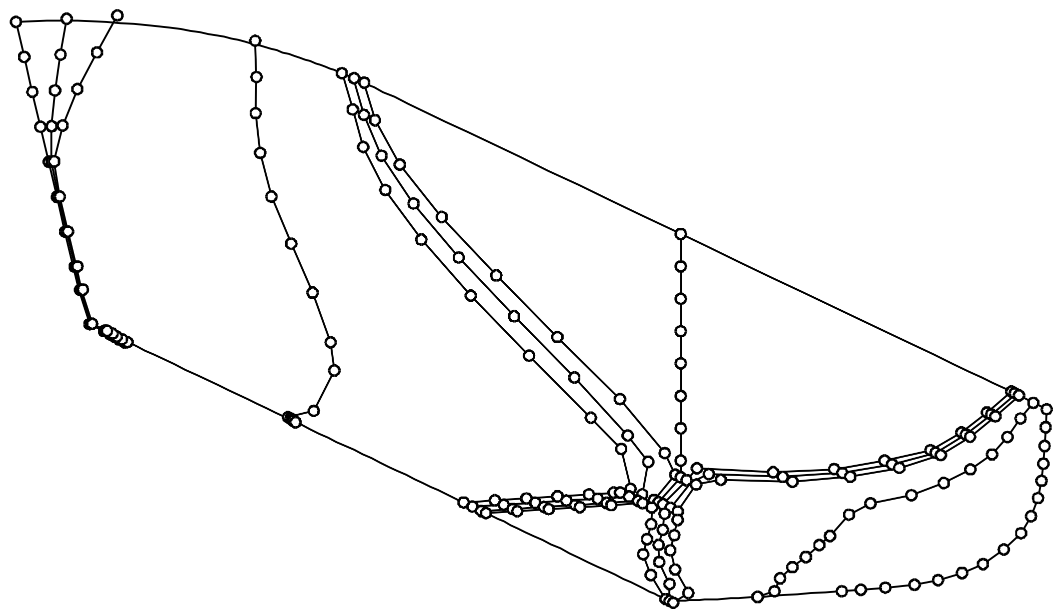


Figure 13.6, thirteen curves are required to represent a basic hull form surface using the SIF.

14. PARAMETERS

14.1. A New Approach for Numerical Parameters in the Hull Design Environment

The recent resurgence in the development of parametric hull design tools, such as those by Friendship Systems [49] and Paramarine [36] must indicate that the present, entirely manually manipulated approach to hull form design is inadequate. However, these recently developed tools have yet to find any acknowledgment in the naval architecture community and there are many reasons for this. Particularly, as these tools do not presently provide a complete ship design package or interface well with existing tools, naval architects will not introduce them into a practical ship design process because no significant advantage is offered. Nevertheless, the continuing development of parametrically based hull design tools suggests that an effective solution is still desired. However, a practical solution will need to be compatible and co-exist with present hull design tools. Without some significant changes to the strategies used in parametric hull design, these techniques will always be seen as interesting, but impractical oddities.

The ability to use numerical parameters in hull form design is a very attractive concept. It becomes very easy to make quick changes to the shape of the surface and intense design optimisation is a practical possibility, especially given the growing trends in the use of systematic optimisation throughout the ship design process. However, as existing parametric hull design techniques use rigid mathematical functions to develop the surface, flexibility in the range of shapes that can be produced is highly constrained. Consequently, the designer has very limited control of the detailed shape of the surface. Developers of parametric hull generation tools have long identified the need for manual manipulation of the hull surface within these techniques. However, a system that could update the mathematical definition of the surface based on changes introduced by direct manual manipulation would be complex and may result in the cumbersome, non-robust and unstable operation of the tool. A different approach needs to be taken to incorporate the use of numerical parameters to control the design of the hull form in an environment that also allows manual definition.

The most practical method of improving the flexibility of any parametric design tool is to add parameters that can control the surface shape in much more detail. For example, detailed parametric control of the bulb can be introduced into these tools. However, if there are too many parameters controlling every small feature of the hull surface, the tool is no longer able to provide the designer an effective hull development tool because every feature will have to be specified at the first instance. Concept and initial design using such a tool would be practically impossible.

It is possible to improve the situation by introducing mathematical rules which can be used to control detailed parameters in the absence of data from the user by, for example, using a scale factor to the overall size of the hull surface. However, the fact that fine surface details exist at the earliest stages of design means that the process no longer consists of gradual refinements of an initially simple surface to a more complex hull form.

An approach where the hull surface shape has an indirect mathematical dependency on numerical parameters has been identified as a solution to the major flaws identified in traditional parametric hull generation techniques in Chapter 9. In this approach, the numerical design parameters are mathematically dependent on the hull geometry and modifications to the hull surface shape, as a result of a parameter change invoked by the user, are implemented through a specific geometric transformation process, i.e. Parametric Hull Modification. Having previously developed a framework that allows a detailed hull form represented by a single NURBS surface, to be defined by, in comparison, a much less detailed number of curves, the development of transformations to change the hull shape, by modifying the small number of definition curves, is quite feasible. Consequently, in comparison to previous parametric hull generation methods, the technique is considerably less complex to implement and much more practical to use. As the shape of the hull surface will not be directly dependant on numerical parameters, there is now the opportunity to make a fresh review of the parametric hull design concept, without needing to consider the practical limitations imposed by the traditional approach to these procedures.

14.2. The Selection of a Practical Set of Numerical Parameters

While naval architects have many ways of comparing ships numerically, the parameters of most interest are those that can be used to compare the viability of a vessel. These are the parameters that the owners and operators are most interested in when reviewing their own or competitors ships with a potential design. Subsequently, the most important performance parameters will be service speed, cargo capacity and passenger numbers. As the mathematical relationships between these performance parameters and geometric measurements of the hull surface are complex and may be based on additional non-hull form related factors, it would be impossible to develop a practical parametric hull design tool incorporating these parameters directly. Although, if used as part of a larger, more integrated design platform, such as a Blackboard system, these parameters may have more relevance. As the relationships between these important parameters and the design are too complex to be modelled directly, at least with current technology, naval architects have developed simpler rules or techniques that estimate the performance of a vessel. For example, resistance interpolation techniques, such as the Delft Series, allow yacht designers to predict speed performance based on data taken from model experiments. Considering, that most of these techniques base predictions on the standard geometric hull design parameters and that the TSCAHDE approach is being aimed at the concept and initial design phases, no overall benefit would be achieved by introducing indirect hull form design parameters.

As a practical example of a possible design scenario, in the initial stages of the design of a Ro-Ro vessel, the deck capacity is going to be of the highest interest. A rectangular deck shape is going to be the ideal shape to allow the best stowage arrangement to be achieved. Globally, the hull can be controlled using length and breadth parameters. Fine tuning can be achieved by adjusting the extent of the parallel deck. Any additional control would not be of any great benefit at the initial design stage.

Based on the arrangement of definition curves that are used to construct the hull surface, a complete set of practical design parameters can be derived to control the shape of a hull form. These parameters can be divided into four groups based the nature of the effect controlled (Table 14.1).

Main Dimensions	Boundary or Regional Dimensions
LBP – Length between perpendiculars LOA – Length overall DWL – Design waterline length BWL – Breadth at waterline T – Draught D – Depth	PMBF – Forward extent of Parallel Middle Body PMBA – Aft extent of Parallel Middle Body PMB – Extent of parallel middle body (PMBF – PMBA) PDF – Forward extent of Parallel Deck PDA – Aft extent of Parallel Deck PD – Extent of Parallel Deck (PDF – PDA)
Volumetric or Hydrostatic Dimensions	**Local Dimensions**
DISP – Displacement CB – Block Coefficient C_B (Note: $C_B = C_P C_M$) CP – Prismatic Coefficient C_P CM – Midsection Coefficient C_M LCB – Longitudinal Centre of Buoyancy	Bilge Radius, Rise of Floor, Tumblehome etc.

Table 14.1, The range of practical design parameters that can be used parametric hull design, with abbreviations.

14.3. Geometric Definition of Parameters

While it is possible to calculate the value of almost all of the numerical parameters directly from the surface definition curves, the fact that the definition curves are user manipulated leads to an increased risk that the parametric values could be incorrectly calculated. Rather than develop a detailed technique which checks user input to minimise the risk of incorrectly calculating the value of a parameter, the values can be found during the calculation of the hydrostatics. As the calculation of the hydrostatics is directly related to the hull surface, any strange values will be due to inconsistencies in the final hull surface rather than in the user entered data.

There are many approaches to calculating the hydrostatics of a single NURBS surface. A panel based approach was selected in preference to hull sections or a more direct calculation approach. Both the hull section and panel based hydrostatic calculations are very reliable and well-tried techniques compared to the direction approach taken by Sanderski [32]. There are many ways of calculating surface contours to find the hull sections. Throughout, the development of YachtLINES and ShipLINES many contouring methods were tested to find the technique which produced the highest quality and fastest execution. Hull design tool developers have obviously managed to solve this problem very effectively and robustly, as most tools can produce high quality sections very quickly without any noticeable errors. However, these techniques are not published and developers will not release any details on internal processes. Most academically published techniques, while producing high quality results do not seem to be robust or fast in execution.

As PolyCAD [50], the tool in which YachtLINES, ShipLINES, and TSCAHDE are implemented, supports many different solid and surface entities, a generic approach is adopted to allow contours to be calculated from any representation. The surface representation is first converted to a triangular facet representation with the density of facets reflecting the quality of contours. Then, each facet is analysed to find intersections with the contour planes [52]. If there is an intersection, the intersection line is calculated and stored. After the facets have been analysed, the intersection lines are combined together to produce the contour lines. The technique can not be considered the most elegant method and there is a great need for optimisation, however, it is robust and it does find all contour loops without any additional processing. As the calculation of surface contours is quite considerable and facet panel are calculated as one of the intermediate steps, the best solution is to calculate the hydrostatics using a discrete surface integral approach on the individual facet panels, trimmed to the waterline. Checks on both panel and hull section based hydrostatics found that there was almost no difference in the accuracy of the two techniques, although it should be noted that an adaptive hull section generation technique, similar to the approach used in NAPA [35], was used to minimise the inaccuracies in the integration algorithm. Further details of the hydrostatic calculation procedures can be found in Appendix 5.

The complexity required to identify local and regional parameters in the panel hull representation makes it necessary to calculate these parameters directly from the definition curves. There are only a small number of regional parameters controlling the extents of the parallel middle body and deck. As these parameters are based on two easily identifiable curves, the value of parameters can be found rapidly. Any local parameters that have been defined require a procedure to locate the correct definition curve and calculate the relevant dimensions. As this could require some complex analysis of the hull definition, it may be preferable to develop local features manually.

14.4. Parametric Modification of Hull Dimensions

There are many possible approaches to implement parametric modification. The standard geometric hull transformations could be implemented, however, the approach would not offer any technical advantages over existing manual hull design tools with hull transformation functionality. The definition curve structure provides an extremely flexible tool for changing the shape of the hull. The separate definition curves provide the ability to change the shape of the hull surface without having to tidy and fair in the surrounding surface to the changes, because the surface interface framework maintains the integrity of surface shape. Consequently, it is only necessary to use basic translation and scaling transformations on individual curves to realise a very powerful modification technique that can achieve the desired parametric change while maintaining the characteristic shapes within user defined curves. The introduction of compound transformations provides the freedom to design hull form modifications that have few restrictions due to the nature of the surface definition or individual transformation functions. The crux of the technique is in the selection of what parts of the surface definition to move, to scale or to leave alone.

As transformations are selectively applied to the hull form definition, each parametric modification must be defined separately. Each modification scheme can be developed by considering the most likely design changes that would require the implementation of the transformation. Three examples of parametric modification will be considered to illustrate how to define parametric modifications using this approach.

1. Lengthening through the increase of LBP

   The most common technique of lengthening a ship hull form is to increase the parallel middle body. Using the arrangement definition curves, this can be simple achieved by translating all curves forward of the midship section by the increase in length and the midship section curve forward by half the increase in length, (Figure 14.1). If there is no existing parallel middle body, then if the curves are available, i.e. the parallel section definition curves exist, then parallel middle body will be introduced. If not, then the final option available is to scale all the curves. The process can be reversed if the surface is to be shortened. The length of the parallel middle body would first be reduced and if this were not sufficient, a scale transformation can be applied to the data.

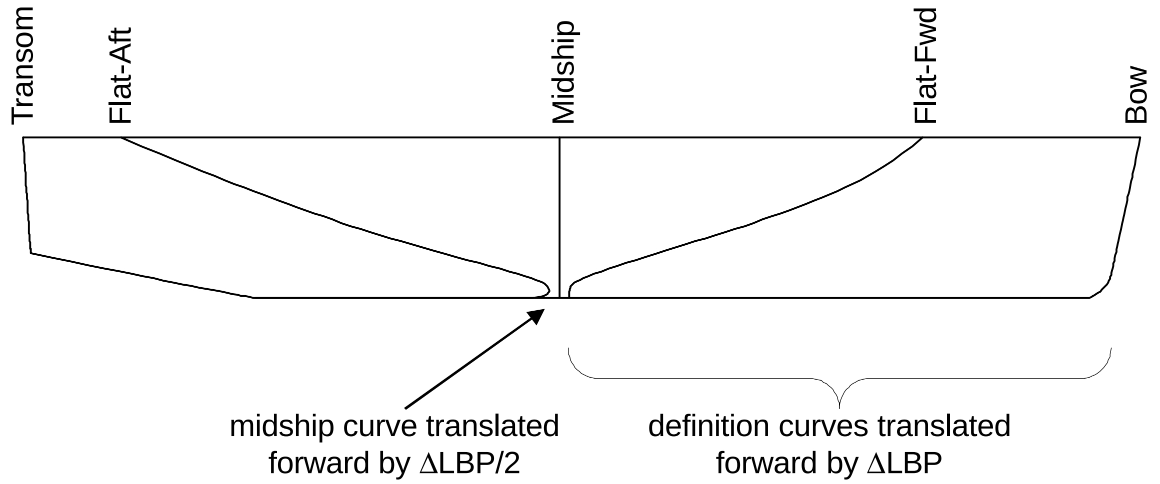


Figure 14.1, lengthening transformations can be constructed from simple concepts. Transformation of a good hull surface definition technique requires very little geometric information to be moved.

2. Changing the beam

   Changing the beam of a vessel defined within existing software tools has always been applied using a standard scaling transformation. However, as illustrated earlier in Chapter 6, this results in undesirable distortion to some hull shapes particularly the bilge radius. A compound transformation, (Figure 14.2), can be used to implement the changes in beam, eliminating distortion.

The curves defining the parallel section of the hull form lie on the same prismatic surface shaped by the midship curve and bounded by the FSF (Forward Surface Flat) and ASF (Aft Surface Flat) definition curves. Consequently, any changes to the midship section shape should be reflected throughout all curves between the FSF and ASF curves inclusive. As these curves, when projected on to the transverse plane, have the same y-z definition, the same transformation scheme can be applied to each curve. Therefore, the modification of the beam becomes a transformation of only one curve shape. Furthermore, in the next chapter, a tool is defined that can be used to relate the transverse shape of all curves on the hull prism to the midship section curve. However, as this is an optional tool, the process assumes that all curves are manually defined and applies the transformation to all the relevant curves.

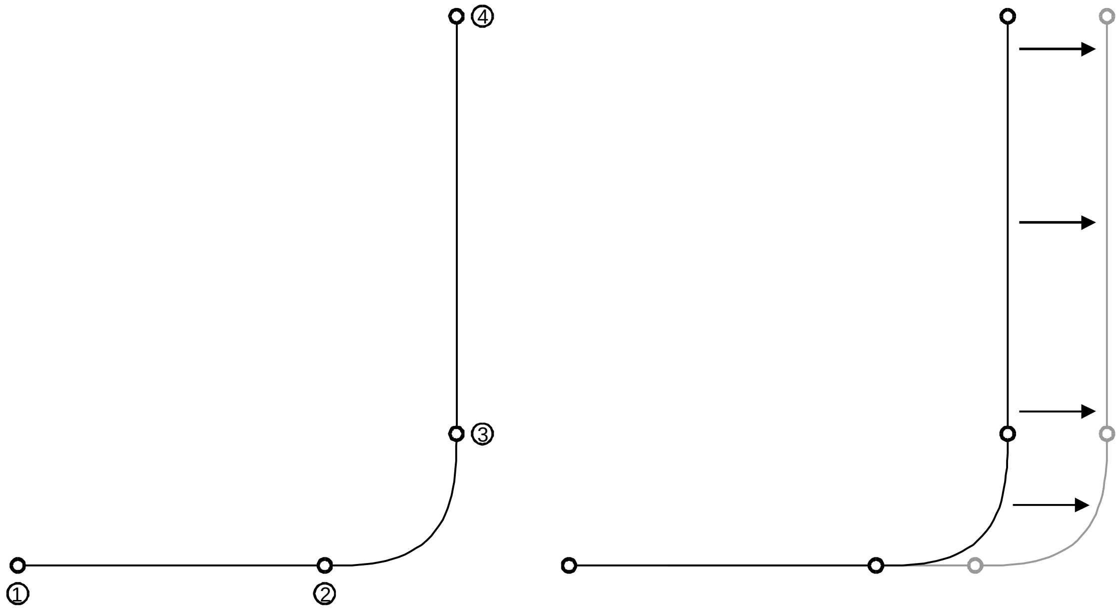


Figure 14.2, compound transformations, where different transformation are applied to the hull definition data selectively, can be used to minimise distortion and modify the surface in a way that is directly compatible with the reasons for instigating the change.

Ship hull sections, for practical reasons, have evolved in to a shape that can be divided in to three segments, (Figure 14.2), the bottom flat (1)-(2), the bilge radius (2)-(3), and the side flat (3)-(4). When the midship section is condensed to this simple context, the obvious way that the beam should be increased is to translate the bilge radius and side flat segments by the increase in beam, scaling bottom flat to fit. The compound transformation is implemented by identifying the control vertex of the B-spline curve that represents the inboard end of the bilge radius arc, scaling vertices before this point and translating vertices after this point. This compound transformation can be further improved by identifying features in the bottom flat, such as a keel flat, which should not be scaled. Changes in the depth of the hull surface can be implemented in a similar fashion.

While this transformation process is simple to understand, the identification of the three segments is not so straightforward, a shape recognition procedure is required. The recognition process can be simplified by reviewing the B-spline control polygon, rather than the actual curve representation, to identify the three segments. If the curvature of individual control polygon vertices is analysed by considering the radius of a circle defined by the three adjacent points, then the location of the three segments of section shape becomes immediately clear, (Figure 14.3). However, as ship sections may have additional features in the sectional shape, such as knuckle points which may mislead the identification procedure, any duplicate vertices are not considered. The three segments in the section shape are identified by comparing the calculated curvature against a template model. The complete identification procedure is presented in Appendix 5.

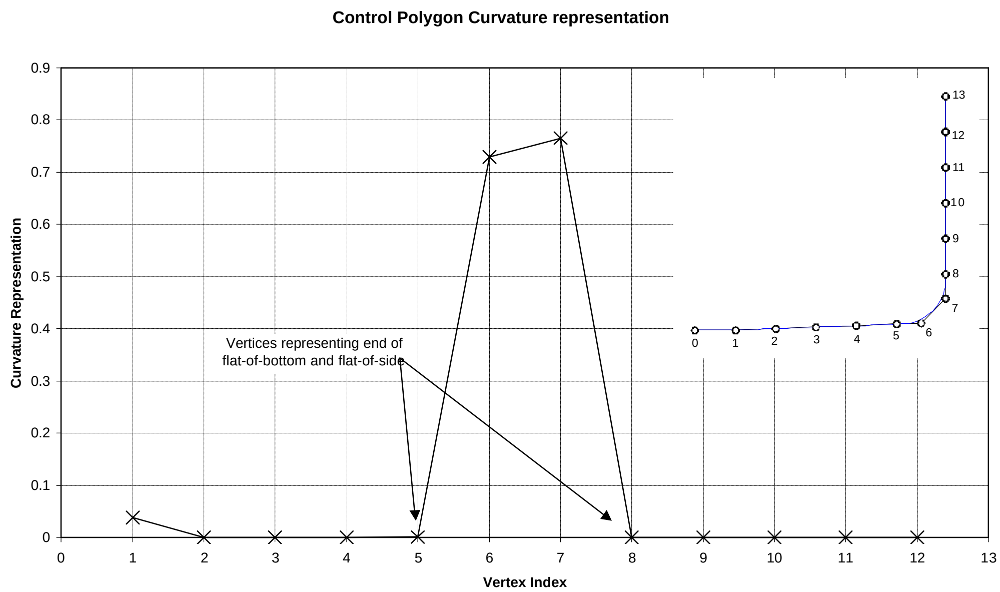


Figure 14.3, the identification of the vertices about which to apply the scale and translation transformations. Identification uses analysis of curvature type information in the control polygon.

3. Changing PDF

Once the technique of compound transformations applied selectively to surface definition curves is introduced, it is possible to use the approach for the other parametric modification transformations. The final example shows how the position of the forward extent of the parallel deck can be modified to change the deck stowage capacity. As this modification needs to maintain any characteristic shape in the definition curve, all vertices on the flat-of-side must be included in the transformation. By longitudinally scaling the curve shape in the segment between the vertices defining the forward extent of the parallel middle body and the forward extent of the parallel deck, the curve characteristic can be maintained, (Figure 14.4). Each definition vertex is longitudinally moved by the ratio of the z coordinate to the vertical distance between the forward deck point, Z_2, and PMB point, Z_1.

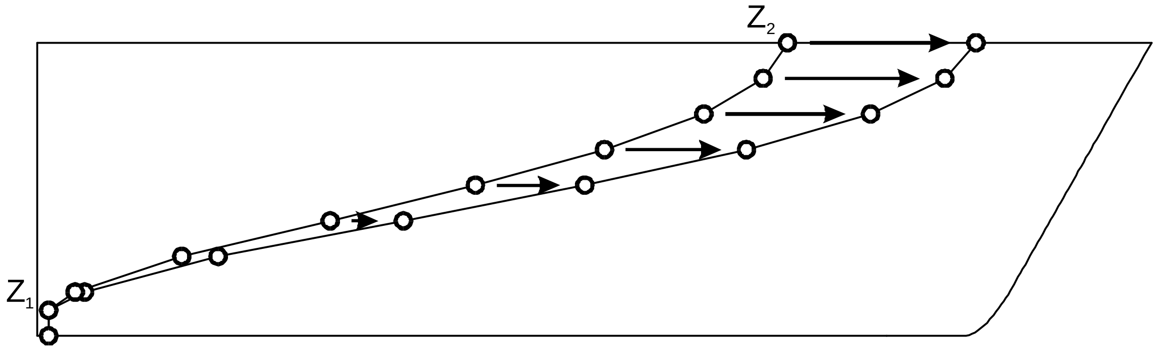


Figure 14.4, if a simple transformation approach is taken, modifications can be made which achieve the dimensions desired by the user and maintain characteristic shape designed in the surface definition by the user.

14.5. Parametric Modification of Hull Hydrostatics

While changing hull surface dimensions can be greatly simplified by using the definition curves and the surface interface framework, control of the hydrostatic properties must be approached with care to achieve a practical solution for this type of parametric modification. Traditionally, parametric hull generation tools have developed the desired hydrostatics properties by deriving a section area curve, which is then used to produce section shapes. This approach has two models of the hull form, the mathematical formulation that generated the two dimensional section area curve and the three dimensional hull surface. As the section area curve is usually a function designed to produce the required shape, the relationship between the two models is purely numerical. Given that there is not a physical shape-based relationship between the two different hydrostatic models, will they always agree at any longitudinal location? Frequently, additional parameters must be included to improve the flexibility in the hull shape definition, such as higher moments of volume. However, these types of parameters often relate to a mathematical property of the volume rather than an effect that can be visualised by the designer. Consequently, the control of these parameters is non-intuitive.

A preferable approach is to base the hydrostatic properties on the actual hull surface representation. However, this introduces a different set of problems. It becomes necessary to introduce an iterative technique to adjust the hull representation to achieve the desired hydrostatic properties. This introduces additional complexity into the hull generation procedure and significantly increases the time required to produce a surface. The performance of the YachtLINES [53] hull generation technique indicates that the use of iterative procedures within hull generation techniques must be approached with great care. YachtLINES is often unable to produce a hull form because the iteration process develops invalid hull shapes, corrupting the calculations.

YachtLINES uses a relatively basic procedure to produce the desired hydrostatic properties. The technique creates hull stations using the local section area coefficient as an input parameter. Once a midship section has been generated, the iteration technique modifies the local section area coefficients (CX) at the quarters stations (5 and 15) until the target hydrostatic properties are met. Two iteration functions are used to adjust parameters driving the volume and longitudinal centre of buoyancy characteristics, P_{VOL} and P_{LCB} respectively. P_{VOL} and P_{LCB} are used differentially to control the local section area coefficients:

$$CX_5 = C_M(P_{VOL} - P_{LCB})$$
$$CX_{15} = C_M(P_{VOL} + P_{LCB})$$

While this approach is able to find a solution, the model assumes that the buoyancy distribution in the Entrance and the Run is similar, which, for a yacht hull form, is far from the case. Consequently, P_{VOL} and P_{LCB} cannot be considered independent parameters. Moreover, the technique completely relies on the abilities of the iteration procedure to find the solution. The iteration has to be performed sequentially because of the dependency between control parameters. Only one parameter can be changed per iteration step otherwise the iteration procedure takes an incredibly long time to find a solution.

The iteration process uses local section area coefficient to drive the shape of the quarter stations. However, the definition of local section area coefficient gives no indication on the range of the parameter. An iteration process is again used to generate station shapes. It must generate a section with the specified area while trying to maintain an appropriate shape. As there are no constraints on the range of the quarter stations local section area coefficient, undesirable shapes often result. The technique cannot interrupt the iteration process to indicate that generated sections are undesirable because the procedure, based on a linear interpolation, would not function effectively. The results of the previous iteration steps used in the interpolation have to be discarded every time an undesirable shape is generated to allow the whole procedure to be reset. Consequently, the iteration may end up in an infinite loop.

The YachtLINES hull generation technique is very cumbersome to use and practically useless for design purposes. The major flaw in the approach is the reliance on the iteration technique to find the solution without any additional information to aid the process. The problems faced when relying on a pure iteration approach to find a solution, based on the experience of the YachtLINES technique, can be summarised as follows:

- No control of parameter ranges leads to the inability to control undesirable hull form shapes.
- Inappropriate design of hull shape parameter control functions leads to inefficiencies in the iteration procedure.
- Dependency between control parameters applies constraints to the design of the iteration process, i.e. only one parameter can be adjusted per iteration step.
- The iteration process is just not capable of efficiently achieving a parametric solution and maintaining a desirable hull shape at the same time.

For parametric hull modification within TSCAHDE, an iterative technique is, unfortunately, the most practical method. More elaborate techniques may limit the designers ability to manually interact with the hull definition. To implement a procedure that can modify the hull form definition to achieve the desired hydrostatic qualities, it is necessary to resolve the problems that make the YachtLINES iteration technique so unwieldy. An approach where the iteration process is heavily supervised to prevent it having ultimate control over the shape of the hull surface needs to be adopted.

The most critical factor reducing the performance of the YachtLINES technique is the close relationship that the iteration procedure has with the hydrostatic properties and the resulting hull form shape. Hull form shape is too complicated to be controlled by iteration routines. By separating hydrostatic modification and hull shape control into two separate tasks, shape can be managed by more rigorous techniques than iteration. As hull shape can be controlled using automatically generated definition curves, developing a procedure to achieve the target hydrostatics now becomes a much smaller modular problem. However, the effectiveness of the system is reliant on the design of the interface between the hydrostatics transformation evaluation and the hull shape control modules, (Figure 14.5).

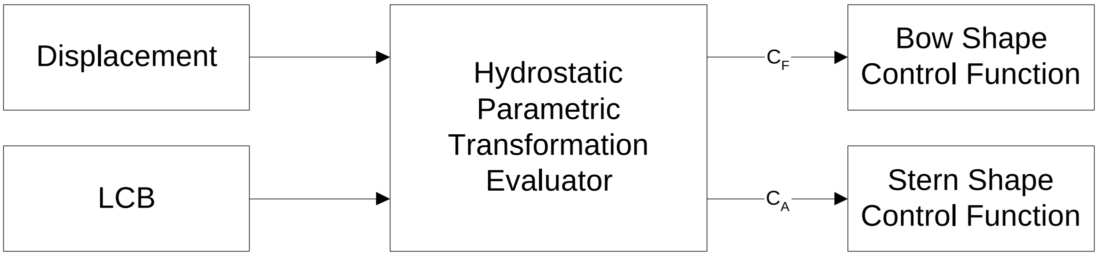


Figure 14.5, if the hydrostatic parametric transformation evaluator is modularly separated from functions controlling the hull shape, the design of the interface between modules becomes a critical factor.

YachtLINES controlled the volumetric characteristics of the Entrance and Run of the hull form using the local section area coefficient. However, valid ranges for this parameter are a function of the shape of the hull and as the shape of the hull is being varied by the iteration procedure, the valid range cannot be established. A circle of dependency exists between hull shape and parameter range. Removing the responsibility of hull shape quality away from the iteration procedure requires an interface to be designed to enable the hydrostatic transformation routines to influence the shape of the hull surface. An interface can be designed using more abstract control parameters, designed to have a specific range. Consequently, the iteration procedure will be able to know, immediately, by interrogating the value of the interface parameter, whether the hull shape will be valid without having to generate and analyse the surface. It can take preventative action in a much shorter period of processing time. The ideal parametric information to send to the shape control functions is a representation of minimum and maximum volume. This information can be limited to a specific range, using for example, a coefficient type parameter with values ranging from zero to one. Two volume coefficient parameters can be defined, C_F and C_A for the Entrance and the Run respectively. The responsibility for producing appropriate hull surface shape with respect to this parametric information now lies with the shape control functions. These functions must identify the limiting geometric surface shapes of the hull form to enable a mapping from each volume coefficient to hull shape to be constructed.

Having defined specifications for the output of the iterative hydrostatic transformations, the iteration technique, itself, must be designed so that it can function within the constraints defined for the input and output information. One of the primary problems influencing the efficiency of the YachtLINES iteration technique is that it uses dependent control parameters. The independent parameters, C_F and C_A, controlling shape within the Entrance and Run of the hull surface both have influence on the Displacement and the Longitudinal Centre of Buoyancy (LCB). A technique is required which allows the Displacement and LCB to be controlled using independent iteration functions, enabling the Displacement and LCB control parameters to be updated on every iteration step. Therefore, the surface solution should be found much more quickly.

It is impractical to build a model of the hydrostatics properties from the actual hull surface. There would be just too much calculation involved. However, by assuming that the parameters affecting hull shape, C_F and C_A, have independent control of the surface shape and subsequently volume, at the respective ends of the vessel (Figure 14.6), a simplified model of the section area can be developed (Figure 14.7).

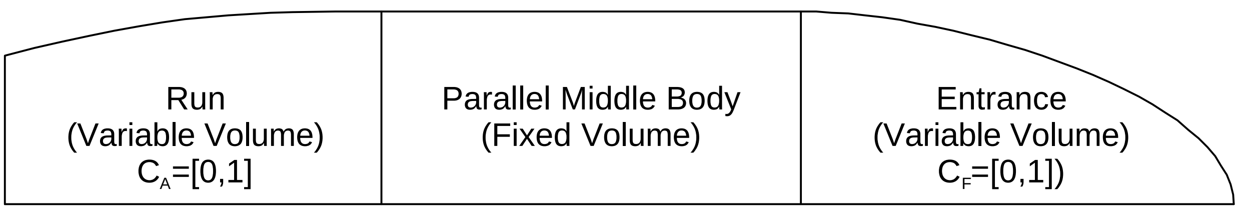


Figure 14.6, the assumption can be made that C_F and C_A have independent control over the shape and volume at the ends of the vessel. Consequently, a simplified model of the section area curve can be developed (Figure 14.7).

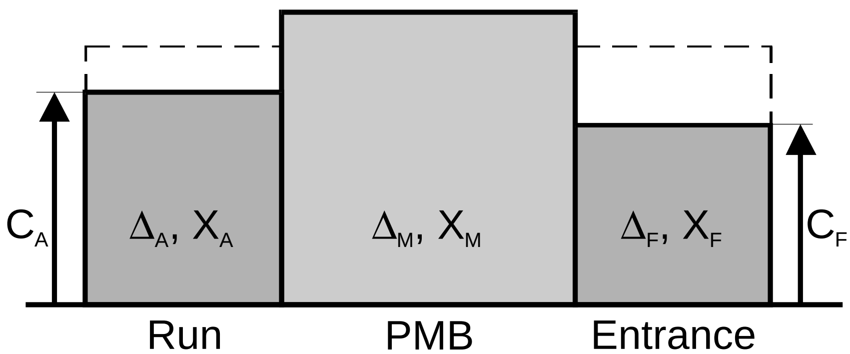


Figure 14.7, the simplified model of the section area curve.

The model uses the assumption to represent the curve using three separate parts, representing the Entrance, Parallel Middle Body and the Run. As the volume of the Entrance and Run is controlled directly by C_F and C_A respectively, the relationships between Displacement and LCB, and C_F and C_A can be defined as follows:

$$\Delta_T = C_F\Delta_F + \Delta_M + C_A\Delta_A$$
$$\Delta_T X_T = C_F\Delta_F X_F + \Delta_M X_M + C_A\Delta_A X_A$$

Where: Δ is Displacement, X is Longitudinal Centre of Buoyancy (LCB) and subscripts T, F, M, A, refer to the Total, Forebody, Midship and Afterbody volumes respectively. These functions can be rearranged to obtain C_F and C_A from Displacement and LCB:

$$C_F = \frac{\Delta_T X_T - \Delta_M X_M - X_A(\Delta_T - \Delta_M)}{\Delta_F(X_F - X_A)}$$
$$C_A = \frac{\Delta_T - C_F\Delta_F - \Delta_M}{\Delta_A}$$

When compared to the actual hull surface, the simplified approach models the abilities of the modification procedures closely, although the analogy of the block shaped section area curve does not fit. Δ_M represents the minimum displacement of the hull surface, i.e. $C_F = 0$ and $C_A = 0$. This minimum volume will include the Entrance and Run for the smallest volume possible. Subsequently, Δ_F and Δ_A represent the maximum increase in volume that can be made to the Entrance and Run respectively. The major deficiency of the model, disregarding the linearity, is the assumption that the location of the Entrance or Run Longitudinal Centre of Buoyancy is constant. However, when the model is used in conjunction with the iteration process, the shortcomings of the model have no significant effect on performance.

One of the major advantages of the model, besides removing the dependency between control variables, is that it creates a guide for the solving process. Once the model is created, it can be used to approximate C_F and C_A. The iteration procedure is used to refine the model, arriving at a solution by identifying the error between the model and hull surface hydrostatics. It modifies the desired (Target) parameters input into the simple hydrostatic model by the difference between the calculated (Calc) and predicted hydrostatics from the previous iteration step:

$$\mathrm{ModelDISP}_i = \mathrm{TargetDISP} - (\mathrm{CalcDISP} - \mathrm{ModelDISP}_{i-1})$$
$$\mathrm{ModelLCB}_i = \mathrm{TargetLCB} - (\mathrm{CalcLCB} - \mathrm{ModelLCB}_{i-1})$$

Consequently, the hydrostatic model becomes more important to the performance of the modification process than the iteration procedure. In contrast, the iteration approach used in YachtLINES searches for a solution using linear interpolation of the results of previous steps. It does not have any technique to guide it to a solution using expert information from a hydrostatics model.

To function, the simplified hydrostatic model requires some boundary information from the hull surface. The modification technique must perform some preamble calculations, constructing the model, before the iteration process can begin. The pre-calculation obtains the constants required for the model and identifies the rectangular extent of the solution domain, (Figure 14.8). The domain information is used to check that the desired hydrostatics are within the extreme limits of the model. However, the iteration procedure may later identify that only a partial solution may be achievable. Model constants are identified by considering the minimum and maximum parametric combinations of C_F and C_A.

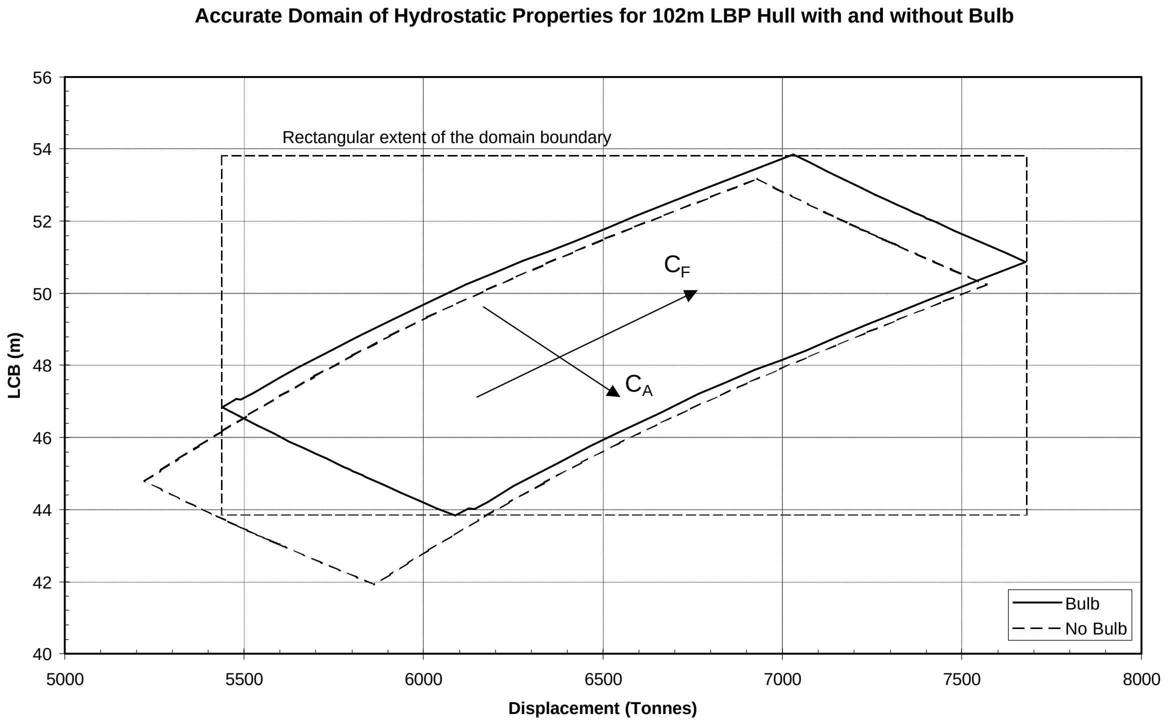


Figure 14.8, accurate domains limits of a hull form with and without the bulbous bow. The initial pre-calculation only identify the extreme rectangular limits of the domain. Consequently, the iteration procedure may only be able to achieve a partial solution.

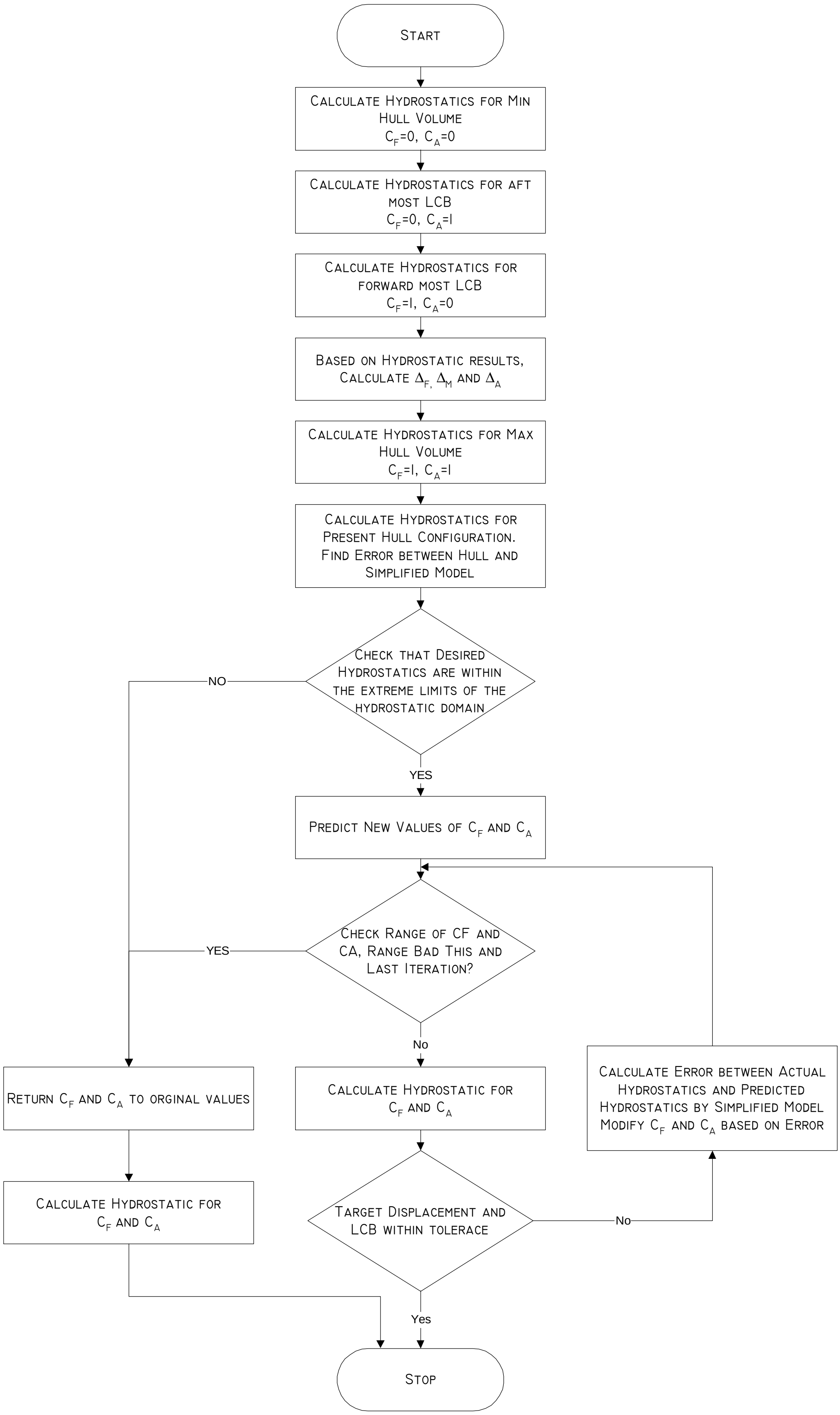


Figure 14.9, the hydrostatic parametric modification procedure.

As the iteration process is not used to create a hull form, only to modify the definition, a hull surface exists before the procedure begins. Consequently, the process can obtain the first approximation of the error by calculating the difference between the existing hull surface and the appropriate hydrostatic model. This information is extremely useful, without it, the process would have to identify some starting location for the error values and further iteration steps would be required to obtain a useful order of error. The hydrostatic modification process used in the implementation of TSCAHDE is detailed in Figure 14.9.

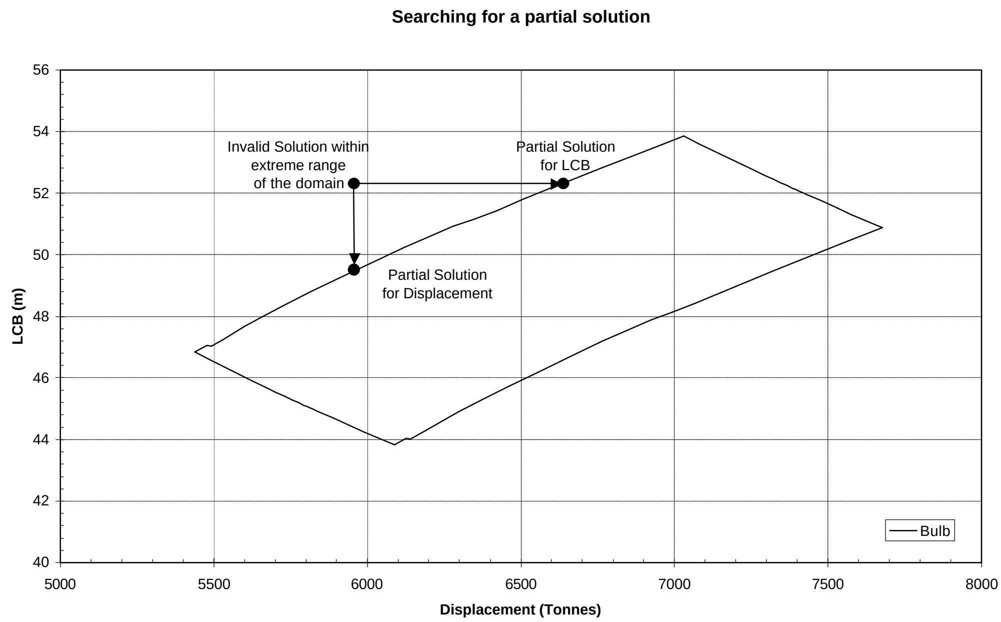


Figure 14.10, partial solutions are achieved by allowing the one of the parameters to be floated on to the domain boundary.

The initial limits calculated by the process represent the extreme range of Displacement and Longitudinal Centre of Buoyancy. The checks of the desired hydrostatics against the possible range of solution do not take account of the relationship between Displacement and LCB. This would require the technique to make a detailed trace of the domain boundary. This can be performed, however, it is impractical to do so. The boundary trace would take longer to calculate than to perform the actual modification. Consequently, the iteration procedure has to identify when some of the parameters can not be achieved. When the desired solution is close to the domain boundary, the iteration procedure may have to specify control parameters that are known to be outside the specified valid range, to close in on the solution location. However, to identify solutions that are outside the domain, the criteria that no two consecutive iteration steps should be outside the domain is specified. Once it has been identified that a solution cannot be reached, the modification procedure can offer to search for a partial solution. Designing the user interface so that the user can only modify one hydrostatic property at a time, either Displacement or LCB, allows the iteration procedure to provide partial solution by achieving the desired value of the modified parameter and allowing the other parameter to float onto the boundary of the domain, (Figure 14.10).

Removing the responsibility of surface definition from the parameters of a hull generation system allows a much more adaptable design solution to be constructed. This cannot be achieved without developing a surface definition technique that is capable of constructing a hull surface with a relatively small amount of definition information, in comparison to the quantity of definition data required for a pure NURBS surface. The benefits of this arrangement can be seen in techniques that can be used to transform the hull definition. It is not necessary to use a single transformation function that affects all definition data. Furthermore, as parameters are not a requirement, the implementation software can be selective about the parameters it displays, hiding those that are not relevant. A more elaborate implementation may even allow the user to define custom parameters and transformations.

The consequence of this new approach is that the design of definition curve functionality has become the most significant factor affecting the success of the approach as an effective design tool.

15. CURVES AND THE APPLICATION OF GEOMETRIC CONSTRAINTS

15.1. Curves in the Control of the Hull Surface

The number of hull design suites using curves to control surface shape is slowly reducing as the NURBS revolution continues. Control curves have many advantages over direct surface manipulation when trying to control the shape of a surface, especially a hull form. The route that a curve takes through Model space is much easier to understand than the shapes of a surface. In fact, hull design applications have to convert the surface representation to sets of curves or contours, to allow the shape of the hull form to be properly appreciated. A surface representation of the hull form is now an absolute necessity in modern ship design. However, this should in no way prohibit the use of curves to control the shape of the surface during design.

Many hull design tools provide the user with curve representation tools. The relational geometry concept has been one of the more successful techniques for combining curve and surface representations together. NAPA has one of the most advanced relational geometry implementations, allowing the construction of complex irregular curve meshes controlling surface shape, discontinuities and surface tangents. However, for hull design suites employing NURBS representations, the technique can only been used to control the locations of boundaries of surfaces with possibilities for applying simple tangent conditions.

One of the key aims of TSCAHDE is to develop an ability to produce a hull surface from, at minimum, the small set of curves representing the major features of the hull form. The surface interface framework develops the hull surface using these feature curves with additional curves being required to control the shape of the surface between the feature curves. The relatively small amount of definition data, in comparison to a complete NURBS surface control polygon, allows parametric modification techniques to be developed which change global dimensions of the hull form by modifying the control curves. However, despite the development of these facilities, the quantity of definition data is still great. The example hull form from Chapter 13, (Figure 15.1), requires thirteen definition curves. Five curves are used to control the features shapes of the hull form, with the seven remaining curves controlling tangents and hull volume. Around sixteen vertices are required on each curve for comfortable control of transverse shape. As all curves must have the same number of vertices, there are a total of 192 control vertices. Compared to the number of control vertices required for a NURBS surface representation of a ship hull this is low. However, this is still impractically large for a manually manipulated definition technique. In addition, as some curves represent tangents in the hull surface, vertices on these curves create effects that are dependent on the respective location of vertices on other control curves.

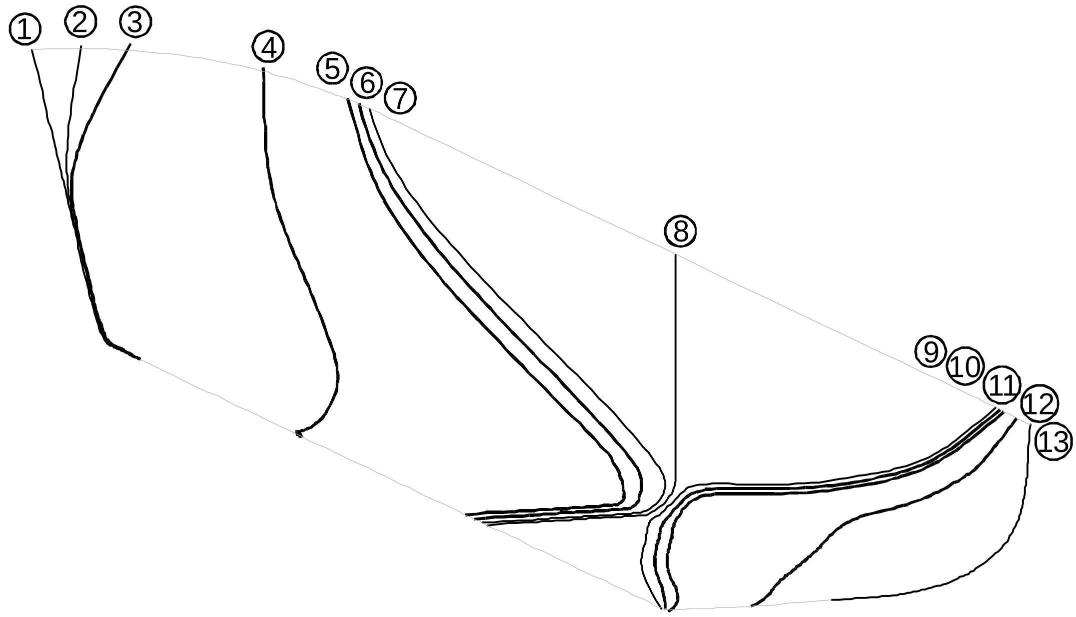


Figure 15.1, the surface interface framework defines the minimum requirement for thirteen curves to define a hull surface.

The main feature definition control curves represent boundaries of particular shapes in the hull surface. The form topology of the hull surface and the dependancies that exist between the vertices of control curves can be used to reduce the amount of definition information that must be provided by the user. Tools can be developed to enable the user to constrain curves to represention of standard hull form shapes and to generate additional definition geometry using the relationships that exist within the form topology definition.

For example, the corresponding control vertices on curves (5) to (11) all lie on the same line, parallel to the x axis, i.e. the value of the y and z components of corresponding control vertices are the same. The range of curves between (5) and (11) contain the FSF, ASF and midship section curves. By inspection, this part of definition represents the parallel section of the hull form. Consequently, the y and z components of corresponding control vertices should have the same value as the vertex on the midship section curve. Furthermore, modifications to the midship section curves should manifest throughout curves (5) to (11), with the vertices on curves (5) to (11), not including the midship section, being locked to changes in the y and z direction and free to move in the x direction, unless additional constraints are applied. A similar relationship is present at the bow between the boundary and tangent definition curves in the construction of a Softnose feature, i.e. the cylindrical waterline shapes entering the stem.

By allowing the user to define the form topology of the hull surface with geometric constraints, the number of manually controlled definition curves is greatly reduced. However, while the reduction of the amount of manually controllable definition is always appreciated by the user, it should not applied to the extent that the user no longer has control over the surface definition. A balance is required between the constraints applied automatically by the technique and those applied at the request of the user. This balance can be appropriately achieved by optionally applying constraints to manually defined curves and automatically applying constraints to generated control curves, using constraints applied to related manually defined curves as a basis. The definition of form topology enables the automatic generation of tangent, flat and volumetric control curves based on the shape of the manually controlled hull feature definition curves.

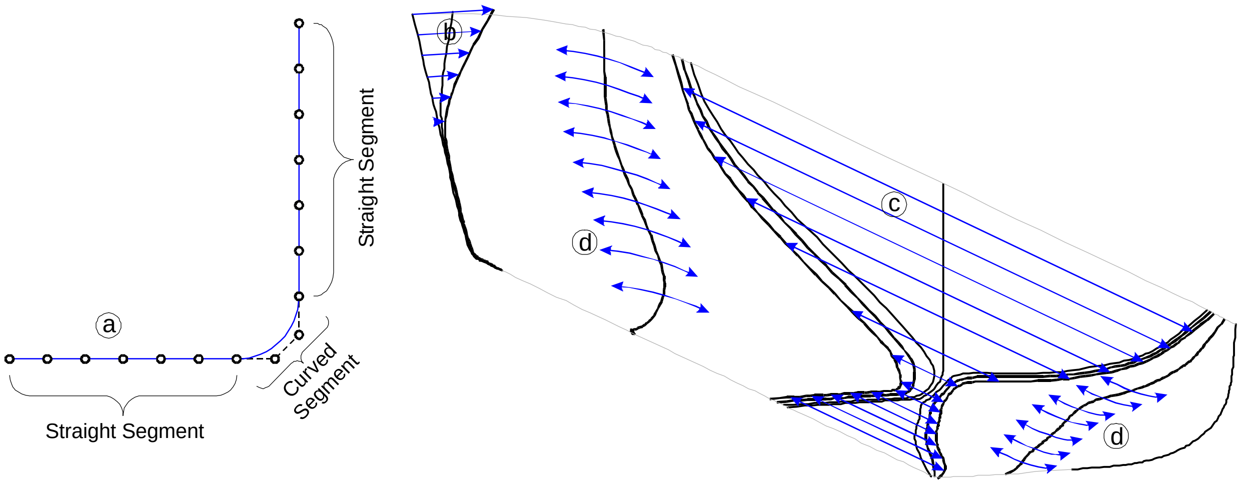


Figure 15.2, the constraint relationships that can be found within the hull surface definition using the surface interface framework.

Figure 15.2 shows the different types of geometric constraint relationships that can be created for curves used in the interface framework.

(a) Simple curve shape constraint tools

(b) Creation of surface tangency

(c) Control of planar regions in surface shape

(d) Control of the blending shape with respect to the desired volumetric properties

These relationships can be separated into two groups: simple relationships, where the position of vertices within a curve are controlled and more complex relations where additional geometry is generated to create the definition required to automatically implement the correct shape features in the surface interface framework defined by the form topology.

1. Individual Curve Shape Constraint Tools
    - Straight curve segment (known as the *Straight Modifier*)
    - Knuckle point (known as the *Knuckle Modifier*)
    - Blended curve segment (known as the *Curve Modifier*)
    - Planar curve (known as the *Plane Modifier*)
    - Axial offset of vertex locations from another curve (known as the *Offset Curve Modifier*)
2. Automatically Generated Shape Constraining Curves Tools
    - Construction of surface tangent representation curves
    - Construction of surface flat representation curves
    - Construction of volumetric control curves

15.2. Individual Curve Shape Constraint Tools

The primary task of the surface interface framework is to reduce the amount of longitudinal definition data required to represent a hull form surface. In the transverse direction of the hull surface, there is no reduction in the data required to produce the surface, ensuring that the designer has the maximum degree of flexibility to design the transverse shape of the hull form. However, most ship hull forms have basic shapes and do not require the high degree of flexibility in the transverse shape provided by the surface interface framework. It is possible to develop template curves based on a parametric definition, however, as previously discussed, this would be a mathematically generated curve and the user would have no way of manually interacting with the shape. While the parametric template curve has been dismissed as an appropriate tool for interactive design, it may become practical to use a curve defined using a template scheme if a purely parametrically generated hull form is desired.

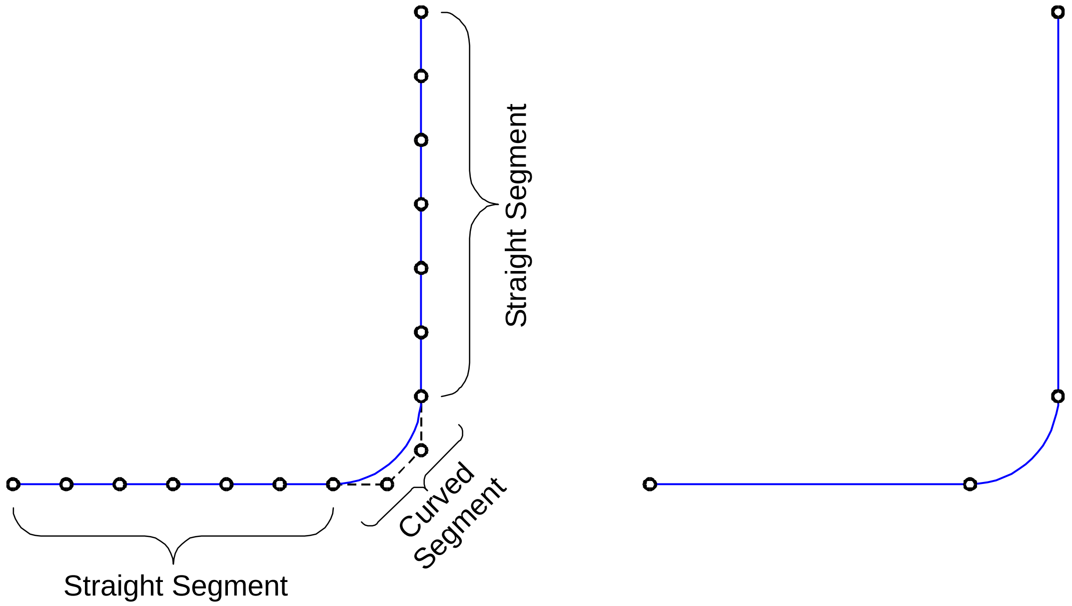


Figure 15.3, curve definition can be constrained using persistently applied tools, (left). If constrained vertices are hidden from the user, (right), the definition appears simpler and can be modified much more clearly.

A preferable option is to interactively apply persistent shape constraints to the local curve shape. Simple tools to produce particular curve shapes are not new. Many existing hull design software suites include tools that can be used to set particular shapes into definition curves. However, in the case of systems using NURBS representations, the designer can usually edit the definition destroying the constrained shape. The constraint produced by a persistent tool cannot be modified. Furthermore, if the display of the definition curve is modified so that constrained vertices are not visible, the curve shape appears to be much simpler, an example of such an arrangement can be seen in the midsection curve Figure 15.3. A wide range of constraint tools can be developed to aid the quick development of a hull form definition. However, as there are many different ways of defining the hull surface not all are appropriate for a particular definition tool. Five practical tools have been identified as being useful for developing hull surfaces in conjunction with the surface interface framework and the corresponding control curves.

15.2.1. A Straight Segment in a B-spline Curve or "*Straight Modifier*"

This is a very simple tool and extremely useful considering the amount of flat areas on a ship hull is large. A straight segment in a fourth order (control curves are fixed to cubic degree) B-spline curve is very easy to create. If the vertices between two vertices P_m, P_n, (Figure 15.4), representing the ends of the straight segment are placed on the line coinciding with P_m and P_n, by the property of the curve lying within the control polygon, the curve will coincide with this line between the vertices P_{m+1} and P_{n-1}. A tool can be developed to impose this feature on the curve by maintaining the control vertices internal to the straight segment at equal intervals using the following the relationship based on the Ratio Theorem:

$$P_I = \lambda P_n + (1-\lambda)P_m$$

Where I is the index of the vertex in the control polygon and,

$$\lambda = \frac{I-m}{n-m}$$

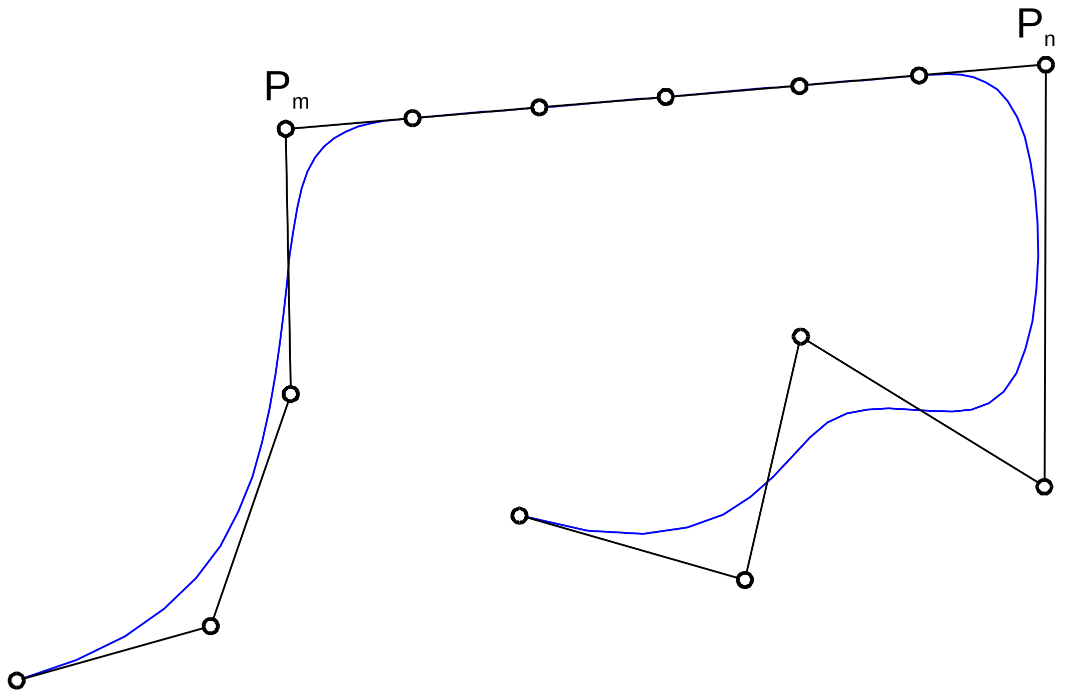


Figure 15.4, using the property of collinear vertices, a linear curve shape can be constructed by linearly constraining definition vertices.

15.2.2. The Knuckle Point or "*Knuckle Modifier*"

A knuckle point is sharp corner in a smooth curve. In a fourth order B-spline curve, this is created by making three control vertices coincident. Knuckle points can also be constructed using a non-uniform knot vector. However, as all control curves must have the same number of control vertices and knot vector, i.e. uniform, this approach cannot be used.

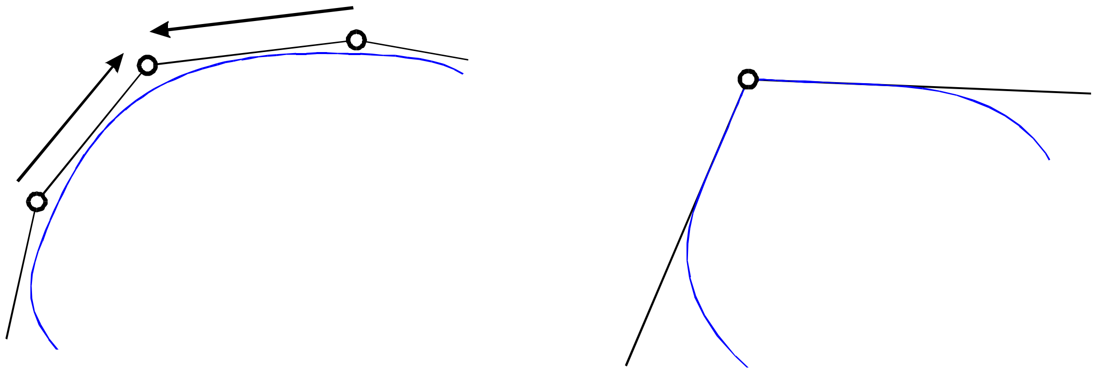


Figure 15.5, knuckle points can be constructed by making three adjacent points coincident.

15.2.3. A Blended curve segment or "*Curve Modifier*"

There are many occasions where it is necessary to develop a shape in a B-spline curve that smoothly transitions from one feature to another, a blend. Often the number of points that must be used to perform the blend is undesirable large to manually develop a smooth shape, particularly in the case of TSCAHDE where the number of control vertices on each curve must be kept consistent. It becomes necessary to use curvature analysis tools to ensure that all the control vertices are in exactly the right location to obtain a smooth and fair curve. This situation can frequently occur in the midship section at the bilge radius, where a circular arc connects the straight segments of the flat of bottom and the flat of side. While this shape can be developed very quickly in standard CAD drawing packages, using lines and fillets, this shape cannot be developed accurately in a basic B-spline curve. To create an accurate representation of a circular arc using B-splines, it is necessary to use the NURBS homogenous weights. However, as each control curve is limited to the same number of points, knot vector and weights, the full features of NURBS cannot be used at this level of the hull surface definition. However, as the tool is primarily aimed at the early stages of the hull development cycle, the lack of an accurate representation of circular shapes is not a major limitation and the blending procedure can be designed to approximate circular shape as closely as possible.

There are many ways of implementing blending procedures to develop closely circular shapes. Initially, the construction of circular shapes, when developing the TSCAHDE implementation, was of prime importance. However, as the circular blending algorithms became complex and limited the application of the constraint tool, a much more simpler and direct approach using cubic blending was chosen. Using a similar approach to the application of the constraint in the straight modifier tool, a cubic function is used to locate the position of the control vertices within the constrained segment, (Figure 15.6). As the tool is a similar style of constraint to the straight modifier, it is applied to the curve in exactly the same way. A complete description of the blending procedure used to develop the location of control vertices is detailed in Appendix 5.

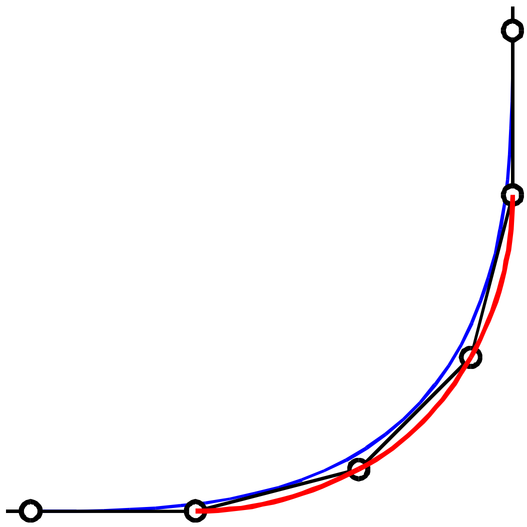


Figure 15.6, a controlled curve shape (blue) can be constructed in a curve by using a cubic spline (red) to locate constrained control vertices. The tangents of the cubic spline are controlled by the end segments of the control polygon within constrained region of the curve.

15.2.4. Curve located within a plane or "*Plane Modifier*"

The constraint of the curve to a plane is a rather trivial tool, however, it can be a great aid to the user when developing the initial definition geometry. It can be considered as a practical extension of the straight modifier tool. The tool can be used in many situations. For example, it can be applied to the midship section to ensure the curve remains in a specific transverse plane or for developing the shape of the transom at a incline. Furthermore, if the location of the curve needs to be changed, it can be interactively dragged or the constraint parametrically modified with the constraining relationship being persistently maintained.

An additional use of this modifier was discovered during the development of hull surfaces using the implementation. The plane modifier tool has additional numerical parameters that can be used to accurately locate the constraint. This feature is also very useful for interactively transforming a curve from one plane location to another, without changing the shape of the curve. For example, the midship section curve can be copied to create the bow profile. The plane modifier is used to rotate the curve from the transverse plane onto the centre plane. Mathematically, the tool is much easier to implement than it is to perform the modification by hand. Apart from the first point of the definition curve, which acts as the origin of the plane, vertices are projected on to the plane after any modification to the curve definition.

15.2.5. Axial offset of corresponding vertex locations from another curve or "*Offset Curve Modifier*"

The offset curve modifier is one of the tools highlighted earlier that can be used to maintain the location of vertices within a curve, orthogonally offset from the position of corresponding vertices in a reference curve. The previous example, Figure 15.2, showed how the vertices of the seven curves defining the parallel middle body of the hull shared y and z coordinate components. This tool takes advantage of the parametric independence between coordinate components in the B-spline function. The shape of the curve in the x direction is independent of the y, z coordinate components of the control vertices. Consequently, the same shape can be maintained across curves in the same plane, perpendicular to the axial direction in which the curves are offset.

This is a powerful tool and another that is implemented rather trivially. It is only necessary to copy the corresponding off axis vertex coordinates from each curve on the referenced curve to the referencing curve. The messaging system within PolyCAD [50] is used by the reference curve to communicate any changes in geometry. This feature is discussed in more detail in chapter 17.

The success of this tool is dependant on the constraint imposed on the similar definition geometry between definitions curves, i.e. the same number of control vertices and uniform knot vector. However, a compound B-spline curve function could be developed by projecting the reference curve on to a surface defined by a curve orthogonal to the reference curve. This technique is found in the NAPA [35] system, in the development of a curve defined with a Location Surface.

Practically, this tool only needs to be implemented in the primary axis directions. However, it would only require a small amount of modification to allow curves to be offset in arbitrary directions, perhaps control by a vector, although, the tool would require an exceptional control system to allow it to be used effectively.

15.3. The Application of Constraint Tools to Curves

These tools are very interesting features when applied individually. However, when used together, more complex curve shape constraints can be applied. Consequently, control of the interaction between these tools needs to be approached very carefully. In some combinations, the curve is constrained to the extent that no further constraints can be applied. These scenarios need to be identified to allow the software to feedback context-sensitive information to the user. Due to the sequential manner in which the constraint tools are applied to a curve, a protocol can be developed to ensure that the shape resulting from the application of these tools is always as desired and to prevent any invalid arrangements of constraint tools. The development of a protocol needs to take account of the needs and consequences of applying the constraint tools to a curve. Some of the needs of the modifier tools are listed below.

- The *knuckle modifier* gathers the adjacent points on each side of the main reference point together. Consequently, this modifier should be applied before those referencing the other points.
- If the *curve modifier* is applied adjacent to a *straight modifier*, it must obtain tangent information from the structure of points created by the *straight modifier*. Consequently, the *straight modifier* must be applied before the curve *modifier*.
- The application of the *offset curve modifier* and a *plane modifier* results in curve that is the projection of the *offset modifier* reference curve on to the plane. Consequently, every vertex on the curve is constrained to one location and no further modifier tools can be applied validly.

Using these rules, the order of precedence for the application of each tool can be developed. Furthermore, query functions can be specified to allow the system to check which constraint tools are currently being applied to a curve. These features are used in combination with vertex tagging, a technique that sets a vertex-stored flag based on the constraints that are being applied to a particular vertex. All this information can be combined to produce a decision table, (Table 15.1), which is used to implement the technique considering the precedence and interaction requirements of each constraint tool. The table identifies two groups of tools based on the extent of the application of the constraints. Vertex level constraints influence only a specified set of control points. Curve level constraints apply constraints that affect the whole shape of the curve.

Rules/Modifiers	Vertex Modifiers			Curve Modifiers	
	Straight	Knuckle	Curve	Offset	Plane
Application	Cannot apply a straight modifier across a knuckle modifier unless the knuckle is on the offset curve parent. ***Define Query: HasStraight***	Cannot apply knuckle modifiers to locked offset curves ***Define Query: HasKnuckle***	Cannot apply a curve modifier across a knuckle modifier unless the knuckle is on the offset curve parent. ***Define Query: HasCurve***	Applying a offset modifier to a curve containing a plane modifier disables all other modifiers. ***Define Query: HasOffset*** ***Define Query: CurveOffsetAxis***	Applying a plane modifier to an curve containing a offset modifier disables all other modifiers ***Define Query: HasPlane***
	Define Query: HasPointModifiers			A curve with an offset and plane modifiers is a special case, the offset curve must be projected on to the plane through the relevant axis. The resulting curve is no longer modifiable because of the intersection. This solution becomes a projection modifier. ***Define Query: HasProjection***	
Interaction	Straight modifiers link to the outside points of a knuckle modifier. Straight modifiers are applied before curve modifiers to ensure tangents are correct ***Sort Order: 3***	Knuckle modifiers are applied before Straight and Curve modifiers to ensure that all points are in position ***Sort Order: 2***	***Sort Order: 4***	Offset modifier must cause point modifiers to function only on the relevant axis. ***Sort Order: 1***	Plane modifier is applied before straight, radius, and knuckle modifiers. These modifiers should function within the plane without problem. ***Sort Order: 1***
Property Change				Cannot change properties once set.	Apply modifier as standard, however remove irrelevant planes when functioning as a projection modifier

Table 15.1, the decision table developed to build the protocols for modifier application and interaction.

While vertex level constraints are able to constrain the curve shape based on the geometry of adjacent control vertices, curve level constraints require additional parametric information to implement the restriction. As these parameters control very powerful constraints, the access to these parameters must be closely controlled to prevent accidental modification, which may destroy the whole model, in addition to any undo tool that may be provided by the implementation. For the *offset curve modifier*, the consequences of changing the offset axis are so serious that this parameter should be prevented from being modified after the initial state has been specified. Interactive modification of definition parameters is an important part of the use of the plane modifier, however this can also be used to ruin the hull definition. Consequently, the plane definition is restricted to a certain range of parameters to ensure that the user cannot apply too large a shape change in one operation. Furthermore, when the *plane modifier* is functioning with the *offset curve modifier* to produce a projected curve, the user should be prevented from selecting a plane axis that would be incompatible with the reference curve location. Care should be taken to ensure that the user is not over restricted. In the case of the *plane modifier*, the tool was found to be such a useful method of transforming curve shape that the modification of the plane definition parameters had to be kept unimpeded.

15.4. Automatic Curve Generation Tools – Automatically Generated Surface Shape Constraint Tools

While the simple and compact localised curve constraint tools are capable of creating the particular characteristics and features of a hull form, the optional format and localised application of these tools renders them inappropriate for applying the geometrically fundamental constraints required to develop the key characteristics associated with ship hull surface shape. Ship hull surfaces have three distinct regions of shape, the entrance, parallel middle body and the run. Development of these regions of shape require constraint tools able to construct the flat regions of the surface, develop controllable tangent shapes at the boundaries of, and within the hull surface and control the fullness of the hull shape with respect to the desired hydrostatic properties of the hull form. The flat and tangential hull features can be developed in the surface using particular structures of control vertices developed from the known properties of NURBS. However, as the structures required to implement these desired effect may involve multiple control curves, an entirely different approach is necessary to allow the application of these geometric constraints to the hull surface.

The surface interface framework is designed to accept the particular structures of points required to implement effects such as the hull flats, surface tangency and the control of the fullness in the ends of the hull form. It provides the facilities for these effects to be created in the surface given appropriate sets of control curves. The structures used to control the key geometrical characteristics of the hull surface can be grouped into two significant different classes. The structures developing the flat and tangent features use relative simple projection and alignment relationships between control vertices. In comparison, the control curve structure required to control the volumetric properties of the hull surface use considerably more complex relationships between vertices to develop a fair and appropriate surface shape using the single hydrostatic control parameter defined in Chapter 14. As both classes of control structure use mathematical relationships to locate the position of all control vertices, it should no longer be necessary for the user to provide the initial definition curves that will be used to develop the features. Automatically generated control curves will accomplish the task in a more refined way and will hide the complexity of surface construction away from the user. The resulting tool appears more concise in the use of definition curves and the user has a direct technique to control the essence of ship hull surface shape, the boundary shapes of the flats, the magnitude of the surface tangents and the fullness of the hull within the entrance and run.

However, while the automatic generation of definition geometry removes the complexity of surface construction, it should be kept in mind that the technique should only apply these constraints if the user has not supplied definition data to implement these effects. If the user supplies definition data in locations where generated data can appear, then it should be obvious that the user wishes to manually control this feature in detail. Furthermore, by allowing the user to extract generated control curves into the design environment, the user can refine the surface definition without having to replicate the generated geometry.

15.4.1. Tangent Feature Control Curves

The control of surface tangents is mainly important at the surface boundaries, the bow and the stern. There are surface tangents elsewhere in the surface, at the boundaries of the surface flat. However, these shapes are controlled as part of the Flat Feature control curve.

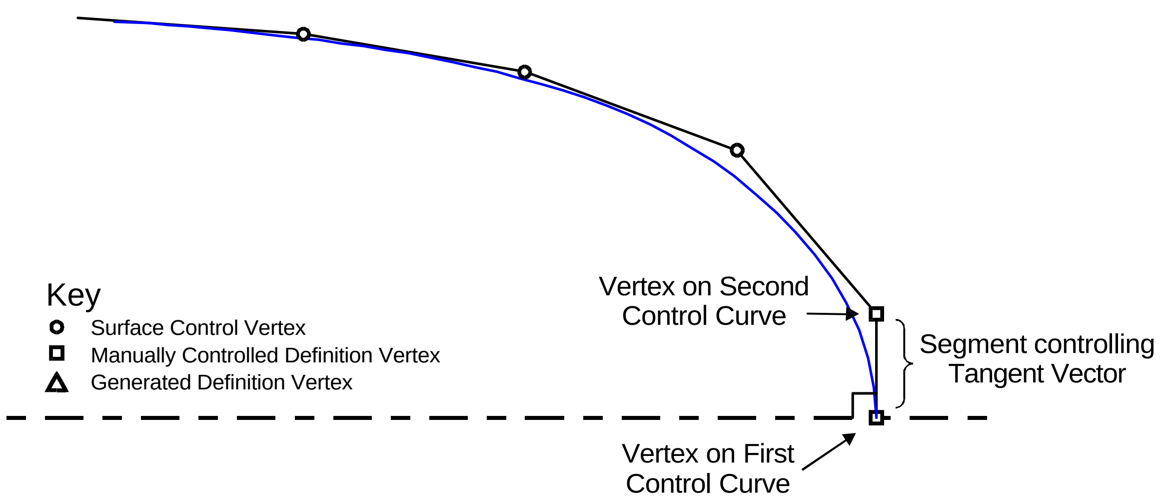


Figure 15.7, the first segment of the control polygon controls the tangent of the NURBS at the first vertex.

From the NURBS properties, it is known that the vector between an end vertex and the next vertex, the end segement, in the control polygon represents the tangent of the NURBS at the end vertex, (Figure 15.7). Therefore, to control the tangent of the surface across a boundary, the position of the first internal set of vertices needs to be positioned with respect to the location of the control vertices on the boundary of the surface. This arrangement suggests two definition curves that must be manually controlled, one for the boundary and one controlling the tangent shape. However, this arrangement only controls the tangent of the surface, as there are many more internal control vertices within the surface definition, the tangency information needs to propagate into the rest of the surface control polygon to ensure that a smooth and fair hull surface shape is developed as the surface leaves the tangent definition.

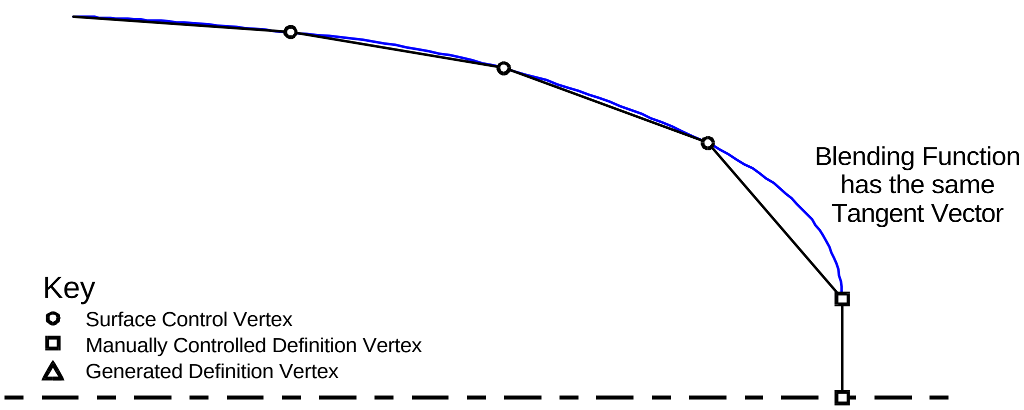


Figure 15.8, to create surface control vertices that smoothly transition away from the tangent feature, the tangent information needs to be included in the blending function.

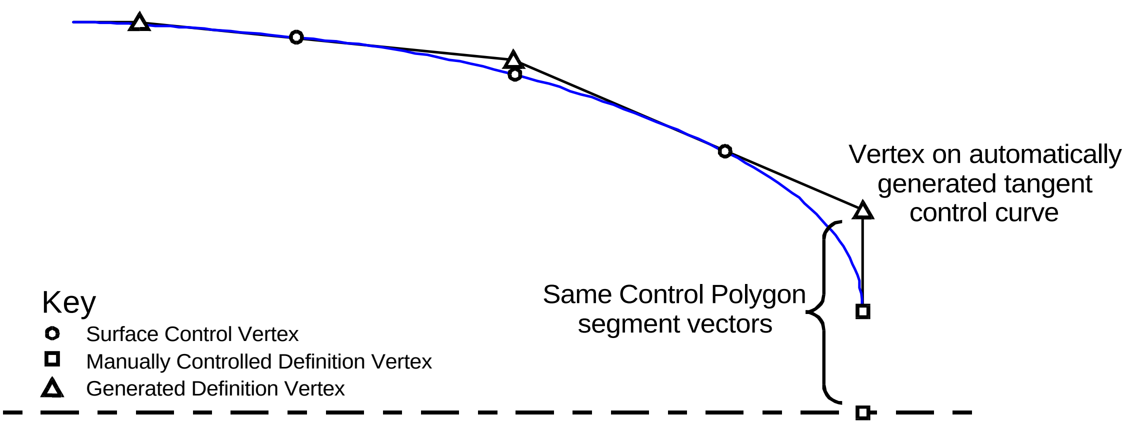


Figure 15.9, Using the same first segment dimensions in the blending function control polygon as in the surface control polygon creates a curve that smoothly transitions the surface control vertices to the tangent feature.

The primary task of the longitudinal blending functions in the surface interface framework is to ensure that the hull surface smoothly transitions across the separate regions of hull form shape. Consequently, any tangent definitions must also be included in the construction of the blending functions. The first two vertices of a control polygon row control the tangent at the surface boundary. The blending function must include the effects of the tangent to ensure that the subsequent control vertices on the control polygon row smoothly transition away from the feature, (Figure 15.8). This can be achieved by defining exactly the same tangent control segment in the blending function as in the surface control polygon, (Figure 15.9).

The tangent control curve controls the surface shape so that it correctly transitions to the tangent feature, however, the technique alone does not make control of the tangent feature easy. Without any additional aids, the user has to control the tangent of each blending function individually with corresponding pairs of control vertices on the first two curves. However, by using the local curve shape constraint tools, control of the tangent feature can be greatly reduced. To create the example in Figure 15.10, firstly, as it is desirable for the surface to be perpendicular to the centre plane, an *offset modifier* was applied to the tangent definition curve using the bow curve as a reference. This constrains all tangent vectors to be normal to the centre plane. Consequently, only the magnitude of the tangent on each longitudinal row of the surface is independent. As hull surface shape below the waterline needs to be sharp into the bow, a *straight segment modifier* is applied, (see Figure 15.10), to implement linear control of the magnitude of all tangents within the selected segment.

To create the shape bow feature, the magnitude of all the tangents within the straight segment is reduced close to zero. Finally, to add flare to the surface at the deck, a *curve segment modifier* is applied to the remaining vertices and the top vertex is drawn out to create sweeping waterlines for the deck shape. As the *curve modifier* is connected to a curve that is controlled by the *straight modifier*, the tangent of the *curve modifier* shape is set appropriately and the flare between the *straight modifier* and deck is controlled with an arc like shape.

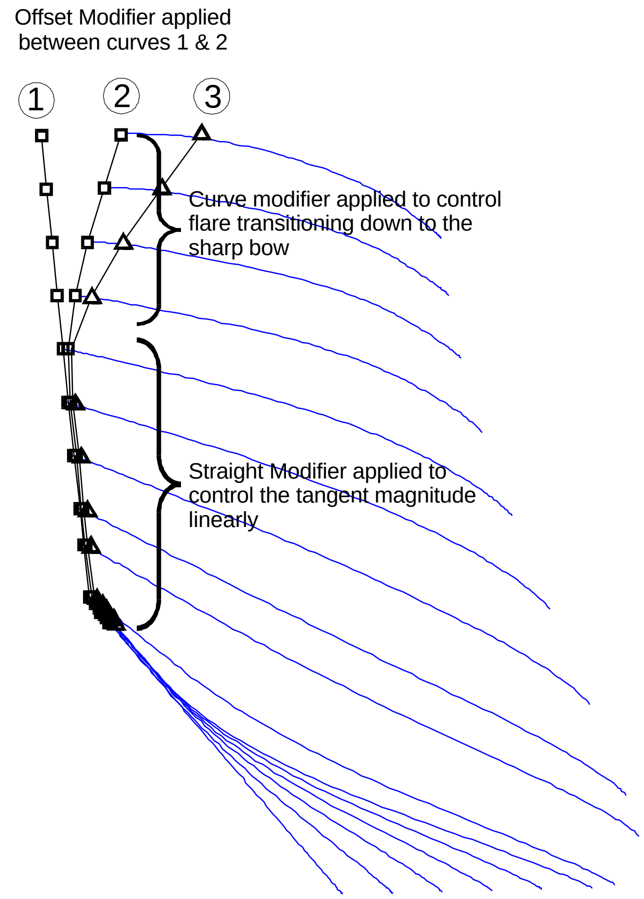


Figure 15.10, Forward view the bow, showing the blending functions and the three curves controlling the tangent of the surface at the bow. Modifiers have been applied to curve 2 to constrain the tangents into grouped shapes and remove the need to control the shape of the tangent on each blender independently.

15.4.2. Surface Flat Control Curves

The parallel middle body shape is a key feature of ship hull surfaces. This prismatic feature divides the hull shape into three different regions of shape, a characteristic that no other vessel type has. The parallel middle body feature is the primary reason why development of ship hull forms is difficult for hull design tools relying on manually entered definition. While the construction of the actual prismatic shape can be developed very easily, the surface shape transitioning to this feature, from the entrance and the run, is considerably more complex to control. Furthermore, as users must control a large area of related surface shape with local definition points, standard manual definition techniques cannot be considered very effective tools for producing this flat parallel shape which is common throughout almost all standard ship hull forms.

However, the existence of the parallel middle body can be used as a great aid to hull form development and, if the existence of this feature is identified by the hull design tool, it can be used to constrain the surface shape at and within the boundary of the prismatic region of the hull. Constraint tools can be developed to ensure that the surface blends smoothly into the parallel middle body as well as maintaining the shape within the bounds of the prismatic section of the hull form. Furthermore, the definitive shape of the parallel middle body boundary can be used to allow the three regions of surface shape to be manipulated independently. NAPA is one of the few hull design tools that allow this type of constraint to be applied. However, the system does not take full advantage of the topological information available within the hull definition curves. For example, the flat of side, and flat of bottom curve are not constrained to always lie on the prismatic shape defined by the mid section. As this is an obvious constraint to make available, it is surprising to find that these practical features have not been implemented, especially as the technology required is already present within the tool in other features.

Hull design tools systems using NURBS surfaces have a more fundamental disadvantage because the representation does not interpolate the definition points. Consequently, the only practical solution is to use multiple patches allowing surface boundaries to be located at the extents of features such as the parallel section. This give the user much better control over hull shape and there is improved flexibility in the variety of shapes that can be accurately represented. However, systems representing hull forms with single surface representation cannot provide these advanced features and have taken to using parametrically defined templates to produce an initial hull surface ready for detailed modification.

The fact that NURBS do not interpolate definition data does present a problem for TSCAHDE. However, as with the development of the tangent feature, NURBS properties can be used to force the surface to interpolate curves representing the extents of the parallel middle body. A brute force method of imposing control vertex interpolation is to use the definition curves to develop a second order discontinuity, or knuckle line into the surface at the boundary between shape regions and use the control polygon segments either side of the discontinuity to control the surface tangents. However, this is not a very elegant solution, particularly as it destroys the mathematical continuity within the surface shape.

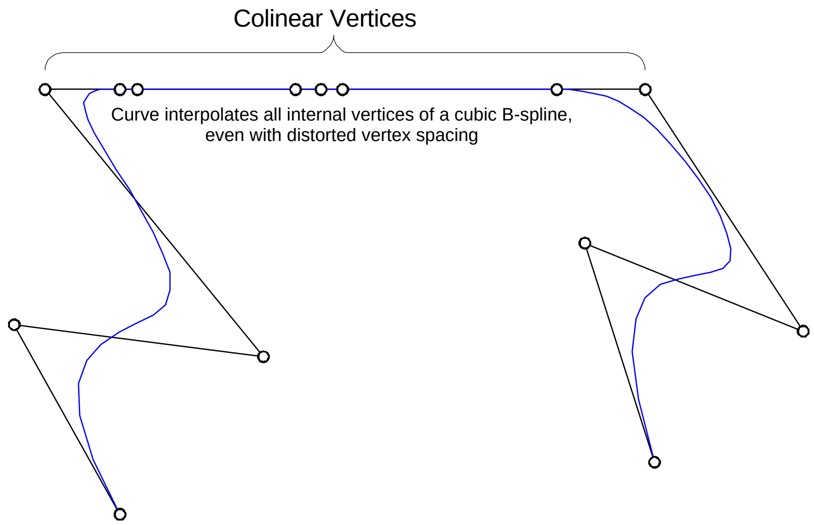


Figure 15.11, the property of collinear vertices. For practical reasons, it can be considered that all internal vertices in the collinear structure are interpolated by the cubic B-spline curve.

The property of collinear vertices is present in the longitudinal rows of the hull surface control polygon, in the vertices aligned to create the prismatic region of hull shape. For a cubic B-spline representing a smooth waterline curve, all definition vertices, except the end vertices, are interpolated even with distorted spacing between points, (Figure 15.11). Consequently, the shape of the boundary between the prismatic regions and the ends of the vessel can be controlled by using the first internal vertices of the collinear segment, (Figure 15.12). Thus, the position of these two vertices on each longitudinal row of the hull surface control polygon should be taken from the definition of the surface flat curves.

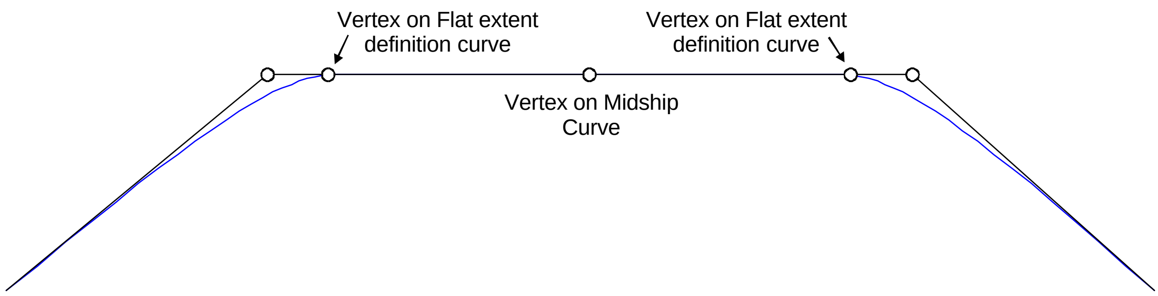


Figure 15.12, the collinear arrangement of vertices used to create the parallel middle body illustrating the geometric source for locating each vertex.

The locations of the end vertices, (b), of the collinear structure have still to be defined, (Figure 15.13). It would be impossible to ask the user to control the locations of these vertices, as this geometric feature has no direct relevance to a user constructing the hull form definition. However, by using the concept applied to create the tangent control curve, it can be seen that these end vertices represent the tangents of the surface as it blends away from the prismatic region of shape.

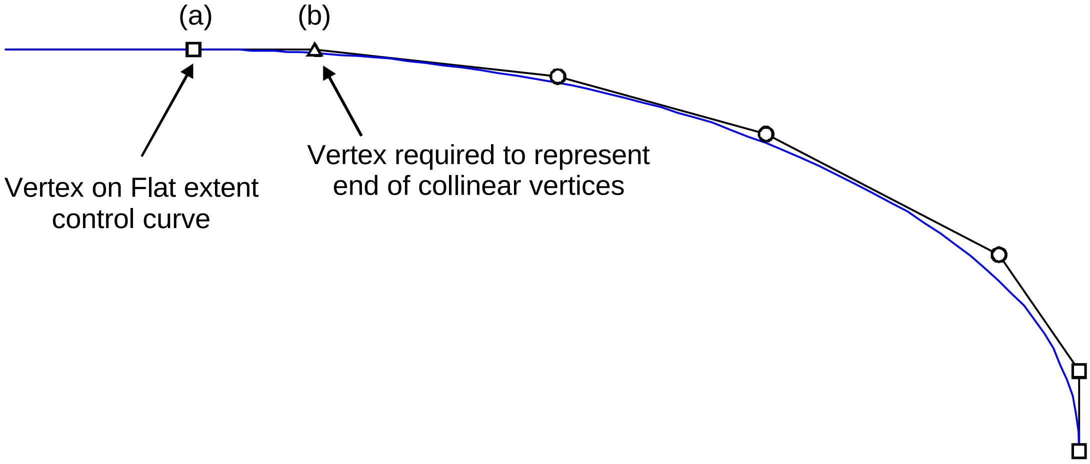


Figure 15.13, the locations of the end vertices, (b), of the collinear structure in the rows of the surface control polygon need to be defined. By identifying that these represent the tangents to the planar shape of the prismatic region of hull geometry, the same concept behind the tangent control curve can be used to generate the two addition definition curves.

Consequently, the blending functions can be constructed in a similar manner, (Figure 15.14), by generating the two control curves required to define the vertex at the end of the collinear structure (b), which also represents the first point of the blending function and the vertex, (c), required to develop the tangent in the blending function. Based on the method used to create the tangent control curve, the vertex representing the end of the collinear structure of vertices and the tangent in the surface, (b), bisects vertices (a) and (c).

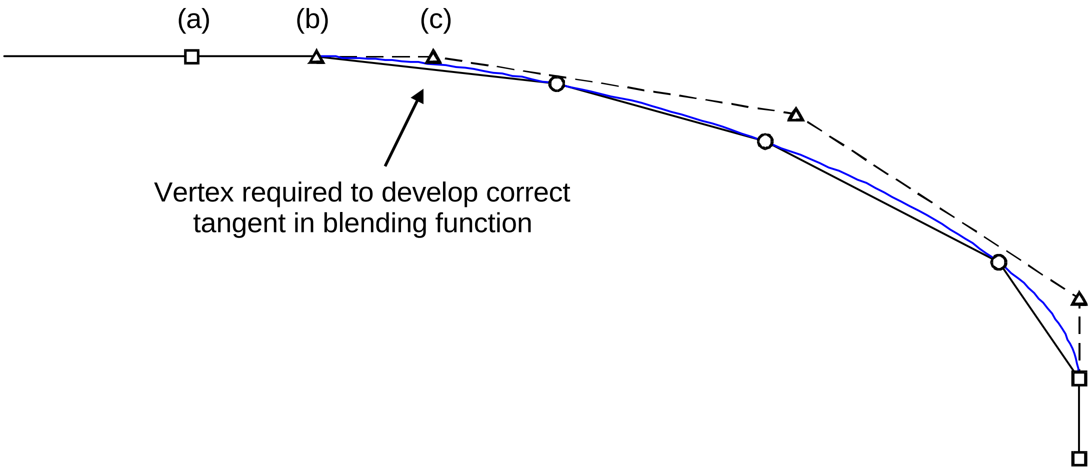


Figure 15.14, blending function are created by generating the additional control curves. The vertices on these curves lie on the lines through the collinear vertex structures in the surface definition and the location of the vertex at the end of the colinear structure, (b), bisects the locations of the vertices on the region boundary, (a), and the blender tangent segment defintion curve, (c).

With corresponding vertices on the three curves defining the surface region boundary lying on the lines through the collinear vertices, the direction of the tangent is defined. However, the magnitude of the tangent is a more complex issue. As a designer using current hull design tool technologies never deals with this surface property directly, the analytical reasons behind the definition of hull shape around the transition of hull shape to the parallel middle body are not common knowledge. Some experienced hull designers may know what form this shape should take, however, as hull design tools require the user to pull the definition vertices around until a desirable form is reached, this feature is hardly even considered in this context.

Based on the lack of available knowledge on the form of the surface shape in this region of the hull, some investigations were performed using TSCAHDE to identify possible relationships between the shape of the entrance and run regions and the magnitude of tangents at the boundary of these and the prismatic region of the hull surface. The software implementation of TSCAHDE is designed to be flexible allowing ideas to be quickly tried and tested. Two tangent magnitude calculations were initially investigated to ascertain if the relationships between the tangent magnitude and region shape were simple. These relationship were based on the following rules:

1. The tangent is a fixed length for all vertices on the control curve, based on some overall proportion of some measured distance between the flat curve and the bow curve, (Figure 15.15).

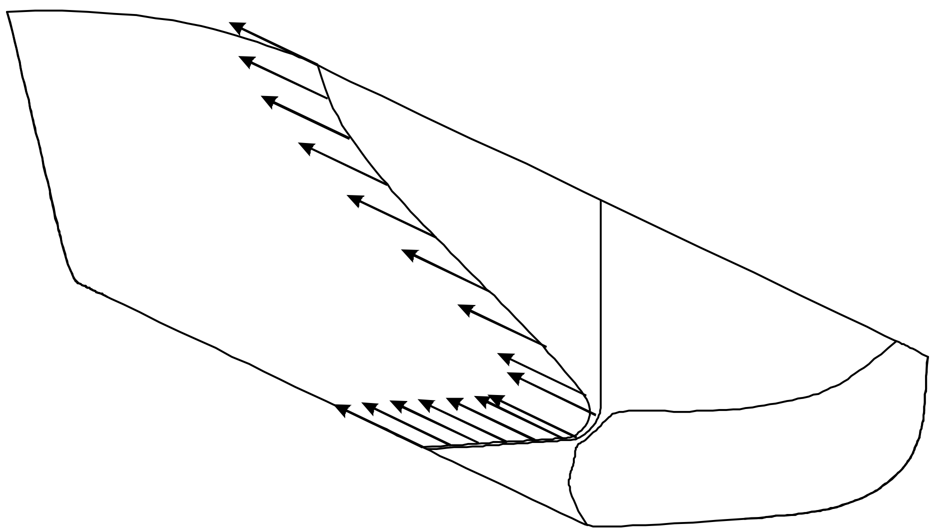


Figure 15.15, an arrangement of constant magnitude tangent vectors.

2. The magnitude of the tangent is a fixed proportion of the distance between the corresponding vertices on the prismatic boundary and bow curve, (Figure 15.16).

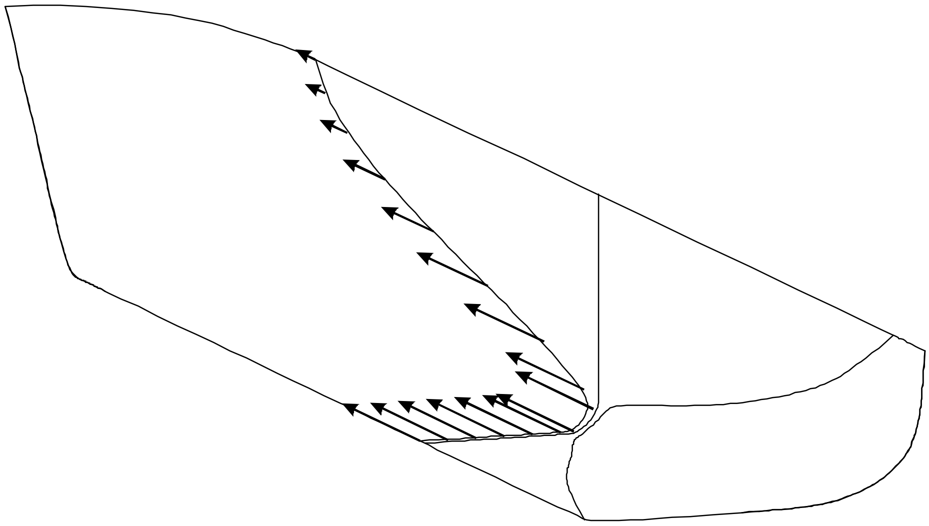


Figure 15.16, an arrangement of tangent vectors of magnitude proportional to the distance between the corresponding vertices on the prismatic boundary and the bow curves.

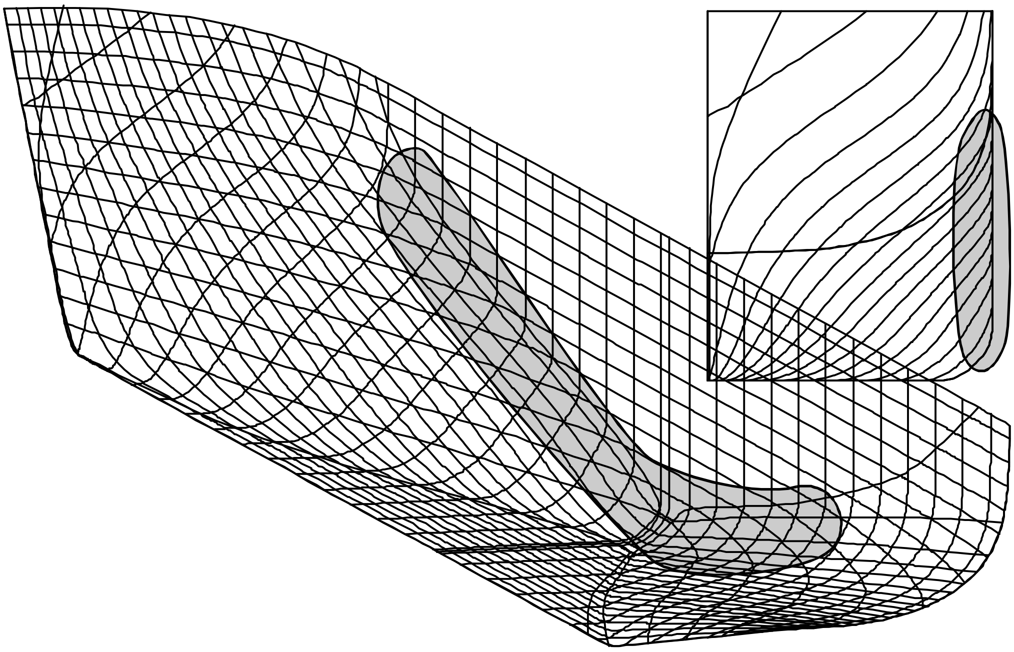


Figure 15.17, the fixed length tangent magnitudes result in sharp section transitions to the flat of side, for an appropriate tangent magnitude at the deck.

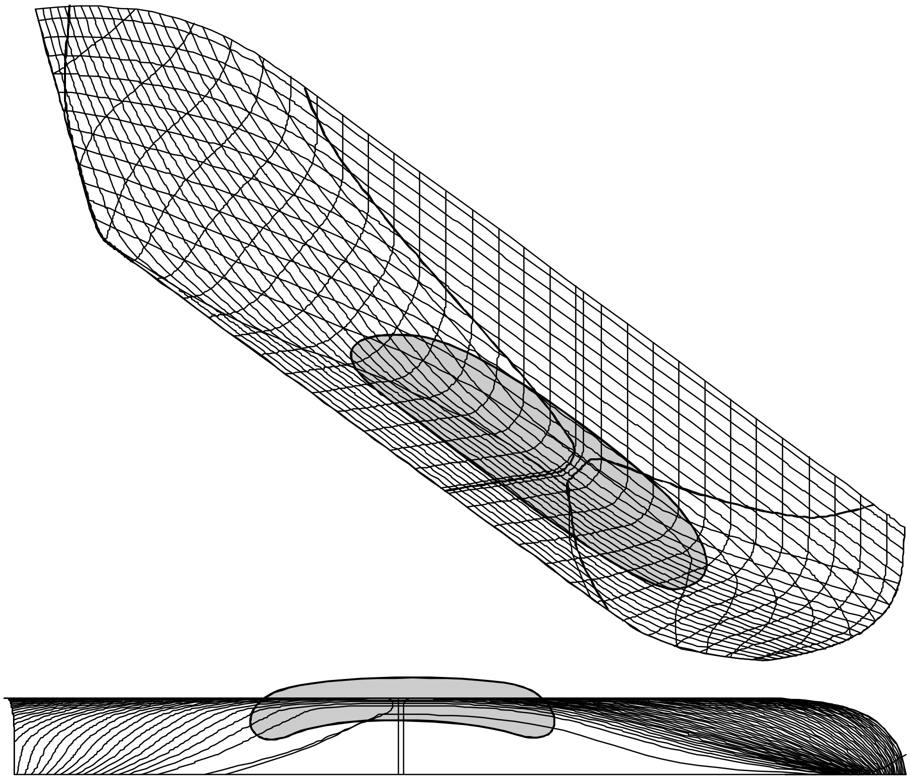


Figure 15.18, proportional tangent magnitudes create very long tangents at the bilge radius. As a result, the waterlines are quite stretched and the parallel middle body appears to be much larger than it should.

In the case of a Ro-Ro vessel, the flat of side curve in the bow for example, takes up a considerable proportion of the length of the hull. This can create, in some cases, an almost circular deck shape from the flat of side to the bow. Using the first method with constant tangent length, the short tangent vector magnitudes required for the deck shape also result in tangents at the bilge radius that are not long enough to create flowing waterlines. Consequently, the sections transition to the flat of side very sharply, (Figure 15.17). In contrast, the proportional tangent magnitudes create very large tangent magnitudes at the bilge radius for the same hull form, (Figure 15.18). As a result, the waterlines are quite stretched and the parallel middle body appears to be much larger than it should.

Closer analysis of a selections of hull forms, (Figure 15.19), shows that, apart from at certain particular features of hull shape, the distance between buttock lines near the flat of side curve is uniform along the length of the vessel. This suggests that the tangents on the flat-of-side curve are of constant magnitude, however, with a direction that is normal to the boundary curve and lies within the plane of the Flat. This tangent arrangement is illustrated with respect to the curve bounding the entrance and the prismatic regions of the hull surface in Figure 15.20. However, as the blending functions run longitudinally along the surface, it is necessary to resolve the surface tangents to vectors aligned with the *x* axis.

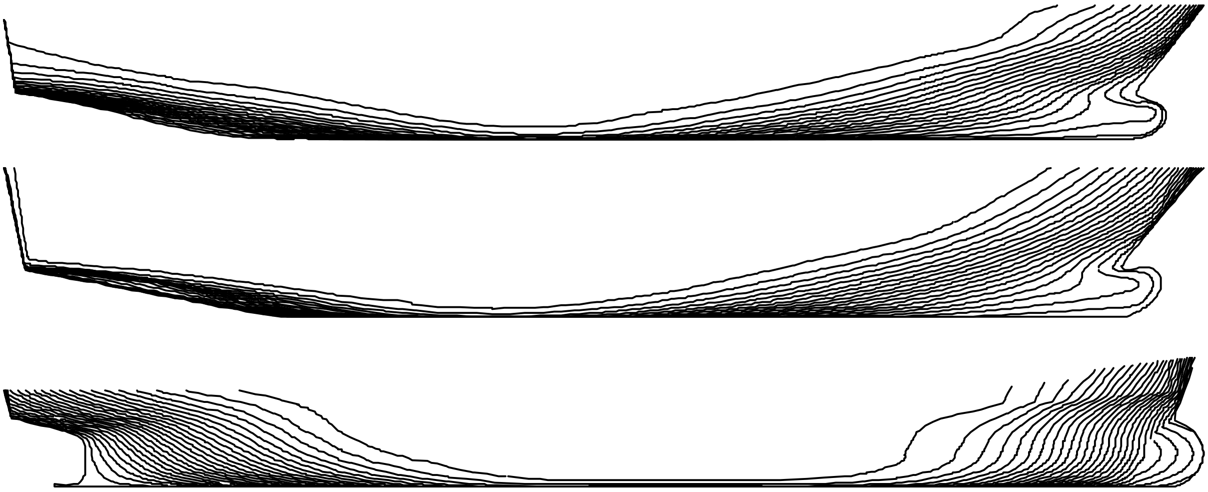


Figure 15.19, Equally spaced buttock curves along the flat of side illustrate that tangents are appear to have uniform magnitudes located normal to the flat of side curve.

It is geometrically difficult to resolve these vectors in Euclidean space. Unfortunately, a calculation cannot be performed on each tangent vector individually. The magnitude of blending function vector may be influenced by many surface tangent vectors, (Figure 15.21). Furthermore, as the curve wraps around the hull form, some surface tangents may not lie within the prismatic surface.

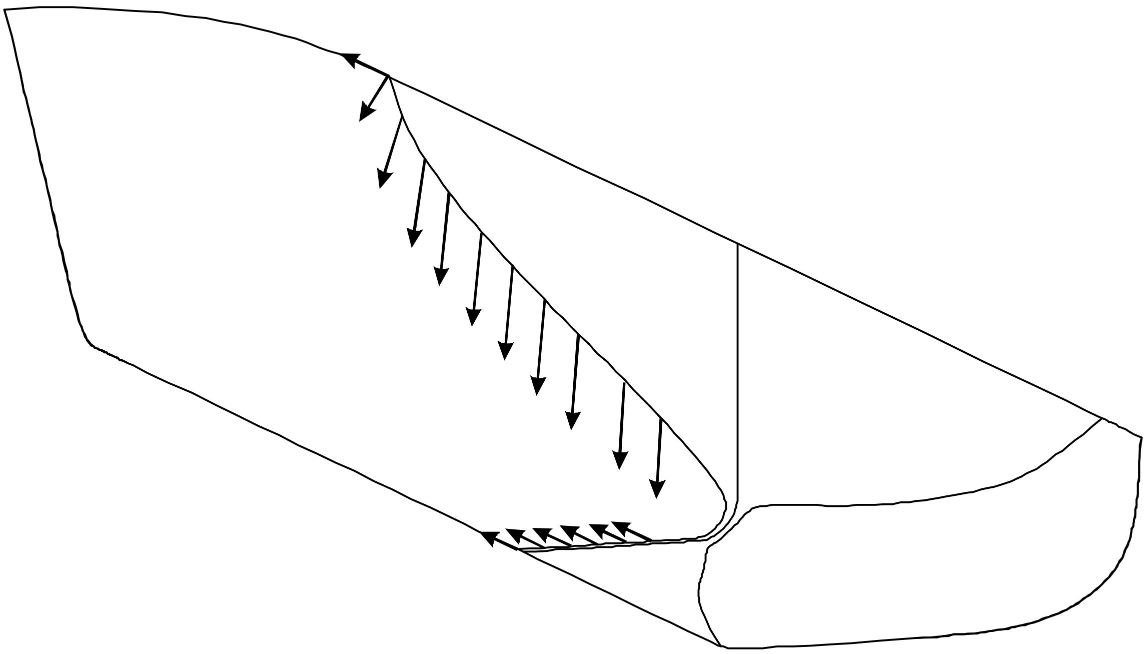


Figure 15.20, Development of tangent vector magnitudes based on fixed length tangents that are normal to the curve shape and lie on the surface.

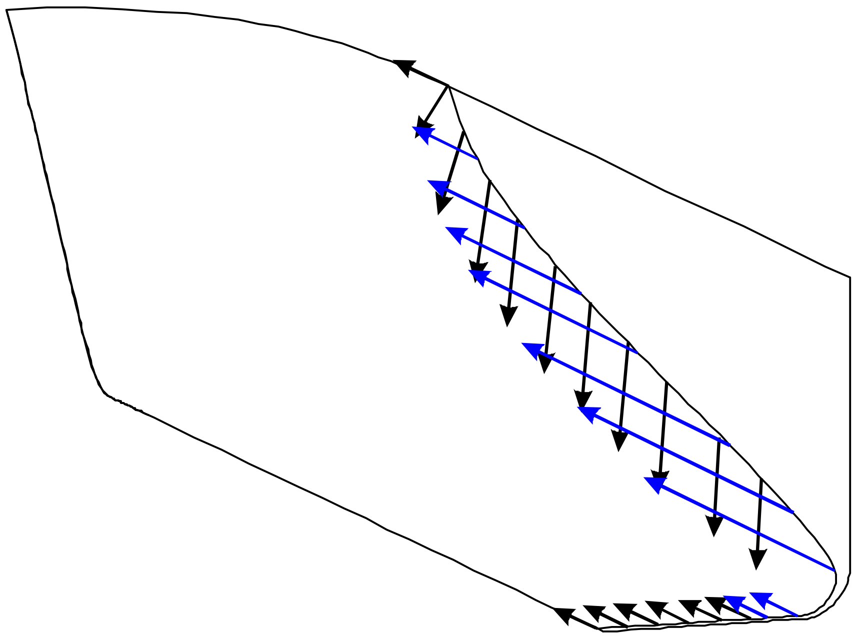


Figure 15.21, Vectors for controlling blender functions (blue) must be developed from the surface tangents, which are normal to the surface region boundary. The magnitude of each blending function tangent may be dependant on many surface tangents

As each surface tangent has a constant magnitude, an appropriate technique is to calculate a curve offset from region boundary by the magnitude of the surface tangent vectors. Once this curve has been constructed, the magnitude of the blending function tangents can be found by considering the distance between the two curves in a parallel direction to the x axis.

A few investigations, made with techniques functioning in Euclidean space revealed that there is a great deal complexity involved in calculating of the offset curve about the bilge radius. Surface tangents on the side and bottom flats can influence the offset distance of the curve at one location. However, as these tangents lie in planes that are perpendicular to each other, standard vector manipulation techniques are not adequate to develop the offset curve.

Calculating planar offset curves for simple shapes is a relatively simple procedure. Most CAD tools incorporate this as a standard function. Therefore, calculating an offset curve from a hull definition curve, if it can be transformed on to a plane, is a much simpler solution. By unwrapping the region boundary curve off the prismatic surface, the calculation of the curve, offset by the magnitudes of the surface tangents, becomes a trivial task in comparison with the three-dimensional case.

The blending function vectors are aligned parallel to the *x* axis. Consequently, an appropriate unwrapping transformation can be constructed by combining the *y* and *z* components of the curve. An intermediate coordinate component r is calculated considering curve length in the *y*-*z* (transverse) plane.

$$r_n = \sum_{i=1}^{n} \sqrt{(y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}$$

The two-dimensional offset operation can then be performed in the *x*-*r* plane. As there will be a limited amount of variation in the shapes of the curves to be offset, a simplified procedure can be used to perform the offset. The two dimensional offset curve calculations are presented in more detail in Appendix 5. Once the offset curve has been calculated, (Figure 15.22), the magnitude of the blending function tangent arrangements can be found by reviewing the distance between the curves parallel to the *x* axis. This tangent information is stored in two curves, the flat control curve and the tangent control curve, (Figure 15.23), is used by the surface interface framework to construct the appropriate blending functions.

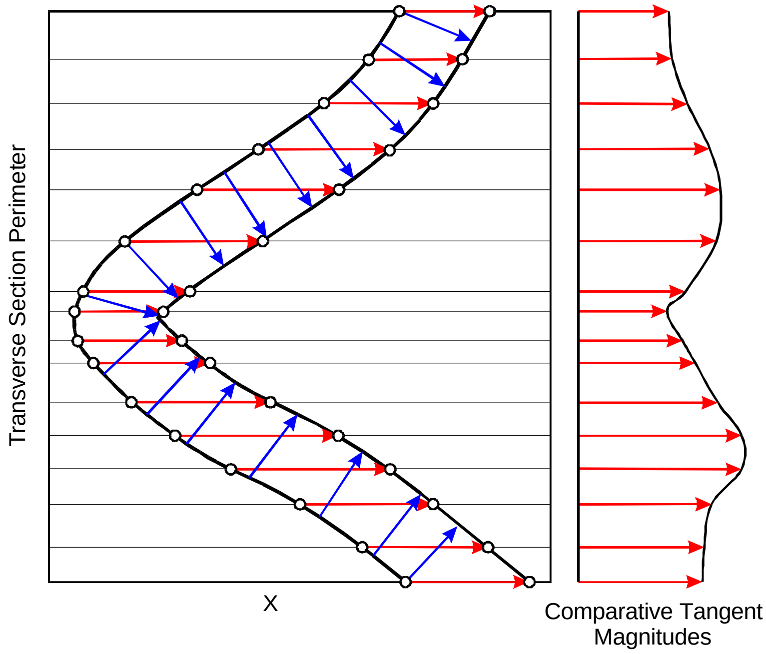


Figure 15.22, the curves bounding the hull ends and the prismatic section are transformed into two-dimensional representation to allow a simplified curve offset procedure to be applied. The magnitude of the blender function tangent vectors can be found by reviewing the distance between the curves parallel to the x axis. The segments at the ends of the offset curve are extended to allow the magnitude of vectors at the extremities of the curves to found.

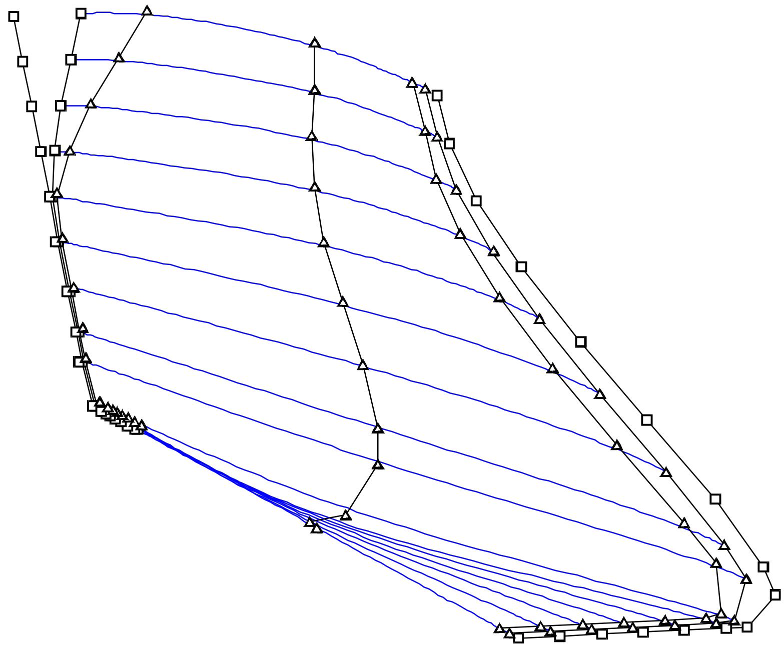


Figure 15.23, The blending function tangent vectors are used to calculate the arrangement of curves required to control the shape of the surface at boundary between the ends of the vessel and the Prismatic region. The figure shows the complete arrangement of curves used to define the Entrance of the hull surface, including the Volume Control Curve defined in 15.4.3.

15.4.3. Volume Control Curves

Modern hull design systems lack any practical tools for controlling hydrostatic properties during the design process. Considering, that achieving certain desired hydrostatic targets is one of the primary goals of hull design, it illustrates another area where modern hull surface design tools are not providing an adequate solution for the task involved. Many have attempted to provide limited control by implementing the traditional transformation techniques. However, these features are not provided in way that allows them to be used in parallel with manual hull development. Usually, the user must develop the desired hydrostatic properties by manually manipulating the hull form in addition to maintaining a correct and fair shape. Parametric hull generation techniques potentially offer a solution to this problem by creating a surface with the desired hydrostatic properties. However, as these techniques have major limitations in the flexibility of surface shapes that can be produced, as previously discussed, the full advantages of this feature cannot be taken.

As TSCAHDE aims to develop the hull shape from curves representing the primary features of the hull shape, the system obviously does not intend the user to control hydrostatics properties by manually interacting with the definition geometry. The surface interface framework includes an additional control vertex in the centre of each blending function that can be used to control the shape of the hull form as it transitions from one region boundary shape to another, primarily to allow the volumetric properties of the surface to be influenced. These vertices, when combined into a curve (B-spline control polygon), can be controlled parametrically. The parametric side of controlling the hydrostatic properties of the hull form was discussed in Chapter 14. The hydrostatic control system provides two numerical parameters, to control the entrance and the run. A method of using these parameters to generate the geometry of these volume control curves now needs to be developed.

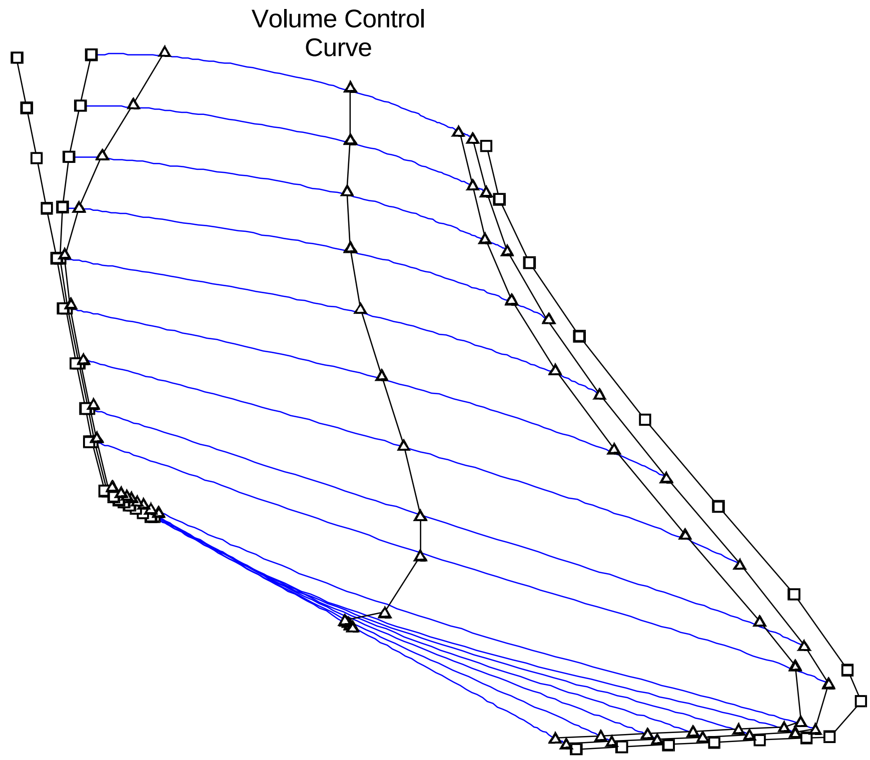


Figure 15.24, the vertices on the forward volume control curve can be located anywhere between the bow and prismatic region borders. It is necessary to constrain the location of vertices to enable practical control of the shape of the curve while maximising control of the curve over the hull form.

The traditional approach taken by parametric hull generation techniques is to model the volumetric properties of the hull form using longitudinal functions representing, for example, section area or vertical centre of buoyancy. The responsibility of constructing the correct hull shape becomes the task of the hull surface generation functions, which often operate in the transverse plane. To develop valid hull forms using this type of analytical procedure, the hull surface generation functions must contain the complete knowledge of the hull form topology. Frequently, these techniques provide the user with additional control over the shape of the sections in the ends of the vessel, allowing the specification of various qualities of U or V type sectional shapes. This indicates that there is an element of independence in the shape of the hull form in the ends for particular hydrostatic parameters, regardless of the bow, flat of side, flat of bottom or transom boundary shapes. Experience gained using various hull design tools and during the development of TSCAHDE seem to suggest that there is a very small range of shapes which produce a fair hull surface with respect to the boundary shapes for specific hydrostatic qualities. Therefore, the lack of success of parametric hull generation tools have suffered must also be a result of inadequate representation of the hull form topology.

In initial design, the exact shape of the hull form is not so important. The main task of the designer is to identify the approximate shape of the surface before beginning a more detailed optimisation process. Consequently, it is not necessary for the technique to provide a high level of flexibility in the shape in the entrance and the run for fixed hydrostatic properties. As there is only one parameter per end of the hull, there will be only one shape for a particular volume, given a certain set of hull definition curves. This does not mean that the variety of shapes produced should be small. On the contrary, to minimise the amount of shape definition that must be provided by the user, the information in the shape of the region boundaries needs to be maximised to ensure that the volumetric control curve represents a shape that is appropriate for the particular hull form definition. An understanding of hull shape needs to be developed to allow rules and subsequently, mathematical functions to be created which allow the volumetric qualities of the hull surface to be controlled with a shape that is dependant on key characteristic features of the hull form. Consequently, a constraint is constructed which keeps the surface to the pattern of the ship hull form. Furthermore, if geometric information in the definition and blending function structure is considered during the development of these rules, a technique can be created which will not exceed the capabilities of the definition functions with regards to the shapes that can be produced.

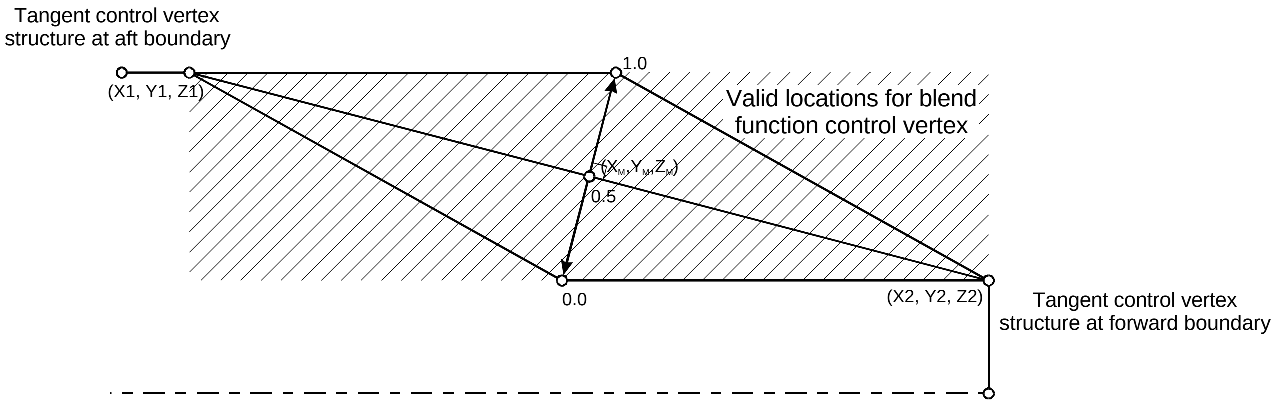


Figure 15.25, the valid range of middle vertex locations used to control the shape of blender functions, constrained to a range that can be controlled by a simple parameter varying between zero and one representing fullness.

Geometrically, there are an infinite number of possible locations for vertices on the volume control curve. However, the range of locations that actually produce a valid hull shape is very small. Vertices can be constraining to a smaller range of locations to enable practical control of the shape, however, care must be taken to ensure that there is still enough flexibility for all valid hull form shapes to be constructed. ShipLINES uses a geometrically constructed volume control curve. However, this requires a significant amount of calculation, particularly for intersections in model space, without providing adequate control over the hull surface. It cannot be seen to be an effective solution. The preferred solution is a parametric approach of controlling the location each vertex. A single parameter can be used to control individual vertices within certain range. This parameter can be simplified, in a similar manner to the parameters used to control the volumetric properties of the Entrance and Run, to a value between zero and one representing the minimum and maximum fullness of the blending functions. This range can be determined from the set of linear vertex locations that produce valid blender function shapes, (Figure 15.25). A structure can be created that constrains the location of the middle vertex on the blending function control polygon to a range that will generally not produce any inappropriate shapes. The diamond shaped structure is located around the position of corresponding vertices on the inner most curves at the boundaries of the region. The vertex is constrained so that it cannot lie outside the transverse range created by the two end vertices. This maintains a hull surface that cannot exist outside the beam defined by the prismatic region of the hull form or inside, creating a hull surface that has an inappropriate concave shape. The location of the mid-vertex controlling the shape of the blender function is defined by the following functions. The functions locate the vertex by considering the location of the midpoint and the line perpendicular to the line through the end vertices, known as Vertex 1 (X_1, Y_1, Z_1) and 2 (X_2, Y_2, Z_2), at the region boundaries.

$$X_M = \frac{(X_1 + X_2)}{2} + (2F_B - 1)\frac{(Y_2 - Y_1)^2}{(X_2 - X_1)}$$

$$Y_M = \frac{(Y_1 + Y_2)}{2} - (2F_B - 1)\frac{(Y_2 - Y_1)}{2}$$

$$Z_M = \frac{(Z_1 + Z_2)}{2} - (2F_B - 1)\frac{(Z_2 - Z_1)}{2}$$

Where Vertex 1 is located on the final curve of the prismatic boundary curve structure and Vertex 2 is located on the final curve on the bow or stern boundary curve structure. Parameter F_B controls the fullness of the blending function. The parameter lies in the range [0,1], as defined, producing fine to full shaped curves respectively.

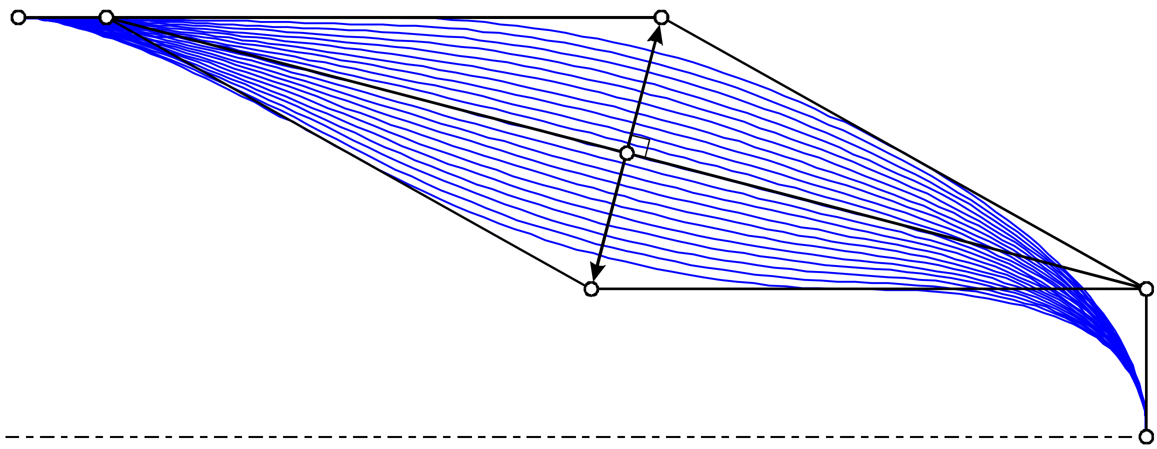


Figure 15.26, the control diamond illustrating all the entrance shapes possible with the vertex structure found at the deck.

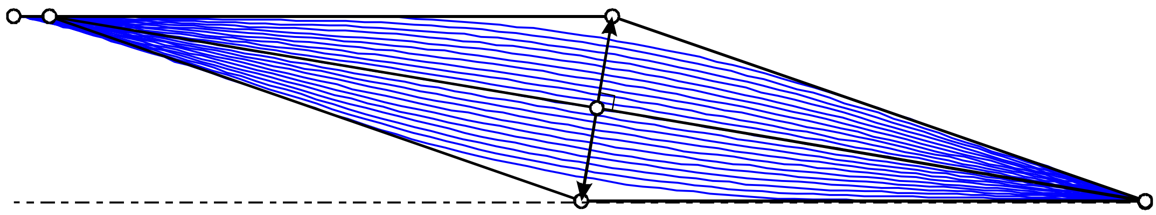


Figure 15.27, the control diamond illustrating the entrance shapes possible with the vertex struction found at the bilge radius.

The various control curves at the region boundaries are used to control the tangents of the blending functions. The control diamond connects these boundary curve structures together and enables some variation of the hull surface shape between the boundaries. Figure 15.26, shows the blender functions shapes that can be constructed with the diamond control structure found in the entrance region of the hull surface at the deck. Figure 15.27, shows the structure from the entrance region at the bilge radius. Note how the tangents control the overall shape of the blending function, allowing the diamond control structure to influence the fullness and hence the volumetric properties independently of the location of the blender function within the hull surface.

As the foundation of diamond control structure is based on the curve structures at the boundaries, (Figure 15.25), the shape produced is always a combination of the region boundary shapes. If all the fullness parameters, F_B, on each blending function were adjusted together, with the same value applied to each, the resulting effect would be a bias or balance control. With F_B set to zero, the volume control curve, (Figure 15.24), would have exactly the same transverse shape as the tangent control curve at the bow or stern. Set all the F_B parameters to one and the transverse shape of the volume control curve becomes the same at the tangent control curve at the prismatic boundary.

With values of F_B between one and zero, the shape of the volume control curve geometry is combination of the curves at the boundaries, the ratio controlled exactly by the F_B parameters.

Having developed a method that allows the fullness of each blending function to be controlled by a simple single parameter F_B, a technique is required which controls the complete family of blending function parameters using the entrance and the run volumetric control parameters C_F and C_A. A diagram, (Figure 15.28), has been designed to allow the blending function fullness parameters to be illustrated. Using the diagram, fullness control functions can be designed to control each individual blender function fullness parameter for a particular region using C_F or C_A as input parameters. The diagram plots the value of the fullness parameter against the transverse length of the volume control curve generated with a fullness parameter of 0.5 throughout. As the spacing between vertices on the definition curves may not be uniform and can be controlled by the user, plotting against curve length ensures that the volume control curve will be smooth when produced in model space.

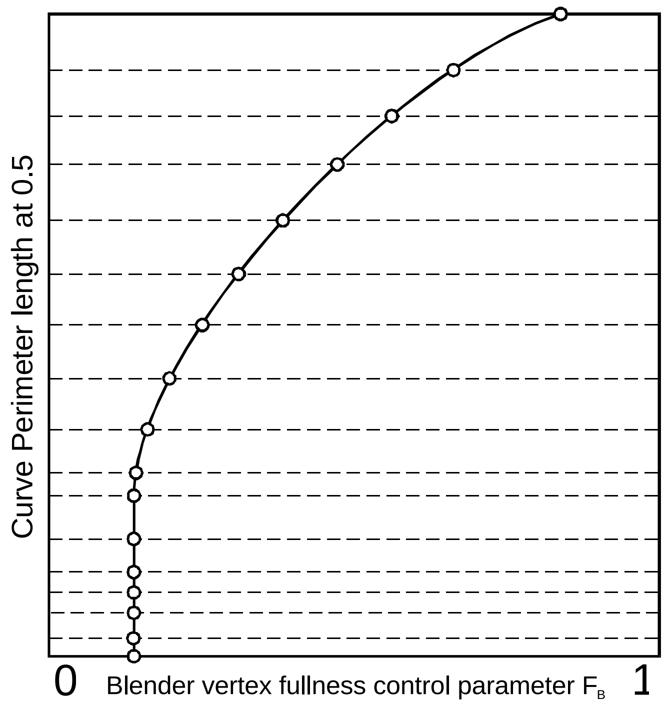


Figure 15.28, a blending function control parameter diagram. The diagram is used to display the pattern of values for the fullness parameter F_B on each blending function.

As the technique produces a hull surface using transverse definition curves linked together by longitudinal blending functions, there is separation between control of transverse shape and control of longitudinal shape. Unfortunately, this arrangement makes it very difficult to manually control longitudinal surface shape features, such as the flat-of-bottom in the Entrance or flat-of-side curves that run to the transom. This is a limitation of the single-surface representation, although it could be handled much more successfully in a multi-surface hull representation. As it is not possible to manually control these features, the fullness control functions must allow these features to be present and control them with respect to the desired volumetric properties of the hull form.

Due to the lack of manual control over longitudinal features, the fullness control functions must deal with up to three different areas of shape, the flat-of-side, flat-of-bottom and the curved surface shape in between. Rather than increasing the complexity of the fullness control function, this allows the problem of controlling surface shape with respect to volume to become easier. The identification of these three different features means that each part can be formed separately. In contrast, many parametric hull generation techniques have problems when these three shapes must be represented. Generally, mathematical functions are unable to represent shapes that include straight and curved features. Furthermore, if the fullness functions are designed so that they are controlled only at the intersections between the flat and curves surface shapes, the technique controlling the fullness function can handle changes in surface characteristics without requiring a method that deals with each case separately. The main advantage obtained when locating the controls of the fullness function at shape intersections is the ability to control how the shape transitions from one shape to another. The most important aspect of this is to ensure that surface achieves the proper tangency across the flat boundaries. However, before this can be achieved, a technique for controlling the flat shape itself must be chosen.

The range of vertex locations available from diamond control structure becomes more limited the closer the line through the two end definition vertices becomes parallel to the x-axis. In the case where the end vertices lie on a line parallel to the x-axis, the middle control vertex will also lie on the line. This is a very useful feature for automatically creating a straight deck line. Furthermore, the structure functions independently in y and z, if the two end definition vertices lie on the same y or z plane, the middle vertex will also lie on that plane. Consequently, for a diamond control structure forming part of the hull surface flat, the middle vertex cannot be moved perpendicularly to the flat. It can only be moved within the plane of the flat. A further feature of the diamond control structure is that families of blender function can be constructed by providing fullness control parameters that are very close in value. Consequently, if all the blender functions across a flat are controlled with the same value, the flat feature will always be correctly maintained and the fullness control function is be kept very simple. Figure 15.29 shows the blender function control polygons constructing the flat-of-side at the aft of a hull. Because the mid vertex must lie within the transverse range defined by the end vertices, the figure shows how the deck is automatically kept straight, how each function lies on the flat and how the family of blending functions is maintained. With the shape of flat features being controlled by constant valued fullness parameters, it is only necessary to define the fullness control function for the curved part of the hull surface. The constant value of fullness applied to the flat can be found from the fullness control function at the intersection between surface shapes. As a result, it is possible to collapse, (Figure 15.30), the diagram illustrating the fullness control function for the areas representing the flat shapes. This allows the control of the curved area of the surface region to be concentrated upon in much more detail.

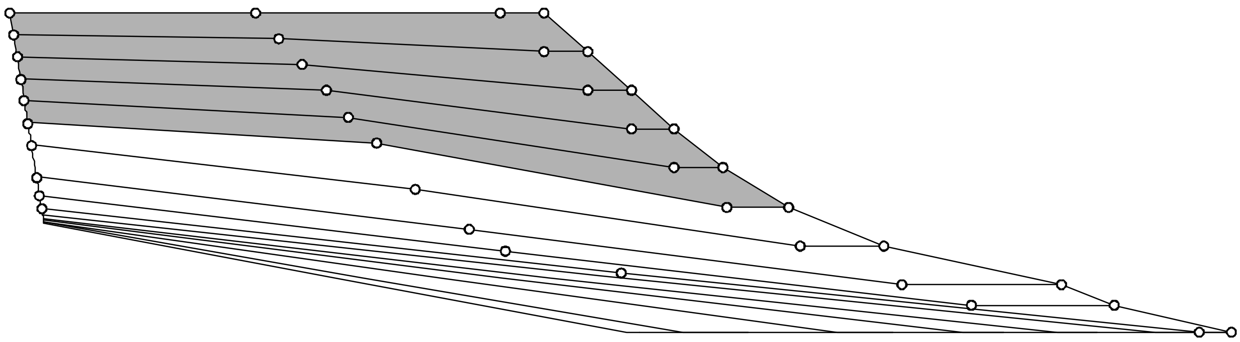


Figure 15.29, a family of blending functions (control polygons only) constructing an aft Flat of Side feature in the hull surface. The diamond control structure constrains the blender functions to form the Flat based on the location of the definition geometry at the transom and the aft end of the prismatic mid-section. Consequently, fullness control function can be kept simple to create this feature.

As it is only necessary to define the fullness control function over the curved region of the surface and not over flats, the most appropriate function is a cubic spline, (Figure 15.31). The cubic spline function can be controlled at the intersections between surface features by the location of the ends of the curve function and the tangents can be used to control shape as it crosses between shape intersections. A scheme can be developed to control the tangents based on the desired volumetric properties. However, when a flat is not present at an intersection, the locations of the ends of the fullness control function are undefined. In this situation, the end of the fullness control function will lie on the boundary of the hull surface. As hull surface boundaries are not very useful locations for controlling the volumetric properties of the hull form, it is possible to select fixed values for these locations to reduce any further complexity in the design of the fullness function. These values, chosen purely on the basis to create pleasingly shaped curves are; 0.5 for a surface boundary at the keel to create a neutral shape transition between the prismatic hull section and the bow or stern; 0.83 at the deck, (Figure 15.32), to create a nice balanced curve shape. 0.83 represents a location 1/3 from the maximum beam and 2/3 from the midpoint of the line through the end vertices.

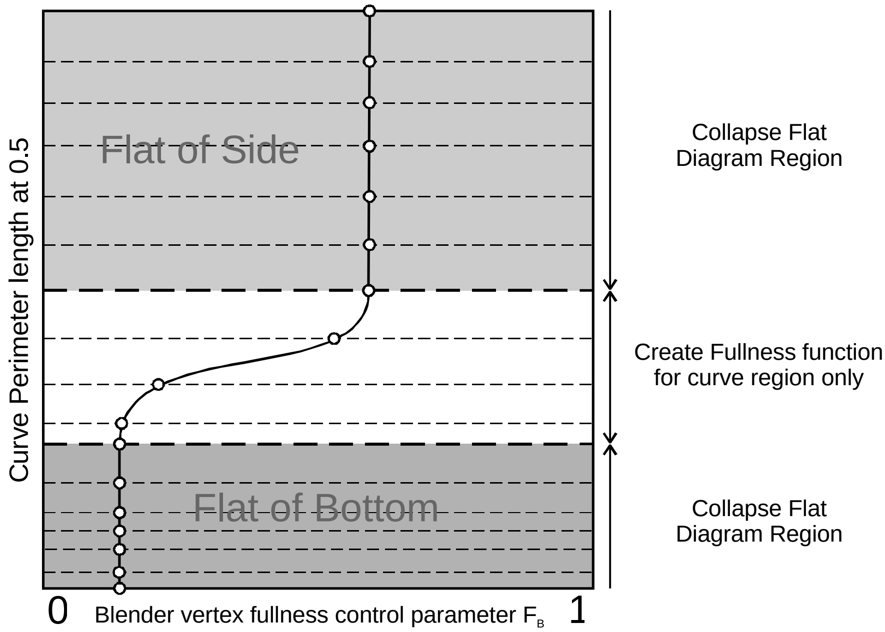


Figure 15.30, the regions representing flats on the fullness control function diagram can be collapsed to show the curved region of the surface in more detail

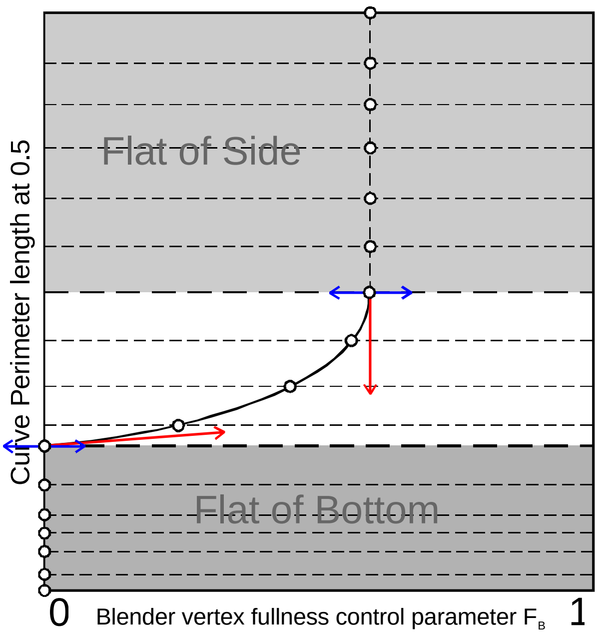


Figure 15.31, a single cubic spline can be used to represent the fullness control function, allowing the control to be situated at the intersection between surface shape features.

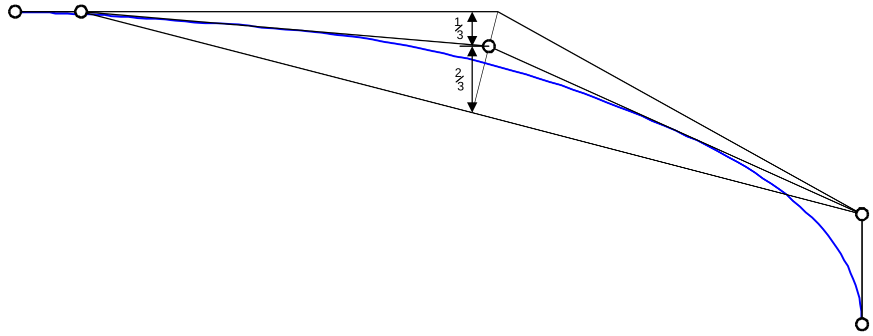
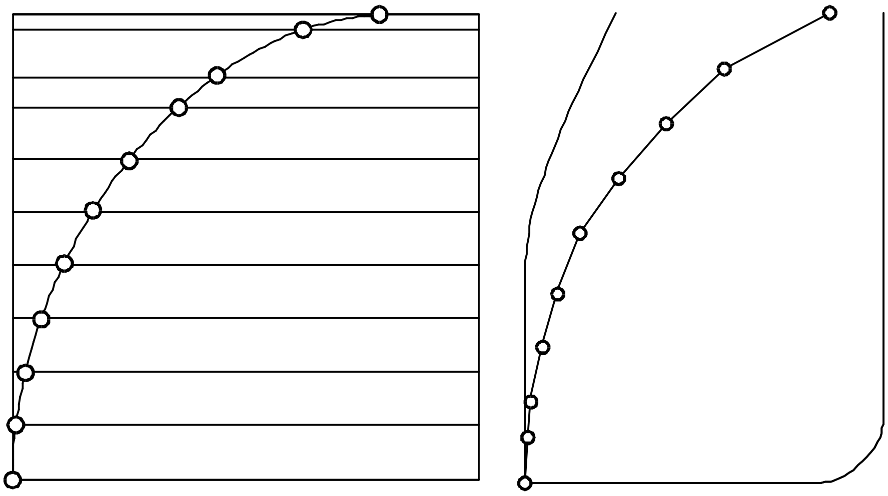


Figure 15.32, a fullness control parameter value of 0.83 is used to create an appropriate deck line shape. This represents a transverse position 1/3 from the maximum beam and 2/3 from the midpoint of the line through the end vertices.

Having developed a technique that can support control of the fullness control parameter on each blending function, the scheme for controlling the fullness function using the tangents must be developed. The most appropriate way to identify what shapes the fullness function is required to represent is to review the shape of the Entrance and Run and the respective shape of the fullness function over shape transitions from fine forms to fat hull forms. Furthermore, by keeping certain parts of the hull shape constant, different shape transitioning modes can be identified. Note, that the curve shown in the following diagrams is a three dimensional curve lying at the centre of the Entrance or Run. It does not, therefore, represent the true sectional shape of the hull surface.

If the shape of the bow is reviewed from, the thinnest to the fattest form, firstly with minimal Flat of Bottom, (Figure 15.33). The thinnest shape is an extremely concave ‘V’, (Figure 15.33a), a form that would be common to a Frigate type hull surface. To create this shape, a concave arc shaped fullness function is required. The volume of this shape can be increased by moving the centre of the shape towards a point at the maximum beam on the baseline. The next stage of shape would be close to a straight ‘V’ shape, (Figure 15.33b). A straight hull shape requires a straight shape in the fullness control function. In the limit of controlling the hull shape with no flat-of-bottom, the largest shape that can be produced is a ‘U’ shape, (Figure 15.33c), with the hull section tangents horizontal at the bottom and vertical at the top of the section. This shape requires a convex shaped fullness control function of similar arc-shaped construction to the fullness function used for the thinnest form, (Figure 15.33a).



a) The finest bow shape, extreme ‘V’.

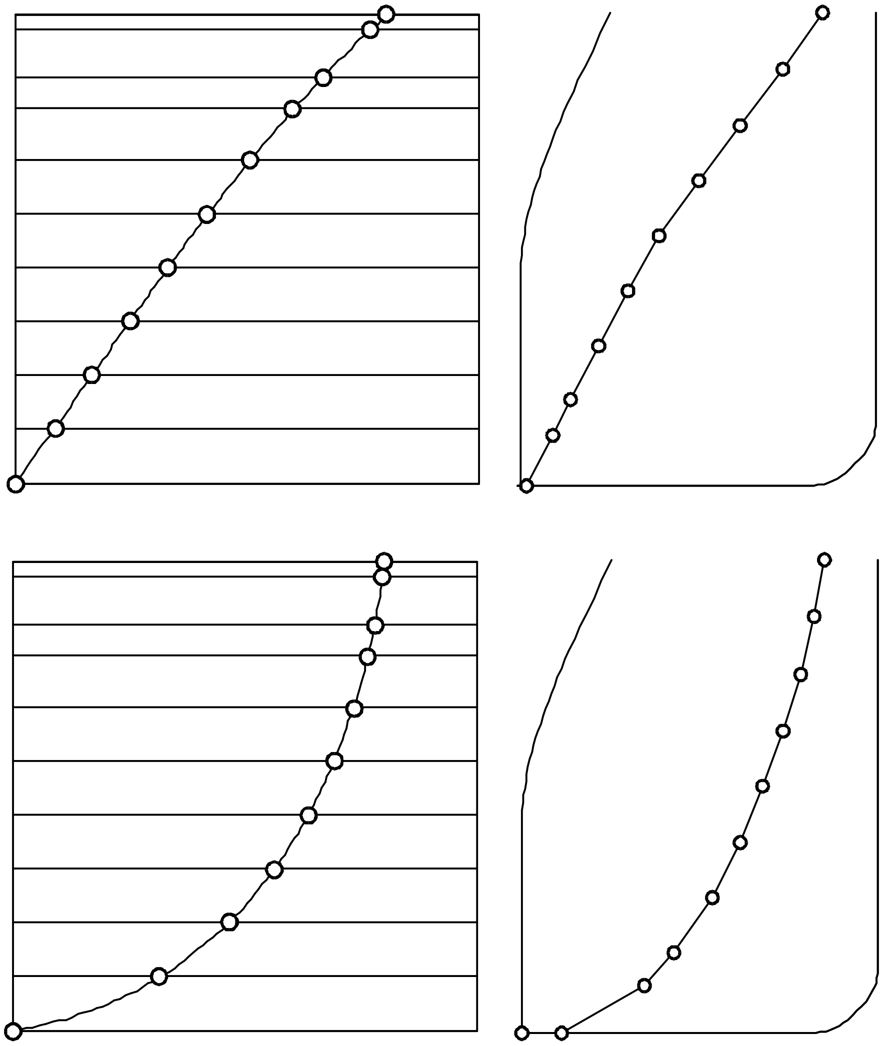
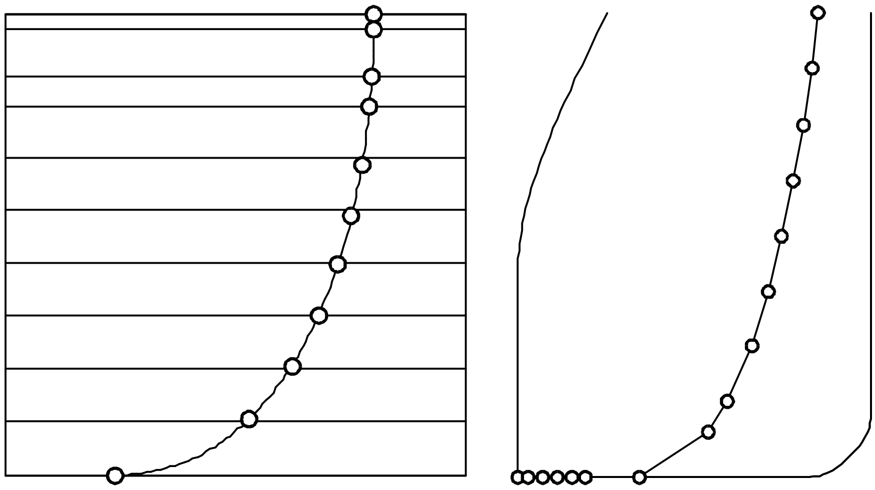
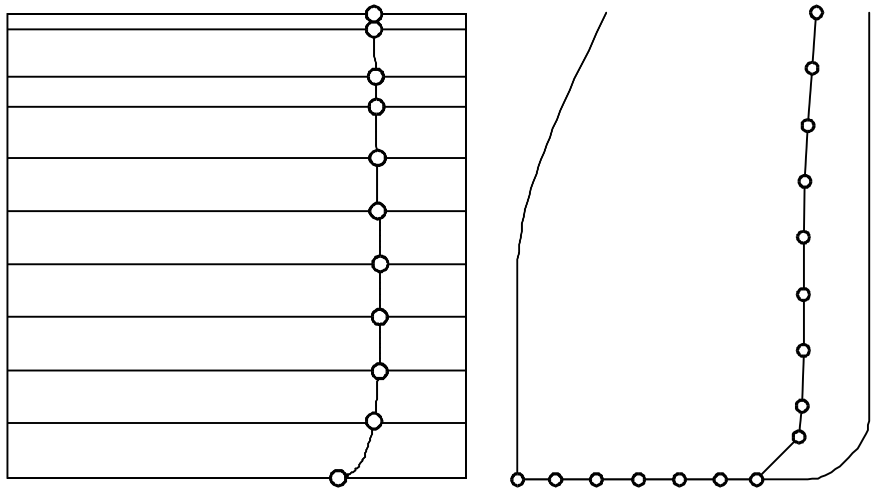


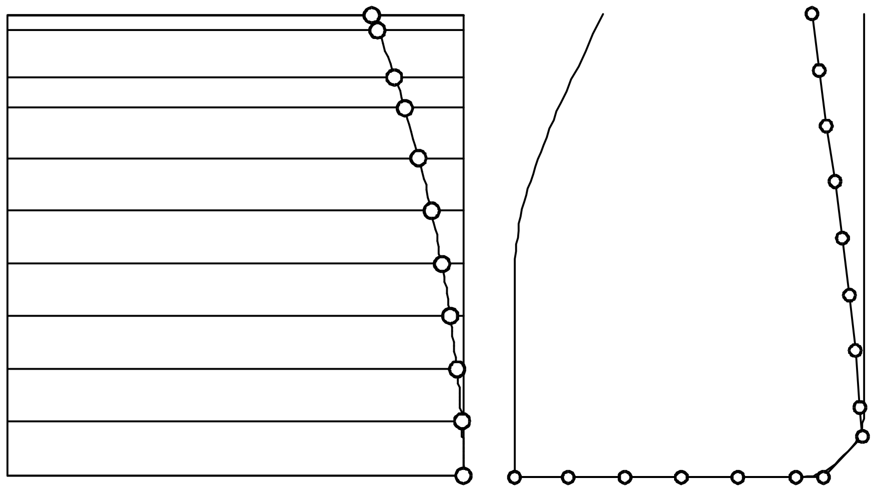
Figure 15.33, control of bow shape without Flat of Bottom.



a) Flat of Bottom added by moving the bottom point of the fullness control function



b) Flat of Bottom further increased in size.



c) The fullness function is adjusted at the top to incorporate the fattest Entrance form.

Figure 15.34, continuing control of Bow shape by introducing Flat of Bottom.

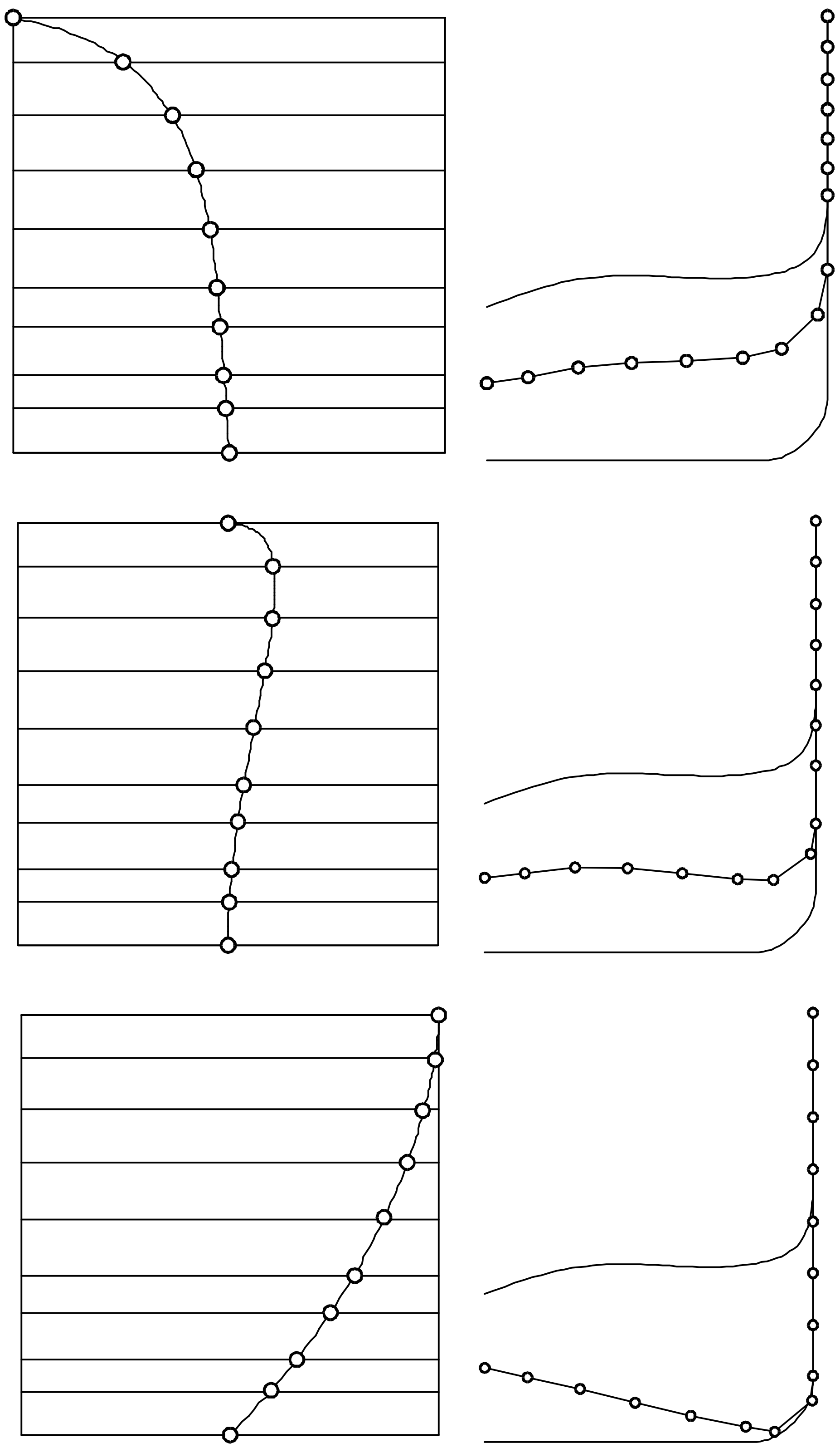


Figure 15.35, control of the stern through changing the size of the Flat of Side.

To further increase volume to the hull form, flat-of-bottom can be introduced, (Figure 15.34). However, note that the lower part of the hull section was made tangential to the bottom flat as part of the previous changes to section shape. The flat-of-bottom can be created by increasing the fullness parameter applied to the blender functions on the flat. The tangent applied at the bottom of the fullness function must be adjusted to reduce the amount of tangent applied to the surface at it leaves the flat-of-bottom. Further adjustment must be made to the tangent at the deck, (Figure 15.34c), as the curve increases, to ensure that a concave shape is not produced in hull section shape.

The shape variation of the run of the hull form is very similar to the shape of the bow, except that the location of the flat has moved from the flat-of-bottom to the flat-of-side, (Figure 15.35). The variation of shape can be achieved in exactly the same way as the Entrance. By increasing the size of the flat, the volume of the hull is increased in the stern.

The review of hull shape with respect to fullness function shape identifies that each end of the fullness function can be controlled relatively independently and that there are two modes in which an end can operate, depending on whether a flat is present. A third mode is required, which is a combination of the other two modes or *end conditions*, when the flat can be reduced to nominal width. These *end conditions* are described as follows.

1. Fixed End Condition

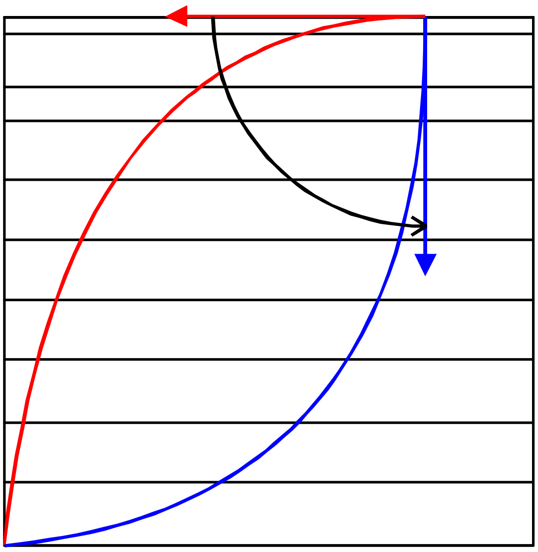


Figure 15.36, the fixed end condition rotates the tangent through 90 to allow concave (red) to convex (blue) section shapes to be generated.

Depending on the arrangement of the flat-of-side and the flat-of-bottom, the entrance of the hull form can have a range of sectional shapes that vary between concave, with flare, to convex. These are shapes commonly described as variations of ‘U’ or ‘V’ shape. Figure 15.33 illustrates that this variation of shapes can be accomplished without changing the shape of the hull surface at the deck or the keel. Tangential control at the end of the Fullness Control function can be used to implement these shape changes. The tangent vector can be rotated from a horizontal to vertical direction using a linear mapping of the range of the volume control parameter, (C_F or C_A, depending on whether the entrance or the run is being controlled), to 0-90° degrees, (Figure 15.36).

The other two end conditions use the full range of the blender fullness control parameter F_B. An additional modification is required to prevent concavity being introduced into the section as the side of a Flat being controlled by the other end of the Fullness Control function is increased to the maximum extent. This occurs once the opposite end of the Fullness Control function is located with a greater F_B than the fixed end condition. To prevent this, the tangent vector is rotated to maintain a convex Fullness Control function. The rotation is such that when the opposite end reaches the maximum position, the fixed end tangent will be directed towards the midpoint of the right boundary of the fullness control function diagram, (Figure 15.37).

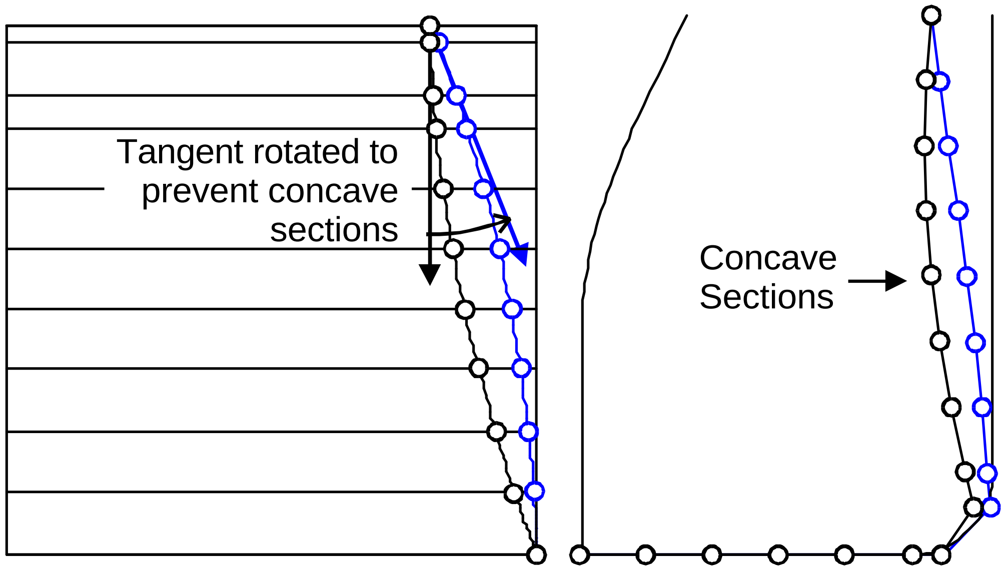


Figure 15.37, to prevent concave sections as the opposite end of the function moves the maximum position, the fixed end tangent is further rotated.

2. Flat End Condition

As the boundary between the flat and curved areas is within the hull surface, it is necessary to change the location of the boundary, as well as the tangent arrangement to enable the shape and the hydrostatics to be adjusted. As previously mentioned, the diamond structure enables the section shape to be constructed from the shape of the boundaries at each end of the surface region. When the shape of a flat is increased to the maximum level of fullness, the shape will be come highly dependant on the shape of the forward boundary of the prismatic region of the hull surface. As this shape is well defined, there will be no need to exert additional control on the tangential shape of surface at the boundary between the flat and the curved areas of the hull surface. However, as the transverse extent of the flat is reduced, section shape becomes less like the shape of the curves at the surface region boundary and it becomes necessary to introduce tangential control.

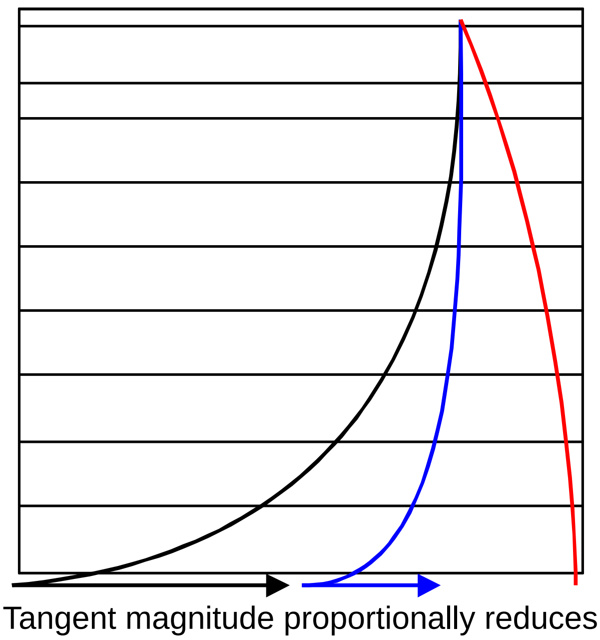


Figure 15.38, to increase the extent of the flat, the location of the end of the Fullness Control function is increased. The tangent magnitude is correspondingly reduced as the shape of the Flat becomes more similar to the boundary at the extent of the prismatic section of the hull form.

As the transverse extent of the flat is reduced, the extent of the curved portion of the surface shape becomes correspondingly larger and sectional shape becomes more circular. The surface must meet the boundary between the flat and curved areas tangential to the plane of the flat. To achieve this, the tangent controlling the surface shape at the boundaries of the flat becomes stronger to maintain a smooth curve shape.

The Flat end condition implements the changes in the extent of the flat by simply moving the location of the end of the control function from 0 to 1, mapping directly from the volume control parameter, (Figure 15.38). To transition the tangent from large magnitude to a small magnitude, the magnitude is proportionally reduced as the location of the end is increased until there is a zero length tangent when the end of the Fullness Control function is located at the maximum fullness. By the property of collinear vertices, the tangent controlling the surface along the boundary of the flat occurs between the blending functions at the boundary and the first function within the curved area of the surface. Using a horizontal vector maximises the magnitude of the tangent when the flat is reduced to the smallest extent.

3. Fixed-Flat End Condition

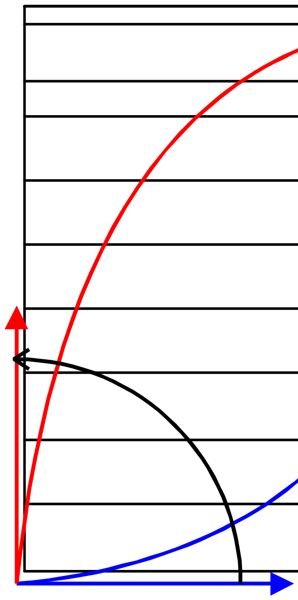


Figure 15.39, if the Flat has negligible transverse extent, the hull sections can be further reduced to 'V' shaped by rotating the end condition vector.

In some cases, particular for vessels with minimal parallel middle body, it is possible to reduce the transverse extent of a flat, (Flat of Bottom), so that it has negligible width for a large proportion of the longitudinal extent. The blending functions become part of the boundary of the surface and it becomes no longer necessary to maintain a smooth transition from the Flat to the curved regions. Consequently, it is possible to further reduce the volume of the hull by applying concavity and flare, resulting in ‘V’ shaped sections. This kind of shape control requires a two stage process, consisting of the ideas from both the Fixed and the Flat end conditions.

Considering the smallest volumetric arrangement of the flat end condition, with the end of the curve located at 0 and the tangent horizontal. With an appropriate end condition at the other end of the Fullness Control function, 'U' shaped sections can be reduced to 'V' shapes, by rotating the tangent vector in a similar manner to the Fixed end condition, (Figure 15.39). This combined end condition is mapped onto the control parameter by subdividing the range into two. In the first half of the range, the tangent vector is rotated, implementing section control from the thinnest 'V' shapes to 'U' shaped sections. In the second half of the parameter, the location of the end of the curve is moved to increase the size of the Flat, (Figure 15.40).

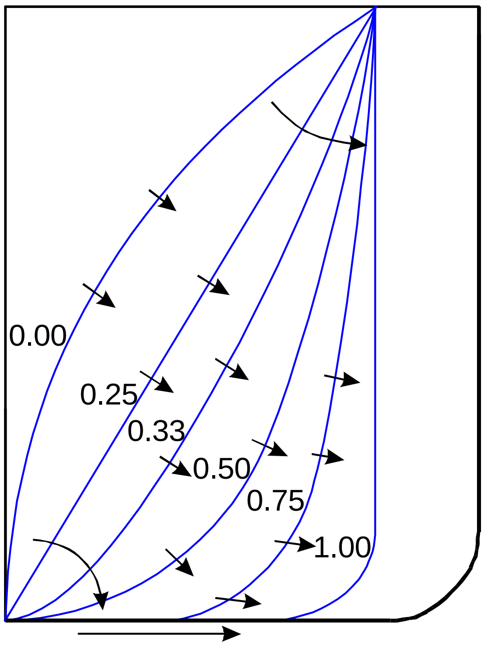


Figure 15.40, the range of sectional shapes available when using the Fixed-Flat end condition on the Hull Form Entrance

As the Fullness Control function will have two end conditions, controlling convex-concave shape or the extent of a Flat or both, a schedule is required to enable the end conditions to rationally affect the volume and shape of the hull form. The two volume control parameters C_F and C_A both have a specific range [0, 1], in which the end condition transformations must be scheduled. By considering the need for surface continuity across the boundaries between the Flat and curved regions of the surface, it can be identified that the convex-concave hull section modifications are applied before there is an modifications to the extent of Flats, when increasing the volume of an end of the hull from a minimum to a maximum state. This sequential process is illustrated in Figure 15.33, Figure 15.34 and Figure 15.35. The range of the volume control parameter can be split into two sub-intervals [0, 0.5] and [0.5, 1] to enable sequential control of the concave-convex shape modification and then modification of the extent of Flats, (Figure 15.41). Furthermore, as the transformation process applied by the Fixed-Flat end condition covers both types of shape modification and therefore the whole range of the volume control parameter, the end conditions can be made to behave cooperatively.

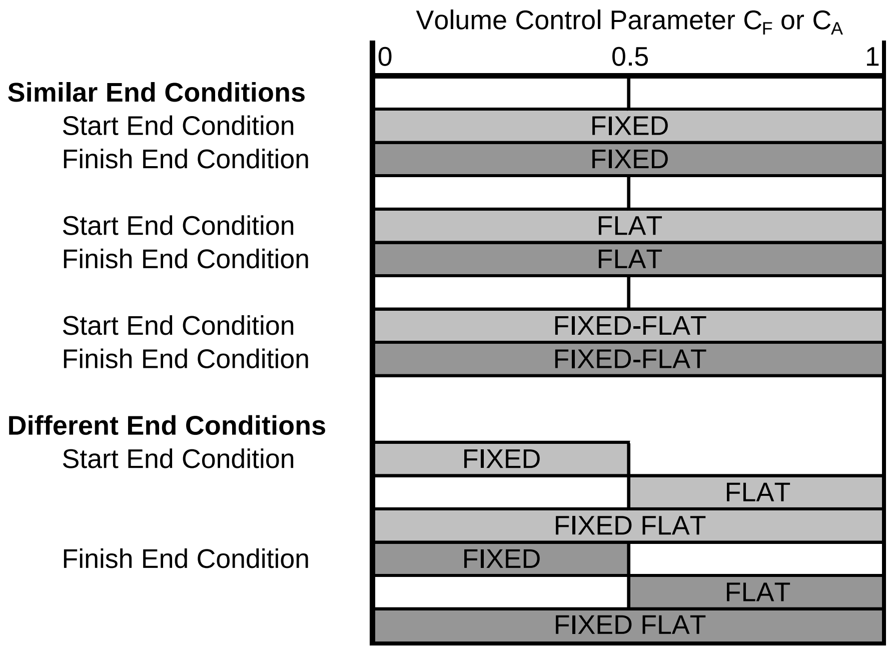


Figure 15.41, the control schedule of the end conditions for the Fullness Control function. Convex-concave modifications occur sequentically before modification of the extent of Flats. Ranges are adjusted to take account of similar end condition states.

The volume control curves are the major link in the relationship between the hydrostatic properties and the shape of the hull form, (Figure 15.43). Compared to other techniques, this approach is quite complex. However, this method develops hull shape considering the major definition features of the hull surface rather than using mathematical functions or iteration processes to develop a shape that satisfies desired numerical properties. Using the components discussed up to this stage, the technique is capable of producing a valid, but basic, hull form, (Figure 15.42). The next stage of development looks at using this initial hull surface as a basis for further manipulation a the local level, to include appendages such as bulbs, for example.

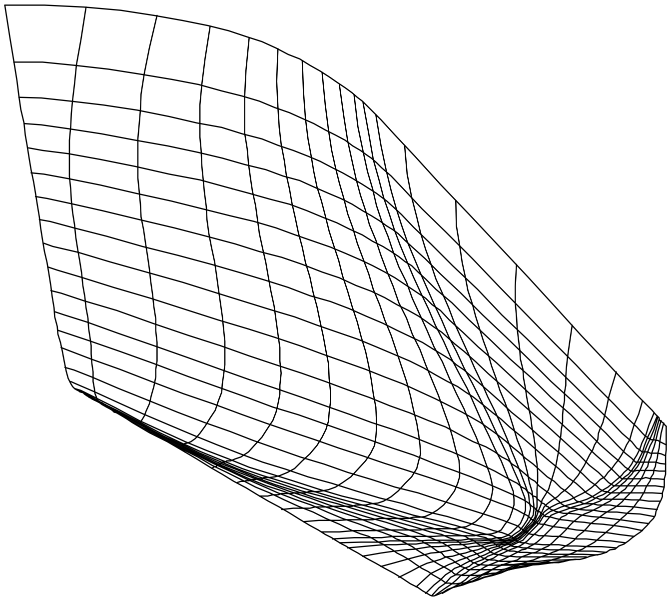


Figure 15.42, hull surface produced by Surface Interface Framework, Parameter and Curves components.

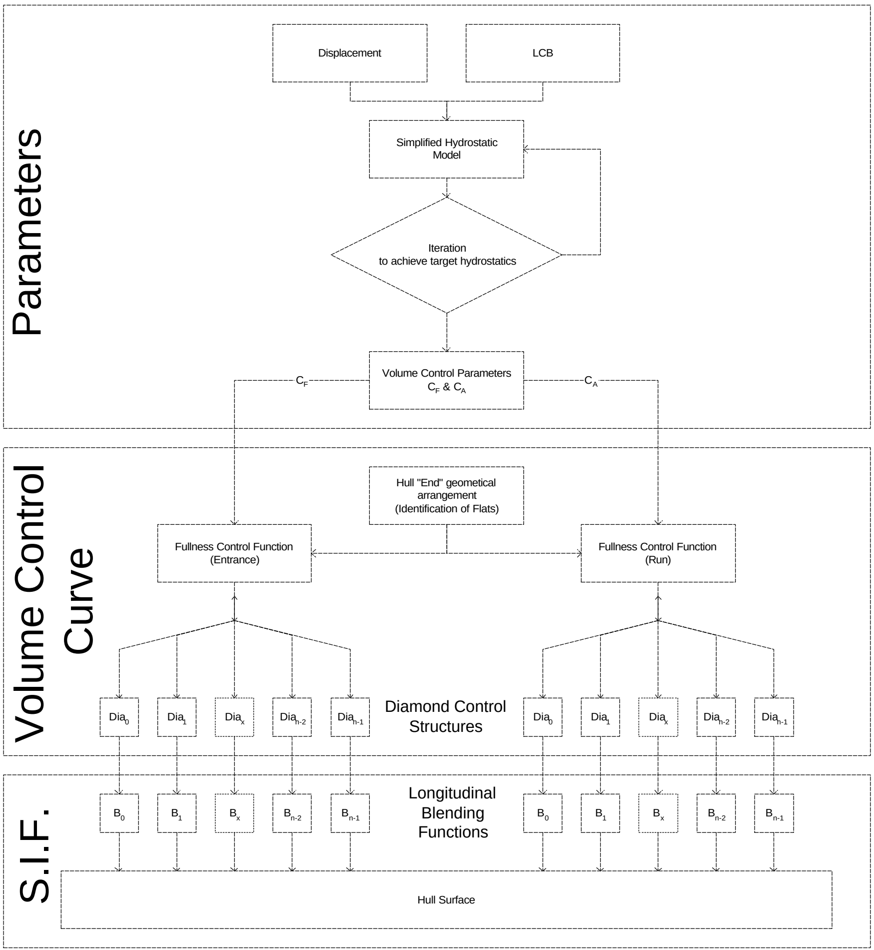


Figure 15.43, the structure used to develop the hull form using the hydrostatic information as a basis and considering the shape of the hull form definition.

16. POST-GENERATION SURFACE MODIFICATION TECHNIQUES

16.1. Surface Modification

The hull surface developed by the tools defined so far generate a basic hull form that can be modified by both parametrically and directly by interactive techniques. However, the hull form can not be suitably used for ship design at any stage, as there are no facilities for adding the common local features present in ship hull forms such as bulbs, keels or propeller shaft bossings. The surface interface framework creates the hull surface using a family of blending functions defined by a low number of definition curves. The blending functions ensure that the transition of the surface shape is fair between the different sections along the length of the hull. However, as a family of curves, blending functions do not allow local features, such as the bulbous bow, to be developed into the surface. The low number of definition curves minimise flexibility to ensure fairness of shape. To be considered as a practical alternative to present hull design techniques, the ability to develop local surface features is an absolute necessity. It may be possible to incorporate functionality into the blending functions to allow local features to be added. However, the simplicity of the technique would have to be sacrificed. An alternative approach, which is in keeping with the hierarchical nature of the hull development procedure, is to include, as part of the generation process, techniques that can be used to modify the local surface definition interactively, after the basic hull form has been produced.

The lack of tools to modify a hull surface after surface generation has been identified as being one of major reasons why the use of these techniques remains low. Developers have yet to offer any solution to the problem and evade the issue by allowing the user to export the hull surface to a system that allows direct manipulation of the hull surface. In hydrodynamic optimisation, it may be necessary to make small local adjustments to the surface shape at the control vertex level. However, such changes during the initial design phases result in time consuming operations to fair the surface around the modified region, with the added disadvantage that the generation technique may not have developed a control polygon mesh that is easy to intuitively manipulate. Furthermore, any possibilities of designing within an integrated environment are lost.

NURBS have matured into the most common mathematical technique for representing surfaces in software design tools. Hull design tools have concentrated on using the control polygon mesh to manipulate the surface. However, other design applications require different editing tools, some of which may be useful in the context of hull surface development. The large CAD systems used for generic product design, such as CATIA and Pro/Engineer, (Figure 16.1), have exceptional surface handing capabilities. However, the user does not need to be aware of the representation technique because surface models are mainly developed automatically from curves using lofting, skinning and lathe techniques and there is generally limited access to modify the surface using the control polygon. In addition to surface creation functions, these systems have developed highly advanced surface trimming capabilities that allow surfaces to be combined in almost any way. In artistic design tools, NURBS have been used as an electronic substitute for stone in sculpture. Special sets of functions have been developed to allow the surface to be manipulated in a similar style to using sculpting tools. Moreover, many more NURBS surface manipulation techniques can be found in academic papers and journals because of the large amount research that is conducted in this area. Unfortunately, as much of this research is directed into graphical design, these tools are often overlooked by developers of engineering design tools due to the lack of apparent accuracy in the approach taken by these techniques.

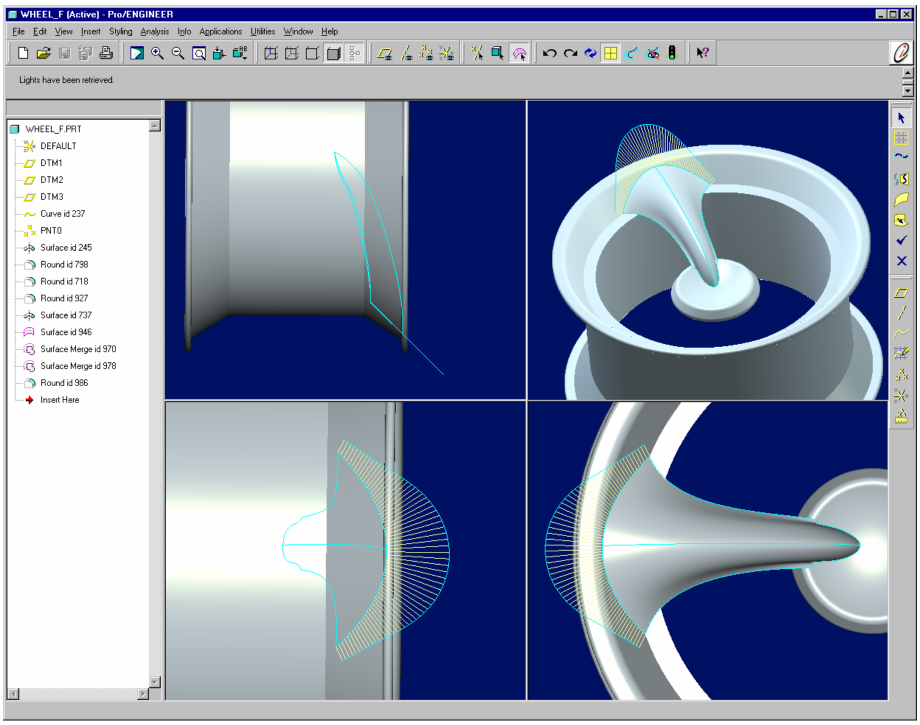


Figure 16.1, An example of a wheel constructed in Pro/Engineer using surfaces of revolution, lathe process and surface trimming.

The basic hull surface generated by TSCAHDE needs to be extended to allow the formation of local features and appendages. Surface trimming is the most appropriate technique to add separate features such as Skeg keels. As many existing hull design tools have surface trimming functionality it will not be necessary to discuss this technique in detail. However, tools to add integral appendages to the hull surface, such as the bulbous bow, do not appear to have been considered by hull design system developers. Techniques used in surface sculpting may offer a way of creating these shapes in a basic hull surface.

16.2. Trimmed Surfaces

Surface trimming extends the functionality of NURBS to a great level and is one of the primary reasons why the representation technique has become such an important tool in modern CAD systems. Surface trimming is a principally a computational task rather than a mathematical calculation. The trimming process is easy to understand, curves are simply used to mark boundaries between visible and hidden regions of the surface, (Figure 16.2). The major computational task is in identifying what parts of the surface to display. When combined with Constructive Solid Geometry (CSG), surface trimming can be used to design very complex parts, (Figure 16.1). Surface trimming is used extensively in ship hull design. The rectangular nature of the NURBS surface often results in situations where some part of the surface boundary must be removed to achieve the correct shape.

a) b) c)

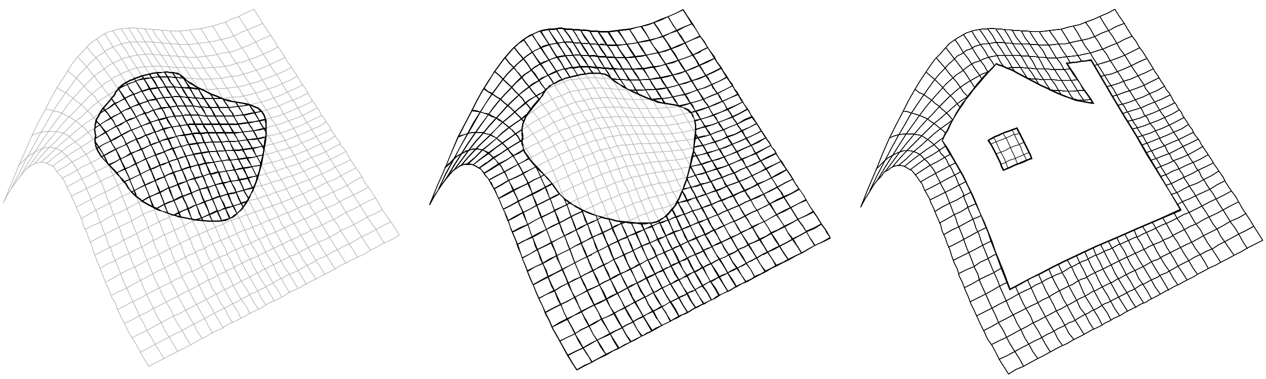


Figure 16.2, Depending on the arrangement of the actual curves, Trimming can be used to create a) islands, b) holes, or c) combinations of both.

There are some arguments between those who advocate relational geometry and those who advocate trimmed surface techniques for creating complex surface models. In the case of relational geometry, complex models are built by exactly matching the surface geometry at intersections, so that one surface boundary lies exactly on another surface. With trimmed surfaces, it is only necessary to trim the surface by the curve generated by the intersection of the two surfaces. While the relational geometry concept appears to be more elegant, it is reliant on the existence of completely compatible geometry. Moreover, models built using relational geometry may not have surfaces meet exactly because of limitations caused by the discretisation of the continuous mathematical surface function. Trimmed surface techniques have the ability to be more robust and have the advantage that mathematically incompatible geometry can be used together, an example of which is a NURBS surface intersected with a facet surface representation.

The rectangular nature of the single NURBS surface hull representation produces the major limitation on the types of hull form that can be developed with the single surface implementation of TSCAHDE. The technique is very capable of creating hulls with pram type sterns. However, hull surfaces with discontinuities in the hull boundary curves but not in the actual hull surface must be developed by extending the surface beyond the desired surface boundary and applying a trimming curve. An example of this configuration can be found in the hull forms of traditionally shaped vessels, with well defined stern posts and a flat transom. A discontinuity is located in the stern boundary of the surface at the base of the transom, where the boundary curve changes from being longitudinal to transverse. The development of this type of hull form using TSCAHDE requires the creation of a canoe stern that is trimmed to develop the transom, (Figure 16.3).

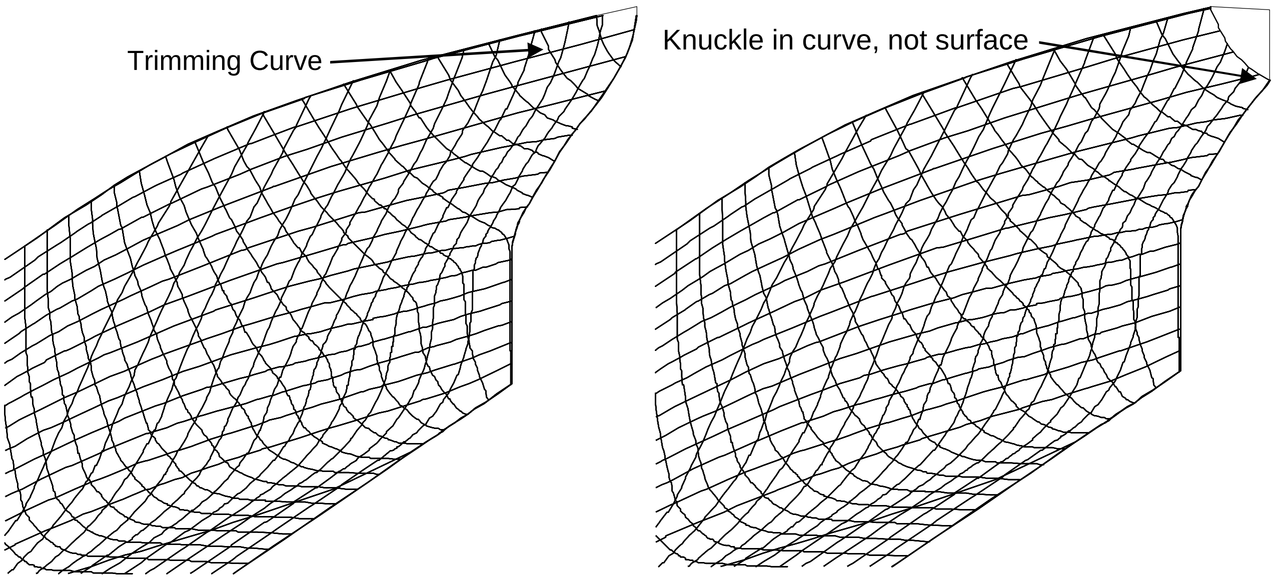


Figure 16.3, as traditional stern types have a knuckle point in the boundary, but not in the surface, the transom can be created by extending the hull aft and using surface trimming.

The incorporation of trimmed surface capabilities allows for the addition of hull appendages to the surface, such as particular types of bulbous bows and skegs, (Figure 16.4). Each appendage surface is developed separately and applied to the hull surface by calculating the trimming curve from the intersection between the hull and appendage surfaces. These appendage surfaces can be developed manually by editing the control polygon mesh, although a more preferable arrangement would be a parametrically defined surface providing a more efficient approach for optimisation later in the design process. There is also no reason why the actual appendage surfaces could not be developed using TSCAHDE approach, providing the implementation is generic and robust enough. However, as surface trimming is not available in PolyCAD [50], the CAD tool in which TSCAHDE is presently implemented, and the development of robust trimming tools would take some time, there are no current facilities to trim hull surfaces developed by TSCAHDE, although trimmed surface capabilities in PolyCAD are planned for future development.

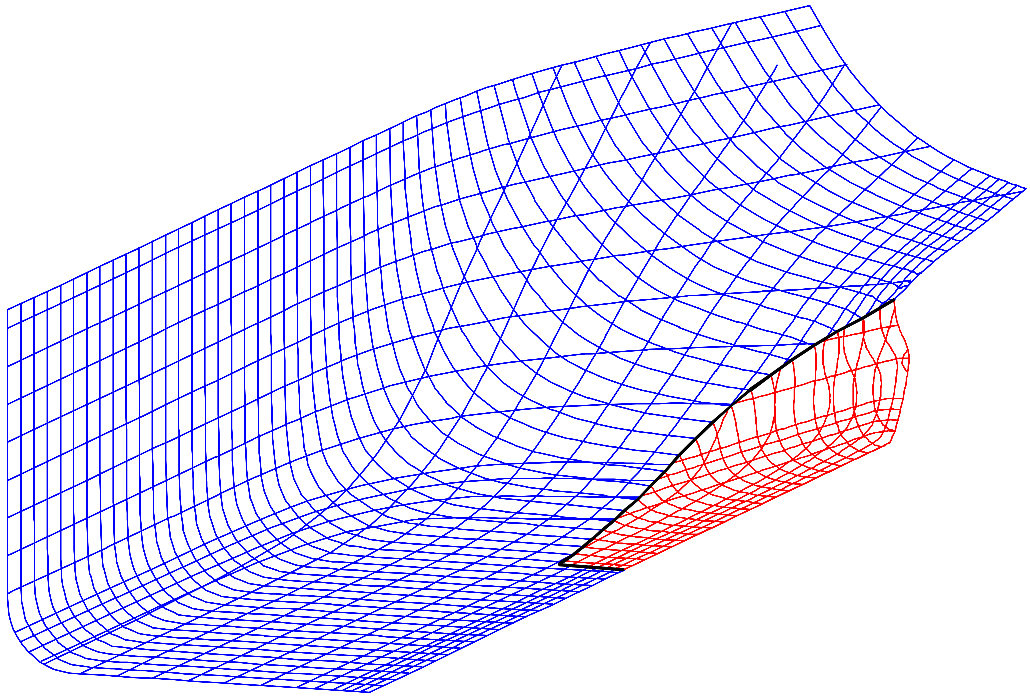


Figure 16.4, hull appendages can be created by defining a smaller local surface and intersecting with the main hull surface.

16.3. Surface Shape Modification Tools

Trimmed surfaces are a very useful technique of extending the functionality of NURBS surfaces. However, intersections in the hull surface between the main surface and appendages introduce knuckle lines. Knuckle lines below the waterline can have a significant effect on the hydrodynamic performance of the hull. Additional analysis is required to ensure that any discontinuities in the hull surface have a minimal effect on the water flow, increasing the costs of the design work. To add appendage shapes to the hull form with smooth transitions it is necessary to look for tools that are capable of modifying surfaces.

Mainstream engineering CAD systems have yet to include any tools that can perform large structured modifications to surface definitions. Given the mass use of surfaces across engineering, it seems that such tools would be a necessary function for all systems. However, as the majority of product design can be achieved with primitive surfaces, such as cylinders, boxes and planes, with complex surface shapes generated through constructive surface tools, such as skinning or lathe procedures, there is no great need for surface modification tools. Hull designers appear to be the only group of engineers where the direct manipulation of the surface definition is the primary technique of developing the hull form.

Tools developed for the qualitative modification of surfaces may offer a solution that can be used to add appendages into hull form surfaces with smooth and fair transitions between the surface shapes. These would normally be ignored by software developers because the application of the tools cannot be controlled accurately when applied interactively. However, with the TSCAHDE hull generation technique these tools can be employed using a systematic approach, resulting in the controlled application of these tools to the surface.

The application of large curved appendages, such as bulbous bows, to the surface will be the primary tasks of the surface shape modifications. Appendages with hard transitions to the hull surface can be accomplished through trimming. Therefore, the appropriate tool is one that can induce a bulge into the hull surface. Not many shape modification tools are appropriate for the task. Many are developed through academic projects for specialist applications and are not described in enough detail to allow robust reproduction of the techniques. However, one very good reference for a practical set of surface shape modification tools is the NURBS book by Piegl and Tiller [39]. This text describes in detail three shape modification tools: Warping, Flattening and Bending.

The Warping tool is the most useful in the context of inducing appendage shapes into the hull surface. The flatting tool doesn't have much relevance to the application of appendages to the surface and the bending tool applies a similar modification effect to the warping tool. The warping tool modifies the control polygon by displacing vertices in a direction normal to the NURBS representation, with the displacement being controlled by a function defined to a local area of the surface. To apply appendage shapes to the hull surface, a similar result is required, however, as appendage shapes are very specific, a better method of applying the appendage shape to the surface is required.

The warping function can be a very simple mathematical relationship that is used to displace control polygon vertices normal to the NURBS surface. This process, although an integral part of the warping modification tool, is too general for the application of appendages to the surface. Furthermore, it would be necessary to define appendages as mathematical surface functions, which requires an additional representation scheme. As warping functions represent a displacement of control vertices, this would not be as intuitive to control as modifying the surface control polygon directly. A more appropriate local surface modification technique is required.

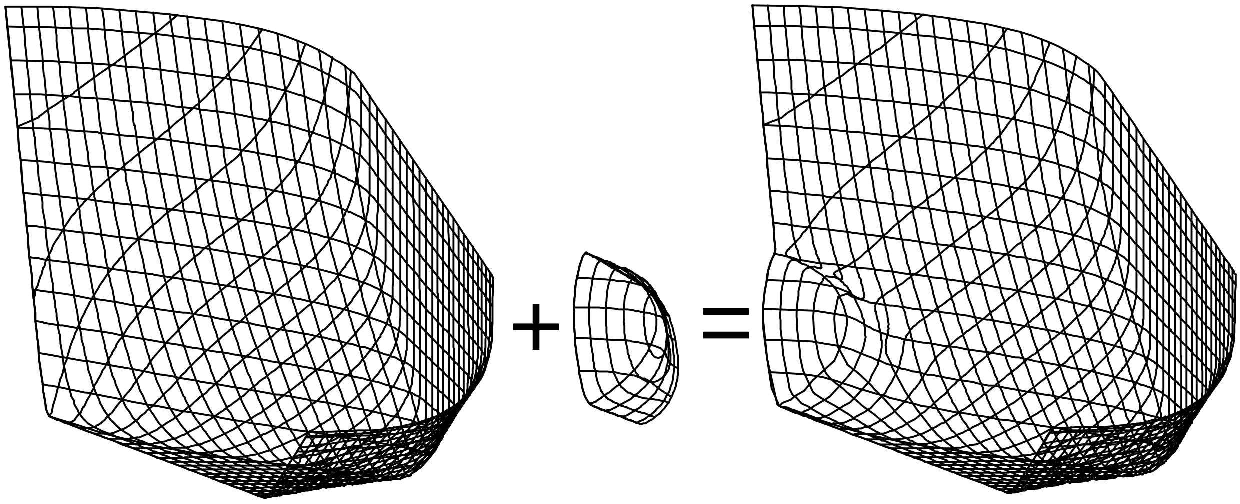


Figure 16.5, using the warping operation, the main hull surface is given a bulb using the local surface control polygon as the warping function.

A NURBS surface representation is, of course, the ideal technique to use to when applying appendage shapes to the surface. Instead of using the representation of the local surface to displace the control vertices of the main hull surface, if the control vertices defining main hull surface were moved to the locations of the control vertices defining local surface, the local NURBS surface appendage shape would be applied to the hull surface. The resulting appendage shape would not exactly match the local surface shape, however, as the NURBS surface has a shape that is approximately the same as the control polygon, the surface shapes would be very similar. The approach should be sufficient for design.

The warping technique, as described by Piegl and Tiller, is simple to develop. However, in contrast, the technique to move the control vertices of the hull surface to the locations of vertices within the local surface control polygon has some very important issues. The most significant problem is that the simple warping technique uses a smooth continuous function located in the parametric space of the NURBS. The local NURBS surface shape is not in parametric space and is defined by discrete vertices. Therefore, the selection of which vertices to move is a function of the intersection between the two surfaces and the results of the modification cannot be guaranteed to be smooth in the interface between the two surfaces without additional analysis and modification. These problems are exacerbated in the bow region of the hull surface because the control polygon can be seriously distorted.

The simple warping technique defines three steps to modify the surface: knot vector refinement, reposition control vertices and finally knot removal. These separate steps can be used as a basis to develop a technique for applying local surface shape to a larger surface. By adapting the procedure to suit the operation it is possible resolve the difficult issues.

The technique to warp the surface using a local surface control polygon takes the same approach as if the operation was being performed manually. However, the computer based procedure is able to apply the modification a lot quicker and with much greater accuracy. The whole procedure hinges around selecting a common grid of control polygon vertices between the two surfaces. The grid establishes a relationship between individual vertices on both control polygons. Consequently, a vertex relocation process can then be used to move vertices within main surface definition to the location of a corresponding vertex in the local surface control polygon. The procedural steps for this process are defined as follows:

Task	Description
1.	Unify Control Polygon Orientation A common grid of selected vertices within the control polygons of each surface needs to be accessed in a coherent manner. Unifying the orientation of the local surface control polygon with the main surface ensures that the increment direction in the columns and rows is the same on both surfaces. Without this, processing would be unnecessarily difficult.
2.	Define the range of a selection grid of control polygon vertices The initial expanse of the common grid is found by querying the location of control vertices. As the surfaces intersect, the vertices of one of the control polygons will be located on both sides of the other control polygon. The query is performed using a BSP (binary space partition) tree, constructed using the facet surface defined by the control

	polygon mesh representation. The control polygons are queried, instead of the actual surface, because the primary objective is to change the definition of the main surface. Control vertices of the main hull surface within the bulb control polygon, (Figure 16.6), and control vertices of the bulb surface outside the main hull control polygon, (Figure 16.7), are selected. The definition of both control polygons is processed to identify a minimum and maximum range of the common grid in rows and columns of both surfaces.
3.	Constructing a common grid of control vertices. It is highly unlikely that the selection grids on the main and local surface will contain the same number of rows and columns. Usually, as the operation will be applying refinement to the hull surface, the local surface selection grid will contain more control vertices than the main surface. Knot insertion is used to locally refine the main surface control polygon so there is the same number of control vertices within both the selection grids, (Figure 16.8). Knots are inserted considering the relative positions between the rows in the selection grids of the main and local surfaces. As Knot Insertion relocates control vertices around the location of the inserted Knot, the procedure must use several passes. One pass for each Knot inserted. The range of the common grid is updated for every Knot inserted. The Knot insertion does not change the shape of the main hull surface. It only refines the definition of the control polygon.
4.	Selection of control vertices on the main hull surface for relocation The common grid is a rectangular structure defining the maximum range of control vertices that will be affected by the operation. However, as the shapes of both surfaces can be quite distorted, not all the control vertices within this range should be moved, only those within the local surface control polygon. Vertices within the main hull surface control polygon are identified and selected prior to relocation, (Figure 16.9).
5.	Relocation of control vertices Selected vertices on the main surface are moved to the location of the corresponding common grid vertex on the local surface control polygon, (Figure 16.10).
6.	Smoothing of the intersection boundary Due to the discreet nature of the control polygon defintion and the BSP trees selection process. Control vertices on the intersection between the two surfaces may be out of

	place. Vertices on the interface are smoothed, (Figure 16.11), by considering the average point on two cubic splines in the row and column directions, through two control vertices on each side of the vertex to be smoothed. A comparison between the original and final hull surface control polygons is shown in Figure 16.12.

Techniques capable of improving the basic hull surface are very important for TSCAHDE. The simple definition curves and surface interface framework are designed to be only capable of developing the basic shape of the hull surface. However, any ship hull design technique must provide the designer with the ability to include the hull appendages in the design. Trimmed surfaces are now a well established technique used to combine numbers of surfaces together. However, as this technique alone cannot be used to refine the shape of a basic surface, the advantages of its use with TSCAHDE are limited. There has been hardly any development of practical tools capable of performing structured modifications to the shape of hull form surfaces. The example surface modification tool demonstrates that even crude procedures are capable of producing representative appendages in the hull form. The inclusion of surface modification techniques takes TSCAHDE from being just an interesting approach of generating a simplistic hull representation to becoming a practical design tool. With some further development of practical and robust surface modification tools combined with some basic trimmed surface functionality, hull design using a TSCAHDE based approach would practically render all existing hull design tools obsolete.

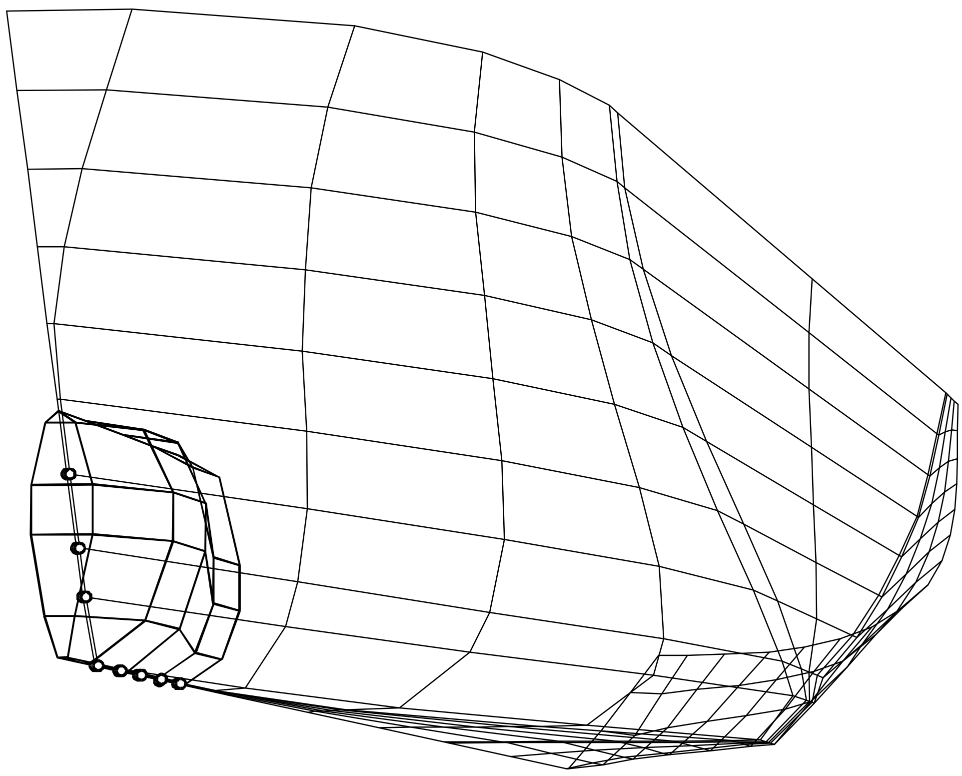


Figure 16.6, main hull surface control vertices are selected only on the inside of the bulb surface.

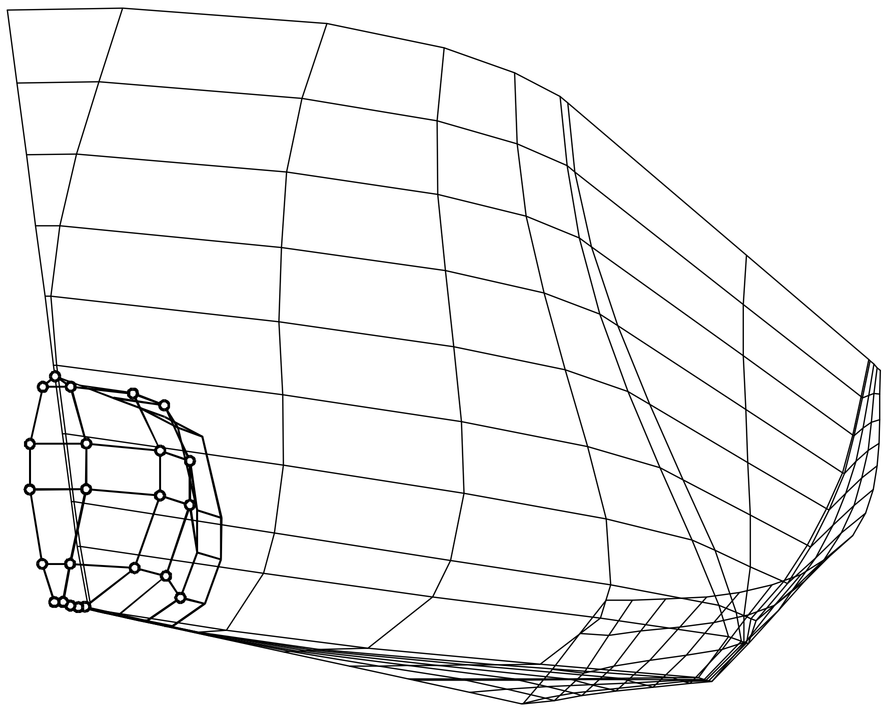


Figure 16.7, bulb surface control vertices are selected only on the outside of the main hull surface.

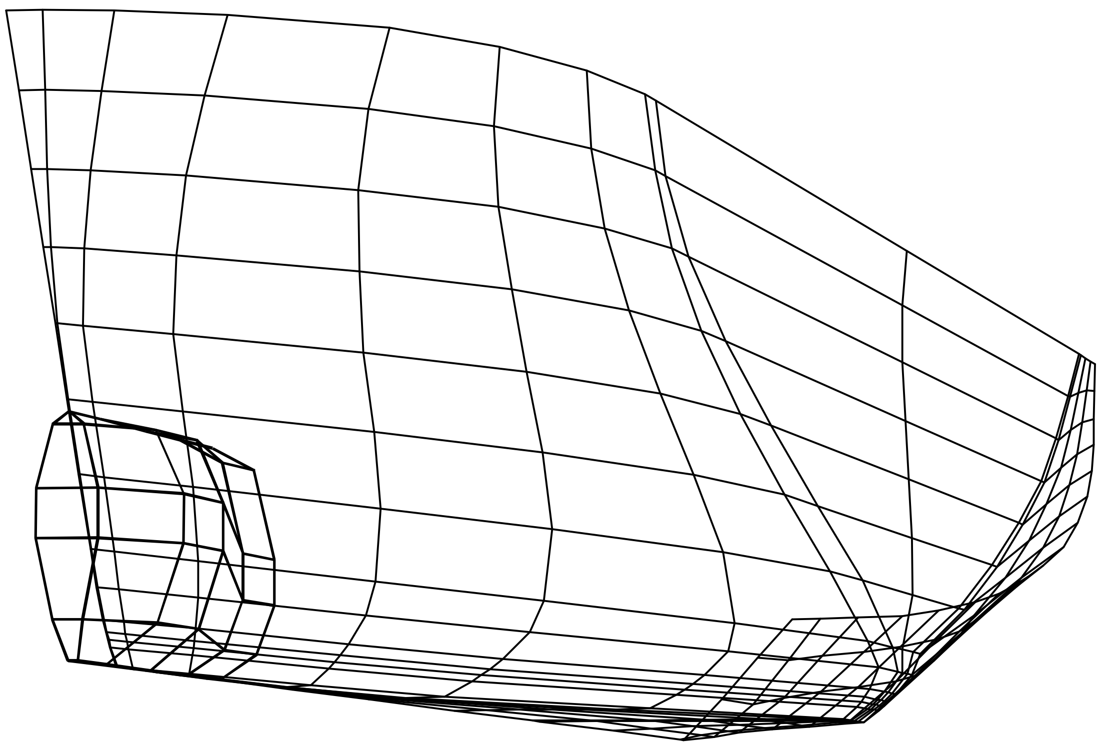


Figure 16.8, main hull surface refinement local to the area around the bulb surface.

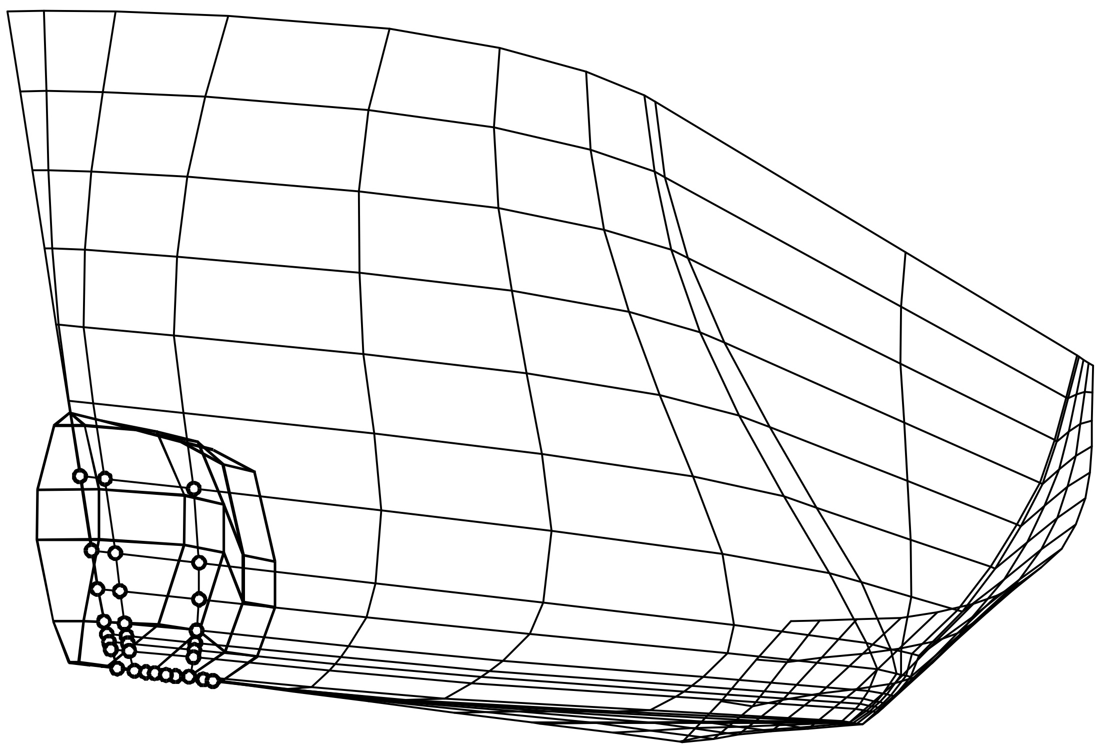


Figure 16.9, Main surface control vertices selected only within the bulb surface control polygon.

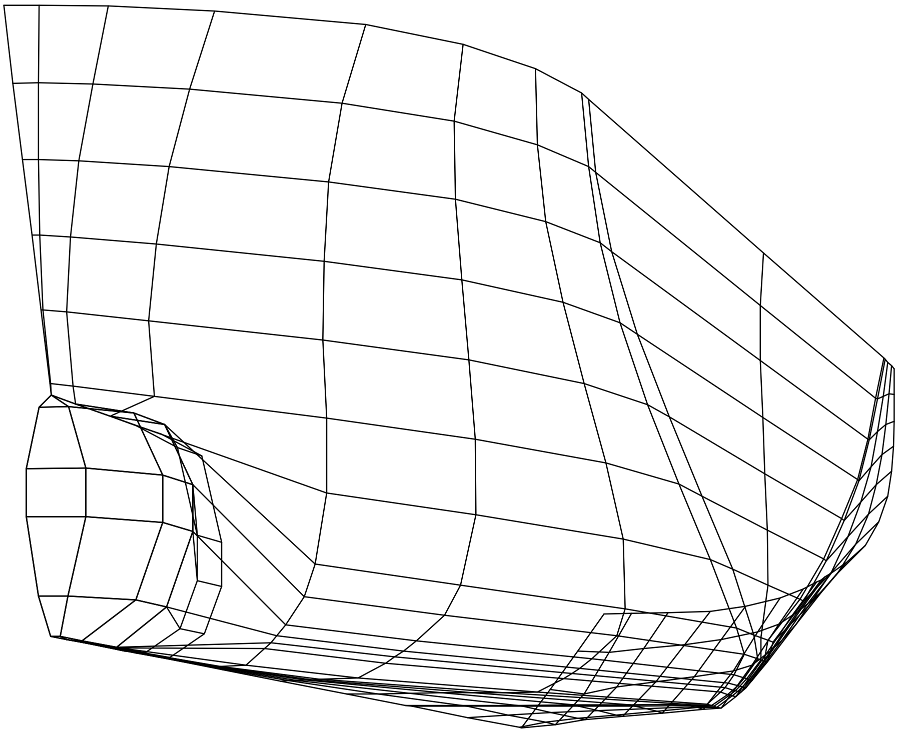


Figure 16.10, control vertices on the main surface are relocated to vertices on the bulb surface control polygon based on the common grid.

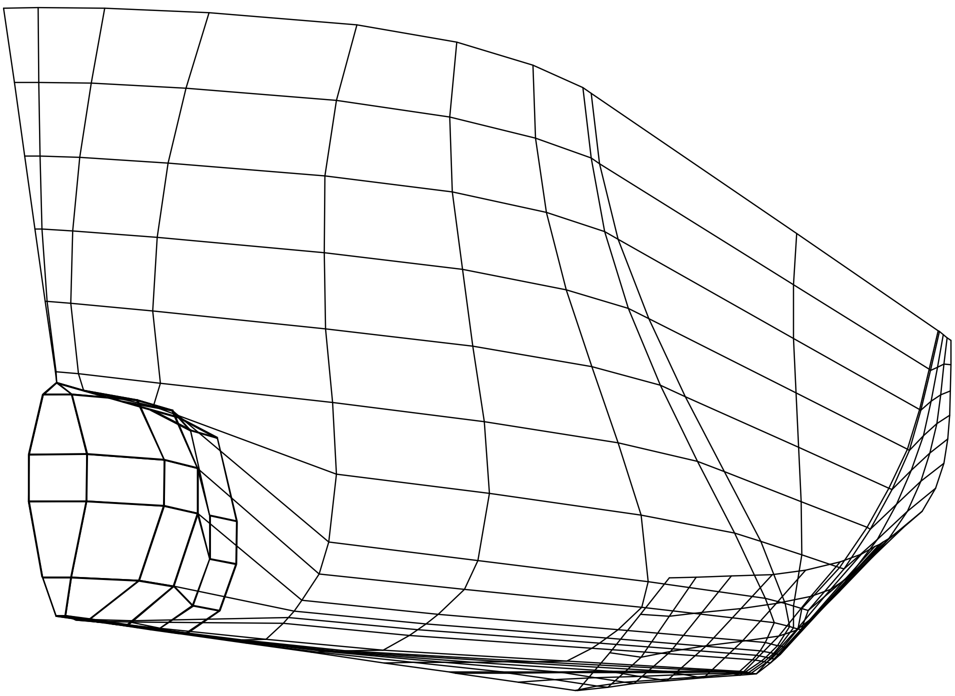


Figure 16.11, control vertices on the border of the common grid are smoothed to ensure a fair transition between the two surface shapes.

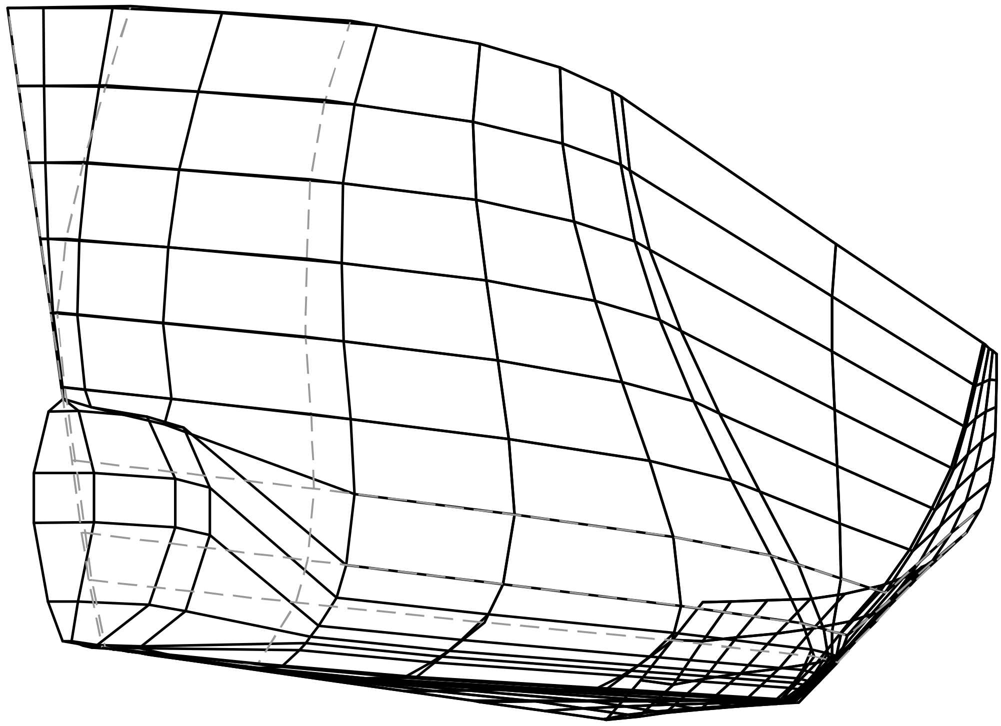


Figure 16.12, the comparison between the initial hull surface control polygon and the modified hull surface control polygon.

17. IMPLEMENTATION

17.1. CAD Platform

When compared with other hull form generation techniques, TSCAHDE cannot be discussed like a direct hull generation procedure, with an identifiable start and end. However, when compared to present hull surface definition tools, there can be considerably more processing after each user interaction. TSCAHDE is a hybrid combination of the two approaches. The user makes a large contribution to the effectiveness of the procedure by constructing the framework of definition curves about which the hull surface is generated. Hence, the implementation must ensure that the user has adequate tools to develop the desired definition shapes.

The structure of TSCAHDE is very similar to that of ShipLINES. However, the user must create some initial definition before a hull surface can be constructed, whereas, in ShipLINES all the definition curves are developed parametrically, although pure parametric hull generation is a possibility using the TSCAHDE approach. As the user must define curves, an appropriate design environment is required. PolyCAD [50] is a CAD application that has been developed as one of the tools used within this project, although its uses are much wider. PolyCAD has been developed with a certain philosophy to geometry manipulation, allowing the user to interact and modify geometry in the cleanest and most streamlined approach possible. Therefore, it is appropriate to implement TSCAHDE within the PolyCAD environment.

PolyCAD is the latest version of a chain of hull design programs developed by the author dating back to 1993. The first version, known as 3DBoat (Figure 17.1), allowed a hull to be visualised in three dimensions. Hull forms could be entered using section offsets and waterlines could be generated by considering planer intersections of the hull stations. In 1994, with access to better computer technology, development of HullCAD was begun, (Figure 17.2). This CAD tool could be used to develop a single B-spline surface. The contours of the hull surface could be calculated and there was analysis of Hydrostatics and Intact Stability. Both applications were DOS based, with the 3DBoat being developed in QuickBASIC and HullCAD in Turbo Pascal. Being DOS applications, restrictions resulting from the variation in graphics hardware and programming language capabilities became major restraints of development. These applications could never match the quality of the facilities provided by commercially developed tools such as AutoCAD.

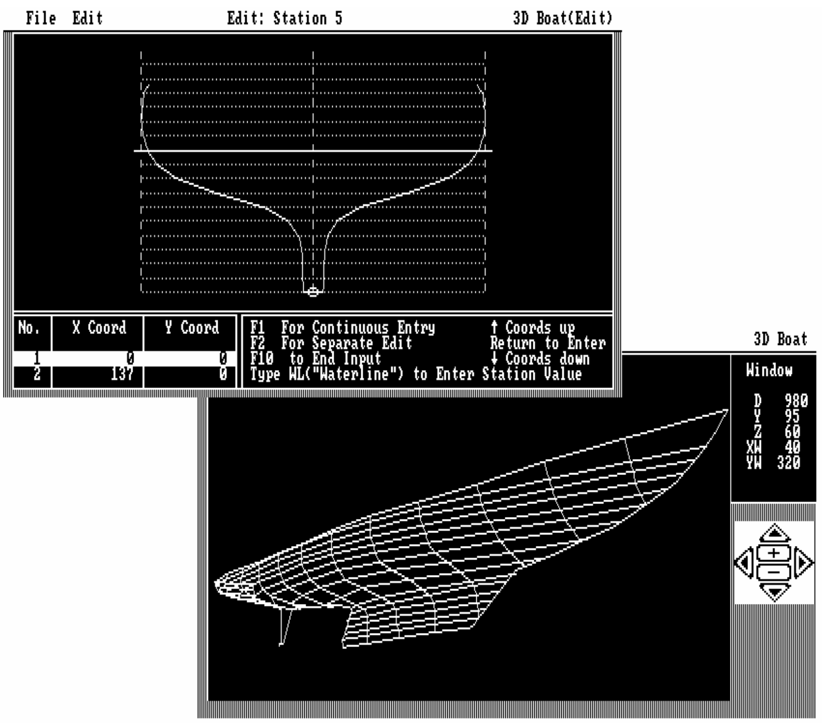


Figure 17.1, 3DBoat (1993-1994), manually entered section offset data displayed in 3D. Waterlines contours calculated.

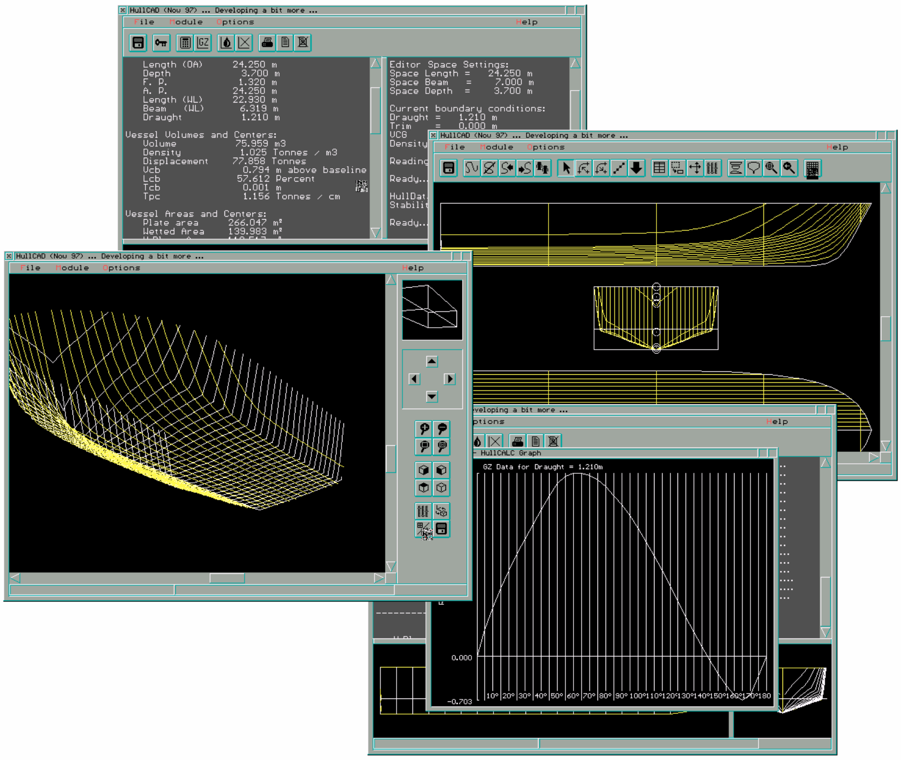


Figure 17.2, HullCAD (1994-1997), Single B-Spline surface hull form, calculation of contours, calculation of hydrostatics and intact stability.

The introduction of Borland Delphi, a Pascal based language, initially only for the Microsoft Windows operating system, removed all programming issues relating to hardware and provided standardised access to modern computer features, such as the Mouse, Printer and 24 Bit Colour. A professional quality geometry manipulation tool could be developed using these features, PolyCAD.

In contrast to the earlier hull design applications, PolyCAD is designed to deal with a wider range of entities and geometric operations, although all are based around operations required in the marine design environment. The PolyCAD software is based on a library of standalone geometric entities that are capable of being modified, drawn and saved to file. Further functions allow entities to be transformed between each other, (Figure 17.3). The library can form the basis of any application requiring geometry capabilities. The PolyCAD application itself is a modern front-end graphic user interface supporting the functions of the library. As PolyCAD implements the library, it is an indispensable tool while developing geometry operations using the library code. It can be used to load and visualise the results of geometric calculation. It can then continue the manipulation process, manually, allowing the development of solutions that can be later coded. It has become the case that PolyCAD is the major tool for debugging the results of geometrical operations performed within the application.

While implementing a very powerful geometry library, PolyCAD is also an ongoing experiment into the user interface design of geometry-based software applications. The best approach to the design of the graphical user has been to develop an uncluttered display environment. This has been achieved through developing a PolyCAD class which "wraps" each geometrical entity class within the library. Wrappers (also known as an Adapter) are an object orientated programming definition within the Design Pattern approach for a class that converts the interface of one class (in this case the library entity) into another which the client (PolyCAD) expects. More information on the Wrapper Design Pattern and others can be found in [51]. The Wrapper class manages all the communications between the User and the Entity. Consequently, as a user is only capable of editing one entity at a time, there is never a situation where the user can be overloaded with too much command information. Furthermore, as there is only one Wrapper editable at any one time, the software is very context-sensitive because the user can only issue commands that are relevant to the entities being edited. All command functions are accessed through a standard "Right-Click" pop-up menu, which further reduces cluttering of the display.

A variety of entities have been implemented mainly concentrating on the basic elements required in naval architecture, such as Polylines, Polyline meshes, Facet Surfaces, B-spline and Cubic Spline curves and B-spline Surfaces. Consequently, PolyCAD can be used in all areas of marine geometry definition from hull surface design and compartmentation to CFD mesh manipulation, (Figure 17.4).

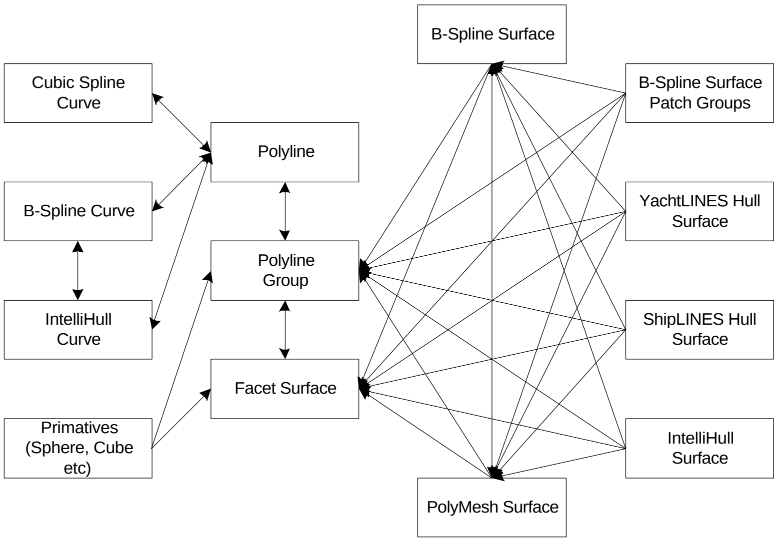


Figure 17.3, One of PolyCAD's most important functions is the ability to transform between the different geometry representations.

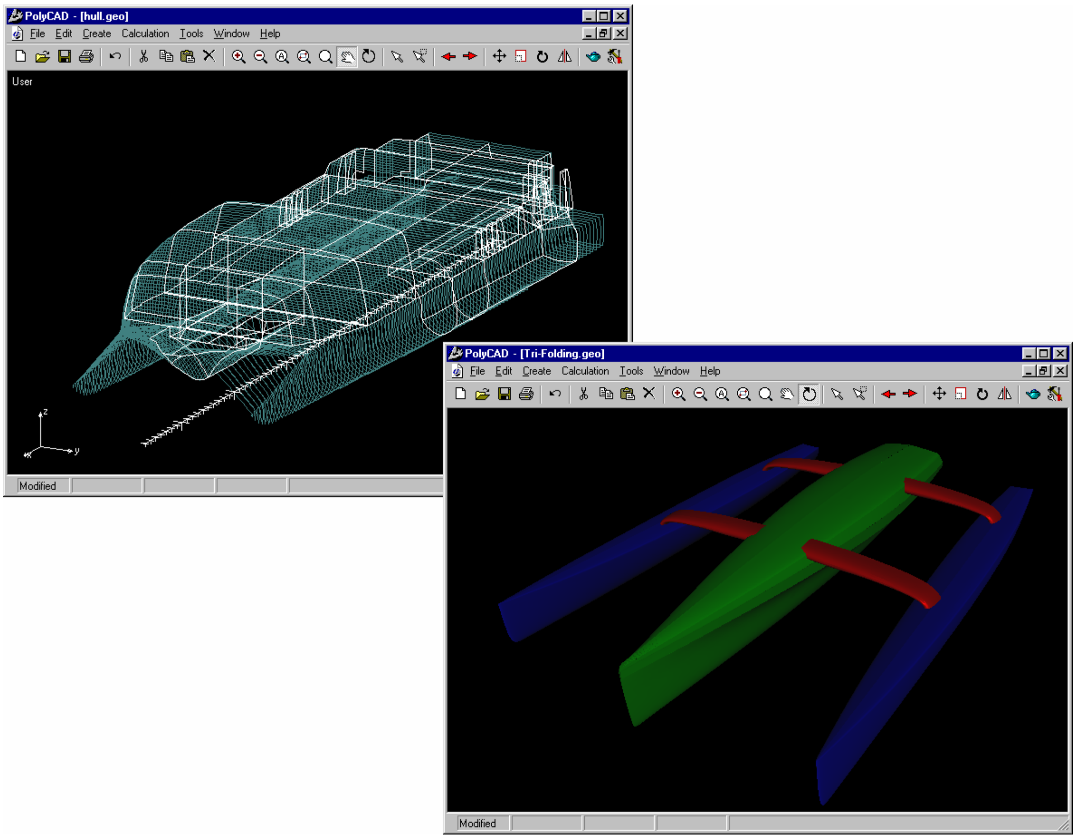


Figure 17.4, Examples of PolyCAD modelling capabilities. Compartmentation within a Fast Catamaran Ferry and a Folding Trimaran design by Svend Vogt Andersen.

17.2. TSCAHDE Single Surface Implementation - Structure Overview

The PolyCAD implementation of TSCAHDE adopts the standard entity-wrapper structure to construct the code used to develop the hull surface. The implementation takes full advantage of the Delphi OOP model. Four main classes have been defined, (Table 17.1), two definition and two wrapper classes. The user can only manipulate data held within the wrapper classes and definition classes do the work of transforming the data into a hull surface. The definition classes are capable of generating a hull surface without the presence of the wrappers. The TSCAHDE classes combined with the wrappers for the user interface to develop the IntelliHull feature of PolyCAD, Figure 17.5. The relationship between these classes is shown in more detail for the definition of the Bow of the surface in Figure 17.6.

Class	Description
THullCurve	A structure of vertices used by THullBlender to represent the control polygon of

	each B-spline definition curve and implement transformations of the curve definition data. A THullCurve can be user defined or automatically generated. If a curve is automatically generated, THullCurve owns (manages the memory) of the structure, otherwise the curve data is owned externally, by the TIntelliCurve structure.
THullBlender	A structure of definition curves (THullCurve) from which a hull surface can be constructed. THullBlender is the class that constructs the surface from the definition curves using the Blending Functions. It implements the hull form parameters and manages transformations when parameters are modified and it handles the iteration to achieve the specified Displacement and Longitudinal Centre of Buoyancy.
TIntelliCurve	A wrapper to THullCurve curves. Implementing user manipulation and modifier constraints.
TIntelliHull	A wrapper to THullBlender, implementing management of the definition curves, user access to parameters and a parametric Bulb surface to demonstrate Local Surface Modification.

Table 17.1, Task descriptions of the OOP Classes that are used to implement TSCAHDE.

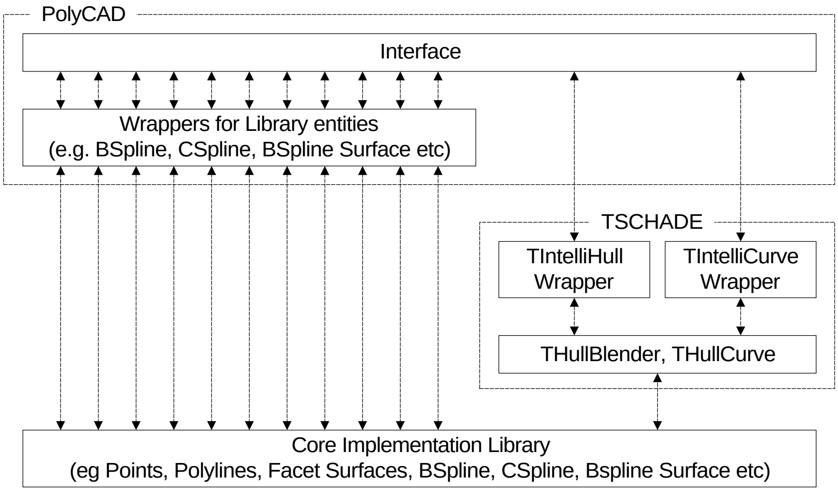


Figure 17.5, The TSCHADE classes (THullBlender and THullCurve) are combined with wrappers (TIntelliHull and TIntelliCurve) and linked as one of the independent modules within PolyCAD.

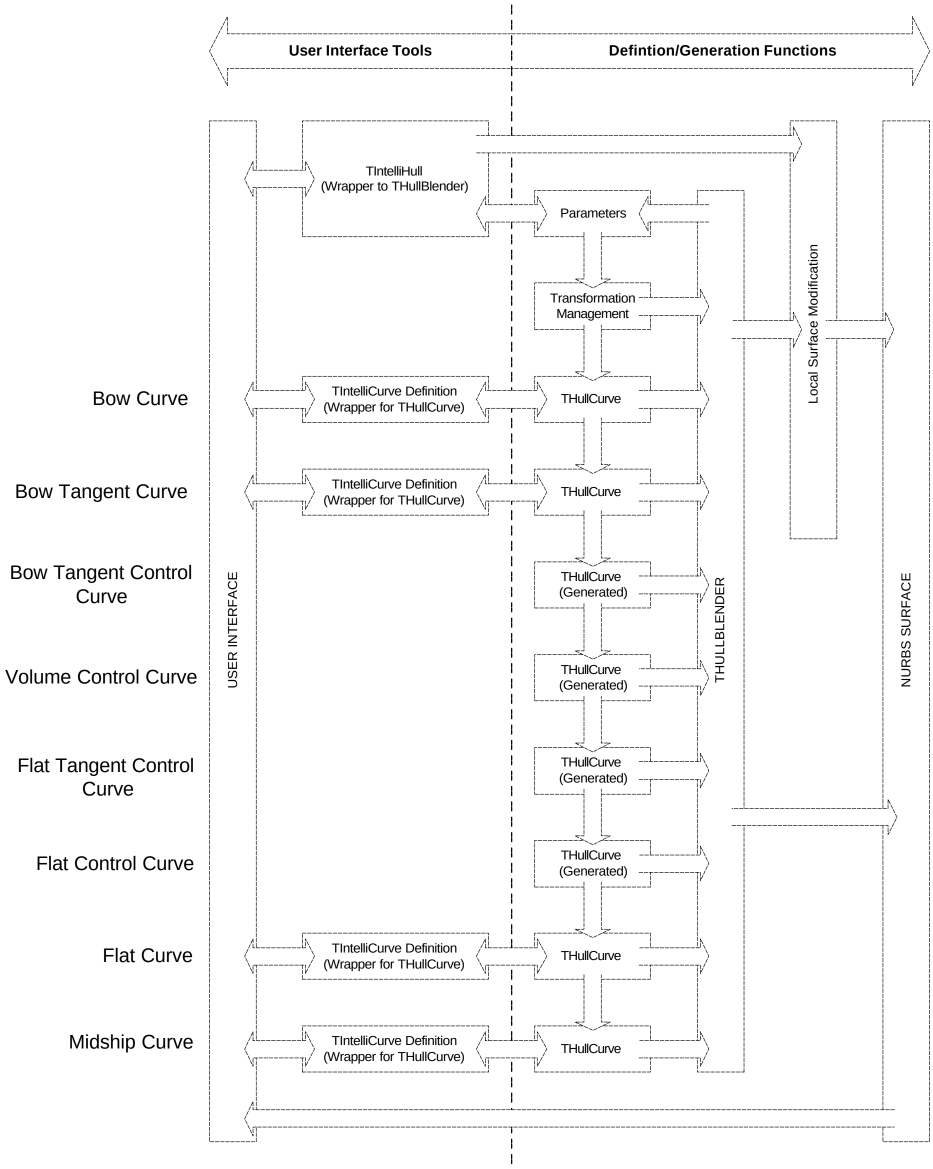


Figure 17.6, Interaction Layers (for bow portion of a hull only)

17.3. The Non User-Manipulated Classes and Procedures

The THullCurve class is the simplest structure in the implementation. It represents the control polygon of a B-spline definition curve. It is not necessary to access the B-spline curve representation, as discussed in Chapter 13, each vertex location on a definition curve forms a control polygon vertex for the Blending Function.

The most important task the THullCurve performs is to transform curve definition data when a parameter is changed. The most efficient use of parametric modification was found to be for changing the global or principle dimension of the hull form and large shape regions within the hull surface. As there are a limited number of principle parameters, only a small number of transformations can be applied to the definition curve data directly. Seven curve transformations are implemented. Three concern the modification of all curves for changes of principle dimensions such as Length, Beam and Depth. The remaining four parametric modifications concern varying the extent of the parallel middle body and the parallel deck and are applied to one curve only. Hence, there are two for each curve defining the extents of the prismatic section of the hull form. THullBlender calls each THullCurve definition with the transformation to be applied and it is the responsibility of the THullCurve to apply correctly.

Each curve is tagged based on the task it performs. This tag is used to decide if a transformation should be applied to the particular curve when a parameter is changed. In the example of lengthening a hull surface, Chapter 14, the transformation increased hull length by shifting all curves forward of the midship section curve by the increase in length, and the midship by the half the increase in length. THullCurve queries the tag and location of the curve to decide whether to translate the curve data by the change in length, half the change in length or not apply a transformation. Transformations are not applied to generated THullCurves because the definition is dependent of the geometry of the user defined curves. The geometry of generated curves is updated directly after the application of a transformation or user modification.

The THullBlender class plays a management role over the structure of THullCurves. Its most important responsibilities concern the construction of a complete set of definition curves and then the development of the initial hull surface, i.e. the surface before local modifications are applied. It also manages the set of principle parameters used to modify the hull form by identifying what features of the hull form are capable of modification and calculating parameter values by querying the definition curves and the hydrostatics of the hull form for a given design draught.

THullBlender contains the array of THullCurves defining the hull surface. Before a hull surface can be generated, it must first identify the curves provided by the user, adding further generated geometry to ensure the Blending function control polygons can be properly completed. A parsing process is used to first identify the main definition curves such as the bow, stern and midship section curves and then to locate the task of the other definition curves between. This process is best illustrated by a detailed review of the parsing procedure tasks for the set of user defined curves shown in Figure 17.7.

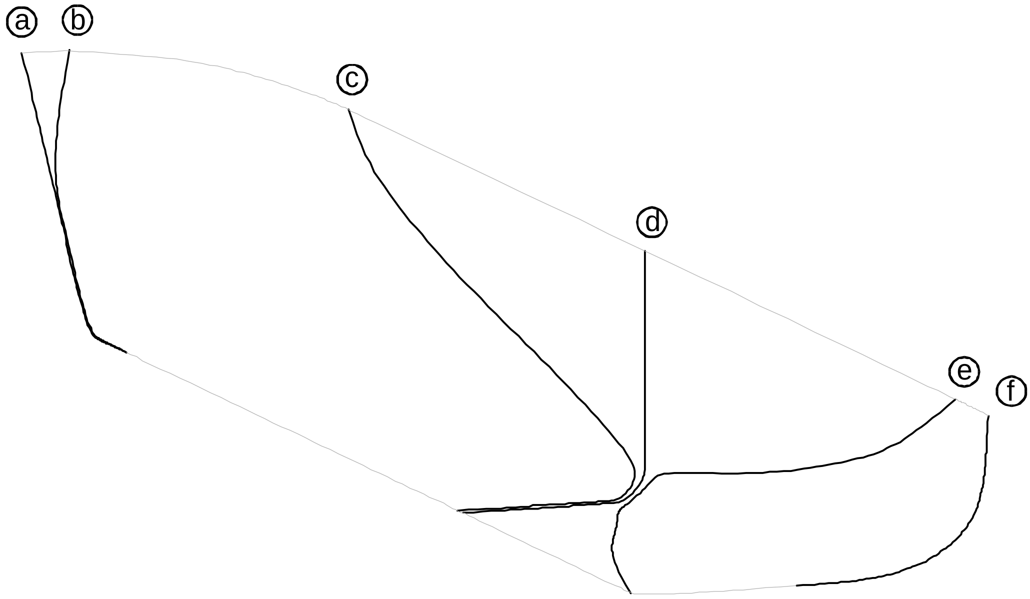


Figure 17.7, a set of initial user-defined definition curves from which the complete hull surface can be constructed.

<u>Task No.</u>	<u>Task Description</u>
1.	Sort Curves For practical purposes, curves are assumed to be in random order. Curves are sorted based on the location of the definition vertex at the deck. The sort compares the *x* component between curves. If the x component is found to be equal, the *y* component of the vertex is compared. The final ordering of the curves must run from the Bow to Stern.
2.	Identify Longitudinal Boundary Curves

	Based on the sort, the boundary curves of the hull surface can now be identified as the first and last curves in the array, ((a) and (f)).
3.	Identify Midship Section Curve The Midship section curve is identified using a weighting system based on the location of a definition curve from the midship (LOA/2) and the respective longitudinal extent of the three adjacent curves, see equation below. The midship section definition is identified as the curve with the largest weighting. $$\text{Weight}_i = L\left(1-\frac{\left\lvert \frac{L}{2}-\text{Curve}_i.X \right\rvert}{\frac{L}{2}}\right)+\left(\text{Curve}_{i-1}.\text{XExtent}-\text{Curve}_i.\text{XExtent}+\text{Curve}_i.\text{XExtent}\right)$$ Where: L is the longitudinal extents of the definition curve data. .X is the *x* component of the first vertex on the curve .XExtent is the longitudinal extent of a definition curve
4.	Identify User Defined Tangent or Flat Definition Curves Curves directly next to Boundary or the Midship section are analysed to see if they represent a tangent or a flat definition with respect to the midship section curve. A curve is considered to be a tangent or flat if 50% of the definition vertices have exactly the same coordinate component values in *x*, *y* or *z*, compared with the boundary or midship section curve. Curves identified as tangents or Flats will be tagged appropriately. Any remaining user defined curves will be defined as “control curves”, i.e. curves that define no specific feature.
5.	Add Generated Flat Control Curves (Marked as (1) in Figure 17.8) Curves defining the collinear structure of vertices to form the extent of the flat are automatically generated and inserted into the array of THullCurves based on the location of previously identified the flat definition curves. Of course, the user has the choice to change this arrangement once the structure of curves has been developed and presented.
6.	Add Generated Tangent Control Curves (Marked as (2) in Figure 17.8)

	Tangent curves are identified, and in a similar manner to Task 5, curves to construct the tangent within the surface control polygon are automatically generated,
7.	Add Generated Volume Control Curves (Marked as (3) in Figure 17.8) If no user-defined control curves are found between the surface region boundaries, i.e between the bow and the forward Flat and the aft Flat and the Stern, curves to control the hydrostatic properties of the hull form are inserted.

The process is completed once the definition has been reviewed for volume control curves. It is now possible to construct Blending functions to allow the NURBS surface definition to be generated. The complete structure of definition curves is shown in Figure 17.8.

In an optimised approach, Blending functions and surface definition generation are processed sequentially for each region. The Blending functions are constructed by adding control curve vertices the corresponding Blending function control polygons until a region boundary is reached, i.e. a hull surface boundary or flat definition curve. Note that the Blending functions are constructed using the definition curves within the boundaries of the region and not the actual region boundary curve data. As the region boundary curves form part of the surface definition, this data is added after each region is processed. Once a boundary has been reached, all blending functions will have been constructed and the appropriate range of hull surface control polygon can be generated. One column of the surface control polygon is created per definition curve using equally spaced intervals of the B-spline parameter *t*, on the Blender functions.

After the basic surface is constructed, the process allows local surface modifications to be applied to hull by external *call back* procedures, (in this implementation local surface modifications are handled by the TIntelliHull class), before calculating the hydrostatics and parameter values. The complete procedure is illustrated as a flow chart in Figure 17.9.

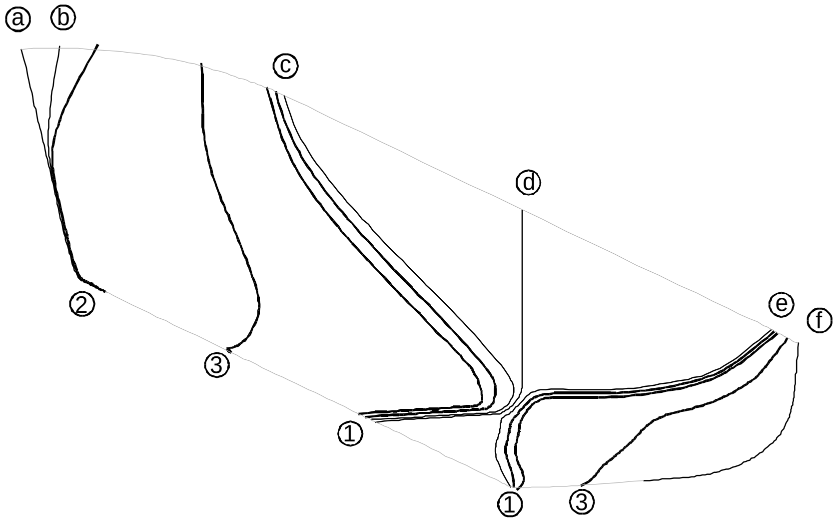


Figure 17.8, Complete set of hull definition curves, automatically generated curves are numbered (refer to task list).

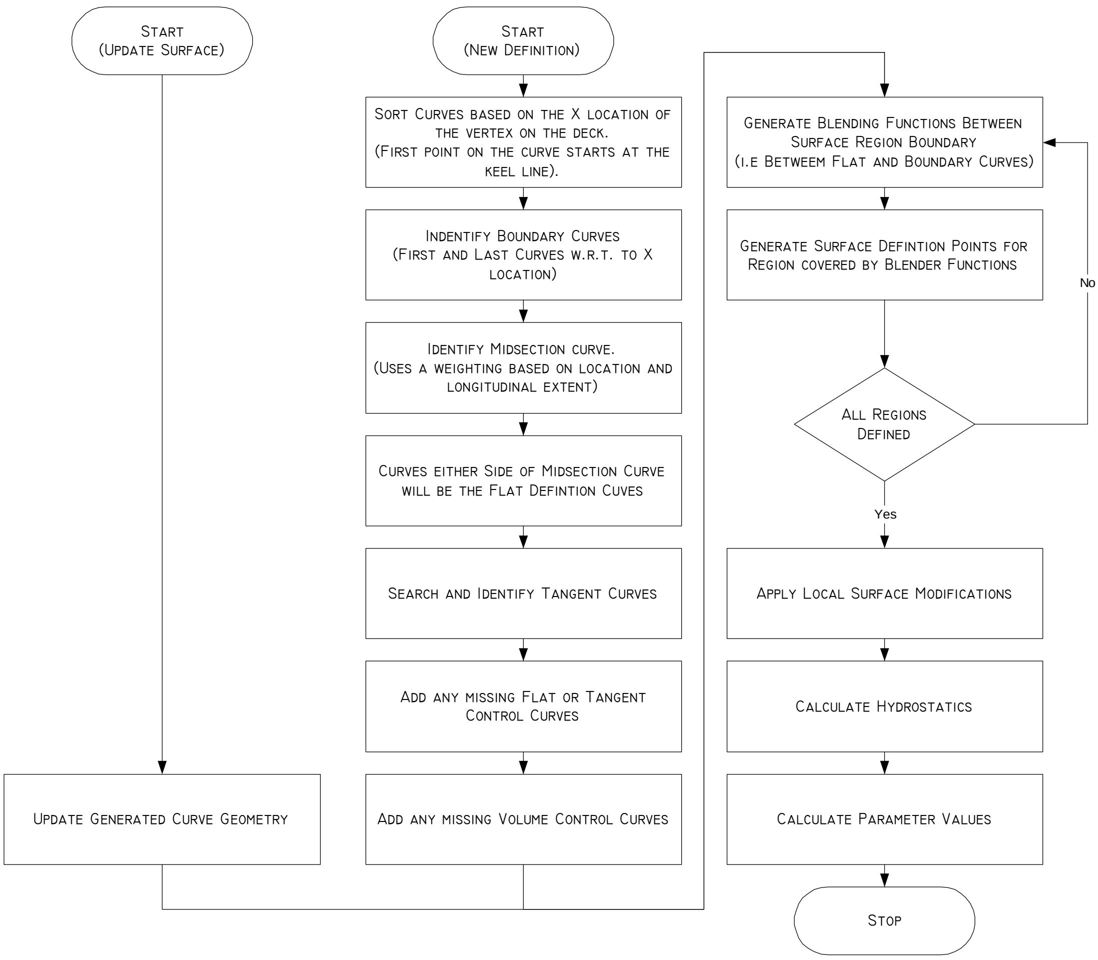


Figure 17.9, the hull surface generation procedure flow chart. Note the start locations when beginning from new definition or an update to a definition curve.

The automatic generation of definition curves does not present much of a problem. The three types of generated definition curves defined in Chapter 15 are implemented as class descendants of the THullCurve class. All generated curves develop vertex geometry based on calculations between other form definition curves resulting in a more elaborate form of the relational geometry concept. The generated curves are provided with the relevant reference curves on construction, allowing the calculated definition to be updated any time. Through polymorphism, THullBlender recognises the curve as a standard THullCurve, hence all curves, user-defined or generated, can be treated in the same manner.

The THullBlender class manages the iteration procedure searching for the Displacement and LCB when relevant parameters are modified. The iteration technique follows the procedure laid down in Chapter 14. The process manipulates the Volume control curve parameters until the hydrostatics of the surface are found to be within tolerance or the procedure decides that the desired hydrostatics are unattainable. If the hydrostatics are found to be unattainable, the Volume Control parameters are returned to the state before the iteration procedure was entered. As only the displacement and LCB hydrostatic values are required, an optimised calculation process is used which does not calculate additional hydrostatic parameters.

17.4. TSCAHDE User Manipulated Tools

The greatest advantage from implementing TSCAHDE within PolyCAD is that all the user manipulation, file and drawing functions are already implemented. PolyCAD provides the capabilities to view the curves and surfaces in three-dimensions, edit curve definition vertices, save data files, manage *undo* operations and view the contours of the hull surface without requiring any additional functions to be provided by the TSCAHDE implementation code. The wrapper classes are developed from a main ancestor wrapper class, which implements all the basic functions such as drawing and editing capabilities. The TIntelliCurve and TIntelliHull classes only have to override the edit and draw procedures to allow the user to view and manipulate the wrapper. While PolyCAD already had most of the required user interface features before TSCAHDE was implemented, the technique required the ability to enable the user to manipulate geometry directly and parametrically. This did not create any programming difficulties, however, a user interface had to be developed which allowed both manipulation techniques to take place at the same time. To implement this, PolyCAD received a major upgrade before the functional version of

TSCAHDE was added, to implement a side panel similar to the style used by 3D Studio Max [54] and furthermore ensuring that manipulation procedures were standardised for all existing entities with the new editing feature.

For the most part, the TIntelliCurve wrapper class is a copy of the basic general B-spline curve entity. However, while the B-spline entity presents the user with all the features capable from a B-spline function, the B-spline curve implemented by TIntelliCurve constrains the user to a much reduced set of features, such as the degree of the curve being fixed to cubic. The primary reasons that a reduced B-spline curve function is presented to the user is because it is not necessary to use the full capabilities of the representation. Furthermore, the reduced capabilities allow the implementations of the constraint tools to be kept as basic as possible.

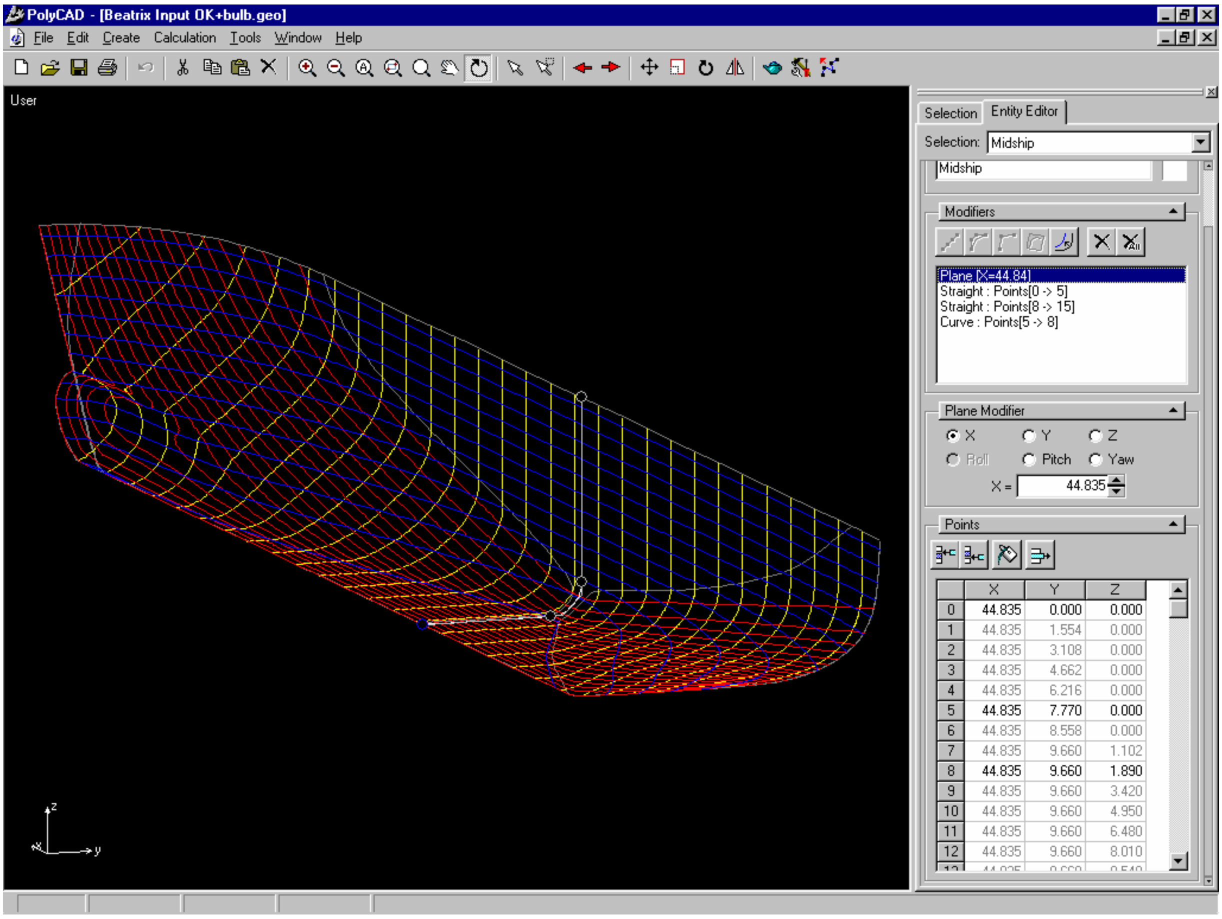


Figure 17.10, the user interface provided for the TIntelliCurve class. Circles on the main screen show the user manipulatable points, other definition points constrained by the modifier tools are disable and show in grey.

The five constraint tools discussed in Chapter 15 have been implemented. The constraint tools are applied sequentially based on the size of constraint applied to the curve. The approach is very similar to the 3D Studio MAX Modifier stack, used for applying different effects and transformations to geometry. In 3D Studio MAX, modifier tools are applied to the geometry in the order of creation. However, as the constraint tools for TSCAHDE have an order of precedence, the technique cannot be considered a stack approach. Figure 17.10 shows the list of modifiers being applied to a midship curve definition, the plane constraint is applied first as this has a global effect on the shape of the curve and localised constraint modifiers are applied subsequently. Furthermore, to illustrate the effect of the modifiers on the definition to the user, constrained vertices are prevented from being manipulated by hiding the edit handles in the main view and showing the vertices "greyed-out" in the point table. This is an example of context sensitivity applied to a much more complex structure than is normally seen. Usually, this technique is only used for menu commands and graphical buttons.

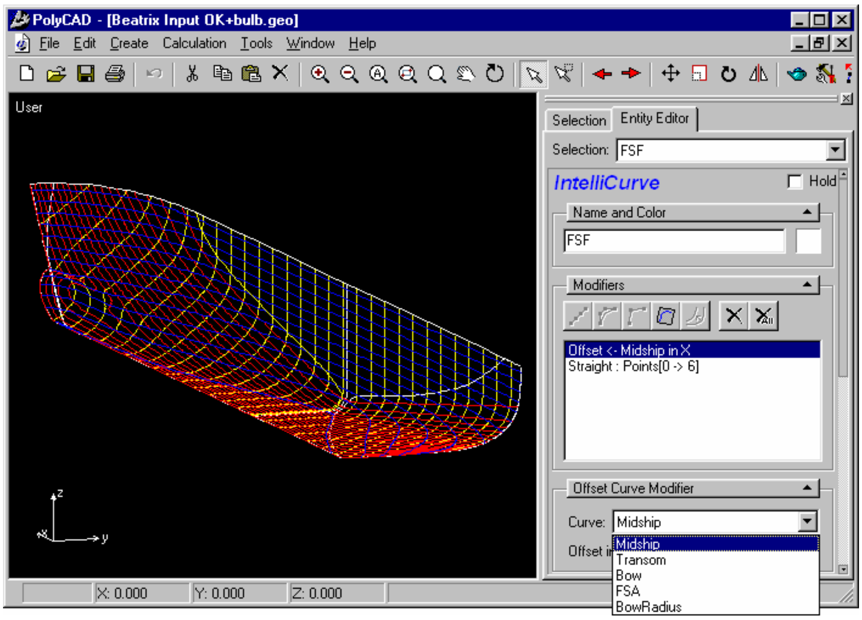


Figure 17.11, the forward Flat definition curve lies on the prismatic section surface defined by the midship curve. The offset modifier maintains this relationship by linking to any curve, except itself.

One of the most important concepts in TSCAHDE is the ability to construct definition geometry based on the shape of other parts in the hull definition. One of the most important areas this can be used is in the definition of the parallel middle body. The offset modifier maintains the curve definition orthogonally to a reference curve. Figure 17.11 shows the definition of the forward flat curves linked to the midship curve. One of the most interesting things about discussions on Relational Geometry is the lack of comment on techniques of implementing the actual relationship. While the actual link process is not of great academic interest, from point of view of program implementation, it is necessary to overcome some challenging problems that occur when data structures have to be robustly linked within a practical software application. Efficient Links between data structures within computer languages can be made using variables called pointers. A pointer stores the memory location of a referenced data structure. This link process is very effective until the link is broken. The most sensitive break in a link happens when the reference data is moved to somewhere else and the pointer is not updated. When the referencing pointer is accessed to locate the data, the program will crash. Careful programming will prevent these situations occurring by minimising the amount of data relocation or by using alternative approaches. However, there are specific circumstances where the pointer link cannot be maintained. This occurs when data is being copied and when data is being loaded from file. These problems were resolved by creating a hybrid referencing pointer, a structure that not only holds the memory pointer, but also the text name of the referenced data. If the pointer memory link ever becomes broken, the structure is capable of searching for the data by name and re-establishing the link. When loading definition from file, the names of reference curves are loaded into the text part of the hybrid pointer. When the data is first required, the structure searches for the reference object by name and then creates the memory link. The structure updates the text name of the referenced data when ever it is accessed to ensure that the name links are not broken.

With a robust mechanism linking definition geometry together, a communication process between entities has to be established to ensure that updating occurs whenever there is a change. The links between data structures, in this case the TIntelliCurves, are one directional. Only referencing curves are aware of the relationship. However, any changes to the definition of referenced curves needs to be propagated through to referencing curves, immediately, to establish an impression of fully interactive editing.

PolyCAD implements a message system similar to the approach used in Microsoft Windows. A PolyCAD message consists of a message identifier and an optional entity reference. The message is sent to all entities to inform of changes to the data structure. The entity reference is used when the message relates to a particular entity. When entities are manipulated, two messages are issued. The first message is issued during manipulation. The TSCAHDE implementation uses the first message to inform any referencing curve of geometric changes so that an immediate update can be implemented. Consequently, referencing curves can be made to appear to change interactively as reference curve is manipulated. The second message is issued when the manipulation operation is completed, i.e. when the user releases the mouse button. This message is used by TIntelliHull to update the surface definition after a curve has been edited.

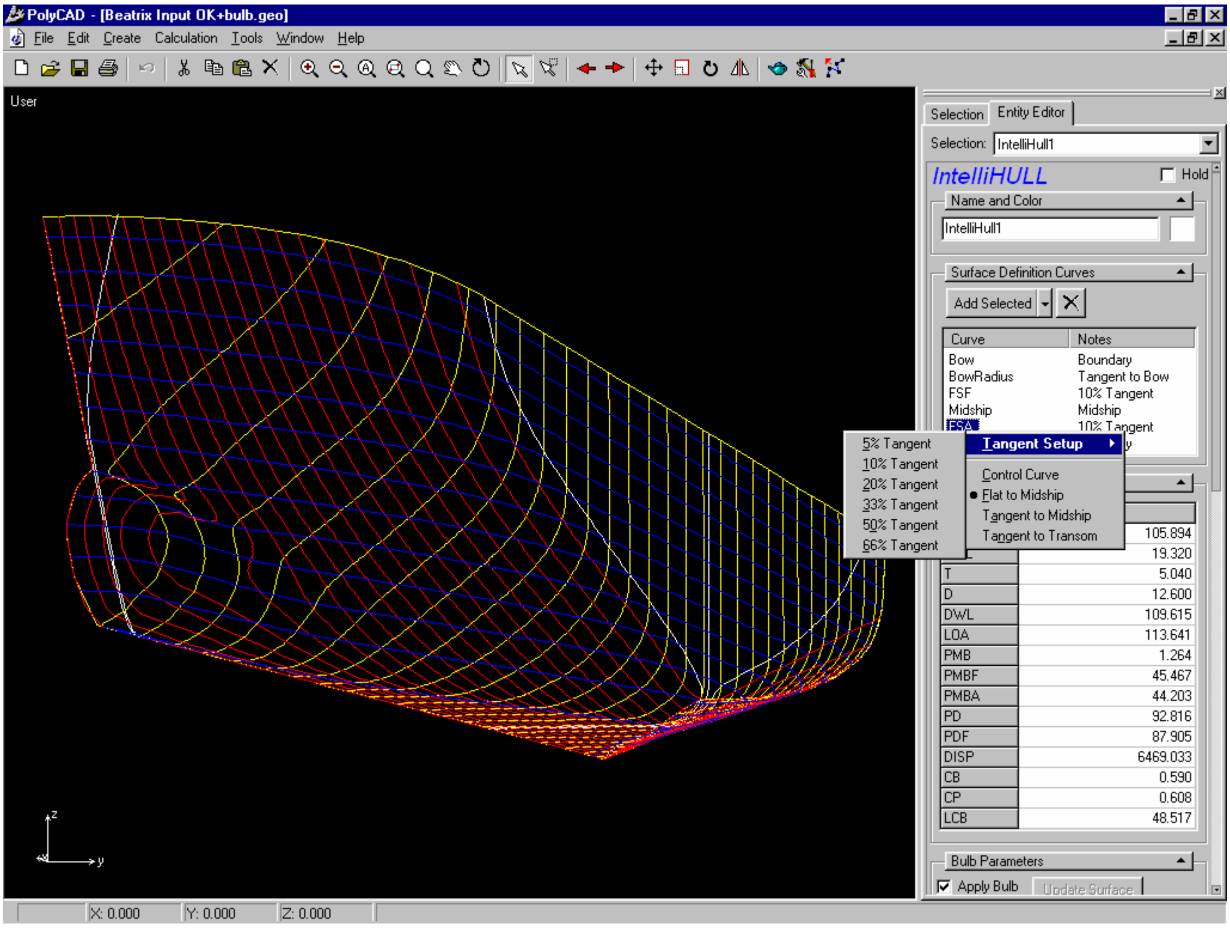


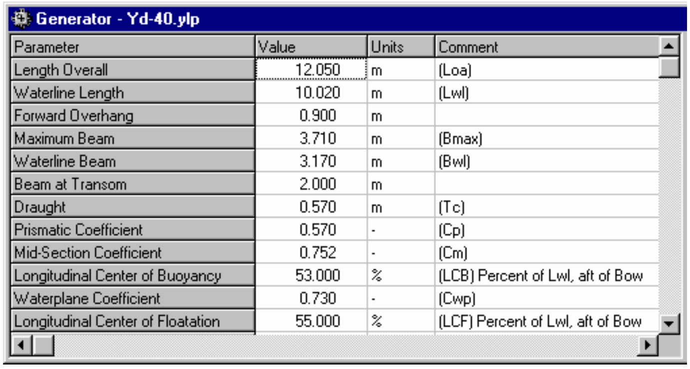
Figure 17.12, the user interface provided for the TIntelliHull class. The user has the ability to change the assigned task of the curve and adjust the influence of the curve on the hull surface.

The messaging technique does not have the capacity to deal with complex structural changes. One of the major limitations of the single surface implementation of TSCAHDE is that each definition curve must have the same number of definition vertices. The offset modifier constraint also requires this relationship. However, this limitation should not prevent from the user from inserting or deleting points from a definition curve, techniques can be found to update the geometry of the other referencing curves. When inserting a point, the action involves additional information that cannot be passed through the message structure to referencing data, the location of the inserted point. Consequently, after a point insertion, curves must perform a search to identify any referencing curves requiring an update, providing the location of the inserted point directly. Deleting definition points also follows the same procedure. It should be possible to identify all the curves taking part in a hull definition from the TIntelliHull structure to allow the number of point definitions to change. However, in the present implementation, only curves that directly reference each other are modified when points are inserted or deleted.

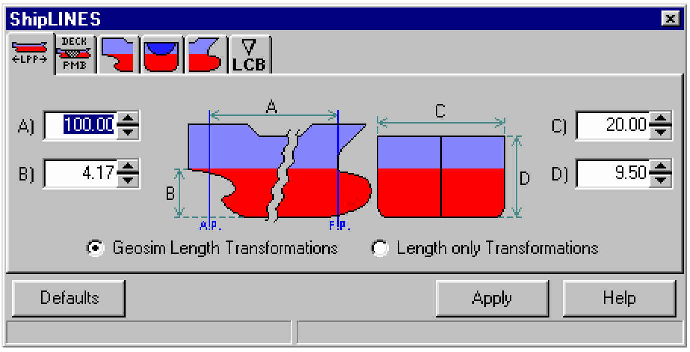
Compared to the definition curves, the user does not have as great an ability to directly manipulate the TIntelliHull wrapper class. Its primary task is to manage the definition curves being used to define the hull form surface and implement access to the numerical parameters provided by the THullBlender class. TIntelliHull also implements a demonstration of the possibilities of Local Surface modification by incorporating an optional parametric bulb surface.

The TIntelliHull surface is created by selecting a group of appropriate curves and then using the option that will appear on the “right-click” popup menu. Once the entity has been created, the user has access to the list of definition curves in the entity frame, (Figure 17.12). The list displays the order in which the curves are used to define the hull, the task that the curves are performing and any related parametric information that affects the shape of the hull form. A popup menu gives the user the ability to change the task that the curves perform and apply some preset values to constraint parameters controlling the shape of the hull surface. Further curves can be brought into the definition by adding to the list.

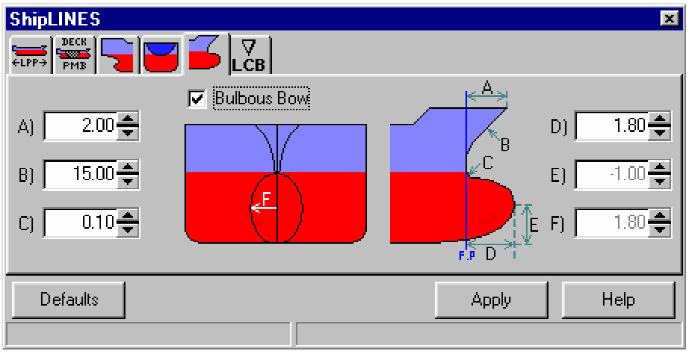
After the development of the parametric hull generators, YachtLINES and ShipLINES, there was some concern over the best way to present the parametric information to the user. YachtLINES, (Figure 17.13a), used a simple approach of list parameters in a table. The user is able to go into the table to modify the value of parameters and can then regenerate the hull surface by pressing a button. This approach is effective, however, as all parameter descriptions are in text, it is difficult to make a good description when referring to unusual dimensions. As parametric hull generation techniques have to completely describe the surface definition with numerical values, the number of unusually named parameters can be quite high. Consequently, new users require a great deal of explanation to be able to use the tool effectively. In the case of ShipLINES, the number of unusually named parameters is greater than parameters with standard names, particularly as many are required to define local shapes within the bulb or the propeller bossing.



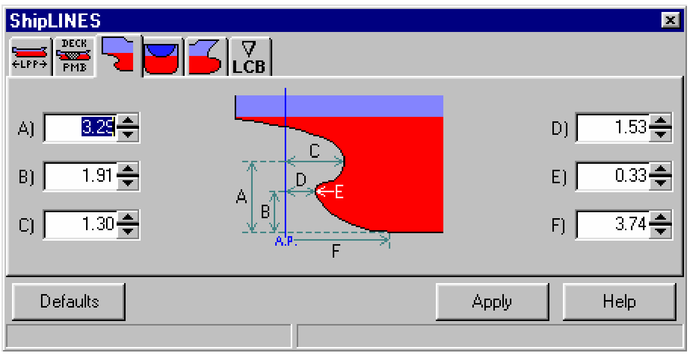
a)



b)



c)



d)

Figure 17.13, Techniques of presenting numerical parameters to the user. a) The table approach taken by YachtLINES. b, c, d) The iconic approach taken by ShipLINES.

A different approach was experimented with for the implementation of ShipLINES. Various diagrams of the hull form incorporating keyed dimensions were used to circumvent the need for parameters to be named. Edit boxes surrounding the diagrams allow the user to modify the parameters indicated by each key. This diagrammatic approach is also used to a limited extent in ShipGEN [29]. Using this technique, parameters can be divided into different categories, enabling parts of definition to be reviewed independently instead of having one large list of parameters. Main dimensions can all be put onto one page, (Figure 17.13b), regions of the hull surface where there are more dense local definition parameter, such as the bulb, (Figure 17.13c), or the propeller bossing, (Figure 17.13d), can be placed on separate pages. However, while this appears to be a very appropriate way of implementing user-friendliness to quite a technical surface definition method, the structure is very difficult to navigate and the diagrams are not easy to understand. Consequently, when PolyCAD was upgraded to accept the implementation of TSCAHDE, the parametric definition interface to ShipLINES was converted to the table approach used by YachtLINES. The table approach makes a better solution because the user can navigate much more easily, other techniques can be used to inform the designer about the definition of each parameter.

The number of parameters defined for TSCAHDE is kept to a minimum, consisting of only the principle parameters. Local definition parameters are not required and are better handled by other surface development techniques, such as manual manipulation or the surface modification techniques detailed in Chapter 16. Consequently, the principle parameters are edited in a table, (Figure 17.14), being identified by standard abbreviations that should be familiar to users working in this field. In contrast, the parameters governing the shape of the bulb surface are more difficult to describe. As the parameter descriptions relate to the geometry of the local bulb surface definition rather than a bulb on a ship, users can access help by clicking on the button, (marked with a ‘?’), next to each parameter value.

Local surface modification is implemented by the TIntelliHull to demonstrate the ability of the system to use the processes laid out in Chapter 16. It should be possible to develop a more generalised approach to local surface modification by allowing the user to apply any local B-spline surface entity to the hull. However, as the local surface modification procedure has yet to prove completely reliability, the facility is not intended as a proper CAD tool and it requires further development. PolyCAD requires the development of further infrastructure before relations between the TIntelliHull surface and general B-spline surface can be made.

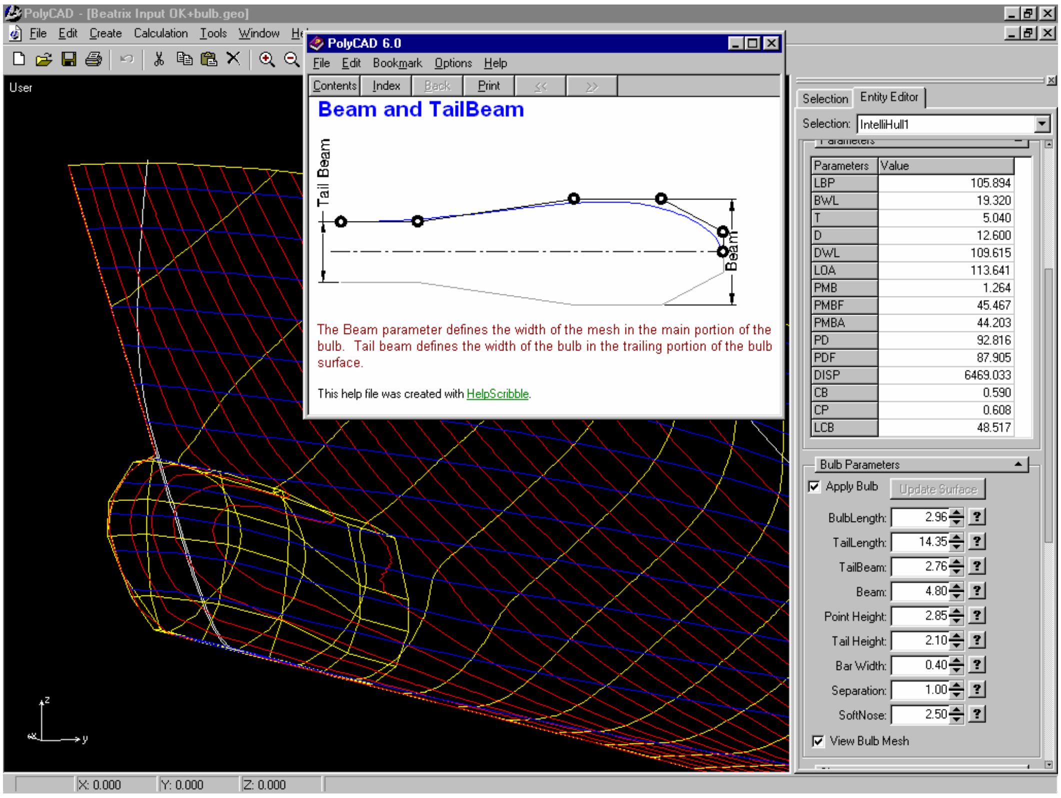


Figure 17.14, the facilities to edit the numerical parameters of the TIntelliHull definition. The principle parameters, with fairly obvious acronym descriptions are grouped in one area. Detailed parameters describing the bulb are grouped separatly with the ablity to display a help window on the definition. Note the local bulb surface control polygon.

While PolyCAD does not provide facilities to apply local surface modification to the hull surface using user defined surfaces, the actual code that builds the surface was constructed with this intent in mind. An approach, similar to the concept of the 3D Studio MAX Modifier Stack discussed earlier, is used to format individual modification operations and create an efficient method of updating the geometry of the final surface. The main hull surface is at the top of the stack, with modification surfaces being below. This structure is efficient because it allows surface changes to take place without the need to re-process the whole geometry structure. Stack entries store the result of previous operations. Consequently, only surface modification operations further down the stack need to be updated. With this type of structure in place and an interface that support the operations, modelling with this technique should be quite effective.

17.5. Developing hull surfaces using TSCAHDE

A hull surface developed using TSCAHDE can be constructed in a very short amount of time while giving the user the ability to customise the shape of the hull surface, (Appendix 6). The best approach appears to be to start with the midsection curve. As the number of points on each curve must remain consistent, the mouse can be used to construct an initial curve with the desired number of points, in an approximate shape. The modifier tools can then be used to constrain the curve to an accurate shape definition. The modifiers effectively reduce the number of definition vertices, by removing the users ability to edit vertices that are constrained. Once the modifiers have been applied, the curve can now be accurately located using the small number of vertices available. In the case of a standard, 'U' shaped, midsection curve, only four definition vertices may be required.

Rather than enter additional curves using the same method, it can be quicker to develop the remaining definition curves by copying and then modifying the definition of the midsection curve. The Flat definition curves can be created by copying the midsection curve, removing all modifiers and applying the offset modifier. The curves can now be manipulated into the desired shape in the knowledge that all vertices lie on the prismatic section of the hull surface.

Curves for the Bow and the Transom can be developed by, again, copying the midsection curve. In the case of the Bow definition curve, the Plane constraint modifier can be used to rotate the curve from a transverse plane on to the centre plane. The Bow tangent definition curve can then be developed by copying from the Bow definition curve and applying the Offset constraint modifier. With all the curves in place, the hull surface is generated by selecting all the curves and using the appropriate command on the "right-click" popup menu.

If full use of the constraint tools has been made during the development of the hull definition curves, the system should be able to detect the task of each curve by the accuracy of the geometric relations between curves. Once the surface has been created, adjustments can be made using the parametric transformations or the strength of the surface tangents applied to the Flat definition curves. Fine adjustments can be made by directly manipulating the definition curves.

The implementation is basic enough to demonstrate that the concept can be realised. During the development, more applications of this approach to hull surface design were indicated and it is possible to develop further tools to aid the development of a surface. One possibility is to develop the means to create the hull surface using only numerical parameters. This may not be such a useful feature for the design process. However, during development there was a request for the ability to generate a hull surface from the basic dimensions of a ship, allowing some basic representative hydrostatic calculations to be made. This could be achieved quite effectively using a "wizard" approach where the user selects predefined curve templates. These curves could be parametrically defined to allow the system to correctly scale each definition curve to the size of the ship and allow the user to make some basic changes without resorting to manual manipulation of vertices.

The TSCAHDE concept appears to be very efficient. The small amount of definition data and low number of parameters, identified by abbreviations, create a very concise editing tool. When it is compared to the commercial hull design tools, the technique must present a very pragmatic approach to the development of a hull form surface.

18. EVALUATION

18.1. Evaluating the Concept

It is difficult to make a detailed evaluation when the subject of the work is a conceptual approach or idea. Any detailed analysis of the material produced by this study would tend to concentrate on the functionality and the results produced by the implemented pilot system. However, as this has been developed as a limited demonstrator rather than an implementation that provides an optimal solution using the concept, there are many detailed functions required for the hull design process that it does not address. Any analysis would have to overcome the limitations of the implementation before the conceptual approach could be reviewed effectively. Consequently, the evaluation has to be more subjective. However, it is possible to make a comparative evaluation of the concept by considering the issues raised by the initial review.

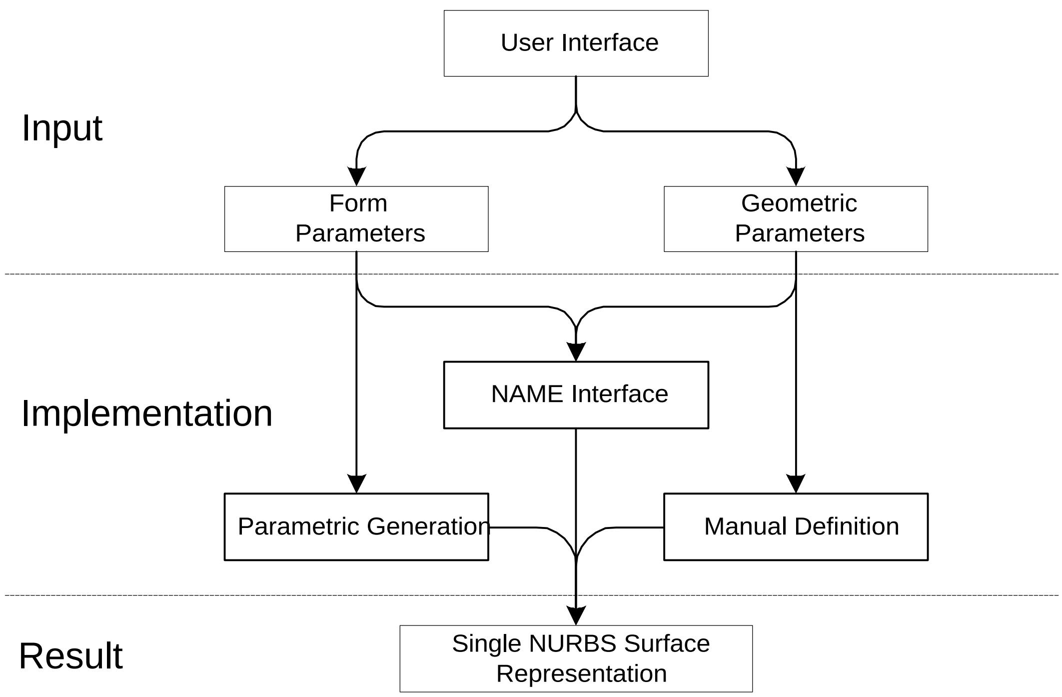


Figure 18.1, PolyCAD incorporates the TSCAHDE, manual and parametric hull development techniques within the same environment. Consequently, it is possible to make an evaluation which minimises consideration for the implementation and user interface functionality.

Furthermore, the pilot system is implemented in a software environment, PolyCAD, which also supports direct manual manipulation of the hull surface representation and the ShipLINES parametric hull form generation technique, (Figure 18.1). Consequently, as all three techniques use a common graphical user interface and produce a B-Spline surface hull representation utilising the same code, it is possible to make a unique evaluation of the new approach without the need to consider any particular implementation or user interface issues. The evaluation can proceed on the basis of considering the input data and the functionality of the design process with respect to an impression that the same hull form representation is produced.

One of the primary objectives of this study was to address inefficiencies in the present hull design procedures that result from inadequate computer-aided hull design tools. Consequently, the ultimate goal of the project has been to make hull form design easier. The review of present hull design tools identified that the major approaches have limitations which prevent the hull design procedure progressing as a development process. Notably, the manual process requires too much detailed manipulation resulting in slow progress through the concept and initial phases of design. Parametric hull generation tools fix the representation to a mathematical development process which limits flexibility by reducing the variety of shapes that can be produced. The user is forced to use the most detailed surface design tools, if there are any requirements to modify the details of the generated hull surface further. On this basis, a set of criteria can be selected with which present processes of hull design can be evaluated with respect to the solution proposed by this study. The criteria can be selected on the basis of the major issues that will affect the design process of a hull surface using a modern software tool at the point when the designer instigates the development of a vessel. Quite obviously, the basic factors affecting the hull surface representation design process are as follows:

1. Quickness: The rate of development of the initial representation and further modifications.
2. Easiness: The level of complexity of the definition with respect to the level of detail required.
3. Flexibility: The number of ways a user can change a surface representation.
4. Dexterity: The range and variety of hull surface shapes that can be accommodated.

The quality of the resulting representation could be considered as a further issue. However, while the quality, or more specifically the level of fairness, of the hull form surfaces produced by each approach is a very important factor, it is more a function of the implementation of each technique rather than the conceptual approach.

18.2. Quickness: Evaluating the rate at which it takes to develop a hull representation.

The practical development of a hull form has traditionally been a lengthy process requiring a great deal of skill. However, as the ship design discipline becomes dominated by the scientific approach, the time it takes to achieve the final hull surface is very important. Modification of existing hull form designs has been the most effective way of producing a new ship. However, as analysis tools are being increasingly used, the design process is shifted to one that invests more in the initial development with a view to producing a more effective vessel and lower life-cycle costs. Consequently, optimisation is featuring more in the design process. It is possible to optimise a hull form in many ways as its design affects so many characteristics of the vessel as a whole. Therefore, the future of hull design lies in tools that will allow the designer to enter an optimisation process as soon as possible.

The amount of time it takes to develop an initial hull form is dependent on the amount of information that must be supplied and the rate at which it can be used to construct the hull form surface. The amount of information that is required to form a hull form can be easily measured. All computer-aided design tools can be considered parametric because all processing is performed on the basis of numbers. The definition control vertices of a surface each have three coordinate numbers which locate the point in space. Hence, for each control vertex there are three parameters. From the point of view of the raw flexibility of the surface representation, this could also be considered as the number of degrees of freedom.

For the purposes of evaluating the quantity of definition information required to form a hull surface, an example of forming a basic hull surface using TSCAHDE will be used. This surface will use sixteen control points on a section. The surface will be generated with a total of thirteen control polygon columns, a total of 208 control vertices. In developing the surface manually, the user will have to modify each of these control vertices separately, considering there are three parameters per vertex. The TSCAHDE approach constrains and automatically generates some of the control vertices, greatly reducing the number of independent parameters. The number of parameters used by ShipLINES to develop the hull form is independent of the surface definition. For the purposes of this example, parameters controlling the propeller bossing will be ignored because the TSCAHDE implementation does not support this feature. A summary of the quantity of the definition information for each technique can be given as follows:

1. Manual Manipulation of Control Vertices:

16 Rows × 13 Columns = 208 Control Vertices = 624 independent parameters

Number of Independent Parameters = 624

2. Parametric Generation of Hull Form using ShipLINES as a basis (Appendix 3):

   Parameters: BowShape, LBP, Beam, Depth, Draught, ParallelDeckF, ParallelDeckA, FOSBow, FOSAft, BowRadiusAtDeck, BilgeRadius, TransomBehindAP, TransomImmersion, TransomBeam, TransomCurvature, BowCurvature, BulbRadius, ForeOverHang, BulbLength, ForeFootRadius, StemRadius, FlatOfBottomStart.

   Number of Independent Parameters = 22

3. Practical Development of a TSCAHDE Hull Surface using the IntelliHULL implementation

   This example will consider the development of a practical hull surface using the IntelliHULL implementation. The use of vertex level modifiers will be limited to forming shapes that are the most frequent such as the rectangular shape of the midship section. Consequently, all vertices on the Stem, Bow Tangent and Transom have two degrees of freedom.

   Considering each definition curve:

   a. Midship Section Curve

      Modifiers: Plane (2 Parameters), 2 × Straight, 1 × Curve. (5 Parameters)

      Control Points: 4 Vertices modifiable in 2 directions (8 Parameters)

      Total: 13 Parameters

   b. Forward and Aft Flat Curves

      Modifiers: Offset (2 Parameters)

      Control Points: 16 Vertices modifiable in 1 direction (16 Parameters)

      Total: 2 × 18 Parameters

   c. Stem Curve

Modifiers: Plane (2 Parameters)

Control Points: 16 Vertices modifiable in 2 directions (32 Parameters)

Total: 34 Parameters

d. Bow Tangent Curve

Modifiers: Offset (2 Parameters)

Control Points: 16 Vertices modifiable in 1 direction (16 Parameters)

Total: 18 Parameters

e. Transom Curve

Modifiers: Plane (2 Parameters)

Control Points: 16 Vertices modifiable in 2 directions 32 Parameters)

Total: 34 Parameters

Surface Constraints:

a. Forward and Aft Flat Tangent Constraints (2 Parameters).

b. Displacement and LBP Hydrostatic Constraints (2 Parameters)

Local Surface Modification

c. Bulb Parameters: (9 Parameters)

Number of Independent Parameters = 148

The number and type of parameters to control each technique is summarised Table 18.1:

Technique	Definition Parameters (Geometry)	Control Parameters (Form, Constraint etc.)	Total
Manual Manipulation	624	0	624
ShipLINES (Parametric)	0	22	22
IntelliHULL (TSCAHDE)	120	28	148

Table 18.1, the number of individual parameters required by each technique to define the same surface representation.

Based on these results when compared with the TSCAHDE technique, the example hull form definition requires 4.2 times more definition when using the manual approach and 14% less definition when using the parametric approach.

While this analysis gives an idea of how much information is required to develop the hull form, it is rather a static view of the process when considering the rapidity that a hull form can be developed. The main reason is that the value of the control parameters is generally known, whereas the positions of the control vertices is generally unknown and must be determined by manipulating the points until a certain level of quality is reached. Consequently, the number of times a control vertex is manipulated becomes a factor in comparison. The number of times that vertices are modified can be included by considering the way that vertices must be manipulated in the fairing process and the local relationships between vertices in the representation. Using the NURBS representations from the example, in changing a vertex of a curve, 'n' surrounding vertices may have to be manipulated to achieve a correct and fair shape. If this vertex was part of a surface definition, 'n^2' vertices would have to be manipulated because of the relationship between the rows and the columns of the definition that produces the shape. 'n' becomes a conceptual measure of dependency between vertices defining the hull form definition. Consequently, the rapidity of the development of the hull form can be estimated using the model shown in Table 18.2. The model is illustrated graphically in Figure 18.2.

Technique	Total Number of Manipulations
Manual Manipulation (Surface)	$624n^2$
ShipLINES (No geometry parameters)	22
IntelliHULL (Curves)	$120n + 28$

Table 18.2, an estimate model of the total number of manipulation required to develop a hull form based on a measure of the dependence between geometric parameters 'n'.

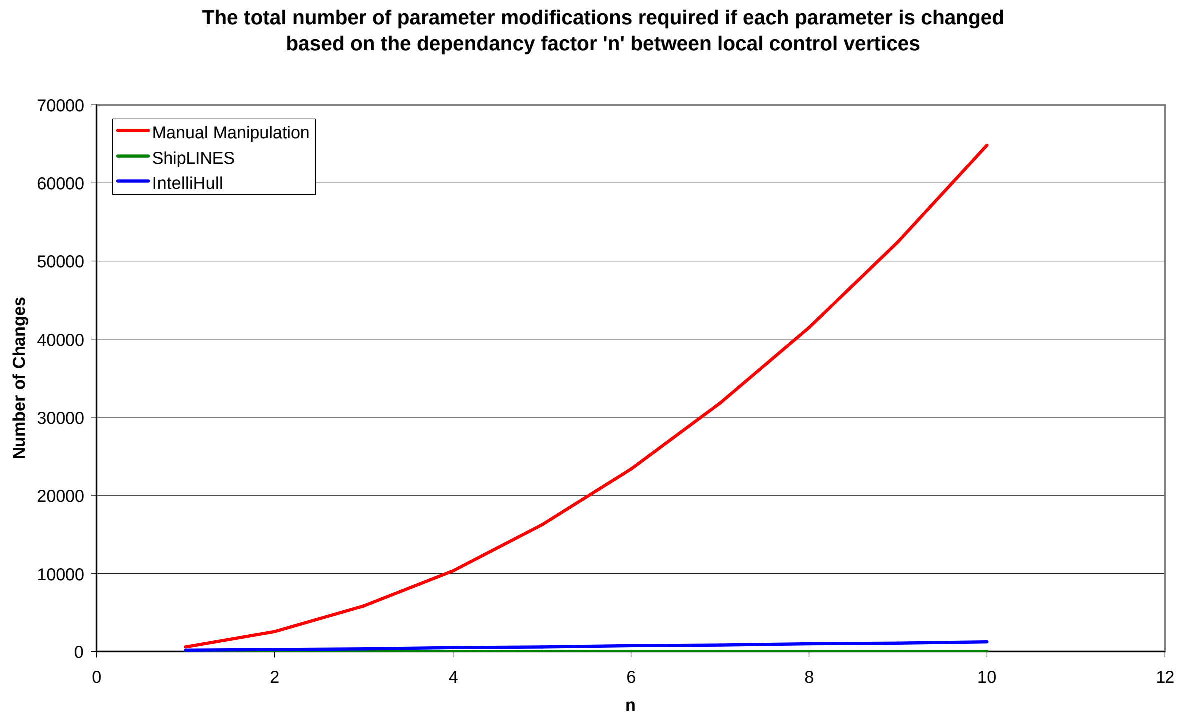


Figure 18.2, a graphical comparison of the level of interactivity required for each approach based on the estimate model developed in Table 18.2

As parametric hull generation tools develop the hull surface representation from independent parameters, each need only be set once. However, as there is a great deal of dependency between the vertices of a NURBS surface definition, the number of manipulations is very large. This comparison illustrates very directly why it takes so long to develop a hull surface using the surface definition directly. When representing a hull form, it is sufficient to use cubic degree NURBS. With an order of this magnitude, each point on a curve is controlled by four vertices. Consequently, a practical value for 'n' could be from four to six, giving a total number of changes of changes of around 10000 to 20000 for the manually manipulated surface definition. However, in practice, the number of changes may be much less accounting for the experience of the user in forming particular shapes in the surface. The number of modification required when using the TSCAHDE approach is much less than manual manipulation and it increases linearly. With an 'n' of four, the number of modifications is approximately 500, resulting in twenty times less work than the manual manipulation approach.

Based on experience gained with manual development and how long it takes to develop a hull form using the IntelliHULL implementation, this comparison is of the correct order of magnitude. It would be expected for the manual development of a hull form to take one or two days depending on the complexity of the surface in comparison with fifteen to thirty minutes being used to develop a hull form using TSCAHDE. While the TSCAHDE approach is much quicker than direct manual definition, parametric hull generation is still the quickest by far. However, as discussed before, relying entirely on numerical parameters severely limits the flexibility of the forms that can be generated.

In evaluating the speed with which the TSCAHDE approach can develop a hull surface representation, it has been proven on the basis of the number of dependent and independent parameters that the technique is much faster than the manual approach, although still slower than a parametric hull generation technique. It can also be proved that the tool is quick in practice. Appendix 6 shows an example of a hull form being developed using a series of twenty-four images. The images were developed over a period of twenty minutes and show the key characteristic curves being developed in the definition of a hull surface. It would not be practical to perform this exercise at the same level of detail for a manually defined hull surface.

18.3. Evaluating Easiness: Can you use available information to define the hull form?

Evaluation of the ease of use of software technology requires a very subjective approach. Traditionally, how easy a piece of software is to use is determined mainly by the implementation and the graphical user interface. As design problems are becoming more complex, the burden of defining these problems is being increasingly left with users. Developers are not always prepared to provide solutions to aid the modelling process if the development of such a tool requires a large investment. Consequently, software tools may be constrained from providing an interface which covers all the tasks that the user wants to perform efficiently. Hull surface design tools are a very good example of this effect. Systems using parametric surface representations have gone no further than allowing the user to manipulate the control vertices of the definition. Parametric hull generation tools have never been found to be very effective for practical hull design as they can only address a limited part of the design process. Consequently, very few have been able to break out from their academic background and there is little investment available for improvements.

Modern hull surface design tools cannot be considered to be complicated to use, the user only has to manipulate control vertices. However, tools that require a lot of simple data are usually more difficult to use than those based around a single complex problem. The main reason is that it is very easy for a user to get bored manipulating lots of simple data. A system requiring a small amount of complex data is likely to retain a user's interest. While tools requiring large amounts of simple data may be uninteresting to use, there is frequently a user who can utilise the systems very quickly and efficiently. These are the users who have developed the skills and expertise to optimise tasks through the development of procedures. Procedures for developing hull forms exist many of which were shown in the review chapters. As these procedures are repetitive, often based on simple actions, there is no reason why they could not be incorporated as part of the software tools implementation. This study has used this approach by identifying problems and similarities between existing hull surface definition techniques and by using the experience gained with these tools, a concept for an interface between the user and the simple definition structures has been developed to make the design process easier and more effective. While the first step is to reduce the amount of definition information, structures and procedures have to be included that ensure the designer can develop the design in ways possible in existing tools. Consequently, an easy to use design tool is one which has the capabilities to produce the surface hull form directly from information that the designer has available at any point in the design process. A tool should give the designer the ability to develop more detailed features as information is generated by the design development process, without requiring the user to provide definition for these features before design information is available.

Considering the present state of hull form design tools, this approach appears far beyond the capabilities of software developers. However, this study has shown that it is possible to develop such a tool by abstracting the user away from the definition of the surface representation by providing an interface between the two. At a basic level, all this does is create a “space” where the processing of data can occur as it passes from the user to the surface definition. How this “space” is used is up to the developer. In this study, the hull surface process has been reviewed to identify structures, processes and procedures that are used by the designer in constructing the surface and in improving the definition process. This experience is combined with the technical knowledge of the surface representation to create a conceptual approach for an interface that can form shape without the user having to directly construct the definition. Furthermore, as the hull design process changes throughout the development cycle, there is no reason why an interface cannot be developed that adapts to the user’s definition requirements, so that it deals with the process of developing the surface definition instead of passing the problems on to the user.

With an interface between the user’s hull form definition and the surface representation definition, developers have an infinite number of methods of processing the user’s data. Parametric hull generation is one solution that uses this approach, by converting numerical form parameters into a hull surface representation. However, parametric hull generation tools are not greatly popular. Numerical form parameters alone cannot be used to develop a flexible hull design tool because there are so many features that cannot be defined in an easily understandable numeric way. Directly interacting with the shape of the hull surface is a very important part of the hull design process. However, the manual construction of the entire surface definition is too great a task to burden the user with, particularly in the initial stages of design. An easy to use design tool will allow the user to control the hull form numerically and interactively in a balanced way that matches the present state of the design process.

A considerable amount of definition information is required to create a basic hull surface. However, in the early stages of design, the user can only provide a very small amount of accurate data. Form topology information and geometric constraints can be used to develop a hull surface. Form topology can be seen as a non-numerical parameter that represents the shape structure of a surface representing a hull form. A small amount of data can be combined with the structure using geometric constraints to control the surface definition. In progressing to more detailed definition, the geometric constraint of the surface definition can be removed and replaced with definition directly provided by the user. On the basis of this analysis, the key issue to making hull design an easier process, given that the definition medium consist of simple geometric points, is in the *clarity* that the hull form definition has with respect to the information that the designer is presently creating. Manual manipulation cannot provide clarity in definition because it cannot adapt the definition to what the user wants to design. Consequently, in forming a hull surface representation for deciding the basic dimensions of the hull form, the bulb must be defined because it contributes to the displacement. A design system that understands that a bulb should be present can automatically add the required definition, perhaps on the basis of some simple parameters to control the size, leaving the user free to control the main dimensions of the hull form.

A hierarchy in the definition process is critical to allowing the design process to be accomplished in simple stages without having to provide the full definition. The hierarchical structure allows for definition to be stored and controlled at different levels of the hull form shape structure. Therefore, unlike a single homogeneous definition structure, procedures for controlling shapes of different sizes and for introducing local features can be accommodated independently of each other. The hull design task is made easier because the designer no longer has to consider how changes to the definition of one feature may affect the definition of another.

The majority of the functions that can be used to control the form topology, constraints and the hierarchy are fairly simple in implementation. In trying to reduce the quantity of definition, simple constraints can be optionally applied to make segments of curves straight or curved. In placing these constraints, the user knows that the affected definition will remain that shape without requiring any further definition. Furthermore, if the implementation hides the constrained control vertices, the definition appears simpler and therefore easier to control and manipulate. Simple constraints can also be used to make definition curves easier to control. Many existing tools employing definition curves allow the use of two dimensional representations. However, when using optional constraints, the user can select how flexible the curve should be. The present pilot system allows curves to be constrained to arbitrary plane reducing the definition to two dimensions. In the case of parallel middle body definition, constraints can be applied to link to the shape of another curve, such as the midship section. Consequently, under this constraint, control vertices are only modifiable in one dimension. Moreover, in making the system easier to use, constraints can be removed at any time by just deleting the relevant definition. Therefore, the user does not have to make any firm decisions on using a particular form of curve representation before definition begins.

In summarising an evaluation on the easiness of use of a hull design tool, the development of a system that provides clarity in the definition medium appears to be the most critical issue in the development of future hull design systems. Present hull design systems involve the user in the development of a surface representation and do not aid the design of the hull form itself. Presently, this task must be carried out by the user. If the amount of definition data can be changed to fit the information it is obvious that the task of designing the hull form will be easier, excepting effects due to the user interface. The pilot system demonstrates that it is possible to create a hull form by combining different amounts of data and, as a result, the process of developing the initial hull surface and subsequent modifications are significantly easier. Appendix 6 illustrates a hull form surface being developed with the tool. The images show that the user only needs to define the key characteristic features, leaving the development of the rest of the surface definition to the software.

18.4. Evaluating Flexibility: Does the concept provide for a variety of ways to control the design?

Flexibility has a variety of meanings when it is discussed with respect to hull form design. Usually, it would refer to the ability of a particular surface technique to represent a range of hull forms. However, as the concept refers to an approach for developing a hull design tool, flexibility in this case shall refer to the number of different ways a user can control the surface representation.

Hull design tools based around manually manipulated surface definition have a very limited degree of flexibility despite the fact that it is possible to represent a very wide range of shapes with these tools. These tools have limited flexibility for design purposes because, as it is so easy to just provide the control vertices to manipulate, it is not necessary to develop any further functionality. Consequently, the user must control all the dimensions, features and shapes within the hull surface with the same tool function, regardless of the structure of the characteristics. The user can only influence the hull form representation at the microscopic scale of definition vertices.

The flexibility of parametric hull form generation tools is a bit more difficult to quantify. If the parametric tool has the capability to exactly represent the hull form that the user desires, then it is likely that the tool provides exactly the right number of parameters required by the user to control the surface. However, as parametric tools are rarely capable of producing exactly the right shape of hull surface for a particular design it is difficult to consider a parametric hull generation tool flexible. Moreover, if a tool was able to produce exactly the right shape of hull form, the tool would not be able to significantly modify the hull form to develop another design in the future. The mathematical procedures used within these systems are hard coded and hence have very real limitations. Attempts to add further parameters may be difficult depending on the completeness of the surface generation solution. A solution that develops a surface using specific parametric information may not be able to accommodate any further parameters.

This study addresses flexibility by considering an approach that allows for a greater variety of techniques for controlling the surface to be incorporated into the design tool by the developer. Consequently, existing techniques can be improved by considering how to remove any limitations and new techniques can be introduced. These new functions can address many of the limitations of existing hull surface development tools driven by the separate parametric and manual approaches. The form topology structure is a critical feature in this concept. It represents a plan of the shape characteristics of the hull surface and can be used to define logical ways of linking parameters to geometric definition and reduce the amount of user controlled definition by implementing optional geometric constraint relationships. Consequently, any tools implementing this approach can provide an extensive range of different controls over the surface by utilising the form topology structure alone.

The pilot system has many examples of how a basic system can introduce these surface development aids. For example, the technique has parameters that will control the magnitude of surface tangent at each end of the parallel middle body. It can control and develop the surface shape in the entrance and the run of the vessel by considering the shapes at the bow, transom and over the parallel middle body with the desired hydrostatic properties. Furthermore, parametric modification techniques have been introduced to change the shape of the hull form by implementing compound transformations. These functions selectively transform parts of the definition reducing the amount of undesirable deformation that occurs when using present transformation techniques which affect the definition across the whole surface. Illustrations of modifications implemented on a hull surface defined within the pilot system are shown in Appendix 7.

The use of parameters within a hull definition could be extended by defining individual components with parameters rather than beginning a design with geometric definition. Key curves could be defined using a template approach and an initial hull form could be created by selecting curve components from a repository. This is an advantage over the traditional technique because it does not constrain the hull form to a particular characteristic shape even though it can be completely defined parametrically. Furthermore, in developing the design's features, each component curve can be converted to geometric definition with or without constraints applied.

While the approach allows for many new design functions to be implemented, the hierarchical structure implements a further design advantage that cannot be matched by present techniques. Because the hierarchical approach develops the hull form by separating global and local definition details, it incorporates a very high level of implicit "undo-ability". It is very easy for the designer to reset the definition to a basic state after unsuccessful interactive modification. This process can immediately return the hull form to a faired state allowing the designer to try another approach. Present design tools have to implement explicit undo functions to handle these cases, by implementing a structure that will stack up previous modifications. Consequently, explicit 'undo' functions are ordered and do not allow the designer to selectively undo changes. Therefore, it can be necessary to reset changes to other parts of the surface definition to return to a point where the design can continue.

18.5. Evaluating Dexterity: Is the tool capable of developing a wide range of hull forms?

The issue of reduced design capabilities can be a major problem for any design tool that uses computer processing to develop a hull surface. This issue could be considered the primary limiting feature for parametric hull surface generation techniques because the range of hull forms that can be produced by these tools is limited by the number of parameters and the generation technique itself. As both these features are hard coded, the tool is constrained by its implementation. Some systems have attempted to provide the user with the ability to develop custom generation procedures. However, these are rarely flexible and require the user to have a reasonable knowledge of the techniques involved in parametric hull development. Improvements to these tools could be made by providing some of the basic generation components to assist the user in building a parametric system. This approach formed the initial basis for the concept developed by this study. However, during development, it was found that the tool could be more effective if manual interaction was used in combination with parametric information to control the definition and generation tools.

There have been many attempts to produce better parametric hull generation techniques. However, the basic capabilities of the manually defined hull surface are far greater than parametric techniques. Furthermore, it should be noted that from the perspective of the developer, manual techniques require significantly less time and skill to develop. However, while these techniques have a great capacity to develop a whole range of shapes, it can take time to produce a surface, involving the user in a lengthy mundane task. The reason it takes so long to develop a hull surface is due to the amount of flexibility in the representation techniques. Hull forms all have very similar characteristics and it should not be necessary for the user to spend a great deal of time developing the surface into a basic form shape. The concept developed by this study addresses this problem by developing an approach that allows design tools to constrain the flexibility of the surface. However, it is not possible to go ahead and develop a system that will always enforce a hull shape on a surface. Such a system would operate very much like a parametric hull generation tool and would not provide the designer with the ability to innovate or design detailed features. Therefore, a key strategy of the approach is to ensure that the designer has the freedom to manually modify the definition directly if so wished. Consequently, the flexibility of the tool is only limited by capabilities of the surface representation and not the processing functions. From this point forward, the concept allows for constraints to be applied to the definition to construct the familiar form characteristics of a hull form surface.

As particular types of hull forms have common characteristics throughout, it should not be necessary for the user to resort to direct manipulation to produce a hull form. By providing basic constraints which can be combined together like building blocks, the tool provides flexibility to produce a range of hull form shapes without being limited to one main processing function like parametric hull generation techniques. Furthermore, if the relationships between the characteristic shape of features of the hull form and the surface definition can be found, constraints can be used to control the shape across large areas of the definition. In the case of the pilot system, constraints are applied to shape the entrance and the run of the hull form. These sections of the hull form are controlled by the shape of the curves at each end and by utilising parametric information to control the hydrostatic properties of the hull form. As features such as the bulbous bow are defined elsewhere in the hierarchy, something as simple as family of curves can be used to attain the proper hull surface shape.

Creating representations of existing hull forms is a procedure to which NURBS surfaces are not particularly suited because the definition control points do not lie on the surface. However, by using form topology, geometric constraints and definition curves, the process can be made significantly easier. The following two examples show that is possible to develop very close matches to existing hull forms by using the same basic geometry structure used to design an initial hull form. Furthermore, parametric hull generation techniques have often been accused of always producing hull forms with the same characteristic shapes. However, as the two hulls used in the examples are very different, it shows that even the pilot system, which implementes the concept to only to a basic level, is capable of producing a wide range of hull forms without any significant limitations or similarities in characteristic shape. Furthermore, it shows that the approach developed by this study could be a great aid when a NURBS representation needs to be developed for an existing surface.

18.5.1. Comparison to a Ro-Ro vessel hull form

The rectangular nature of the single NURBS surface patch makes the development of Ro-Ro hull forms with pram type sterns very easy as there are no particular areas of the hull that require the surface to become significantly deformed away from its uniformly rectangular definition structure. A ferry hull form was chosen from one of the databases at the Ship Stability Research Centre. The vessel has what would be considered a very standard Ro-Ro shape. There is a considerable amount of planar surface shape and there are two cut-outs in the stern surface shape to allow for propeller flow. The vessel has the following particulars:

- LBP = 132.7m
- BWL = 28.0m
- Depth = 19m
- Bilge Radius = 2.5
- $X_{MIDSECTION}$ = 53.4m
- No parallel middle body, PMB = 0
- The forward extent of the parallel deck starts at 116.5m and runs all the way to the transom.
- The vessel has an operational draught of 6.3m resulting in a displacement of 15380 Tonnes and a LCB at 61.547m

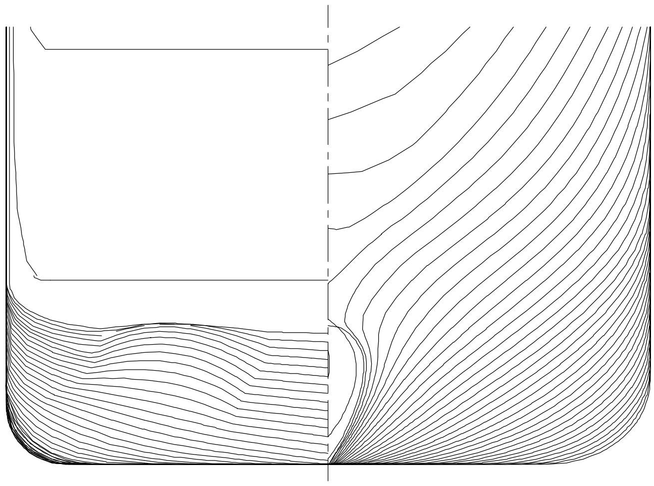


Figure 18.3, the hull form sections of the conventional Ro-Ro ferry hull form.
Sections are spaced at intervals of 2 metres.

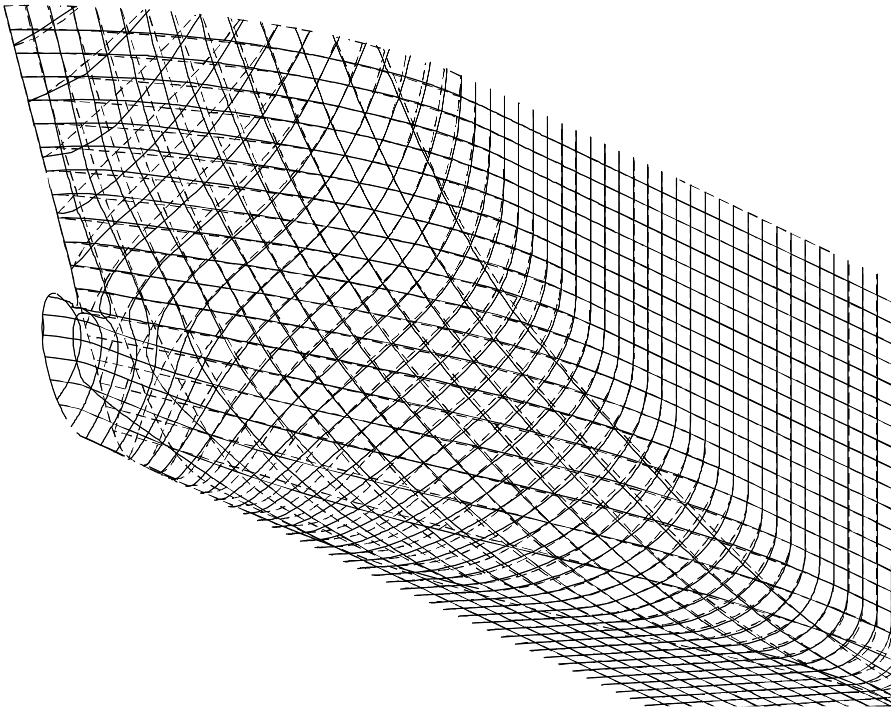


Figure 18.4, the entrance of the compared Ro-Ro ferry hull forms. Ignoring the bulb, the only difference major difference in shape is in a local region close to the stem. (NAPA definition – Solid Lines, IntelliHull definition – Dashed Lines)

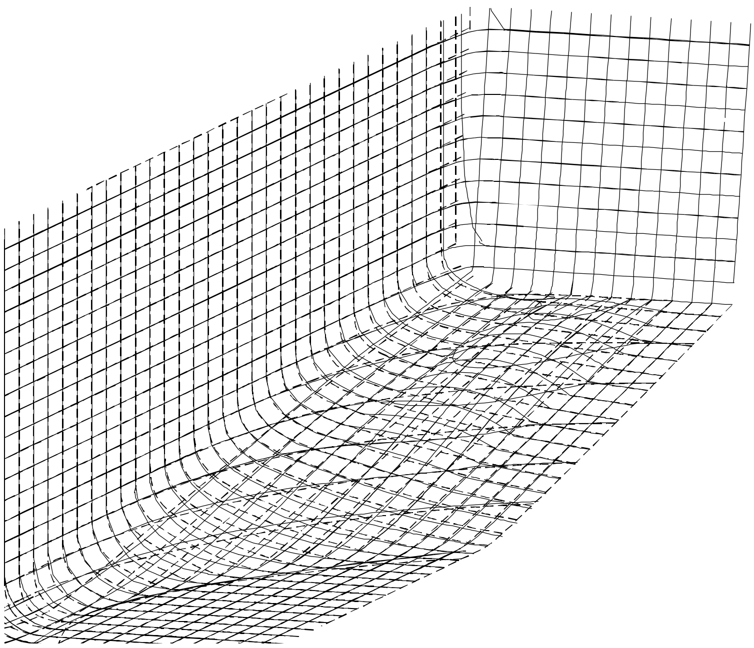


Figure 18.5, the run of the compared Ro-Ro ferry hull forms. The match is again very close, particularly considering that there is very limited control of the shape of the surface on the flat of bottom curve. (NAPA definition – Solid Lines, IntelliHull definition – Dashed Lines)

T	DISPM	VOLM	LCB	VCB	KMT	WPA	LCF	TPC	CB	CP
0.50	-13.933	-13.593	-1.444	-0.001	4.856	-38.396	-1.198	-0.393	-0.011	-0.010
1.00	8.193	7.993	-1.947	0.000	1.468	15.618	-1.348	0.160	-0.022	-0.017
1.50	19.285	18.814	-1.536	0.000	0.746	21.532	-0.524	0.221	-0.026	-0.019
2.00	39.906	38.934	-1.290	0.005	0.354	63.433	-0.659	0.651	-0.047	-0.039
2.50	48.226	47.050	-0.732	0.001	0.203	29.732	0.756	0.305	-0.029	-0.021
3.00	62.326	60.806	-0.421	0.004	0.130	56.832	0.609	0.582	-0.048	-0.041
3.50	104.517	101.968	-0.174	0.007	0.022	74.978	0.646	0.769	-0.048	-0.039
4.00	92.216	89.966	0.267	-0.002	-0.181	9.797	1.879	0.100	-0.029	-0.023
4.50	88.353	86.198	0.564	-0.002	0.186	17.545	1.731	0.179	-0.030	-0.024
5.00	78.917	76.993	0.802	-0.006	0.602	65.001	0.623	0.666	-0.033	-0.026
5.50	55.834	54.472	0.993	-0.012	0.106	-1.402	1.396	-0.014	-0.023	-0.018
6.00	39.852	38.881	1.076	-0.018	0.332	46.064	0.029	0.472	-0.025	-0.019
6.30	**42.707**	**41.666**	**1.050**	**-0.018**	**-0.055**	**10.585**	**0.464**	**0.109**	**-0.025**	**-0.020**
6.50	45.519	44.409	1.026	-0.018	-0.044	10.374	0.416	0.106	-0.025	-0.020
7.00	52.907	51.617	0.966	-0.016	-0.015	11.799	0.322	0.121	-0.026	-0.021
7.50	61.920	60.409	0.908	-0.015	0.002	13.478	0.273	0.138	-0.026	-0.022
8.00	69.105	67.420	0.848	-0.014	0.003	11.461	0.216	0.117	-0.018	-0.013
8.50	77.232	75.349	0.802	-0.014	-0.002	11.200	0.217	0.115	-0.018	-0.014
9.00	85.447	83.363	0.763	-0.013	-0.003	12.567	0.245	0.129	-0.018	-0.014

Table 18.3, the direct linear comparison between the hydrostatics of the original form and the matched form.

T	DISPM	VOLM	LCB	VCB	KMT	WPA	LCF	TPC	CB	CP
0.50	2.06%	2.06%	2.30%	0.38%	-5.63%	2.60%	1.88%	2.60%	2.22%	1.94%
1.00	-0.54%	-0.54%	3.11%	0.00%	-2.96%	-0.89%	2.13%	-0.89%	4.37%	3.23%
1.50	-0.78%	-0.78%	2.43%	0.00%	-2.03%	-1.11%	0.82%	-1.12%	5.06%	3.53%
2.00	-1.14%	-1.14%	2.04%	-0.46%	-1.18%	-3.01%	1.03%	-3.01%	8.99%	7.13%
2.50	-1.04%	-1.04%	1.16%	-0.07%	-0.79%	-1.31%	-1.19%	-1.31%	5.42%	3.78%
3.00	-1.07%	-1.07%	0.66%	-0.24%	-0.57%	-2.35%	-0.97%	-2.34%	8.84%	7.31%
3.50	-1.47%	-1.47%	0.27%	-0.36%	-0.10%	-2.94%	-1.02%	-2.95%	8.62%	6.78%
4.00	-1.09%	-1.09%	-0.42%	0.09%	0.93%	-0.36%	-3.03%	-0.36%	5.14%	3.97%
4.50	-0.89%	-0.89%	-0.90%	0.08%	-0.99%	-0.61%	-2.85%	-0.61%	5.28%	4.12%
5.00	-0.69%	-0.69%	-1.28%	0.21%	-3.25%	-2.15%	-1.05%	-2.15%	5.62%	4.33%
5.50	-0.43%	-0.43%	-1.59%	0.39%	-0.61%	0.04%	-2.40%	0.04%	3.95%	3.03%
6.00	-0.27%	-0.27%	-1.74%	0.53%	-1.93%	-1.38%	-0.05%	-1.38%	4.22%	3.14%
6.30	**-0.27%**	**-0.27%**	**-1.71%**	**0.50%**	**0.33%**	**-0.32%**	**-0.82%**	**-0.32%**	**4.13%**	**3.25%**
6.50	-0.28%	-0.28%	-1.68%	0.49%	0.27%	-0.31%	-0.73%	-0.31%	4.08%	3.21%
7.00	-0.29%	-0.29%	-1.59%	0.40%	0.10%	-0.35%	-0.56%	-0.35%	4.13%	3.29%
7.50	-0.31%	-0.31%	-1.50%	0.35%	-0.01%	-0.39%	-0.47%	-0.39%	4.04%	3.36%
8.00	-0.32%	-0.32%	-1.41%	0.31%	-0.02%	-0.33%	-0.37%	-0.33%	2.74%	1.95%
8.50	-0.33%	-0.33%	-1.33%	0.29%	0.01%	-0.32%	-0.37%	-0.32%	2.68%	2.06%
9.00	-0.34%	-0.34%	-1.27%	0.25%	0.02%	-0.35%	-0.41%	-0.35%	2.64%	2.02%

Table 18.4, the hydrostatic comparison presented as a percentage difference of the values obtained from the original hull form.

The technique is very capable of producing hull forms of this type, (Figure 18.4 and Figure 18.5). With the way the technique functions, matching of this type of hull form has become just a case of entering the definition curves and reviewing the results. A small amount of manipulation may be necessary to ensure that the surface is totally fair. These hull forms are easily developed because the surface does not need to be greatly deformed from a rectangular mesh shape. The only areas that are not matched well are smaller local regions which the technique does not have the definition to produce. The hydrostatics, (Table 18.3 and Table 18.4), show a very close match. As the basic technique forms these hull forms so well, improvements would consider enhancing the local surface modification techniques to introduce more features.

18.5.2. Comparison to a Bulk Carrier Type hull form

The development of hull forms, such as those similar to bulk carriers, are much more difficult to replicate because of the reasonable amount of deformation found around the stern of the vessel. The following hull form has a stern post instead of the stern propeller bossing arrangement that made the development of hull forms using the method developed for ShipLINES, (Appendix 3), so difficult. Even so, the number of changes in the shape of the stern curve still makes the replication of this hull form a challenging task. To fully develop this hull form, trimming tools are required to enable the appendage bulb and the transom to be formed. Consequently, in the replication, the bulb is not formed and the stern curve continues aft of where the transom would be located. There is going to be a reasonable mismatch in the hydrostatics because of these differences. The origins of the vessel are unknown, it was obtained as a lines plan of an example of a hull form with a stern post, of which there are few vessels designed today. Based on the lines plan drawing, the design is probably from the 1960's and was probably drawn with the aid of a spline batten rather than modern computer software. The sections of the hull form are shown in Figure 18.6 and the particulars of the vessel are listed below:

- LBP = 163.1m
- BWL = 25.6m
- Depth = 14.0m (at the midship section)
- $X_{MIDSECTION}$ = 90.0m
- Bilge Radius = 1.829m
- There is 25.1m of parallel middle body (PMBF = 102.565m; PMBA = 77.457m)

- The extents of the parallel deck are from 132.654 (PDF) at the bow to 37.067m (PDA) at the stern

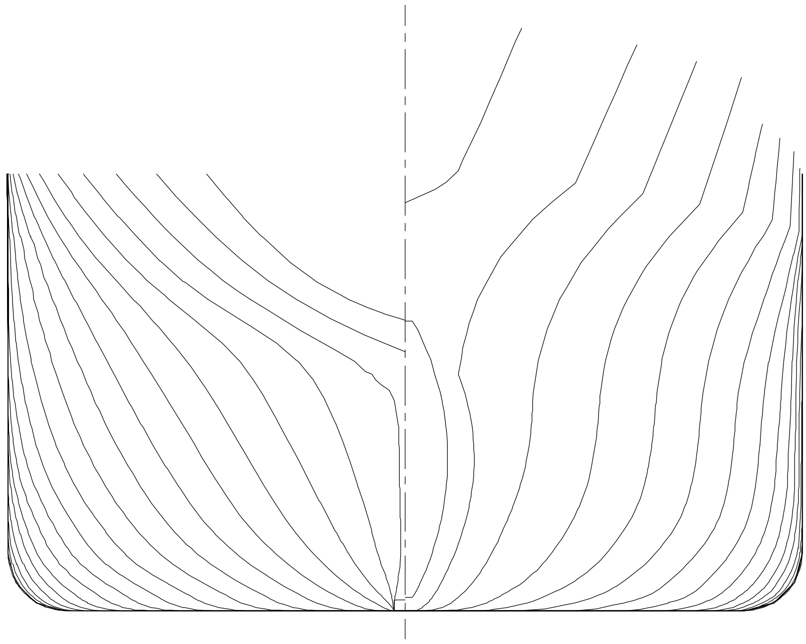


Figure 18.6, the hull form sections of the bulk carrier hull form.
Sections are spaced at intervals of 4 metres.

During development, a variety of hull forms were used to check how the technique performed in creating a hull form with similar characteristics. Based on the experienced obtained with ShipLINES, the technique was not expected to perform very highly. However, after some initial problems in the definition, the closeness of the match was quite surprising compared to the expectations. The hull, (Figure 18.7 and Figure 18.8) was developed using the same standard approach for the Ro-Ro hull form. The bulk carrier hull form consists of many more specific features, such as knuckle lines, that cannot be defined in the hull forms produced by the present implementation. As the knuckle lines are very close to the level of the deck at the midship section, the best approach was found to be to ignore the knuckle line and continue the hull form higher with the same characteristic shape. The development of the stern shape was initially quite complex. The nature of the stern requires the hull form surface to be quite deformed. In development, the stern shape was formed by using the control vertices corresponding to the bilge radius to form the highly curved segment at the top of the rudder. Getting a match with the hull form was found to be very difficult and the aft volume control curve was manually modified in an attempt to improve the surface. However, after three and a half hours of manipulation, the solution did not appear to be in reach.

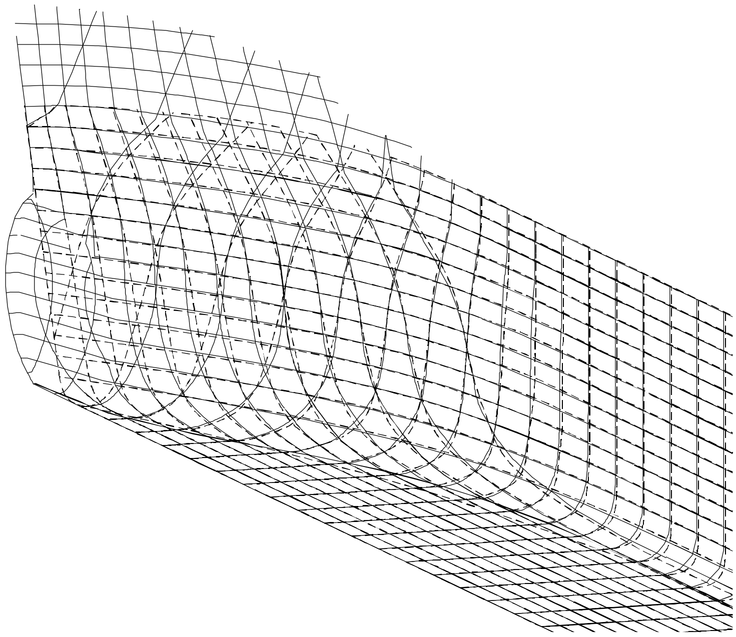


Figure 18.7, the entrance of the Bulk Carrier hull form. Again, there is quite a close match. However, this was obtained by manipulating the flat of side curve until the lines matched up. The hull protrudes toward the deck due to the knuckle line in the original hull form that was not considered. (NAPA definition – Solid Lines, IntelliHull definition – Dashed Lines)

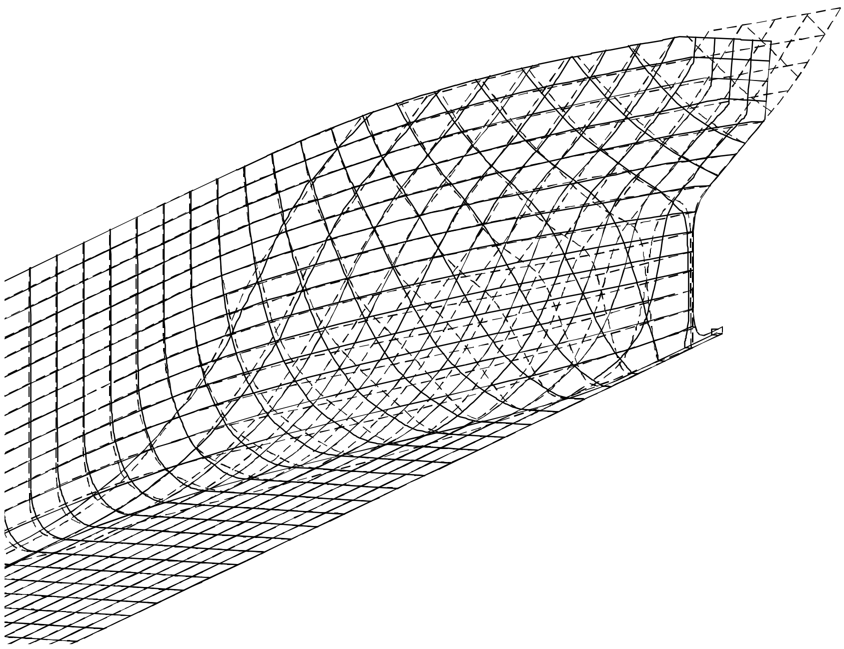


Figure 18.8, the run of the Bulk Carrier hull form. Here there is a significant difference between the original and replicated hull forms. However, the replicated stern is considerable better that what the technique was expected to produce. A different arrangement of points was found for the stern curve compared to the layout used during the development of the implementation. (NAPA definition – Solid Lines, IntelliHull definition – Dashed Lines)

T	DISPM	VOLM	LCB	VCB	KMT	WPA	LCF	TPC	CB	CP
0.50	-589.798	-575.413	-25.925	0.005	-0.078	-1165.204	-25.194	-11.943	-0.153	-0.152
1.00	-1137.021	-1109.289	-26.230	0.017	2.733	-1050.462	-25.445	-10.768	-0.158	-0.151
1.50	-1649.172	-1608.949	-25.660	0.032	4.262	-960.512	-24.350	-9.845	-0.165	-0.154
2.00	-2114.392	-2062.822	-25.298	0.053	4.814	-848.425	-24.131	-8.696	-0.171	-0.158
2.50	-2500.817	-2439.822	-25.149	0.075	5.009	-736.027	-24.032	-7.544	-0.171	-0.159
3.00	-2858.806	-2789.079	-24.952	0.102	5.136	-624.469	-24.207	-6.401	-0.172	-0.162
3.50	-3149.360	-3072.547	-24.767	0.129	5.055	-539.053	-23.856	-5.525	-0.167	-0.155
4.00	-3384.727	-3302.173	-24.791	0.155	5.034	-427.748	-24.487	-4.384	-0.167	-0.158
4.50	-3582.599	-3495.218	-24.786	0.187	5.325	-297.909	-25.493	-3.054	-0.170	-0.162
5.00	-3731.217	-3640.212	-24.859	0.217	5.821	-164.558	-26.738	-1.687	-0.158	-0.148
5.50	-3816.198	-3723.120	-25.074	0.250	5.376	-73.595	-27.436	-0.754	-0.168	-0.160
6.00	-3845.469	-3751.678	-25.389	0.283	5.565	69.406	-29.183	0.711	-0.163	-0.155
6.50	-3821.986	-3728.766	-25.740	0.319	4.878	61.473	-28.004	0.630	-0.150	-0.143
7.00	-3786.943	-3694.579	-25.961	0.350	4.534	73.088	-27.047	0.749	-0.138	-0.132
7.50	-3748.680	-3657.249	-26.052	0.376	4.206	74.842	-25.918	0.767	-0.122	-0.115
8.00	-3711.027	-3620.514	-26.039	0.398	3.900	71.981	-24.791	0.738	-0.113	-0.106
8.50	-3677.839	-3588.136	-25.940	0.415	3.623	61.514	-23.587	0.631	-0.092	-0.086
9.00	-3649.522	-3560.509	-25.764	0.428	3.372	48.062	-22.414	0.492	-0.073	-0.066
9.50	-3512.883	-3427.203	-25.823	0.450	3.231	79.738	-22.346	0.817	-0.073	-0.066
10.00	**-3476.742**	**-3391.943**	**-25.606**	**0.460**	**3.006**	**64.000**	**-21.396**	**0.656**	**-0.068**	**-0.062**
10.50	-3458.538	-3374.183	-25.324	0.465	2.798	34.522	-20.003	0.354	-0.038	-0.032
11.00	-3446.798	-3362.731	-25.016	0.469	2.607	11.491	-19.068	0.118	-0.034	-0.027
11.50	-3446.586	-3362.523	-24.691	0.470	2.434	-10.608	-18.296	-0.109	-0.031	-0.025
12.00	-3466.148	-3381.607	-24.332	0.469	2.275	-46.777	-17.298	-0.479	-0.003	0.003
12.50	-3497.527	-3412.221	-23.977	0.465	2.126	-75.753	-16.547	-0.777	-0.001	0.005
13.00	-3543.894	-3457.458	-23.617	0.460	1.988	-105.172	-15.843	-1.078	0.002	0.007
13.50	-3605.270	-3517.336	-23.256	0.454	1.859	-134.424	-15.199	-1.378	0.003	0.008
14.00	-3831.295	-3737.849	-23.138	0.431	1.730	-206.176	-15.503	-2.113	0.025	0.031

Table 18.5, , the direct linear comparison between the hydrostatics of the original form and the matched form.

T	DISPM	VOLM	LCB	VCB	KMT	WPA	LCF	TPC	CB	CP
0.50	-46.63%	-46.63%	-29.19%	1.95%	-0.09%	-44.15%	-28.33%	-44.15%	-23.57%	-22.79%
1.00	-42.65%	-42.65%	-29.49%	3.29%	5.84%	-37.49%	-28.64%	-37.49%	-23.90%	-22.27%
1.50	-39.96%	-39.96%	-28.89%	4.12%	13.12%	-33.20%	-27.53%	-33.20%	-24.30%	-22.22%
2.00	-37.58%	-37.58%	-28.53%	5.11%	19.17%	-28.71%	-27.41%	-28.70%	-24.64%	-22.41%
2.50	-34.96%	-34.96%	-28.42%	5.79%	24.18%	-24.50%	-27.43%	-24.50%	-24.22%	-22.24%
3.00	-32.84%	-32.84%	-28.26%	6.56%	28.74%	-20.50%	-27.75%	-20.50%	-24.06%	-22.41%
3.50	-30.65%	-30.65%	-28.11%	7.11%	31.74%	-17.47%	-27.46%	-17.47%	-23.07%	-21.23%
4.00	-28.52%	-28.52%	-28.20%	7.47%	34.66%	-13.70%	-28.30%	-13.70%	-22.85%	-21.44%
4.50	-26.58%	-26.58%	-28.25%	8.01%	39.47%	-9.43%	-29.57%	-9.43%	-23.04%	-21.77%
5.00	-24.70%	-24.70%	-28.39%	8.37%	45.80%	-5.16%	-31.14%	-5.16%	-21.21%	-19.76%
5.50	-22.79%	-22.79%	-28.70%	8.76%	44.39%	-2.28%	-32.09%	-2.28%	-22.37%	-21.19%
6.00	-20.89%	-20.89%	-29.12%	9.08%	47.75%	2.13%	-34.30%	2.13%	-21.56%	-20.39%
6.50	-19.03%	-19.03%	-29.59%	9.44%	43.15%	1.87%	-33.09%	1.87%	-19.69%	-18.67%
7.00	-17.38%	-17.38%	-29.91%	9.61%	41.06%	2.19%	-32.15%	2.19%	-17.99%	-17.12%
7.50	-15.95%	-15.95%	-30.09%	9.63%	38.78%	2.22%	-31.02%	2.22%	-15.93%	-14.95%
8.00	-14.70%	-14.70%	-30.16%	9.55%	36.40%	2.11%	-29.87%	2.11%	-14.66%	-13.70%
8.50	-13.62%	-13.62%	-30.13%	9.36%	34.06%	1.78%	-28.63%	1.78%	-12.06%	-11.23%
9.00	-12.68%	-12.68%	-30.01%	9.10%	31.79%	1.37%	-27.40%	1.37%	-9.66%	-8.71%
9.50	-11.48%	-11.48%	-30.17%	9.05%	30.41%	2.24%	-27.49%	2.24%	-9.59%	-8.65%
10.00	**-10.72%**	**-10.72%**	**-30.00%**	**8.78%**	**28.15%**	**1.78%**	**-26.45%**	**1.78%**	**-8.88%**	**-8.06%**
10.50	-10.08%	-10.08%	-29.77%	8.43%	26.01%	0.94%	-24.96%	0.94%	-5.09%	-4.27%
11.00	-9.52%	-9.52%	-29.50%	8.10%	24.02%	0.31%	-23.91%	0.31%	-4.52%	-3.58%
11.50	-9.04%	-9.04%	-29.21%	7.75%	22.20%	-0.28%	-23.03%	-0.28%	-4.09%	-3.29%
12.00	-8.65%	-8.65%	-28.88%	7.40%	20.51%	-1.22%	-21.94%	-1.22%	-0.41%	0.40%
12.50	-8.32%	-8.32%	-28.55%	7.03%	18.93%	-1.95%	-21.09%	-1.95%	-0.13%	0.67%
13.00	-8.05%	-8.05%	-28.21%	6.67%	17.46%	-2.68%	-20.29%	-2.68%	0.27%	0.93%
13.50	-7.83%	-7.83%	-27.86%	6.33%	16.10%	-3.38%	-19.55%	-3.38%	0.40%	1.06%
14.00	-7.94%	-7.94%	-27.72%	5.77%	14.75%	-5.07%	-19.79%	-5.07%	3.37%	4.18%

Table 18.6, the hydrostatic comparison presented as a percentage difference of the values obtained from the original hull form.

After a break, it became apparent that using the vertices corresponding to the bilge radius resulted in the part of the surface forming the flat of bottom to start horizontally at the midship section and finish vertical at the stern post. This level of deformation was obviously too much. The solution, particularly as this part of the hull form shape distribution is orientated towards flowing waterlines, much like the entrance, was to locate all the vertices forming the flat of bottom at the first point of the stern curve. Once the change was made, the surface immediately changed to represent to correct stern shape without the need for any further manual influence. Some experience in how hull form surfaces deform is required to use the technique effectively. Furthermore, the need to know about the correct arrangement of points to create the appropriate surface is always going to be a major issue for hull form development tools relying on single surface patches. Tools representing the hull form using multiple patches have a lot more flexibility. However, it is very satisfying to see that once the technique is provided with the correct definition information, the appropriate hull form shape is produced without the need to manually manipulate the hull form further. As the shape of the stern of bulk carrier type hull forms are reasonable deformed, the fact that the technique can represent the correct shape using the basic geometric rules and the resulting blending curves must identify that this technique has a particularly effective approach to the development of hull form surfaces.

Despite forming the appropriate shape for a hull of this type, a perfect match could not be achieved particularly in the stern, (Figure 18.8). This was mainly due to the lack of control the technique has over surface tangency along the flat of bottom parts of the definition curve. The arrangement of the blending curves results in very effective control of the surface shape in the longitudinal direction. However, the blending curves are not able re-orientate themselves when the control curves are stretched in the longitudinal direction. Consequently, the transverse shape of the surface cannot be controlled as effectively. Again, a multiple patch surface implementation would be able to handle this more successfully.

The hydrostatic differences, (Table 18.5 and Table 18.6), between the two hull surfaces are quite considerable. Apart from the lack of the bulb definition, the differences in the volume at the stern are large. This bulk carrier was most likely defined by hand and there are additional complications when trying to match to the hull form as a result. Today, the designer manipulates a surface definition to create the hull form. As a consequence, the surface representation imparts a constraint on the designer by not allowing certain shapes to be formed. The surface definition functions in three dimensions, whereas a considerable amount of time was spend making the lines plan function three dimensionally. The designer of this hull form was not constrained by a three dimensional representation. The design was created in two dimensions and adjusted to fit three. Therefore, the task of matching a generated surface constructed from limited definition to a manually defined hull form may not be able to achieve a close match without using additional definition. While the technique may not be able to match this particular hull form exactly, it is able to represent the characteristics very closely. This must illustrate that the technique is capable of forming a variety of hull form shapes without the need for any further modification to the approach.

18.6. Summary

The concept proposed by this study allows for better utilisation of existing hull surface definition techniques by bringing them together in an environment that will allow them to coexist. By ensuring the user can still develop the hull form using existing surface development processes, the approach aims to prevent the introduction of limitations that would ensure that tools relying on the manual definition technique must still be available, as with the case of parametric hull generation tools. Furthermore, by allowing the hull definition to be constructed from different tools, the approach allows the designer to change the level of detail in the definition as the design process of a vessel is progressed.

While it is difficult to identify any particular limitations that concept introduces, the approach may require tool developers to be more innovative. Some surface definition techniques that have been optimised to aid the designer in manual definition and may need some adaptation to allow them to function well when using form topology and geometric constraints. However, by utilising the hierarchical approach to the way definition is used to the fullest extent, a great range of different tools can be introduced which allow any limitations to addressed in the representation technique. Consequently, any limitations can be considered to be the result of a lack of imagination and skill on the part of the developer and not so much by the concept or representation technique.

.

19. DISCUSSION, APPLICATIONS AND FUTURE DEVELOPMENT

19.1. Discussion

After approximately a century of the development of mathematically based hull surface design and representation tools, the desire to find better solutions still continues. It took a long time to achieve the first major milestone in effective hull representation technology with the introduction of parametric curves and surfaces.

However, without the development of application technologies, such as shipyard automation and computer-aided manufacturing (CAM), these tools would not have brought any realistic benefits to the actual hull design process. Hull design still requires a considerable amount of time, skill and manual manipulation to produce a surface of good quality. After thirty years of use, tools employing modern surface technologies have yet to provide effective means to aid the designer in producing the hull form. This project has tried to bridge the gap between the needs of modern design practices and surface representation techniques, using an approach that makes more effective use of presently available tools and knowledge of hull form surface shape.

Most present hull surface design software is essentially implementations of surface representation techniques. Hull form design methodologies, such as transformation or parametric hull generation approaches, are added as separate tools or features and are not directly integrated into the surface development environment. This curious oversight on the part of the software developers is probably due to the idea that to provide an implementation of a surface representation technique capable of representing a hull form is enough for design purposes. Consequently, there has been hardly any development of hull surface *design* tools which truly help the designer.

As detailed optimisation becomes an increasingly important role in the ship design process, parametric hull generation techniques are again being seen as the way forward for hull surface development. However, as an approach that integrates this tool with the other hull surface design methodologies has yet to be found and as the technique, alone, is not capable of producing a detailed hull surface, it will never become a frontline design tool. The formulaic approach used in hull form generation tools results in very deterministic hull form shapes. Subsequently, certain techniques have become notorious for producing hull forms with particular shape charateristics.

Some software developers have tried to implement radically new approaches. Paramarine [36] is a tool capable of parametrically defining the whole vessel. The user interacts with the tool through a tree-like interface structure. While this change in the direction of development is commendable, a radical approach can alienate the tool from experienced practical designers.

The future of hull design is likely to involve more performance based design methodologies which are likely to offer much potential to improve the qualities of the hull form. However, unless the practical limitations found in the manufacture and operation of a vessel are taken into account, these tools may not find much use.

This project has approached the hull surface design problem from the viewpoint of present ship design strategies. Much credence is now being given to concept design and optimisation procedures, as the design process is able to make better use of emerging technologies such as time domain simulations and CFD. The present design solutions provided by hull surface development software, based around labour intensive and non-integrated tools, is no longer compatible with the modern approach to commercial ship design. A solution has been proposed in this thesis to this problem by utilising an approach which combines the strengths of the parametric and manual methods of hull generation. The resulting pilot software clearly showed that the idea has much to offer in providing designers with capable tools, which are effectively based on the characteristics inherent in the product. By reviewing the problem from the earliest stage of the hull surface design process, it was identified that the designer needs the ability to quickly create and modify the hull form. However, control of the surface shape must remain with the designer. Moreover, the designer wants to spend as little time as possible manipulating the hull form, to maximise the time that can be spent identifying the best ways of meeting the owner specifications and changes that will make this possible.

The identification that ship hull forms are composed of simple shapes and that there are relationships between these shapes, allow a topological representation of the hull form to be defined. The form topology structure can be seen as the basic shape definition of the hull surface, the shape features. Consequently, the information will be in evidence in even the most basic representations of the hull form, including the designer's initial sketch.

If the designer is provided with an ability to represent the form topology and features, by utilising constraint tools which control the surface by referencing to the topology structure, the designer will be able to realise the hull surface quickly, at a rate that is appropriate with other design tasks. A rapid hull form prototyping and development tool which had been introduced here will be invaluable in this context.

While this approach, and the consequential tools that result, may appear quite radical, the methodology can be traced to the natural extension of current techniques used in hull surface representation. The relational geometry [47] concept can be seen as a precursor of the approach, allowing relationships between different CAD entity definitions to be constructed, implementing constraints affecting position. The surface definition scheme in NAPA develops this a stage further, allowing tangential control of curves and surfaces through side conditions, tangential constraints that apply along the length of a curve or at curve intersections. TSCAHDE takes this approach yet a stage further by allowing the designer to build patterns of relationships to constrain the surface topologically. Consequently, a curve may influence the shape of the surface over a much wider area than just the locality of the definition, based on what part of the hull form the curve represents. By applying geometric form constraints to the surface definition, the designer can specify the shape of the surface over much large area of the hull form by, in comparison to present techniques, a minimal expanse of physical definition data.

To implement a scheme of geometric form constraints, it is not necessary to introduce any new technologies. In fact, many present hull design tools have the ability to introduce a limited amount of form constraint relationships. The technique does not rely on any particular class of surface representation mathematics. In the case of the implementation, it has been shown that, given a difficult representation technique for a ship hull form, such as a single NURBS surface, a capable solution can be developed requiring a considerably smaller amount of manual manipulation compared to the definition of a hull surface without the technique.

Constructing the hull surface by using form topology and geometric constraints creates an opportunity to integrate the different hull form design methodologies together in a cooperative manner. Constraints develop a structure using geometric relationships with many levels (hierarchy) of definition complexity. Consequently, it is possible to develop the means to apply modifications, such as global transformations or parametric modifications, using the different levels within the definition structure as interfaces. Modification operations on the hull surface no longer have to consider ways of preventing distortions or disruptions in the hull surface. In fact, it has been shown that discontinuous or compound transformations can be used to apply better changes in form, for much smaller initial changes in the geometry definition.

The many levels of dependency created by the relationships between the geometric elements in the definition develop a supporting framework that the designer can use to modify areas of hull shape with the knowledge that other areas will remain unchanged and that areas influenced by the changes will update and re-fair into the features of the surface. The framework makes for a very comfortable tool for the designer to work with. As the primary task of the form topology structure is to represent the overall shape of the hull surface, it is not capable of supporting local features in the surface. However, surface modification tools, such as trimming or the warping process developed by the project, can be applied as an additional level of definition. The strong constructional approach of the geometrical structure allows various definition techniques to be used to the fullest capabilities with separation to prevent modification influencing each other detrimentally.

One of the most important advances that the form topology and geometric constraint approach has enabled is the introduction of parametric hull design to a surface that is also capable of interactive manual modification. The rigorous mathematical relationship between numerical values of parameters and the shape of the hull surface geometry should not feature in a flexible design process. The traditional parametric approach constrains hull shape as well as dimensions because shape has to be constructed to achieve the dimensional constraints. However, the designer needs the ability to change freely dimensions, as the design progresses without necessarily affecting the shape, and to change the shape without affecting dimensions. Consequently, these operations need to function independently of each other, especially as shape can have a profound effect on performance for similar measurable hull dimensions.

The introduction of automatic methods of developing definition is a feature unique to this project. While geometric constraints improve the construction of a hull surface, the use of these functions should always be provided as additional definition options, requiring the user interaction to assign the initial application of the relationship. If this were not the case, the tool would override any user changes, always applying default constraints. Automatic definition generation is an excellent way of controlling surface shape and can be further adapted to aid the fairing process by actively constraining the surface definition structure, reducing the need to adjust individual definition vertices. It can also be developed to provide a less structured approach to surface definition. Most parametric surface representation techniques require the user to control many definition vertices to get the exact shape. The number is generally high, because the representation technique requires a specific structure, such as a mesh of definition vertices, to generate the surface. Once the shape of the hull form surface has been defined using a structure of constraints, areas between could be refined using further geometry developed by through the subdivision of existing definition. It may be possible to continue to interface any new definition, as a level in the hierarchy, to the form topology structure to ensure that the parametric modification techniques remain fully effective.

As TSCAHDE uses a separate definition structure to that used to produce the hull surface representation, it provides facilities to implement standalone techniques of controlling surface shape. Procedures can be developed that review surface shape, as part of the constraints functionality, in order to define form constraint relationships. These procedures could be quite detailed, employing a high level of mathematical analysis of the surface shape. However, as procedures become more technically complex, the amount of processing required to execute these functions will increase. If the processing load is too high, the performance of the interactive user interface may suffer significantly. A simpler approach is to develop passive means of controlling the shape using the arrangement of other geometric definition components and mathematical rules. These techniques are not capable of detailed control of surface shape. However, as the diamond control structure constraining the shape of the volume control curve demonstrates, (Figure 15.26), simple mathematical mechanisms can be used to prevent inappropriate hull form shapes from occurring. This particular technique requires minimal processing as all the analysis of the surface definition is developed into the mechanism of the structure. It demonstrates that a simple approach to constraint application is probably more effective than methodologies involving complex shape analysis functions.

The project demonstrates that an iterative procedure can be used to generate hull form surfaces if an appropriate implementation can be found. An iterative approach will be ineffective if the process is allowed to have an overriding control over the shape of the hull surface. In the implementation, control of surface shape and dependencies between parameters was removed from the iteration procedure. Consequently, the procedure is much more effective at achieving solution when compared to performance of other hull generation techniques, such as YachtLINES, (Appendix 2), when processing the same hydrostatics problem. The choice of the driving function is always a big factor in the performance of the iteration procedure. The Newton-Raphson technique, for example, lays down a framework that can be used to construct, generally, effective iteration functions. However, in the complex situations often found when solving problems computationally, accurate mathematical models can be practically impossible to construct. In these situations, approaches like the Newton-Raphson technique are not very helpful. However, it has been shown that it may not be necessary to calculate an accurate model of the problem. A simplistic model can greatly improve the performance of an iteration procedure. It would be interesting to see what the improvement in performance would be if this approach was now incorporated into the YachtLINES hull generation procedure.

With all these new features, how does a tool developed using the form topology and geometric constraint approach measure up to the criteria defined for more appropriate hull surface design tools? In Chapter 7, criteria defined by early developers of hull design tools, working before the introduction of NURBS, were used to review present software tools. Kuo [6] stated that development should be directed toward tools that minimise the amount of manual involvement, but were not a computerisation of the present hull surface design procedures of the time. TSCAHDE has the potential to meet these criteria. However, it should be noted that this can be achieved and is dependent on both the approach concept and implementation of the design tool.

Present commercial tools providing integrated ship design solutions could greatly benefit from the introduction of the form topology and geometric constraint approach to hull form design. However, when present ship design tools are reviewed, it can be seen that these systems continue to use the approach of high manual involvement throughout the ship design process, requiring the user to implement every analysis and modification task. As research is now developing towards an approach that considers more automated optimisation, TSCAHDE has a great deal to offer in an integrated design tool capable of optimisation. Developers of these systems are looking for the capabilities of integrating parametric control while allowing the form of the hull surface to be specified independently by the designer. Success will ultimately depend on the approach taken to implement the concept.

Benson [4] was interested in the implementation of hull design tools. He stated that a successful system should cater for practical users. Those within the draughting office must be capable of understanding how the system works. While the draughtsmen of Benson's era may not be familiar with the tools of the computer age, present designers would be familiar with the approach used within TSCAHDE.

While the approach offers a better technique of defining the hull surface, one of the major areas that this project particularly set out to address was to make better effective use of NURBS properties. However, while the IntelliHull tool gives the user the ability to apply constraints that implement NURBS properties, the designer still needs to understand the concepts involved in the construction of the properties to apply the constraints to the geometry. The question of how to resolve the gap between detailed knowledge and better effective use of the representation technique is difficult. It is possible to further package NURBS technology to hide the behaviours of the representation. However, some of the basic qualities of NURBS may begin to be lost and this may have a detrimental effect on the tool's ability to represent, flexibly, what the designer wants. A better approach may be more education. Now that the IntelliHull implementation gives a designer the ability to construct structures present in NURBS properties, investigation and experimentation can be used to understand the behaviour of the representation in conjunction with traditional methods of teaching.

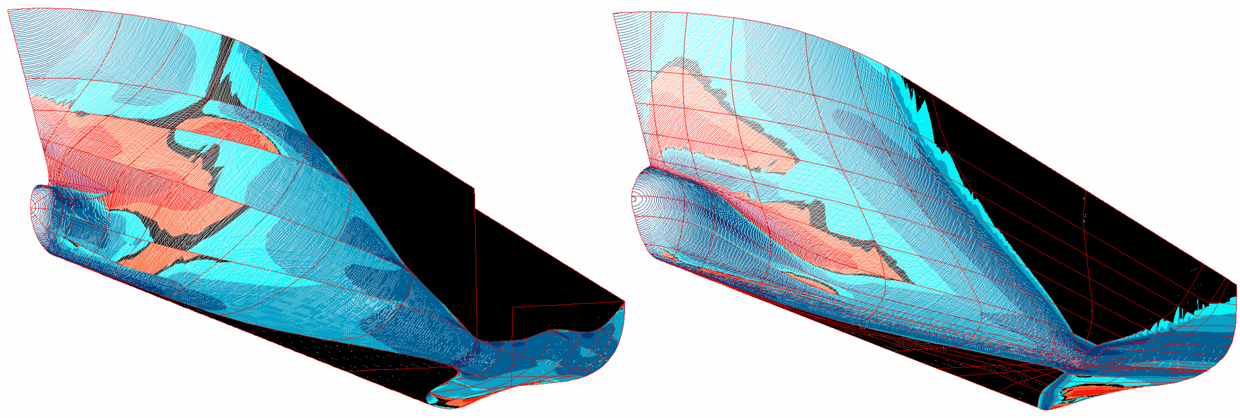


Figure 19.1, displays of section curvature calculated by the NAPA system. The left hull form was produced in NAPA over three days by a new user. The hull on the right was produced using the form constraint methodology to be a similar representation of the left hull. The displays show that the hull produced by the form constraint has much smoother variations in curvature, although the hull on the left has a much better definition of the flats, most probably as it is a multi patch representation.

Due to practical limitations of software development, the presented implementation is not capable of competing on the same terms as commercial systems developed over many years. The most efficient implementation of the approach would have required a multiple patch representation of the hull surface to be used. This would have taken a great deal of development time without any additional benefits in the demonstration of the concept. Consequently, a more simplistic approach was taken using a prototype system to demonstrate the fundamentals of the concept. It is interesting to find that because present hull design tools require so much manual effort, the implementation creates an impression that the hull form produced by the approach is much lower quality. Despite the limited use of hull surface representation capabilities, it can be shown that the implementation is capable of developing quality hull forms (Figure 19.1).

The choice of the single hull surface representation and the approach taken to allow the technique to have overall control of longitudinal shape makes it very difficult to influence certain areas of the hull surface. The transverse definition technique only allows surface shape to be subdivided transversely. While it is possible to distort the subdivided surface into an arrangement that can almost represent all the feature of a ship hull form, the approach limits the number of variations that can be produced. The designer does not have any ability to introduce any longitudinal constraints that could achieve more detailed control over shapes such as the flat of side or flat of bottom. Furthermore, if the designer is provided with the ability to introduce knuckle points into the keel boundary using the constraints, hull forms like frigates and vessels with stern posts or heavily faired skegs could be defined. Now that it is possible to demonstrate the capabilities of the concept, an implementation that can accommodate better hull surface control can be developed using the patch surface approach.

One of the major limitations, which can be resolved with more time, is the constraint that all definition curves must have the same number of vertices. Some flexibility is lost with this limitation because some shapes become more difficult to construct depending on the arrangement of the control vertices. It would have been possible to overcome this limitation by making automatic adjustments to the control curve definitions before the surface is created. This procedure could be based on the standard CAD procedures, such as lofting or skinning, which can function with definition curves that have dissimilar control polygons and knot vectors. However, further development to remove the limitation would have made the surface generation more complex and more difficult to understand for the purposes of this project.

The project has concentrated on finding a better approach of developing the hull form surface using present representation technique and design methodologies. Consequently, some of the more detailed points of the definition process within the implementation were not researched in great depth. As there are many parts of the implementation that can be researched in more detail, a modular development approach was used to ensure separation between different areas of detail within the tool. Processes that could have been investigated in more depth are implemented using a functional, but not necessarily efficient, approach. Heuristics have often been used, although it should be noted that this approach often offers a more effective solution when compared to more analytical approaches. The modular construction allows any particular details of the hull generation procedure to be investigated in more depth in the future. However, as has been already illustrated, a multi patch surface representation technique is required to make the best use of the TSCAHDE approach and any improvement in the subsystems of this implementation is unlikely to make the tool any more effective.

Having had experience using NURBS for some years before beginning this project, it was interesting to find that much more can be learnt about the technique when new things are tried. The NURBS technique offers so many possibilities and it is not always necessary to understand the detailed mathematics to make full use of the tool. Perhaps this is the main reason why tools using NURBS have failed to make any significant development since its introduction. The development of the surface modification technique that inserts the bulb feature, using the warping technique, required some detailed understanding of knot insertion. Having previously had the impression, from the structure of the technique, that it was possible to use subdivision and knot insertion to, globally, change the number and location of definition vertices without changing the shape of the curve, it is now understood, from the nature of the piecewise basis functions, why this is not possible. The implementation of the knot insertion procedure for local surface modification was quite difficult because it was necessary to understand where the control polygon geometry was located before and after the operation. Knot insertion is generally not considered in this respect in standard texts. Furthermore, the simple functions used to insert the knots hide the concepts involved. Knot insertion uses a linear calculation to modify the knot vector, regardless of the order of the function. It was necessary to create a test program to understand what happens when a knot is inserted, to understand where the new control vertex would appear, where surrounding control vertices move to and how to predict their behaviour so that the desired arrangement of refined vertices could be achieved. The test program illustrated the piecewise nature very well, once the operation could be reviewed interactively.

PolyCAD [50] has been one of the most useful tools during development. It implements the NURBS curves and surfaces in a basic state, allowing an understanding of how these techniques function. As it has been developed as a tool for the manipulation of general geometry for naval architecture applications, the design of the package contributes much to this project by implementing an already tried and tested user interface into which the IntelliHull implementation could be directly inserted. Despite the high general use of PolyCAD, additional development was undertaken to create an editing interface that allows entity manipulation both interactively and parametrically at the same time. The development of this interface created a standardised approach that could be used to access any parametric type of information related to entities used for IntelliHull definition and was subsequently, developed for other entities in PolyCAD. It shows that, particularly in hull design, the user needs to have completely open access to definition data, to modify it in any way at any time. Consequently, PolyCAD has developed a very comfortable and businesslike design environment. The user feels that full control over the design can be maintained. A feeling that just cannot be achieved with feature-orientated interfaces found in applications such as Paramarine.

As the TSCADHE approach has essences of other techniques, it is difficult to clearly identify any disadvantages. It has been designed to address disadvantages in present hull surface design methodologies by integrating previously incompatible techniques. If it could be incorporated into a presently available hull design tool and the implementation and interface was developed to provide the full capabilities of the technique effectively, the resulting tool would be very powerful. Although it makes the hull design approach very much more streamlined, it may take some time for designers to get used to the idea that the concept of an approach that constrains definition does not necessarily mean that the user is prevented from exploring a wide range of solutions. It is important for the design tool to ensure that the designer is in always in charge of the surface shapes being created. This technique implements this by providing the designer with a greater range of tools and functions to control the hull surface representation.

A subsequent stage in the development of the hull design framework will be to identify techniques that will allow constraints to be refined in a practical manner, methods that can control the amount of complexity in the definition framework between levels. Without this ability, the approach will still face problems when removing the hull form from the surface development environment, although these issues are not as limiting as those faced by present parametric hull generation techniques.

19.2. Application of the Approach

Once it is possible to use form topology and geometric constraints to control a hull form surface, the number of applications of such a technique start to grow because the hull form can be manipulated in so many more ways. Despite the limitation of parametric hull generation approach, there is still a great desire within the industry to find a functional technique. Applications of parametric hull generation can be in non-design related areas. It can be used to produce a hull form representation for an existing ship without documentation. Such requirements often arise in the ship repair and salvage businesses. The form topology and geometric constraint approach could be used to create these hull form surfaces in a matter of minutes, from a sketch or photographs, supported by a minimum number of main particulars.

One of the major problems with NURBS hull surface representation techniques is the difficulty involved in producing a hull surface representation from existing data. Consequently, brute-force surface interpolation techniques using goal seeking methods such as genetic algorithms are being developed. These approaches may spend many hours iterating over a hull form until the surface matches offset data to the required standard of accuracy. Considering the ease with which simple geometric constraints functions have been established in this project, it would be very easy to use this approach to minimise the amount of calculation by first establishing the form topology, defining the specific geometric constraint functions and then work on improving the accuracy of the areas of the surface, controlling geometric constraints by modifying parameters or subdividing definition. Of course, round bilge hull forms, due to the nature of the surface shape, are going to require more analysis than other forms. However, as re-representation generally occurs for hull forms that have been built, the data is likely to include many construction features that the TSCAHDE approach can aid in representing.

Nevertheless, the biggest application of the technique will be in design based activities. The implementation developed by the project has been mainly aimed at the concept design stage, where the hull form surface can be used as a basic model on which initial calculations of hydrostatics, estimates of hydrodynamics, stability and weight performance can be made. A COM (common object model) interface could be developed to allow the implementation to be connected to spreadsheet programs like Microsoft Excel. The designer could set up the basic design calculations within the spreadsheet, allowing the hull form to be parametrically modified through the COM interface to meet the specified target. Many naval architecture packages are now offering these facilities and they are not very difficult to implement using modern software development tools. If the system was implemented using a multi patch hull surface representation, the opportunities for more technical optimisation exist. As this is an emerging field of design, requiring the types of modification techniques available in the TSCAHDE approach, it is likely that this area of development may be targeted foremost.

19.3. Future Work

As the technique has a very large number of potential applications, the possibilities for future work are endless. The present implementation leaves many of the detailed areas of a basic form topology and geometric constraint approach unexplored and it could be used to investigate better methods of controlling surface tangency at the boundaries of the parallel middle body and the control of volume with respect to shape in better detail. These elements were tackled adequately by the project, but better solutions may exist. However, as the pilot implementation does not have all the capabilities possible with patch hull representation techniques, the most appropriate route forward is the development of a multi-patch based technique first.

Implementation will form the initial development task of a multi-patch hull surface representation. The basic functionality of patch surface structure needs to be developed before any progress can be made into the application of form constraints. There is some information available on creating multi-patch surfaces in the public domain and some initial contact with regards to this design approach has been made with commercial developers implementing multi patch hull surface definition tools.

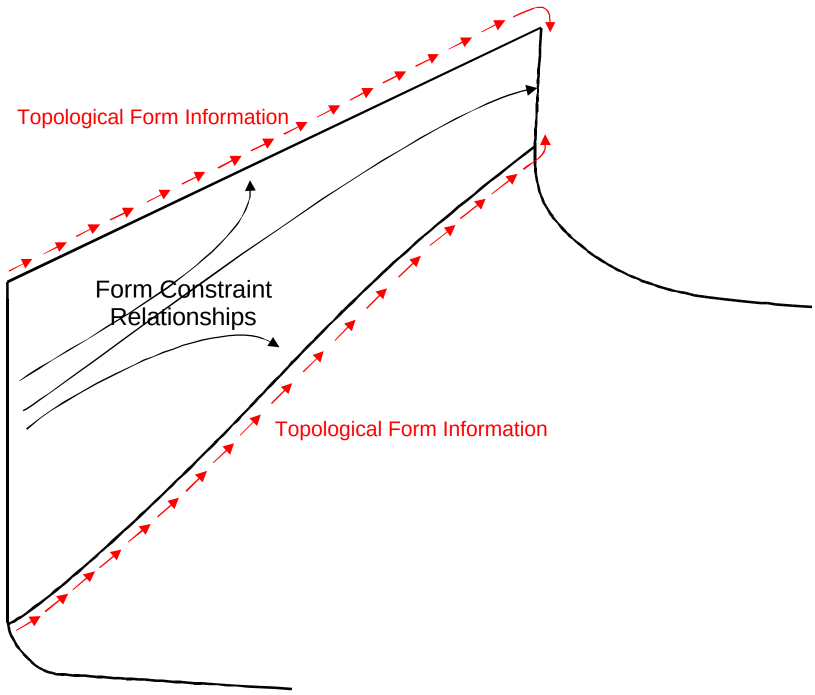


Figure 19.2, a generalised approach to form constraints would be able to take more advantage of the curves. For example, if the flat of side curve connected with the transom curve, some of the shape would be controlled by the midship curve. The network of interconnected definition curve can be used to communicate the topological information to form constraints to be applied automatically.

Once a structure of surface patches can be constructed, the primary goal would be to develop a more generalised approach to the application of geometric constraints. The present implementation has an almost monolithic style with separate definition curves about which the surface must pass. The mesh of definition curves used for a multi-patch representation allows a more integrated form topology structure to be used where it is possible to build up more fundamental geometrical constraints and implement general topological relationships. The primary goal for the next stage in the development of the TSCAHDE approach is to create a structure for a multi-patch representation that can automatically establish constraints based on the form topology and the relationship of a definition curve to its neighbours. This type of structure would allow many different regions of shape to be formed and constrained without relying on any predefined definition structures defined within the tools implementation, as in the case of the pilot system. The structure will allow the user to take more advantage of the geometric constraint relationships. For example, in an arrangement where the flat of side curve is connected from the midship curve to the transom, (Figure 19.2), part of the transom shape and the longitudinal shape of the aft deck boundary is defined by the midship curve. Consequently, any changes to the midship definition curve are transferred throughout the network of curves, updating the shape of related curves.

Once generic relationships can be established, the introduction of interactive interface tools, such as object snapping, and other tools of relational geometry, such as beads, can be incorporated. At this stage the surface design system should be capable of performing all the basic tasks that the present implementation can do, although it will be much more functional. Further progress is required to develop a tool appropriate for optimisation. The following areas will need to be addressed:

1. Better control of surface tangency. Enabling control of the surface around knuckle lines or Chines. Practical mechanisms for controlling surface tangency have yet to be defined.
2. Techniques of refining the definition of the form topology structure
3. The introduction of user customisation facilities. So more custom constraints can be applied and to allow the application custom parametric modification procedures.

The development of these features should produce a tool capable of meeting all the demands of a hull surface optimisation analysis tool. However, the needs of the naval architect should be continually kept in mind to make sure that the design tool remains practical to use.

20. CONCLUSIONS

In the effort to understand and improve hull form surface design tools this study has reviewed an extensive range of techniques that play a role in the process and has proposed a solution by identifying ways of integrating presently incompatible techniques and providing methods and tools for controlling definition consistent with the designers needs. The following conclusions can be drawn:

- Over the years many different hull representation techniques have been used in design and construction. However, today, parametric surfaces, such as NURBS, are almost exclusively used to represent the hull surface. These techniques are flexible enough to be used across concept design, all phases of construction and in through life support. Consequently, disadvantages due to the amount of surface definition data that must be provided are insignificant compared to the benefits. Hence, a more effective hull form design process can only be achieved by improving the design tools.
- Despite improvements in technology, the methods used to design the hull form have remained fairly standardised. Even today, the process predominantly revolves around the re-use of existing designs because it is so prohibitively expensive to develop designs from scratch. As the use of performance evaluation tools becomes more frequent, optimisation will play a greater role in hull form development. In both design processes, modification is generally orientated around changes to form characteristics. In present hull surface design tools, changes to form characteristics can only be achieved through extensive changes to the surface definition which must be implemented manually, regardless any existing relationships between form characteristics and hull surface shape.
- Although the idea of the modern *computer aided hull design package* gives an impression of a highly advanced and effective tool, the reality is that these programs are no more that software wrappers for the mathematical functions contained within and do not provide features to assist the designer achieve the desired surface shape. The designer must manually manipulate significant numbers of definition points to produce a basic hull form representation. This process is time consuming and may require detailed information which is unavailable in the early stages of design. Consequently, designer may restrict or postpone modifications until they are absolutely necessary.

- Parametric design tools have always been seen as an alternative to the manual surface definition approach. However, these techniques are highly dependent on the mathematical functions and processes used to create the hull surface. As these are usually quite complex, they are not open to a significant range of modification and change. Consequently, these techniques cannot provide the level of flexibility required for practical hull form design.

- Parametric hull surface generation design tools are good at quickly producing a hull surface and manual hull surface definition tools are effective when it comes to detailed changes. It would seem that these tools could have an apparent complementary relationship in the hull surface design process. However, parametric hull generation tools rarely produce a surface that is easily edited by hand and is not possible to accurately move the hull surface from the manually manipulation environment back into the parametric hull generation tool. Furthermore, even if present versions of these tools could be combined together, they would not support the kind of form characteristic changes desired by the designer. Consequently, there exists a large technology gap between tools effective in the concept and detailed design phases of the hull form.

- Through exploratory development of manually and parametrically orientated hull design tools it has been possible to identify some common areas to begin an integration process. Presently, hull surface representation is achieved using one homogeneous definition structure in which all the features, shapes and appendages must be included. By separating the definition of the hull surface around shape and function, a "divide and conquer" process can be instigated. Computing power can be used to combine separate definitions together in a hierarchical structure to produce the data required by the surface representation function. A hierarchical definition structure provides an excellent platform to introduce new software methods that can assist the designer.

- By evaluating the approach the designer takes to develop an initial design, a hierarchical definition structure can be orientated around the topology of the hull form. This structure is closer to the designer's mental representation of the hull and forms a framework on which techniques for controlling form characteristics can be implemented.

- Between the structural elements of the topology lie shapes governed by simple relationships when compared to the overall complexity of the shape of the hull form.

Tools can be developed to constraint the hull surface definition into the correct shape. Software can directly assist the designer, where appropriate, by automatically applying constraints based on a knowledgebase of valid hull form topologies.

- As the basis for this technique is a hierarchical definition structure, it can be designed to accept as much or as little data as the designer is prepared to give. The knowledgebase of hull form topology structures can be used to identify missing parts and automatically produce definition based on relationships to other parts of the structure. Consequently, this approach can form the basis for a hull design tool that is effective throughout all design phases.

- The pilot system was produced to investigate and evaluate certain aspects of the approach. Specifically, the hierarchical definition structure incorporating a representation of hull form topology and constraint tools. Considering the resources available for this study, the tool exceeded expectations on the range of hull forms that could be produced. Given the expertise available to commercial software developers, this technique could be developed into an incredibly powerful hull design tool.

- When compared to techniques used within existing hull design tools, this technique excels not only because it contains the existing techniques. By taking a pragmatic approach, considering the designers requirements a priority, a tool can be produced which allows the features of the hull form to be controlled using the appropriate means. Numbers can be used to control dimensions and geometry can be used to control shape. Furthermore, the hierarchical definition structure allows for a robust design environment that can readily accept changes at any level and are just as easily reversed if the changes did not improve the effectiveness of the design.

- Considering all the different method parametric surfaces can be used to represent the hull form, the ideal approach is thought to be a multiple patch arrangement of about ten to fifteen NURBS patches, similar to the format used in DFform [29]. This allows for a wide variety of hull surface topologies to be implemented within the form topology without requiring a complex definition evaluation algorithm.

21. REFERENCES

[1] F. H. Chapman, A Treatise on Ship-Building, 1760.

[2] J. Scott Russell, The Modern System of Naval Architecture, circa 1870.

[3] D. W. Taylor, Calculations of Ships' Forms and Light Thrown by Model Experiments upon Resistance, Propulsion and Rolling of Ships, Intl Congress of Engineering, San Francisco, 1915.

[4] F. W. Benson, Mathematical Ships' Lines, Trans R.I.N.A., Vol. 82, 1940.

[5] J. Kerwin, Polynomial Surface Representation of Arbitrary Ship Forms, Journal of Ship Research, Vol. 4, No. 12, 1960.

[6] C. Kuo, Computer Methods for Ship Surface Design, Longman, London, 1971.

[7] C. von Kerczek, The Representation of Ship Hulls by Conformal Mapping Functions, Journal of Ship Research, Vol. 13, No. 4, December 1969.

[8] G. Kuiper, Preliminary Design of Ship Lines by Mathematical Methods, Journal of Ship Research, Vol. 14, March 1970.

[9] FORAN, Sener Ingenieria y Sistemas, s. a., http://www.foransystem.com/

[10] A. M. Reed and H. Nowacki, Interactive Creation of Fair Ship Lines, Journal of Ship Research, Vol. 18, No. 2, June 1974.

[11] P. Bézier, Numerical Control, Mathematics and Applications, John Wiley & Sons, London, New York, 1972.

[12] R. F. Riesenfeld, Applications of B-spline Approximation to Geometric Problems of Computer Aided Design, Ph.D dissertation, Syracuse University, 1972.

[13] G. Creutz and C. Schubert, Interactive Curve Creation from Form Parameters by Means of B-splines, Schiffstechnik, Vol. 25, 1978.

[14] F. C. Munchmeyer, C. Schubert and H. Nowacki, Interactive Design of Fair Hull Surfaces using B-splines, ICCAS III, North-Holland, 1979.

[15] N. G. Fog, A B-Spline Surface System for Ship Hull Design, ICCAS V, North-Holland, 1985.

[16] M. Pommellet, Advanced Use of Bezier Surfaces for Computer Aided Hull Definition, ICCAS V, North-Holland, 1985.

[17] I. M. Yuille, The Forward Design System for Computer Aided Ship Design using a Mini-Computer, Trans R.I.N.A., Vol. 120, 1978.

[18] A. J. Keane, A Computer Based Method for Hull Form Concept Design: Application to Stability Analysis, Trans R.I.N.A., Vol. 130, 1988.

[19] M. I. G. Bloor and M. J. Wilson, Using Partial Differential Equations to Generate Free-Form Sufaces, CAD 22, 1990.

[20] J-H. Jorde, Mathematics of a Body Plan, The Naval Architect, January 1997.

[21] Toisa Perseus, Significant Ships of 1998, RINA.

[22] Bold Endeavour, Significant Ships of 1999, RINA.

[23] Disney Magic, Significant Ships of 1998, RINA.

[24] Voyager of the Seas, Significant Ships of 1999, RINA.

[25] D. J. Eryes, Ship Construction.

[26] H. Lackenby, On the Systematic Geometrical Variation of Ship Forms, Trans. RINA, Vol 92, 1950.

[27] Maxsurf, Formation Design Systems Pty Ltd., http://www.formsys.com/.

[28] Autoship, Autoship Systems Corporation, http://www.autoship.com/.

[29] DFform and ShipGen, DEFCAR Ingenieros, S.L, http://www.defcar.es/.

[30] Tribon M1, Kockums Computers Systems, http://www.tribon.com/.

[31] D. E. Goldberg., Genetic Algorithms, Addison Wesley, 1989.

[32] N. E. Standerski, The Generation and Distortion of Ship Surfaces Represented by Global Tensor Product Surfaces, Dissertation, TU, 1988.

[33] A. H. Day & L. J. Doctors, Minimal-Resistance Hullforms for High Speed Craft, Trans. RINA, Vol 138, 1996.

[34] FastShip, Proteus Engineering, http://www.proteusengineering.com/.

[35] NAPA (Naval Architecture PAckage), Napa OY, http://www.napa.fi/.

[36] Paramarine, Graphics Research Corporation Ltd., http://www.grc-ltd.co.uk/.

[37] D. F. Rogers and J. A. Adams, Mathematical Elements for Computer Graphics, McGraw-Hill Inc, 1976

[38] D. F. Rogers, An Introduction to NURBS: with Historical Perspective, Morgan Kaufmann, 2000.

[39] L. Piegl and W. Tiller, The NURBS Book, Springer, 1995.

[40] M. Mortensen, Geometric Modelling, Wiley, 1997.

[41] G. Farin, Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide, Academic Press, 1990.

[42] O. Nørskov-Lauritsen, Practical Application of Single Curved Hull Definition: Background, Application, Software and Experience, ICCAS, 1985.

[43] J. J. Jensen and J. Baatrup, Transformation of Ship Body Plans into a B-spline Surface, Danish Centre for Applied Mathematics and Mechanics, The Technical University of Denmark.

[44] S. M. Hollister, The Dirty Little Secrets of Hull Design By Computer, New Wave Systems Inc. http://www.newavesys.com.

[45] The Initial Graphics Exchange Specification (IGES) Version 5.2, ANSI Y14.26M, U.S. Product Data Association, Fairfax, VA, 1993.

[46] Standard for the Exchange of Product Model Data (STEP), ISO 10303, A series of documents, Part 42: Geometric and Topological Representations, ISO 10303-42, ISO Secretariat, National Institution of Standards and Technology (NIST), Gaithersburg, MD, 1994.

[47] John S. Letcher Jr, D. Micheal Shook and Simon G. Shepard, Relational Geometric Synthesis: Part 1 – framework, Computer-Aided Design, Vol. 27, No. 11, pp. 821-832, 1995.

[48] J. Leather, Colin Archer and the Seaworthy Double Ender, 1979.

[49] S. Harries and C. Abt, Parametric Design and Optimisation of Sailing Yachts, 14th Chesapeake Sailing Yacht Symposium, 1999.

[50] M. Bole, PolyCAD, http://www.polycad.co.uk/.

[51] Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma et al., Addison-Wesley, Massachusetts, 1995

[52] P. Bourke, CONREC – A Contouring Subroutine, BYTE Magazine, July 1987.

[53] M.Bole, Parametric Generation of Yacht Hulls, Final Year Project, University of Strathclyde, Glasgow, 1997.

[54] 3D Studio MAX, Autodesk Corporation, http://www.discreet.com/

[55] Prolines 98, Vacanti Yacht Design, http://www.vacantiw.com/.

[56] Prosurf, New Wave Systems, Inc. http://www.newavesys.com/.

[57] FastShip, Proteus Engineering, http://www.proteusengineering.com/.

[58] MultiSurf, Aerohydro, Inc., http://www.aerohydro.com/.

[59] I. M. Yuille, The Forward Design System for Computer Aided Ship Design using a Mini-Computer, RINA Trans, 1978.

[60] Fairline, Aerohydro, Inc., http://aerohydro.com/.

[61] J. Gerritsma, J. A. Keuning and A. Versluis, "Sailing Yacht Performance in Calm Water and in Waves (The Delft Series)" Delft University of Technology, 11th Chesapeake Sailing Yacht Symposium, SNAME.

[62] A. Rosenfeld and E. Johnston, Angle detection in digital curves, IEEE Transactions on Computers, 22, 875-878, 1973.

[63] Circumradius, http://mathworld.wolfram.com/Circumradius.html.

[64] O. Gonzalez: Global Curvature, http://lcvmwww.epfl.ch/curvature.html

[65] M. A. Kulczycka and L. J. Nachman, Qualitative and quatitative comparisons of B-Spline offset surface approximation methods, Computer-Aided Design 34, 19-26, 2002.

[66] L. A. Piegl and W. Tiller, Computing offsets of NURBS curve and surfaces, Computer-Aided Design 31, 147-156, 1999.

[67] B. K. Choi and S. C. Park, A pair-wise offset algorithm for 2D point-sequence curve, Computer-Aided Design 31, 735-745, (1999).

[68] Drawing Exchange Format (DXF), AutoCAD, Autodesk Inc, http://www.autodesk.com.

22. APPENDIX 1 – INDIVIDUAL SOFTWARE PACKAGE REVIEW

22.1. Prolines 98 – Vacanti Yacht Design

Prolines [55] began development in 1985 as one of the first PC based hull design programs to feature B-Spline surfaces. The software has been developed over the years; more recently using Delphi, into a graphically oriented surface design system running on the Windows platform. A demonstrator version of Prolines 98 was downloaded from the Vacanti Yacht Design website for the purpose of this review.

Prolines is a package tailored to small craft hull design. The package allows the design of hull forms using B-Spline surfaces. Additional software B-PLATES is used to generate sheet metal patterns from the hull surface. B-PLATES performs analysis of the Gaussian curvature of the surface, allowing the hull to be corrected in to developable surfaces allowing easy construction, especially in smaller boatyards where the level of forming technology is not so high. The graphical interface to PROLINES is standard for hull design software, (

Figure 22.1). Up to four different screen windows allow the hull surface to be viewed and edited from different directions.

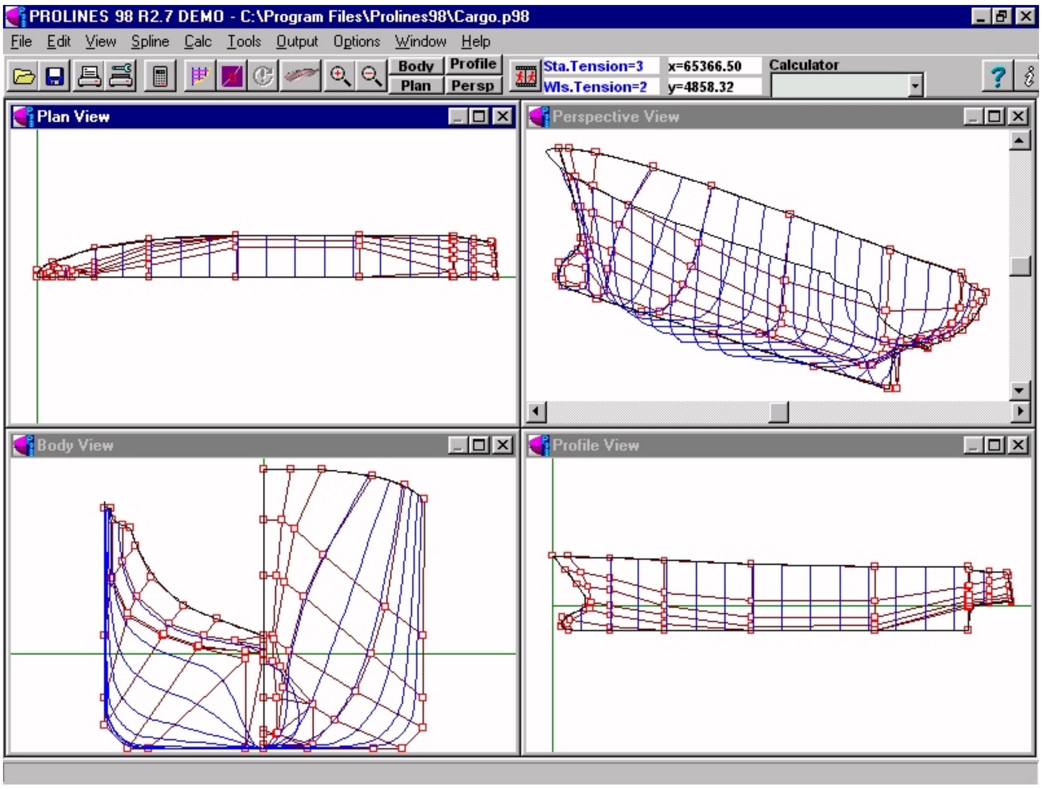


Figure 22.1, Prolines Interface Window

To begin a hull design, the initial hull surface shape can be generated using parameters in a wizard style interface that allows the choice of several different hull types: -

- Sail boat – Round Bilge
- Power boat – No Chine
- Power boat – 1 Chine
- Power boat – 2 Chines
- Cargo Ship

A choice of Stern type allows the shape of the surface to varied further. The size of the hull surface is modified with a set of parameters consisting of: -

- Length overall
- Half Breadth
- Min Freeboard
- Maximum Draught
- Bow Angle
- Stern Angle

Some parameters are only valid for certain hull types. Once the surface has been created, the mouse can be used to modify the shape further by interactively moving the control polygon of the B-Spline surface. The program provides a basic selection of tools to modify the surface properties, such as changing the number of control polygon row and columns, changing the tension (mathematical degree) of the surface, adding knuckle lines and tools to position the control polygon vertices. The software provides what could potentially be a very efficient method of fairing the surface. A dialogue box is provided which allows the user to modify sections of the surface while viewing the resulting curvature on a graph. This system can be difficult to use as the vertices of the surface can only be moved absolutely by typing the new position of the point or by using button to incrementally displace the position of the vertex. If the interface to this were to be further developed to allow more analogue interactivity such as using the Mouse, the system would be easier to use.

Once a surface is created, the resulting design can be analysed with a basic set of functions, which allow the review of Hydrostatics, Wave and Friction Drag, GZ Curve, Curve of Areas and the Curve of Wetted lengths. The surface can also be exported to other CAD systems using the DXF [68] and IGES [45] file formats. The hull can also be exported to the IMS VPP to analyse the performance of a sailing yacht. To aid the designer in the visualisation of the hull form, the software can render a shaded image of the surface, (Figure 22.2).

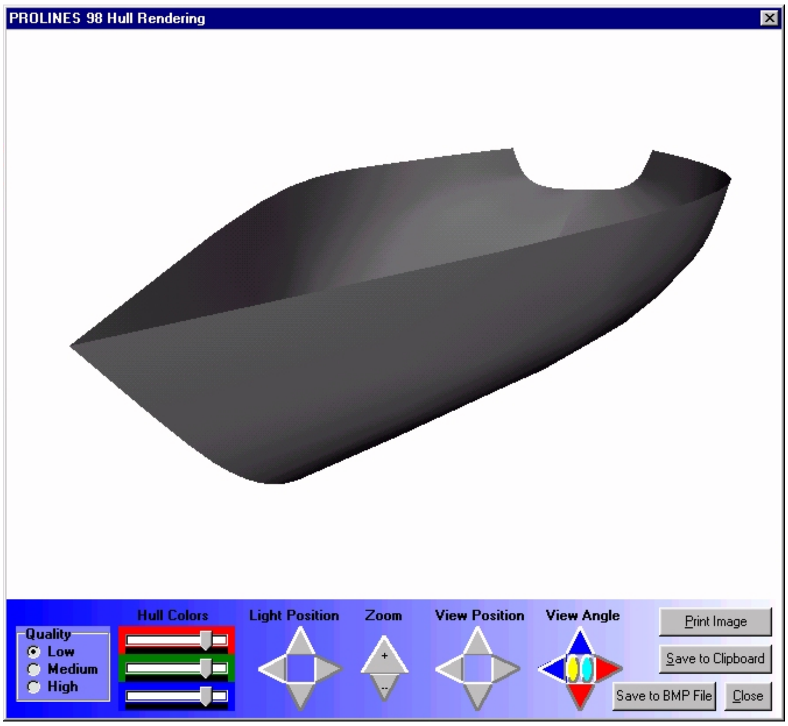


Figure 22.2, Prolines hull rendering interface.

This software is very basic and provides all the tools necessary for a small craft designer to produce simple free form hull surfaces. It is possible to create more complicated hull shapes, however, with increasing vertex numbers it becomes difficult to see the effect of the modifications to the surface especially in areas of complex curvature, such as around the bulbous bow for example. It also becomes more difficult to interact with the surface as the computation necessary to display the surface and calculate contours increases with the number of vertices.

The user interface for this program is simple and provides a good range of tools to design simple hull forms. It is an ideal package for small craft or amateur designers.

22.2. ProSurf – New Wave Systems Inc.

ProSurf [56] is part of the Nautilus system developed by New Wave Systems, Inc. The company, founded by Stephen Hollister in 1985, is a developer of CAD/CAM/CAE software for boat and ship design, analysis and construction. ProSurf is a Windows application in a very similar style to Prolines 98 in which a hull can be designed through the manipulation of the surface control polygon by the mouse. ProSurf has a number of entities that can be used to design a hull. Points, curves and surfaces can be created using B-Splines.

To create a surface, the four corner points of the control polygon are picked using the mouse. This creates a basic 2D surface. Further rows and columns can be added to the control polygon using the editing facilities of the program. Once the vertices of the control polygon have been creates the surface can be manipulated in to a hull type surface by moving the control vertices. This process can be very time consuming and the manner in which the vertices are moved can be critical to achieving a good surface representation of the hull. As the vertices of the control polygon are moved, the surface can become very deformed and the user may find it increasingly difficult to visualise the hull form within the surface at early points of the design, especially as this software only allows one vertex to be moved at a time.

To help the user generate an initial hull shape, a “Create Boat” tool is provided. This allows the user to create an initial hull surface based on a chine or round bilge sailing yacht hull. Numeric parameters are used to control the shape of the surface through control of the length, overhangs, sheer and beam at different positions. The tool creates a hull using a 3 x 3 control polygon grid. This tool allows the user to create an initial surface of the correct size with vertices in good locations for surface manipulation. It is then up to the user to modify the surface further through moving and the addition of control vertices until the desired hull shape is attained.

A rudimentary system of relational geometry is built into ProSurf. Relational geometry is where the shape of an entity depends of the shape of another generally less complex entity. This allows a user to build up a complicated shape from simple parts. Many hull design programs use this technology to control the boundaries of a surface through links to separate curves. ProSurf does not have these features. Points can be linked to curves or surfaces so that they always remain on or in the entity. Similarly, curves can be linked to surfaces. However, as points cannot be linked to curves or curves to surfaces this realisation of relational geometry provides no real benefit to the user. ProSurf does feature a method of bonding two surfaces together, but it is necessary for the two edges to exactly match each other before bonding will take place.

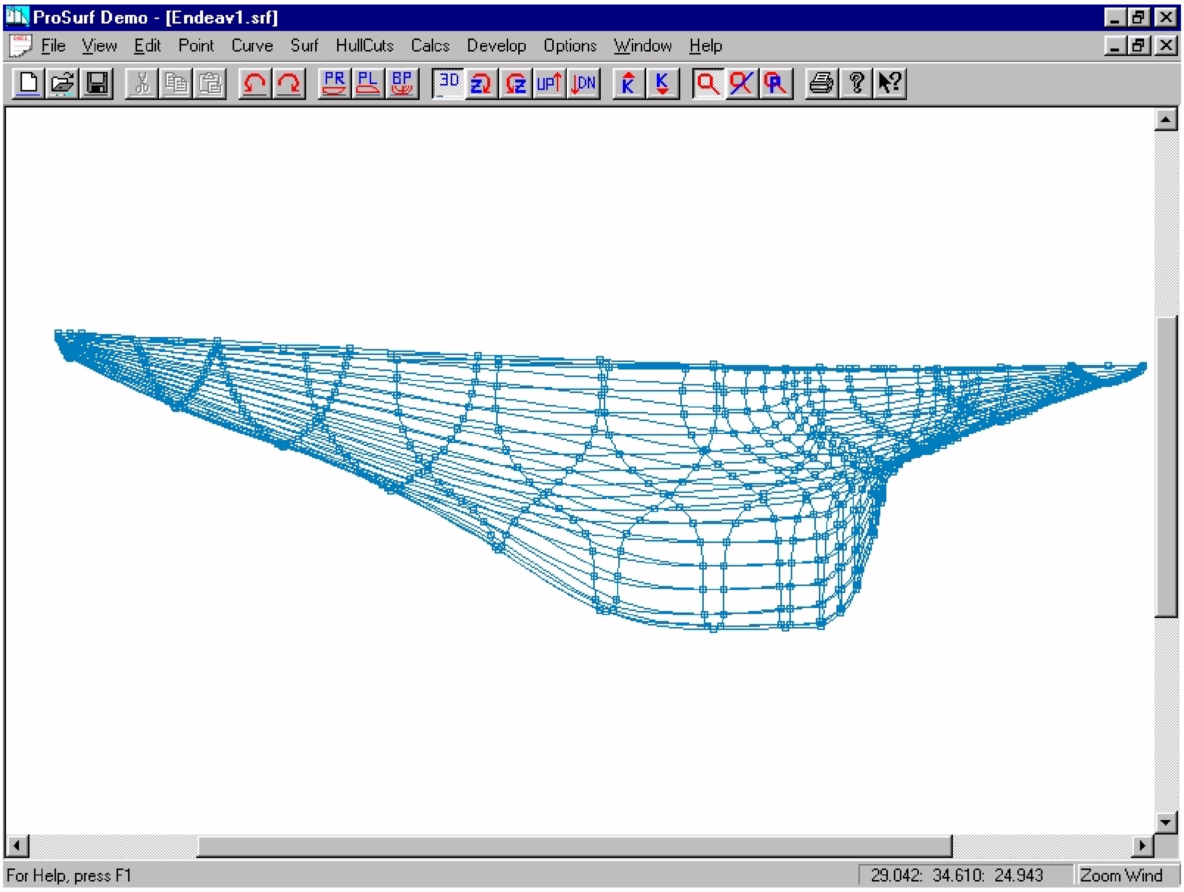


Figure 22.3, Endeavor as digitised in ProSurf

As Figure 22.3 shows, complicated shapes can be made with this software, however, the number of points in this particular surface is so great that any modification to the surface will require the movements of many vertices.

ProSurf, as do many of the smaller hull surface design tools, relies on a "blank screen" approach to surface design. On initiating the software, the user is faced with a blank screen on which to design the surface. This gives the user no feeling or indication of the location and scale of the surface, even with a display of the co-ordinate location of the mouse cursor. This lack of on-screen information is exacerbated by the way that all of the surface manipulation tools are located in the main menus and are, therefore, hidden from the sight of the user most of the time. New users can find it difficult to find these tools, as they must search through the all the menus to find the right tool. As some of the names of the tools are abbreviated, the search for the correct tool is made more difficult. An interface in this style goes against the philosophy of modern software design. Furthermore, each tool operates in a modal fashion, a tool remains in operation until the user selects another. Although this seems like a good approach to take, the operation of the program becomes more difficult. For example, once a surface is created, the next instinct of the user is to modify the surface by manipulation the vertices shown on the screen. However, as the program is still in the "create surface" mode the vertex of the next surface to be created is added to the screen.

ProSurf does not provide any interactive modes for changing the view on the model, such as panning and rotating the projection. This can only be achieved by clicking the controlling buttons on the toolbar. Aside from the necessary mouse use to manipulate the surface vertices, the program requires a lot of mouse manipulation to operate. This makes the program very uncomfortable to use.

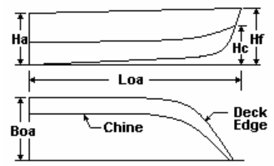
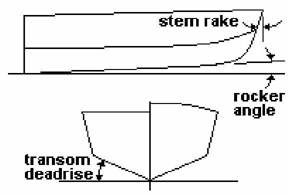
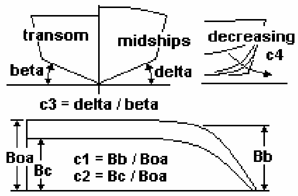
As a hull design package, the software does not provide enough detail in the hydrostatics calculations to be useful and the inefficient interface does not make this an application that could be recommended to any designer looking for an effective small hull design package.

22.3. FastShip – Proteus Engineering

FastShip [57] is currently being developed by Proteus Engineering. Development began in 1985 by Design Systems & Services, Inc. and the package now has many users across the world. The package provides a complete set of tools for vessel design and performance evaluation.

FastShip uses NURBS surfaces to represent the hull form. The surfaces are created from the main menu and can be manipulated on screen using the mouse. However, as the initial surface provided by FastShip requires a lot of manipulation before it is detailed enough to model a hull form, the software provides additional tools to generate hull shaped surfaces.

FastShip provides two versions of tools known as hull wizards to create initial surfaces, for planning hulls, (Figure 22.4), and container ship forms. The hull wizard system is based upon stored surface data, which is modified based the information provided by the user. Once the user is satisfied with the information entered into the hull wizard, the software loads the basis surface and makes modifications on screen. The planing hull wizard accepts the following parameters and gives a diagram for the location of each parameter:

Gross Dimensions	Length overall, Loa Beam overall, Boa Deck Height Forward, Hf Deck Height Aft, Ha Chine Height Fwd, Hc Chine Width, Wc	
Angular Dimensions	Transom Deadrise, beta Rocker, psi Stem Rake, alpha Deadrise Limit, betalim	
Shape Factors	Bow Fullness, c1 Transom Width, c2 Bottom Twist, c3 Forefoot shape, c4	

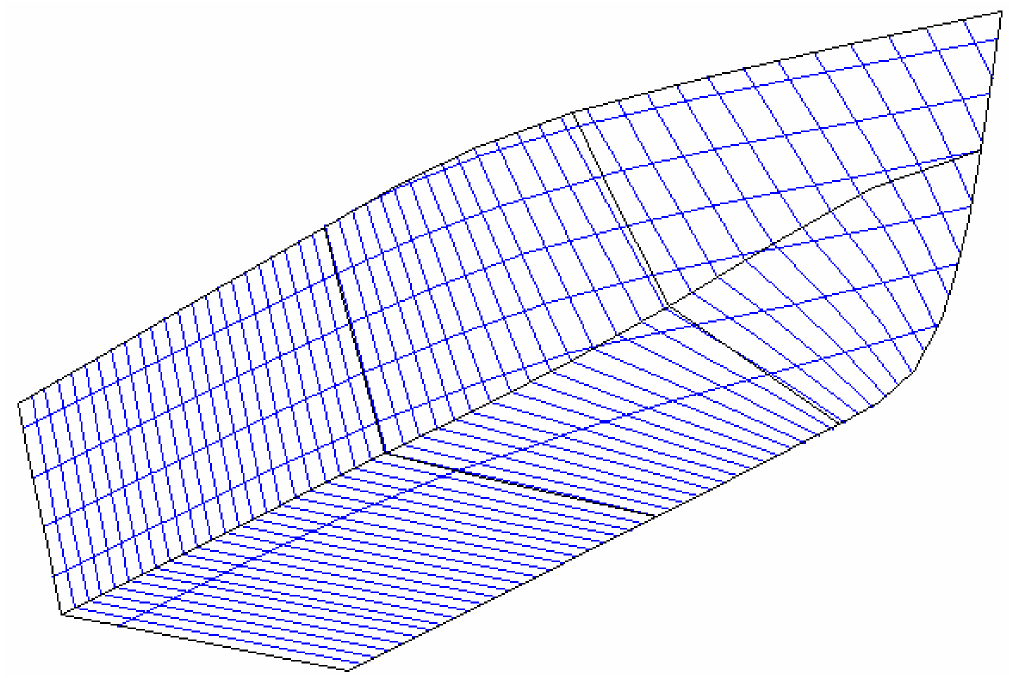


Figure 22.4, the surface generated by the Planing Hull Wizard.

The container ship wizard uses the following parameters, however, it does not provide diagrams to locate each parameter:

Gross Dimensions: Length Overall, Loa Beam Overall, Boa Deck Height, Hd Bilge Radius, Rb	Hull Features: Forward extents of PMB Aft extent of PMB Fwd extent of FOS Aft extent of FOS
Hub Features Aft Location of hub, Lh Centreline height of Hub, Hh Radius of Hub, Rh	Bow Features Location of stem/bulb intersection, Lsb Height of stem/bulb intersection, hsb Location of bulb tip, Lcb Height of bulb centreline, hcb
Transom Features Height of Transom, Ht	

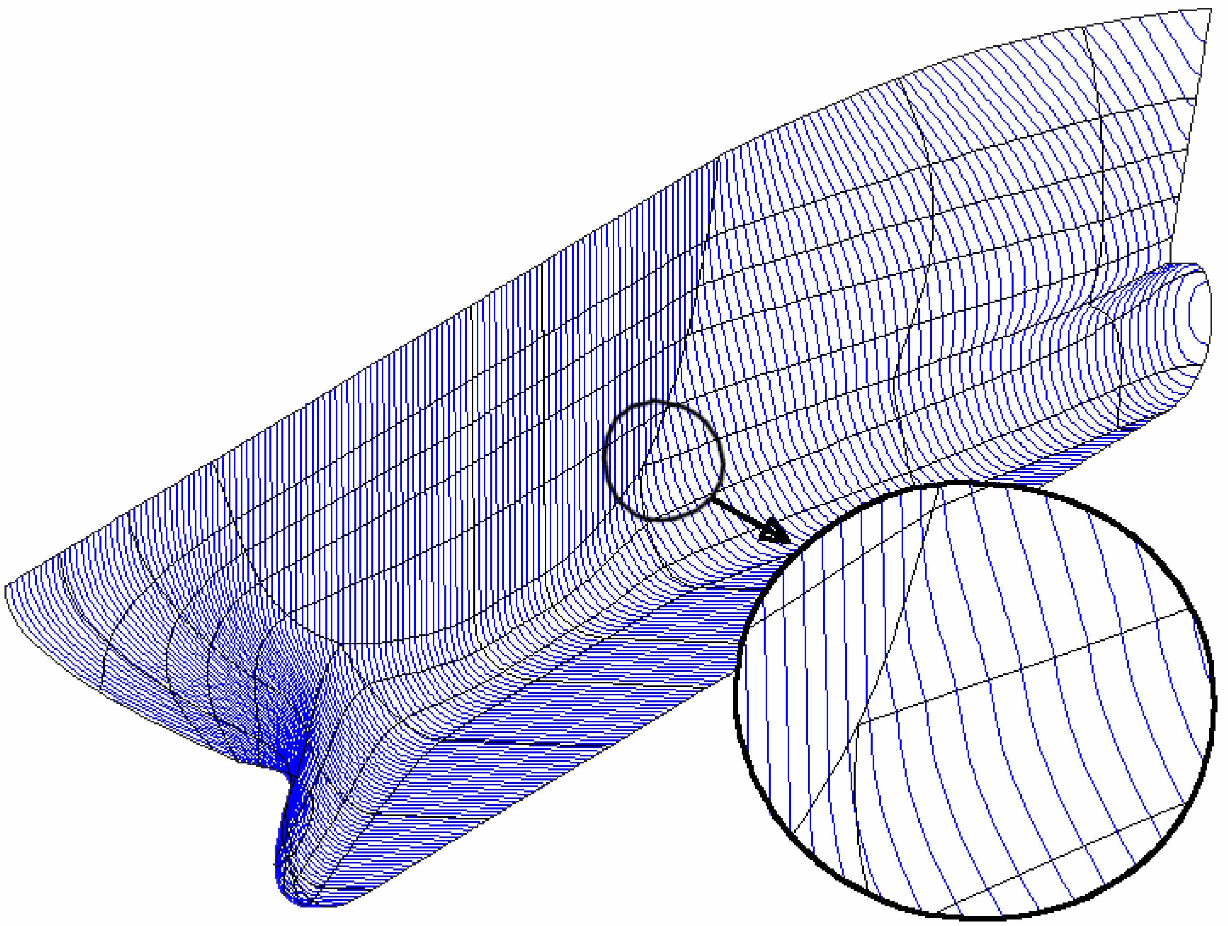


Figure 22.5, the generated by Container Ship Hull Wizard. Note: Unfairness in the surface.

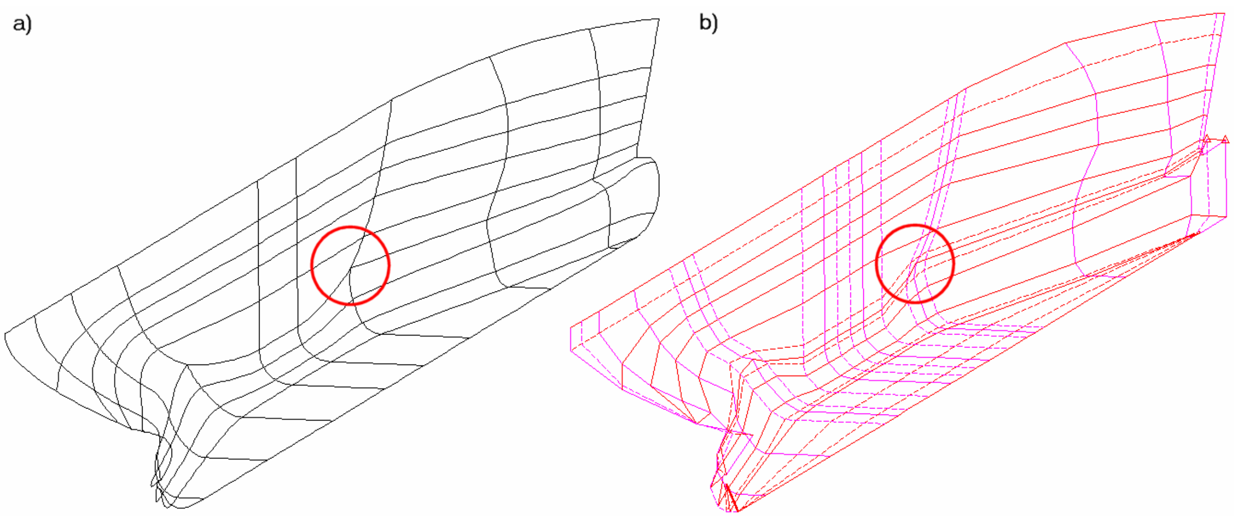


Figure 22.6, (a) Parameter lines and (b) Control Polygon Mesh of the container ship hull from Figure 22.5.

On initial inspection the hull wizard has produced a good hull, however, the sections do not seem to be fair around the transition to the side flat, (Figure 22.5). A close look shows that there is a region of high curvature along the flat of side curve, possible a line of first order discontinuity along the flat of side curve and there is a knuckle along one of the parameter lines. A view of the parameter lines only, (Figure 22.6a), shows that there is also a knuckle point on the flat of side, this appears to be related to the increased number of definition points in the control polygon, (Figure 22.6b). It appears that discontinuities have been used in the definition to make sure the surface forms features such as flat of side and flat of bottom, this generally creates an unfair hull form and is therefore undesirable. A review of the hull definition in FastShip did not indicate if discontinuities were used to create this surface, as the software could not provide this information.

FastShip features a scripting system allowing the automation of some of the editing processes. The scripting system is based on a programming language called PERL (Programming Extraction and Report Language). PERL is a full programming language and has all the usual traits including an unnatural syntax. Consequently, most users are unlikely to develop scripts directly. The scripts are most likely to be used for creating complex surface parts, within the Hull Wizards and in the FastGen tools. FastGen creates a new hull from a parent hull form through the modification of the main dimensions, the midship section coefficient and the Section Area Curve. FastShip can also record the user’s interaction with the program into a script if an operation is required repetitively.

FastShip may have some interesting features but the design of the interface is very poor. FastShip has been developed from a command line system and the toolbar buttons have generally been used as short cuts to enter commands. The current version of FastShip has too many buttons, many of which are seldom used and other important actions are not included in the tool bar. FastShip gives very little information on the current state of the program. The properties of surfaces cannot be reviewed to find out if there are any discontinuities. The positions of control vertices are not listed and can only be found by clicking on individual vertices. Furthermore, there is no indication of the position of the mouse cursor in the editing space.

The editing features of FastShip are not user friendly. Vertices are edited in an unusual way for a Windows program. The first click of the mouse button selects a vertex for edit. The location of the vertex now follows the mouse until the mouse button is clicked again. The editing system has quite a large tolerance for vertex selecting and the user can often initiate an edit operation by clicking in an empty part of the screen. In this case, the user must now find the option to cancels vertex editing. It is generally common for the Escape button cancel editing, however, in FastShip, one of the many toolbar buttons is used to accomplish this task. Moreover, this situation is exacerbated because the vertices of all the displayed surfaces can be edited.

The many installations of FastShip show that designers are prepared to use this software. However, the interface not easy to use and does not facilitate the design of an accurate and fair NURBS surfaced hull. It is unlikely that larger vessels could be created efficiently with this software package, as the design process requires a level of quality, precision, flexibility and performance from tools to remain within deadlines. For smaller projects, the lack of flexibility within software is likely to impair the design process and restrict the final design solution.

22.4. Multisurf – Aerohydro Inc.

Multisurf [58] is an a surface design system from Aerohydro Inc. Although developed for hull surface design the package does not provide many naval architecture type functions. Despite this, the software has been used to develop such high profile vessels such as the America's Cup Yachts *Stars and Stripes* and *Black Magic*. Aerohydro was founded in 1973 by Dr John Letcher to develop marine applications for naval architects and boat builders. Aerohydro have produced more than one hull design product. From Fairlines, which creates yacht hull shapes from cubic spline longitudinals that interpolate user defined sections to the current product, Multisurf, which provides a design environment where many different types of surface formulations can be used to develop a hull form.

At the core of Multisurf is the Relational Geometry Kernel. Dr Letcher has pioneered relational geometry, writing papers [47] and owning patents on the concept. Relational geometry is an object-oriented, relational surface modelling system, which allows complex 3D surface models to be built from a hierarchy of points and parametric curve and surface entities. The relational geometry system is a very powerful approach and many other surface development tools use definition techniques which are similar.

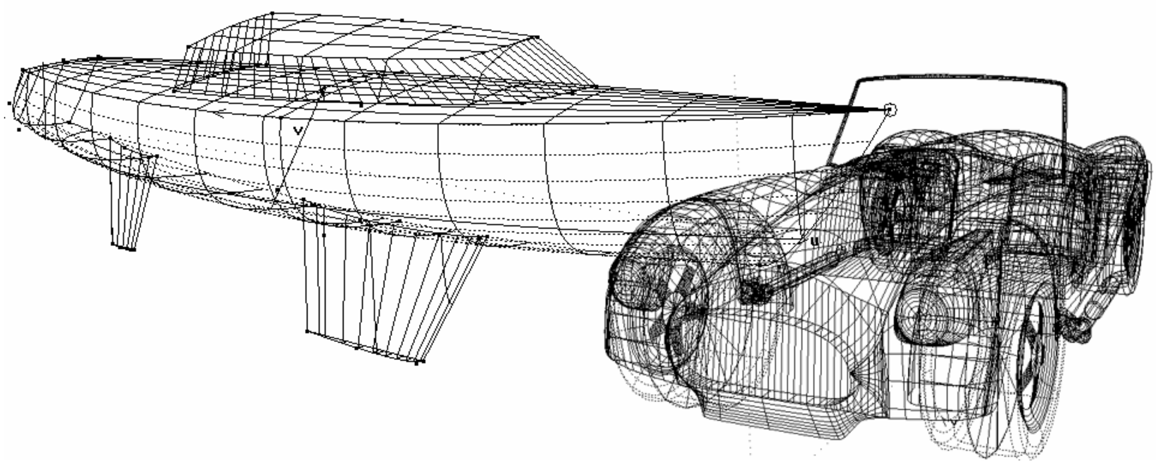


Figure 22.7, geometry designed using Multisurf.

The interface to Multisurf is very simple. As with the previous hull design packages, Multisurf starts with a black screen although an axis is displayed showing the orientation of the view. To add an entity to the design it is selected it from the Create menu, (Figure 22.8). As many of entities are available, it can be difficult to find the right entity to start with. All of Multisurf entities are found on the Create menu and its substructure. When entities are created, the system takes a very exact approach. A dialogue box is displayed for each new entity allowing the user to directly adjust all of the numeric and other properties. This may be necessary for the components that rely on relational geometry for their shape. However, for simple points it is not necessary. As you require many simple points before you can start creating more complex entities, the initial design process is slowed, preventing the definition from progressing in a fluent and interactive manner. Once points are created, they can be manipulated directly using the mouse.

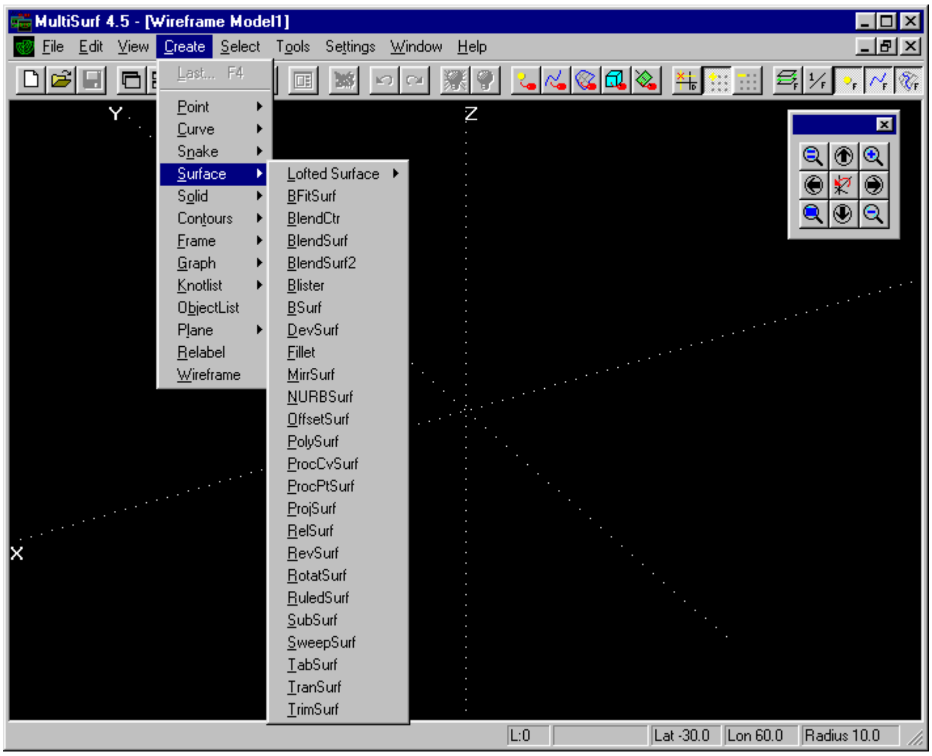


Figure 22.8, The Create menu showing all possible surface entities.

Despite the powerful relational geometry system used in Multisurf, it is also the packages greatest weakness. The relational geometry system is the only way complicated geometry especially surfaces can be created. Subsequently, the structure behind the relational geometry system is so prevalent that the mathematical functionality of the behind individual entities is hidden. The control polygons behind NURBS curves and surfaces are not displayed, (Figure 22.9), and you cannot tell how the surface will react when an individual point is manipulated. Moreover, if more that one curve or surface is shown on screen at once you cannot tell which points belong to which entity. This problem becomes more apparent as the screen fills up with geometry and it becomes quite difficult to select the part you want despite the software giving you a choice of object names based on the screen distance to the location of the mouse selection point.

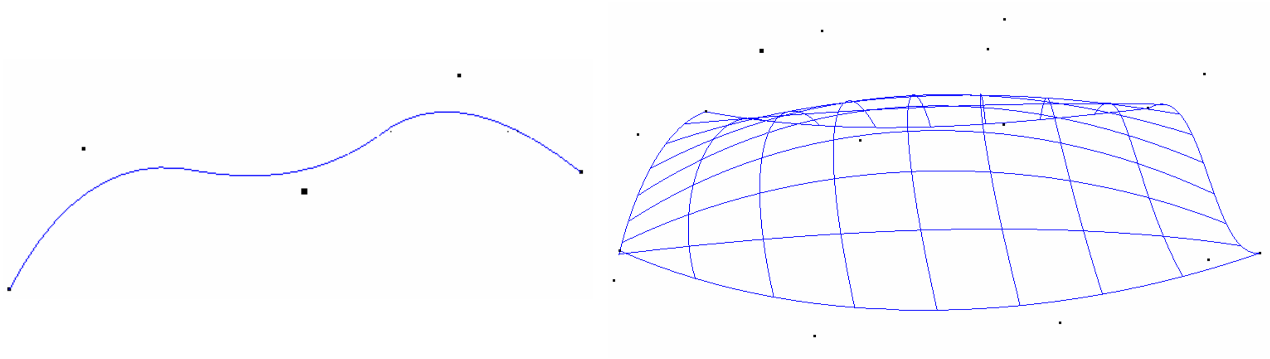


Figure 22.9, Multisurf does not display Control Polygons for Curves and Surfaces.

As there are difficulties managing entities reliant on a large numbers of points, those requiring fewer points or using simpler relations are more attractive. It is interesting to note that the software provides no examples of NURBS surface use and many examples of developable and ruled surfaces, as these entities can be easily created from two curves under the relation geometry framework.

With some initial effort, Multisurf can be used quite efficiently to design a hull surface. It is most appropriate to use the package when there are likely to be many changes to the hull surface. In the cases of the America's Cup Yacht designs, Multisurf was used to develop a set of systematic hull forms, which could be analysed to find the one with the best performance. The relational geometry system aids this type of development as one point can be used to drive changes on the whole surface shape resulting in a more parametric approach to the design.

Multisurf handles the relationships between the geometry well, the software updates all dependant geometry immediately when interactive modifications take place. However, there are limits to this feature and when the number of entities becomes high the operation of the program is slowed.

Unless a particular design operation is planned which can take full advantage of this type of relational geometry implementation, Multisurf is not an ideal piece of software to be used flexibly for the design of a hull surface. Too much effort has been put into the realisation of the relational geometry system and to the extent that the user-interface makes it difficult for the designer to create the desired surface shape. Multisurf appears to be more of a forum to demonstrate the concepts of Relational Geometry.

22.5. Maxsurf – Formation Design Systems

The Maxsurf [27] suite of applications has been available since 1984. The package provides programs, which can used to design, perform basic calculations and produce constructional details of a vessel. Maxsurf is widely used by many companies to design pleasure craft up to the size of workboats and small ships. Demonstration copies of the individual programs in the Maxsurf suite can be downloaded from the company's website.

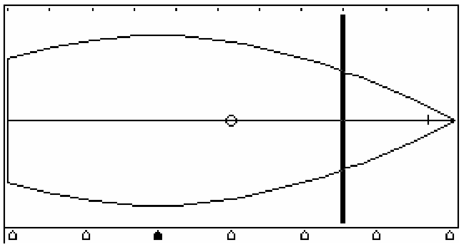
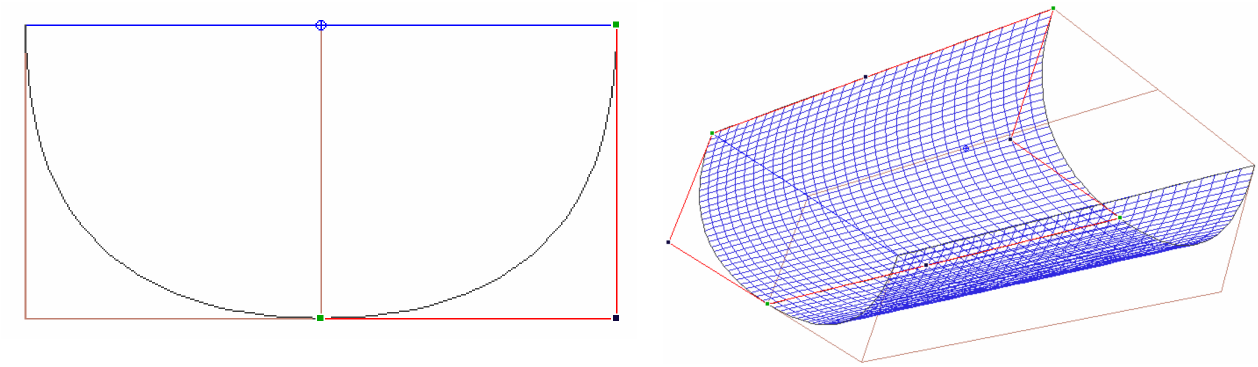


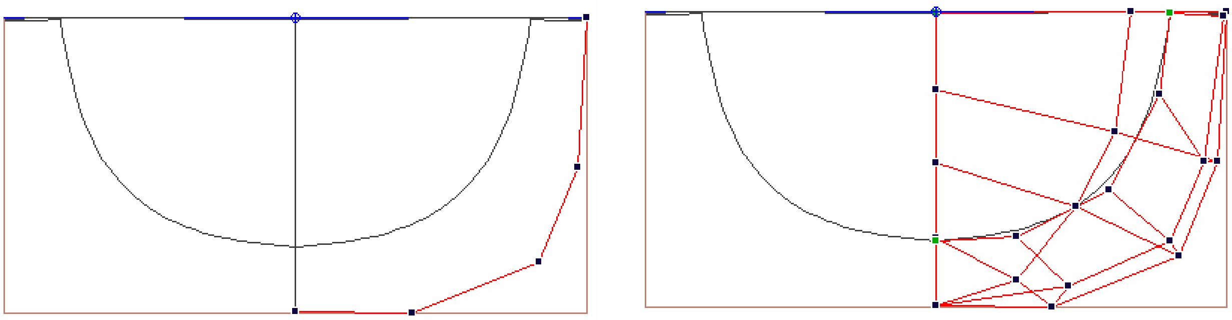
Figure 22.10, The tool used select control polygon columns.

Compared to other hull design packages the range of entities available within Maxsurf is very small, all entities are based on the NURBS surface. This has allowed Formation Design to concentrate on providing a user interface that gives the designer one of the best possible tools for creating hull form surfaces using the NURBS representation. The design of a NURBS surface is based around the control polygon. Maxsurf provides a new and unusual way of dealing with the control polygon, (Figure 22.10). This tool allows the user to select one column of vertices from the control polygon to view and edit on screen. The user can then only modify the vertices of this column without being distracted by the position of vertices in other columns of the control polygon. This approach gives the user a better sectional appreciation of the surface, which is especially useful when trying to design a hull shaped surface. Maxsurf also provides a table that allows users to modify the vertices of the surface directly.

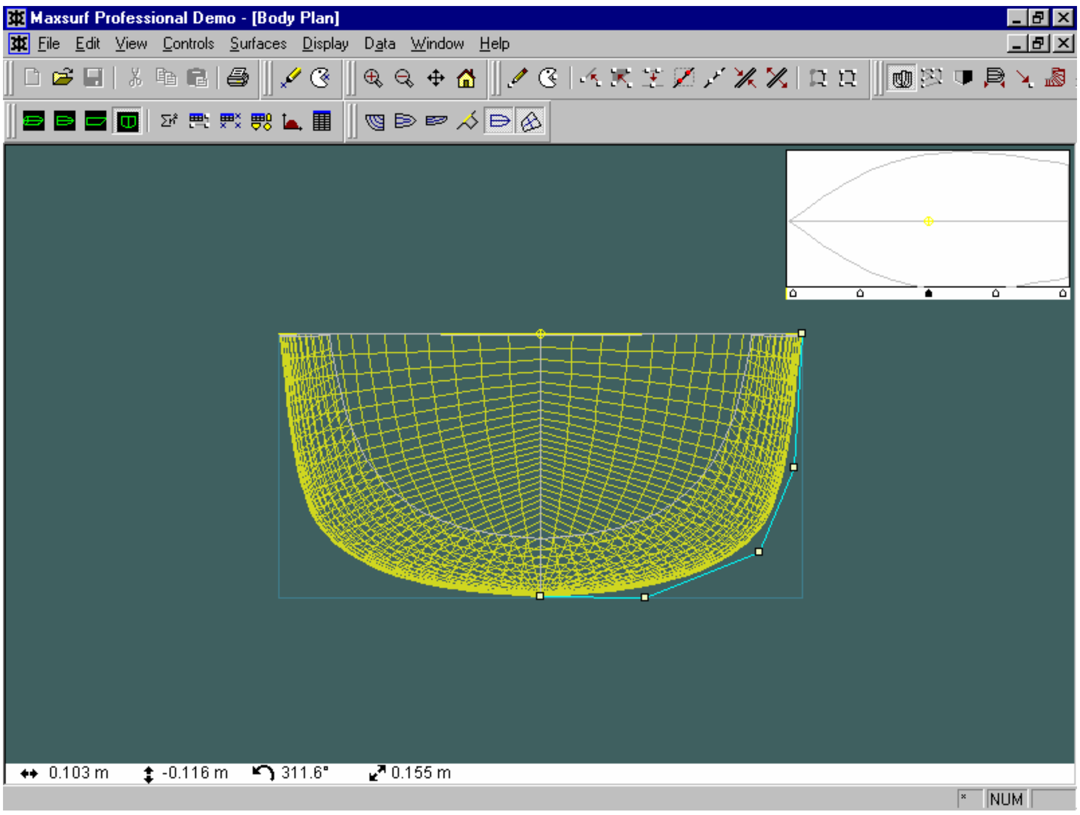
The initial surface created by Maxsurf is a semicircular mesh consisting of three rows and three columns of control vertices, (Figure 22.11a). The surface is defined only on one side of the vessel but is shown mirrored on the screen. The surface can be further subdivided to increase the number of control vertices. The editor has two modes for editing the control mesh. The complete mesh can displayed allowing the user to edit the whole surface at once. Alternatively, individual control columns can also be displayed so that the user edits the hull by controlling the sectional shape, (Figure 22.11b). Once the hull has been edited, the user can display surface parameter lines or contours, (Figure 22.11c).



a) The initial hull surface, body plan and perspective view.



b) Single control columns or the whole mesh can be edited.



c) Surface parameters. The control mesh column selector is shown in the top right of the window.
Figure 22.11, creating a hull in Maxsurf.

During the development of a hull form, the surface can be analysed using section area curves and a template system similar to a spreadsheet allows the user to define and customise the set of hydrostatic calculations. The interface provides all the tools that would generally be associated with the design of NURBS surfaces giving control of the surface curvature and the addition of knuckle points etc. A small level of relational geometry definition is incorporated into the software to allow the edges of surfaces to be bonded together. Consequently, deck surfaces, for example, can be accurately attached to the hull surface with ease.

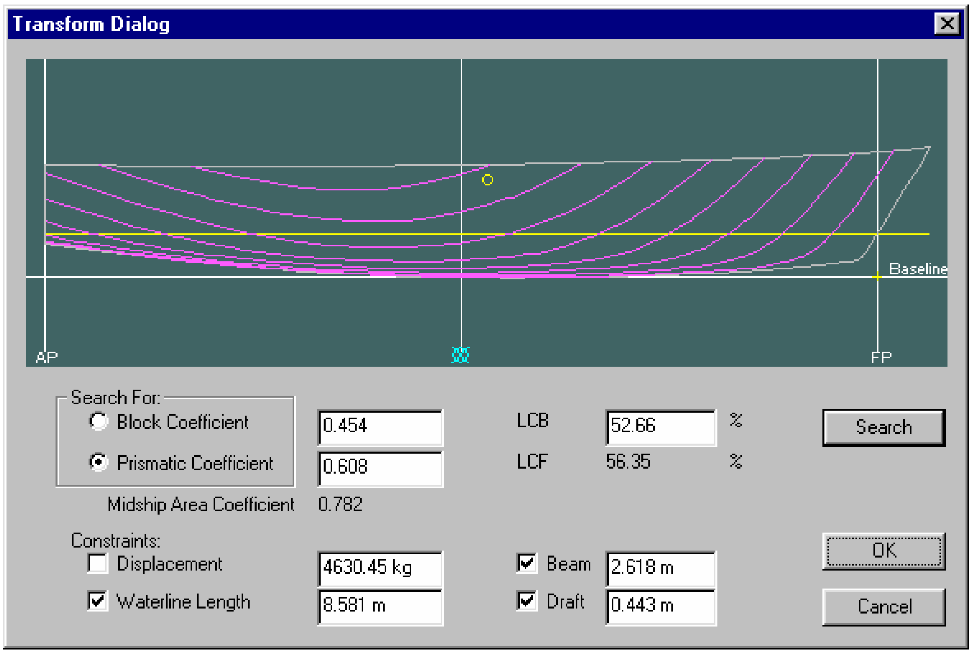


Figure 22.12, the Transformation Tool.

Maxsurf is capable of performing geometric transformation on the hull in many ways. The dialogue box shown in Figure 22.12 is capable of scaling and performing volumetric changes to the hull form.

Maxsurf is a very good program for hull surface design. Despite the fact that it does not have a large range of features, the interface is well designed resulting in a powerful and flexible tool for designing and controlling NURBS represented hull forms.

22.6. Autoship – Autoship Systems Corporation

Autoship [28] systems have been developing their suite of Naval Architecture software since 1980. Over this twenty year period the company has managed to build up a customer basis of over 1,700 installations in fifty countries. The Autoship suite is package that provides all the basic tools for the design and analysis for vessels up to medium sized ships. The software package covers activities such as hull design, hydrostatics and stability, structure and plating and power prediction.

The software supports points, NURBS curves and surfaces and can connect these entities together using a relational geometry technique. Unlike some of the other hull design systems reviewed here, Autoship defines a systematic approach to the creation of a hull surface using the hierarchical structure of relational geometry.

The first step of hull design is to define the scale of the vessel. Autoship as with other software packages starts with a blanks screen, which gives no real feed back of the scale of the design space. The first steps of design are to define the limits using point entities at the extremities of the vessel, (Figure 22.13). Using a relational geometry framework, curves can be attached to the points and their shape adjusted by the manipulation of the control vertices, (Figure 22.14a).

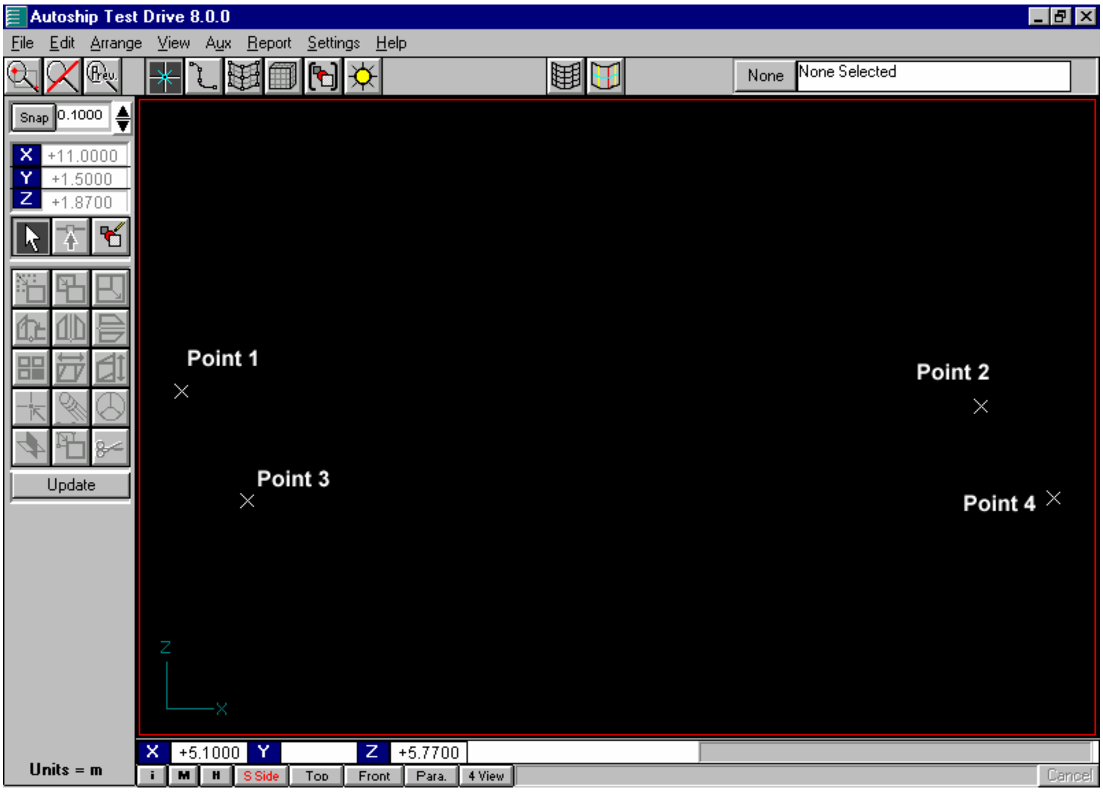
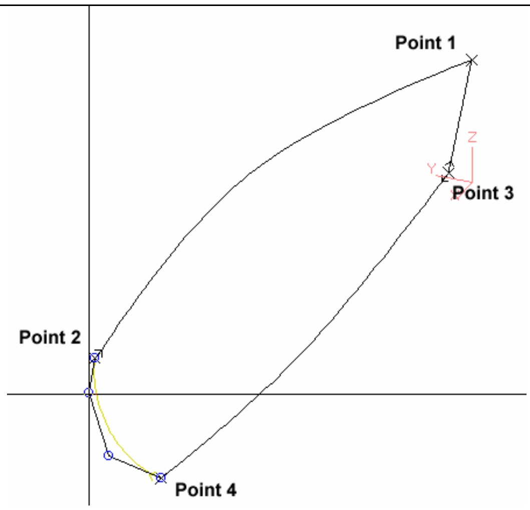
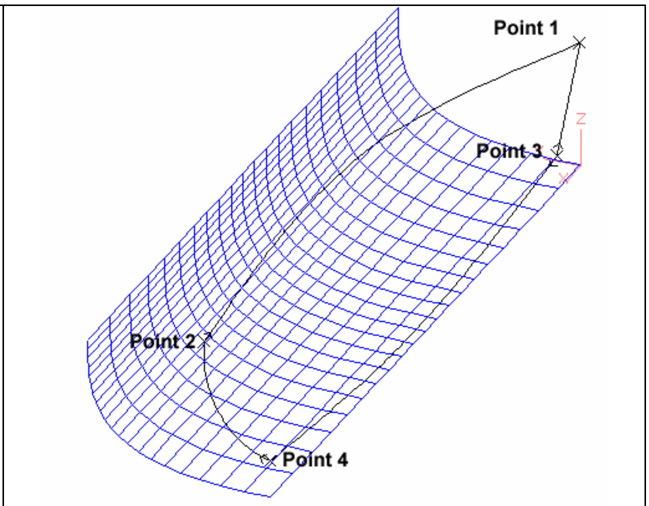


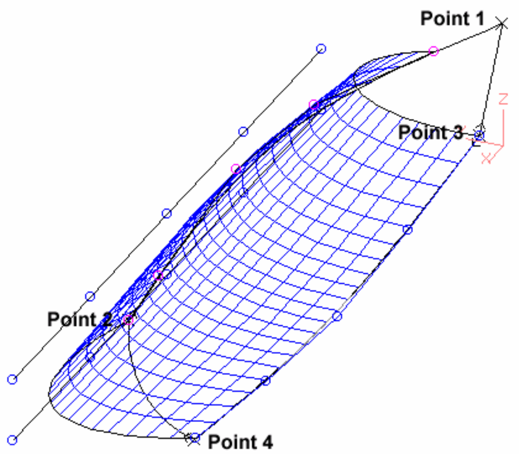
Figure 22.13, the initial definition points for a yacht hull surface.



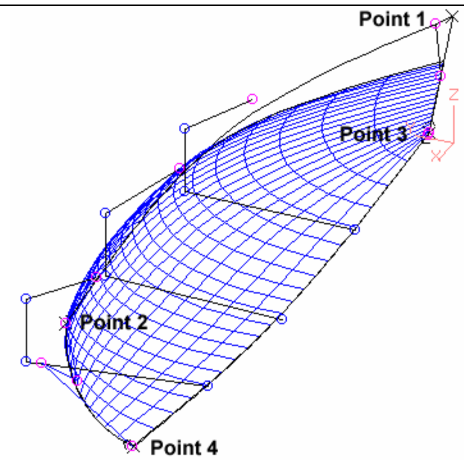
a) Curves attached to definition points.



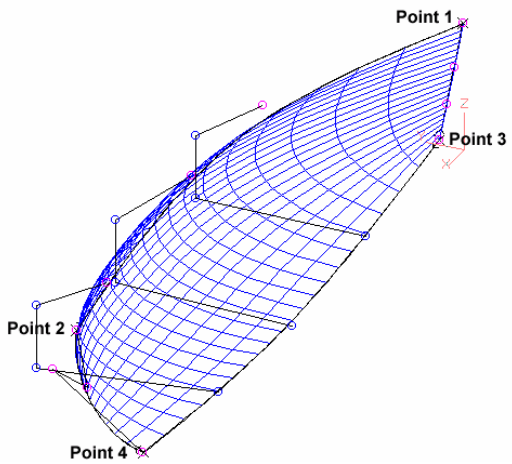
b) A new surface is added to the definition.



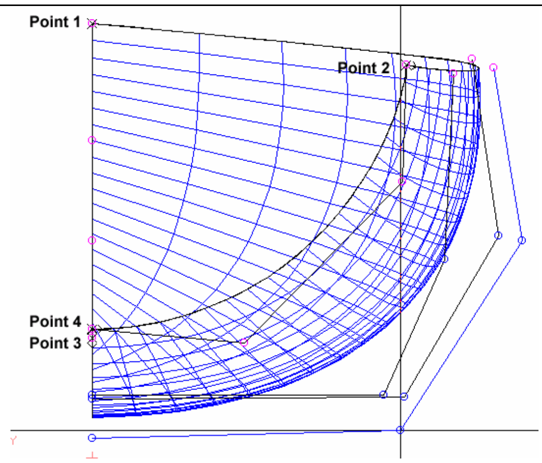
c) Upper and lower surface boundaries attached to curves.



d) Forward and Aft boundaries attached to curves. Note that upper boundary is now unattached from the curve.



e) Bow control vertices a adjusted to reattach upper boundary to curve



f) Control vertices are adjusted to form the hull shape

Figure 22.14, the stages of attaching and manipulating curves and a surface to form a basic yacht hull.

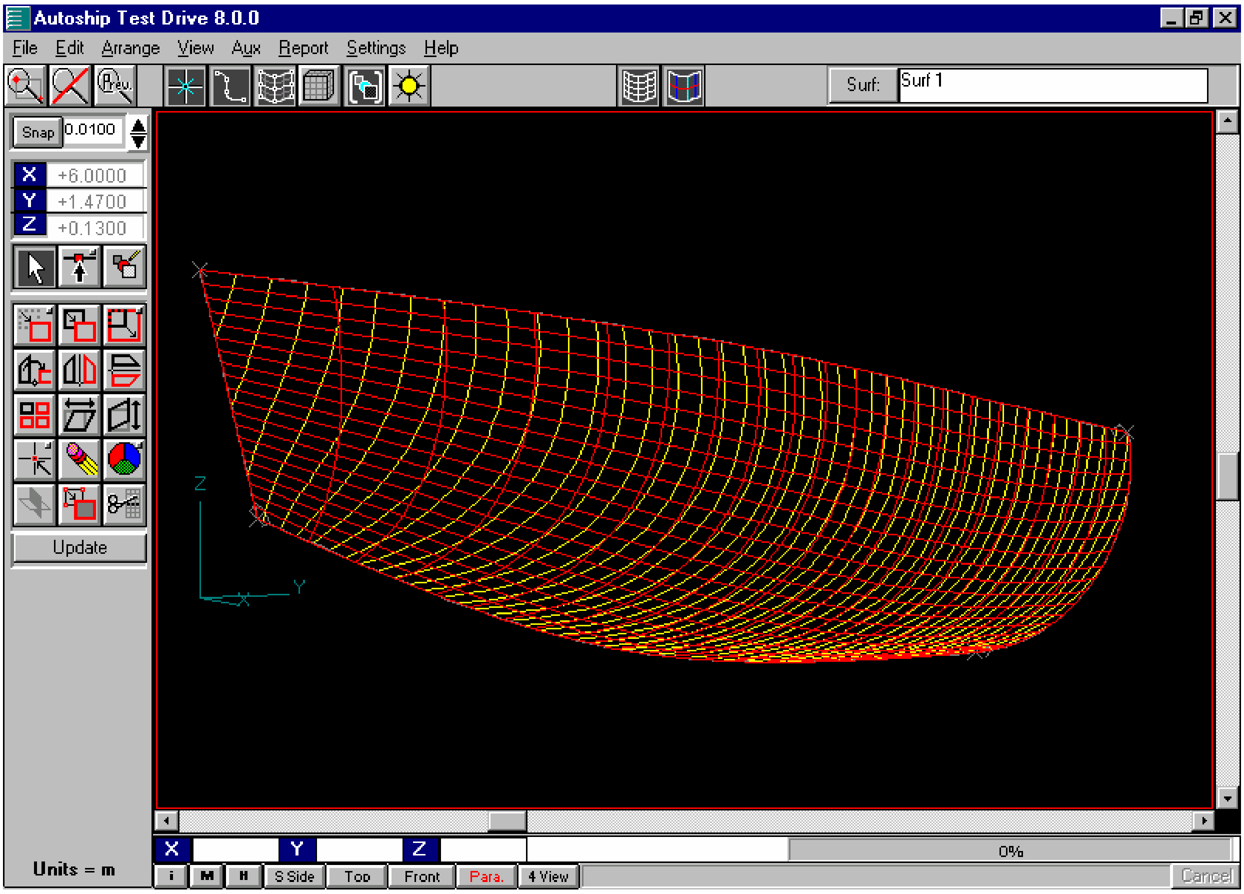


Figure 22.15, the sections and parameter lines of the yacht hull created in Figure 6.12.

The Create Surface dialogue box, (Figure 22.16), is used create a curved surface of size similar to the curve structure. The surface boundaries are attached to the curves in Figure 6.12c and Figure 6.12d. Notice that when the surface attaches the stem of the hull the upper boundary becomes unattached. This appears to be a bug in the software and it can be rectified by reordering the control vertices on the stem surface boundary, (Figure 6.12e). The shape of the surface can now be controlled using the surface control mesh, (Figure 6.12f). The completed hull form, with sections is shown in Figure 6.13.

Points and control vertices are modified using the mouse. However, the modification procedure is not as interactive as it is in other systems. The user selects a point and then drags the cursor until the new location of the point is reached. The geometry only updates once the mouse button is released. This system is difficult to use and is not very accurate, as changes in geometry resulting from modifications cannot be judged. However, Autoship also allows the user to interactively modify points and control vertices with the keyboard arrows keys. This is a particularly accurate method as the distance moved by a point during one key press can be accurately controlled and the hull geometry is updated after every press. For small modifications to adjust surface fairness, this feature is invaluable.

If the surface is part of the relational geometry hierarchy, a change in the location of any of the four initial points will propagate through the curves to the surface boundaries. However, changes in the geometry of points or curves will only influence the surface boundaries and the control mesh must also be adjusted to maintain fairness and shape in the hull surface. Further surfaces can be added to this definition such as the deck and transom. The beauty of the relational geometry structure is that any changes to a curve will be reflected in any of the surfaces attached to the curve.

Autoship supports other relational geometry features such as points attached to lines and lines within surfaces. The names of entities are required to create relational links. As Autoship automatically names new entities, the user does not have to explicitly remember any. Therefore, it is necessary for the user to investigate the names of the entities before a linking operation to be sure that the correct relation will be created. The relational geometry system could be improved by allowing the user to select the geometry for a relational link on-screen.

Once the hull has been created, it can be analysed through coloured rendering based on shape or different curvature analysis techniques. A brief set of hydrostatics can be calculated and the results are displayed in a spreadsheet. This would be quite useful for other calculations were it not for the fact that every time the hydrostatics are updated a new set of results appear below the last. For deeper analysis, the hull can be saved in a format compatible with the other software in the Autoship suite.

The modeller within Autoship is quite powerful and with a reasonable amount of time, complex ship shapes can be created. However, the interface is the weak link in the programs operation. The program provides the tools to access and operate on every feature of the geometry. However, tools are presented in a way that makes the system very uncomfortable to use. Most features are crammed in too close together. For example, the dialogue window used to create surfaces has options to allow the user to create a surface in 11 different ways, (Figure 22.16). Consequently, the more complex surface creation operations are arranged in such a way that it is difficult to understand how the software intends to create the surface. As a result, some of the operations especially those using the relational geometry system, do not always work as intended.

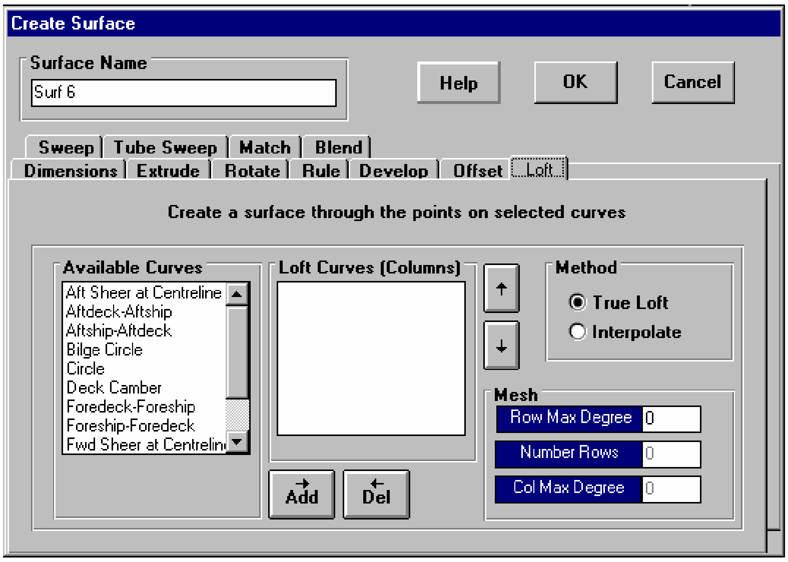


Figure 22.16, the dialog box used to create surfaces in AutoShip.

At a fundamental level, the user interface is badly designed. It does not follow the ideals of modern interface design. The controls of the dialogue box shown in Figure 22.16 are too close together. They are not well aligned and there are too many boxes around the controls. Other interface problems can be found on the main screen where some of the buttons have two operations, one when the left mouse button is pressed and one when the right is pressed. Many of the screen controls have been programmed especially for the software and although they are similar to some of the standard Windows controls, they function quite differently. There have even been occasions in the previous versions of Autoship where some of the buttons have disappeared off the side of the window.

The Autoship suite is ideal for the small design office dealing with small to medium sized vessels. Autoship is best used on hulls that do not require a lot of surface modification to maintain a fair shape, as the software does not allow efficient editing of large number of control vertices. Autoship has many areas where the design of the system could be improved, mostly in the interface. However, compared to other packages Autoship appears to be one of the better medium sized Naval Architectural packages based on its apparent popularity.

22.7. DFform and ShipGen – DEFCAR Ingenieros, S.L

DFform and ShipGen [29] are hull surface design tools of the Naval Architecture package developed by DEFCAR Ingenieros. The other components of the package cover Hydrostatics and Stability, Shell Expansion, Structural Steelwork and Nesting and NC Cutting. DFform is the hull surface design package allowing interactive manual modification. ShipGen is a tool that used to generate new hull forms by making transformations using parent templates.

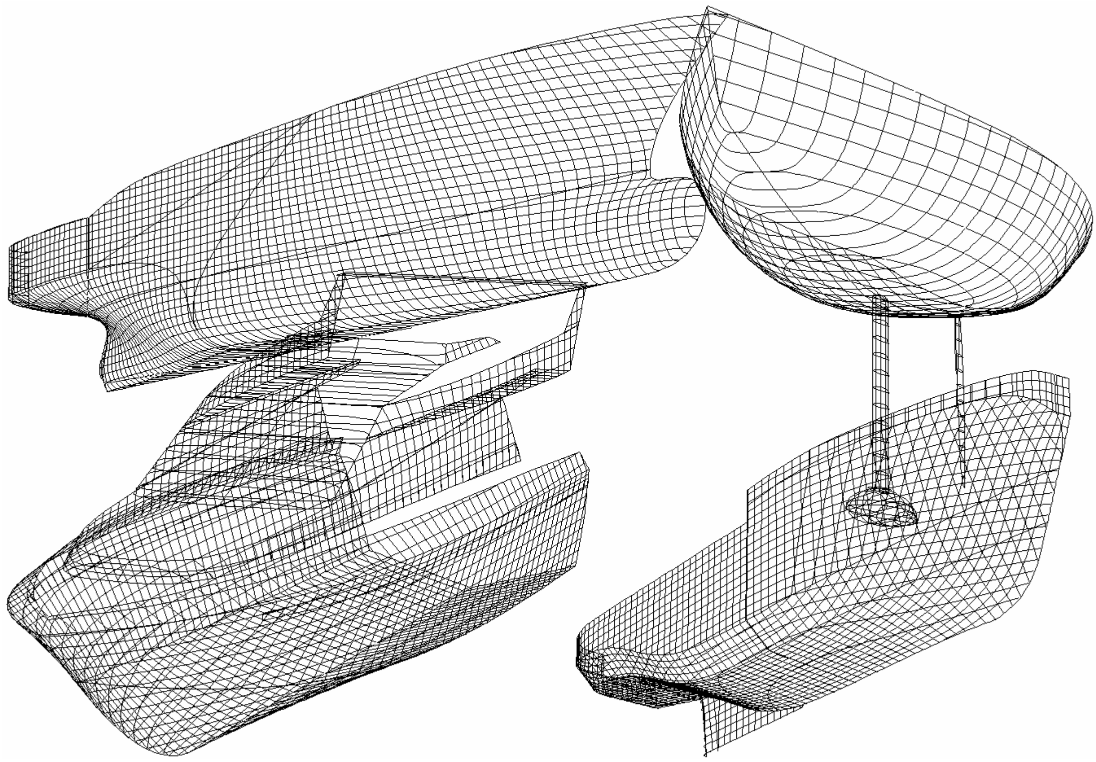


Figure 22.17, Hull form surfaces developed with DFform

DFform uses Bézier surfaces to represent the hull form. Figure 22.17 shows some examples of hull surfaces designed with this software. The hull surface is created using a structure of Bézier surface patches, (Figure 22.18). Each patch can be used to represent properties of the hull surface such as flats and patch boundaries can be used to represent knuckles. When hull flat boundaries and knuckle lines lie on surface patch boundaries, the modification of these features reduces to the manipulation of a Bézier curve instead of the surface.

This software has been created with three different types of hull creation and modification processes in mind, Identification, Transformation and Design.

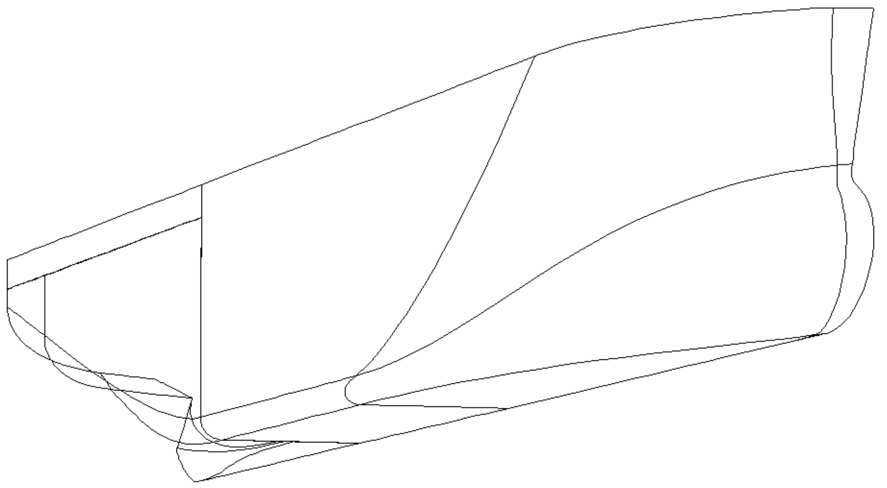


Figure 22.18, the patches of a DFform hull surface.

Identification is when an existing hull form is to be inserted into the software, a frequent task in practice. Reference points on the existing hull form are introduced into the software and the system, helped by the user, adjusts the surfaces to the reference points. As the control polygons of Bézier surfaces do not lie on the surface, it can take a considerable amount of time to manually adjust the definition vertices, until there is a good surface fit with existing hull data. Any help that the software can give in the process is always welcomed.

Transformation is when a new hull form is generated from an existing hull form using simple transformation techniques. Transformation of existing hull forms is a common process, however, only the basic geometric transformations of Move, Scale and Rotate, are provided in this software. A fully functional transformation system needs to provide operations to modify the section area curve and the midship section area.

Design is when a new kind of hull that does not have a similar shape to a previous project is to be created within the software. The software does not provide any useful facilities for helping the user generate a new hull form. Individual Bézier patches can be created from scratch by entering all the vertex data into the software. However, as it is necessary to build large structure of patches modelling the properties of the hull, it is a time consuming task. An alternative solution is to take an existing project with a surface structure similar to the one desired and manually modify the surface until the shape of the new hull form is reached. As this can be interactively achieved with the use of the mouse, it is a more intuitive approach for the designer. Both methods largely consist of data manipulation, a process that the software could handle much more efficiently than the designer.

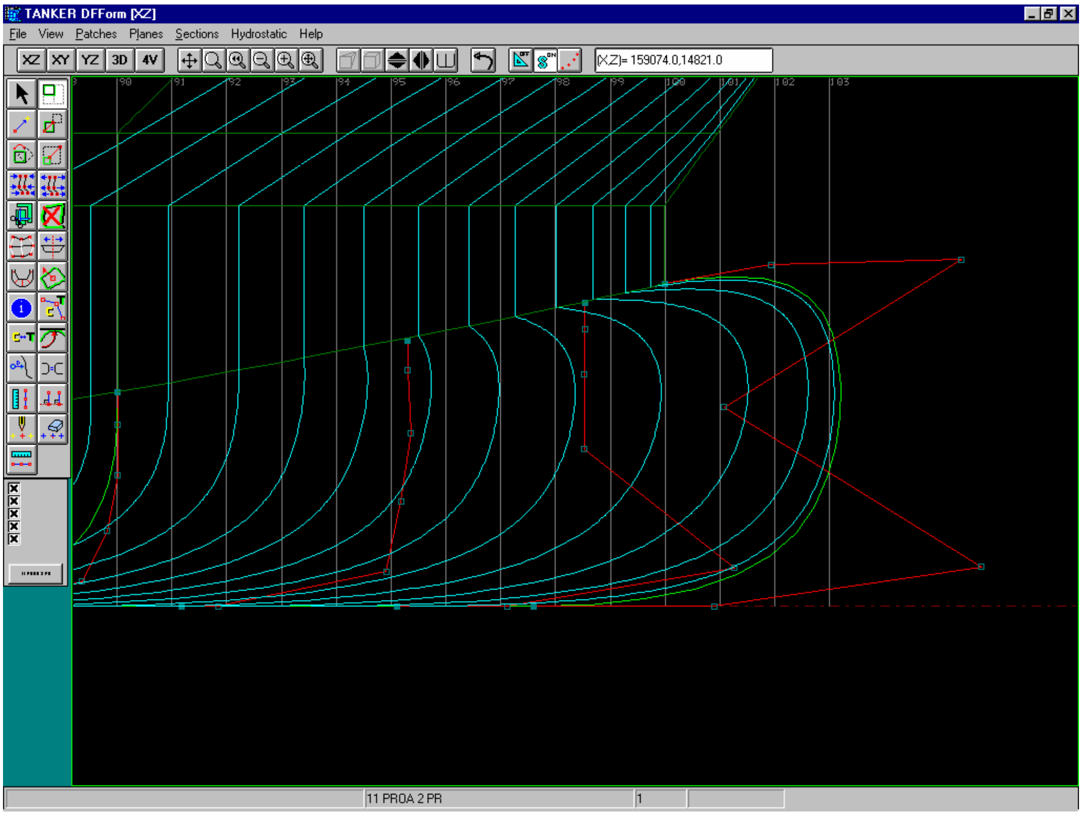


Figure 22.19, the graphical Interface of DFform.

DFform gives the user full control of the patch surface shapes and provides many ways of editing the properties of the surface. Figure 22.19 shows the DFform editing the control polygon of a bulbous bow patch. DFform can display the control polygons in three ways giving the user the ability to edit the patch borders only, the entire control polygon net or just selected control columns. DFform also provides tools to implement medium levels of control over the hull surface. One useful tool is used to set patch boundaries to be perpendicular to the centre plane.

There is a good level of functions to analyse the results of hull surface modifications. The software can mark lines between the surface and the control polygon to show the points of maximum influence on the surface. This can be very useful when editing complex hull shape around the bulbous bow or shaft bossing. After a modification of the surface, the contours can be recalculated and drawn over the previous set of contours. This gives an impression of how much the hull changed since the last time the contours were drawn. Porcupine plots can be generated from each section to show the curvature of the hull. The hydrostatics of the hull can be calculated at any time of the design process as long as there is a valid hull on which calculations can be performed. The section area curve can also be displayed.

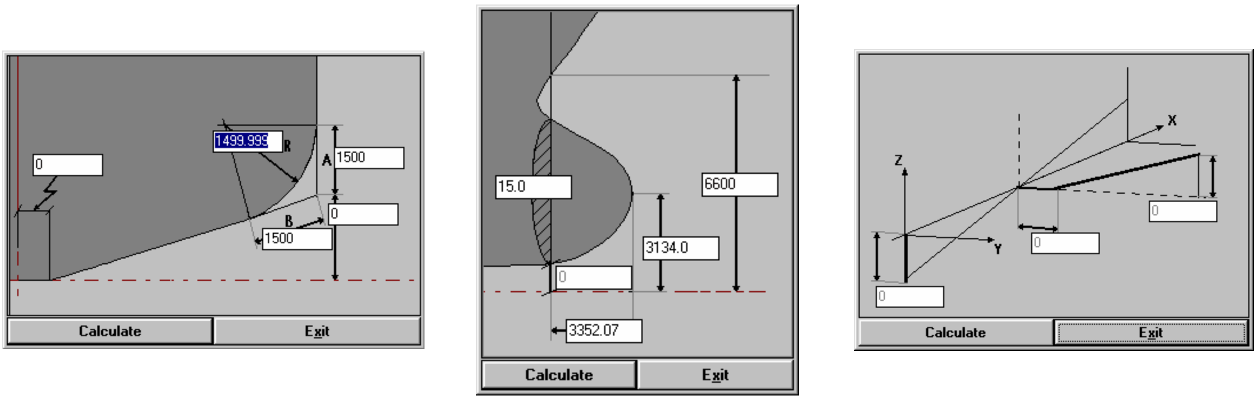
ShipGen is a hull form generation tool that implements some of the functions missing from DFform. ShipGen creates new hull forms by transforming parent hull forms from stored templates. Transformation functions for changing the global shape of the hull are provided along side parametric controls for changing the local geometry of the hull. ShipGen is only a hull generation tool and patches cannot be edited directly. However, the hull definition can be saved for editing with DFform or other systems supporting the available file formats once the transformation process has been completed.

ShipGen is capable of up to fourteen hull transformations depending on the hull type. Table 22.1 shows a complete list of the transformations available.

Global Hull Transformations	Local Transformations
1. LBP	8. Trim
2. Moulded Depth	9. Keel Half Width
3. Moulded Breadth	10. Rise of floor
4. Draught	11. Bilge radius
5. Parallel Middle Body	12. Bulb Area
6. Displacement	13. Height of the bulb maximum length
7. LCB location	14. Maximum Bulb length

Table 22.1, parameters controlling the ShipGen hull transformations.

The software can provide small diagrams for each of the parameters modify local hull properties. Examples of these diagrams are shown in Figure 22.20.



a) Bilge Radius Parameters b) Bulb Parameters c) Trim and Section Parameters

Figure 22.20, displays used to edit parameter in ShipGen

There are currently twenty-five templates available for ShipGen covering a range of vessels from tankers and Ro-Ro's to small fishing vessels. The user cannot create new templates and must contact DEFCAR to add a new hull form. There appears to be a lot of complex information in the template definition files besides the hull form patches, ShipGen may require further information besides the hull definition before it can implement the hull form transformation functions.

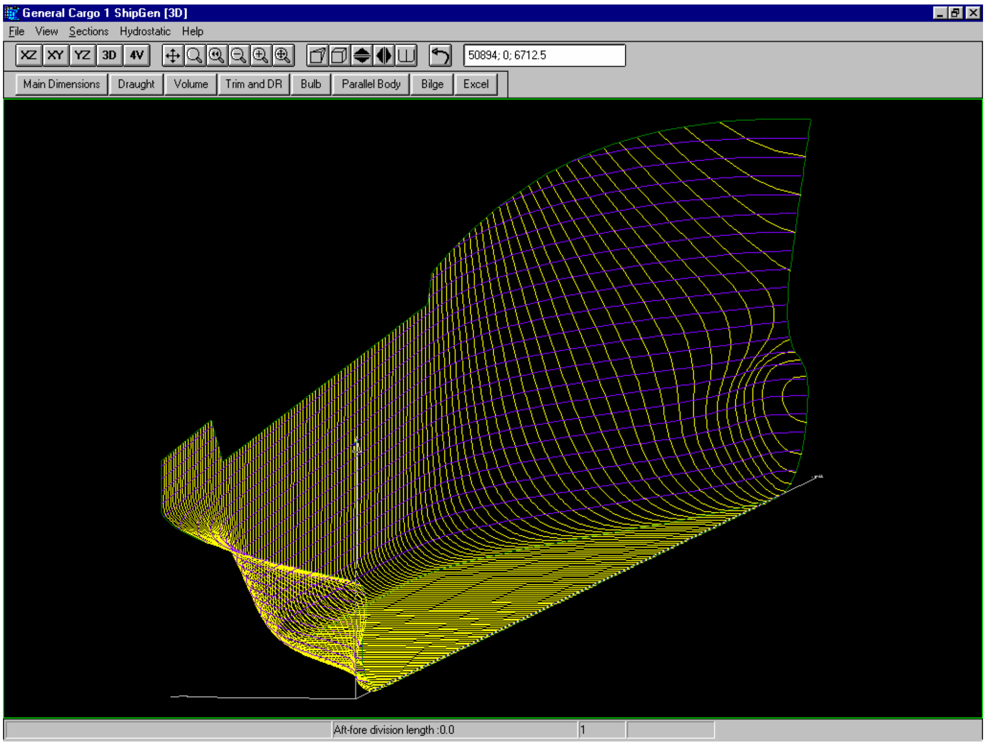


Figure 22.21, the user interface of ShipGen.

The look of the ShipGen user interface is very similar to that of DFform, (Figure 22.21). The software does not perform great range of tasks and therefore only requires a small number of tools functions to be displayed on the interface window, allowing the user to change the transformation parameters and the hull form view. Access to the contouring function is available from the menu bar. Like DFform, the software provides similar functionality to allow the user to view the hull surface contours before and after a hull transformation can be reviewed, (Figure 22.22).

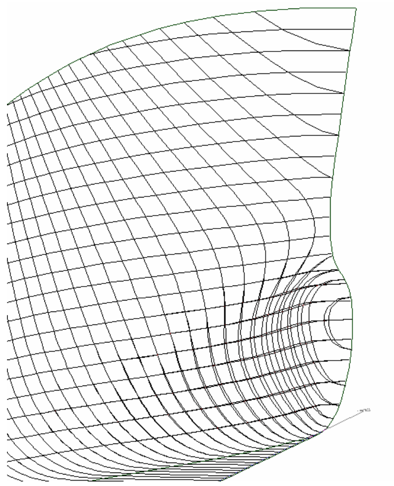


Figure 22.22, Hull lines showing changes to the bulb region of the surface.

ShipGen has a dynamic link to the Microsoft Excel spreadsheet. This feature could be very useful as it allows the designer to set up simple calculations, which could be used to select the size of vessel, based on cargo capacity for example. An interactive loop can be created where the hull design is interactively adjusted based not only on hydrostatic calculated by the hull design software, but also on custom calculations defined by the designer in Excel. Many other software packages would benefit from such a feature.

Both DFform and ShipGen are capable design tools and appear to be usable for the purposes of most modifications. DFform provides one of the best sets of tools for manipulating NURBS type surfaces compared to other Naval Architectural packages. The tools are simple to use and easy to understand. However, DFform does not have enough functionality to be used for the design of new hull forms as almost all of the tools are used to edit the surfaces at the vertex level. The level of data manipulation for a new design is likely to be so high that a designer would probably lose interest and effectiveness during the process.

ShipGen has an impressive range of transformation functions, however, it operates as a one-off program and can only use custom defined templates. It does not give the designer the ability to directly modify any shape of the hull surface. For these reasons this software would be rarely used. The DEFCAR package would benefit greatly if some of the functions implemented in ShipGen could be integrated into DFform software.

22.8. NAPA – Napa Oy

The Finnish shipyard, Wartsilä, began the development of NAPA [35] (Naval Architecture PAckage) in the late 1970's. As the package became a marketable solution a separate company, Napa Oy, was created to develop, market and provide technical support on the software. Today the software is one of the major packages being used for design, construction and evaluation of ship designs, especially passenger vessels. NAPA provides a complete range of fully integrated tools covering Hull Design, Compartmentation, Hydrostatics and Stability, Damage stability, Propulsion and Seakeeping and Steel Work design. NAPA is not a standard product modelling system and it cannot be used to design the intricate details of a ships outfitting. However, vessel data stored for calculations is presented in a similar style to standard product modelling systems.

NAPA is one of the few Naval Architectural packages not to use NURBS surfaces for hull definition, instead NAPA uses cubic patches and a custom formulation of cubic splines to define the hull shape. The definition system is very powerful and the hull definition can be controlled at all levels from the global shape of the vessel right down to intricate local details in the surface. Figure 22.23 shows some examples of different hull forms defined using NAPA.

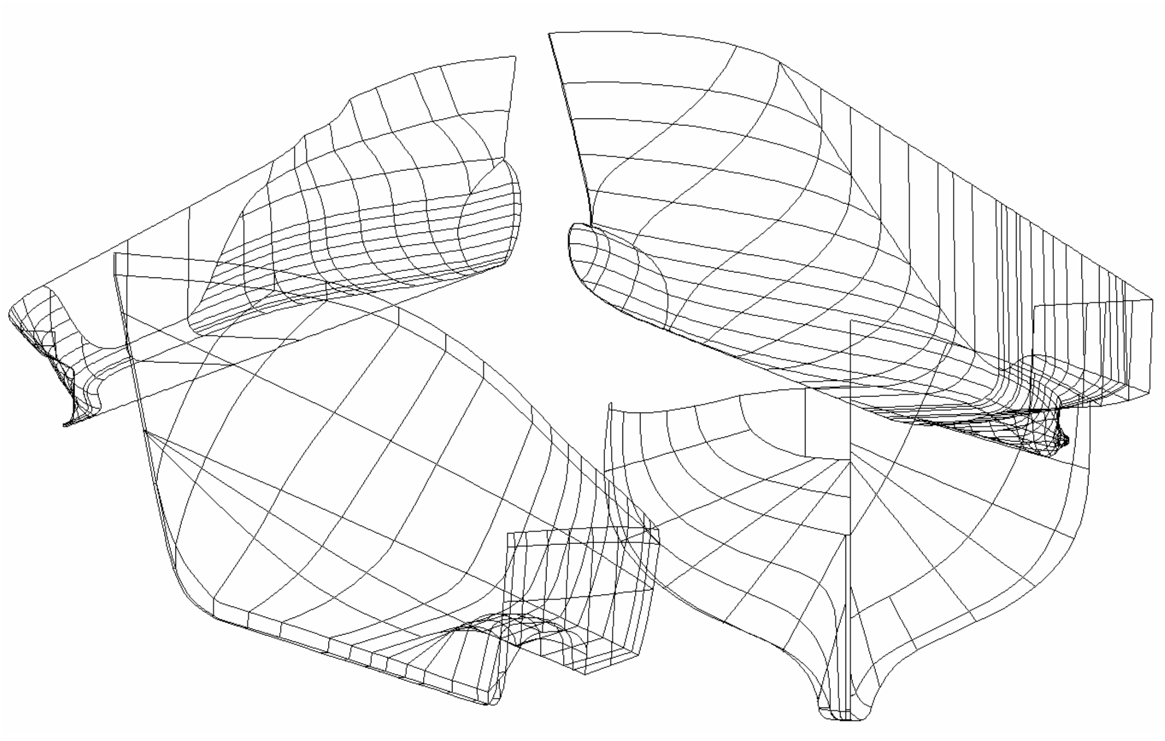


Figure 22.23, hull shapes created with the NAPA system.

The ability of NAPA to create complex hull forms is due to its unusual and powerful hull definition technique based on a non-manifold structure of curves constructed using relational geometry.

Earlier Naval Architectural packages 0[17] using patches for hull definition required the user to control the patches directly. A major disadvantage to this type of direct definition is that the three different quantities used to define patches, position, tangent and twist vectors are generally all of different order of magnitudes and must be specified to describe the patch. As the user ordinarily lacks the intuition to understand the behaviour of the twist vectors, these system were difficult to use and not ideal for the design process. NAPA creates surface patches from a mesh of cubic splines defined by the user. As the mesh is defined using mostly positional data it can be created much more easily by the designer. Once defined, the software calculates the tangent and twist vectors for each patch from the mesh.

NAPA has been based on a command line system from its inception. Even the interactions with the newer graphical user interface are transferred to the command line. Hence, all user data must be defined in a text-based format. This may be an antiquated style of interface, however, the definition system has been developed in such a way that objects, especially, curves can be created in a very flexible and powerful manner, much better than most other graphically orientated definition systems.

The curve definition system is very powerful allowing the user to easily create many shapes. The curves are based on a custom formulation of cubic splines. The curves are designed to produce a circular approximation so that the shapes are easier to construct in the shipyard. Curve definitions in NAPA rely on a two-dimensional system, simplifying the definition of all but the most complex curves used to define a hull. Curves are defined in two parts, the Location Surface and the curve shape. The Location Surface is a two-dimensional surface in which the curve exists. For most applications this is a plane, however, for more complex shapes such as deck sheer shape, the location surface can be itself defined by a curve, (Figure 22.24).

NAPA uses text in an interesting way to define the shape of a curve. Besides numeric information to define the position of vertices, text is used to control the tangent at the vertices. The text symbols, “-/” or “/-” create relaxed tangents. More control can be gained by specifying the angle of the tangent at the vertex, e.g. 10/ (10 degrees). Combinations of these symbols can be used to create effects such as knuckle points or straight segments in the curve. There are also symbols to specify how the points will be ordered or whether every vertex is a knuckle point. Figure 22.25 shows examples of some NAPA curves and the associated text definition. Vertex positions can be based on intersections of the location surface with other curves, vastly simplifying the creation of the hull surface mesh. As some intersections can be complicated, NAPA provides further text based definition commands to control intersections in cases, for example, where a reference curve intersects a location surface more than once.

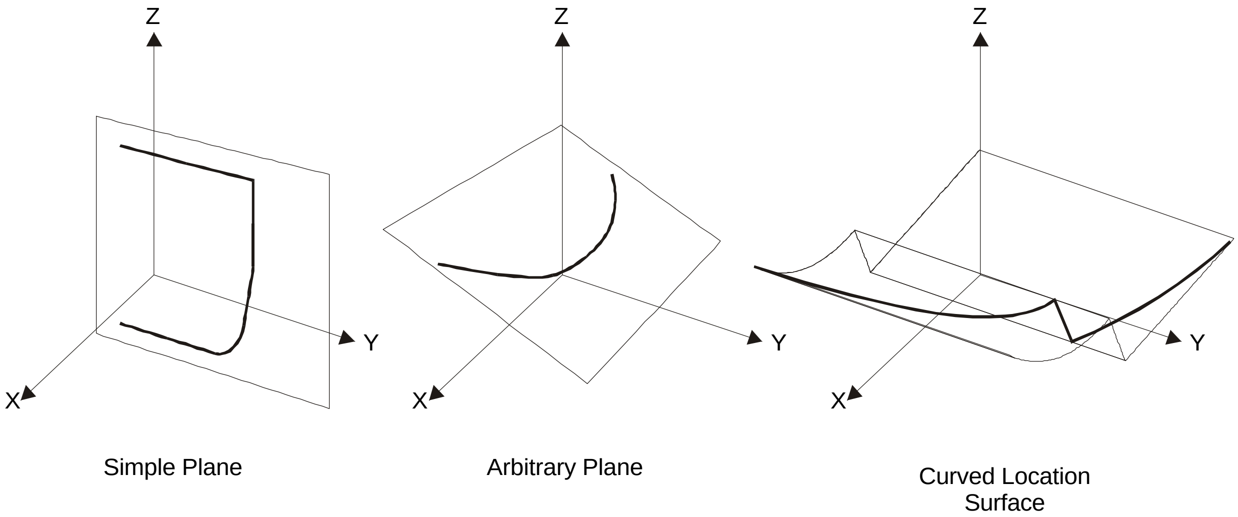


Figure 22.24, different types of location surfaces used to 3D curve construction.

Curves can be assigned properties known as Side Conditions. The Side Conditions are used to control the shape and curvature of the patches attaching to a curve. This allows curves to define knuckle lines in the surface or curves can be used to fix the surface at a certain angle. NAPA gives further Side Conditions to aid the creation of ship hull shapes. Midship sections curves are assigned a Side Condition forcing the surface to be perpendicular to the midship plane. Hull Flats curves are assigned a Side Condition forcing the surface to be parallel to hull flat plane when attaching to the curve.

A standardised approach has been developed for creating the hull form definition in NAPA. The procedure states the order in which the hull definition curves should be created and the naming system that should be used. This speeds up the definition process, as the designer does not have to spend time thinking about which curves are required and how to name them. A further benefit of the standardisation is that all hulls are created similarly and can be easily understood when passed between the different parties working with a hull definition. Figure 22.23 illustrates the hull definition procedure. Definition starts by defining the boundary curves (a). The hull flats and features curves are added next (b). At this stage, the curves defining the shape of the hull are added, building up from the curves added previously. It advisable to define the shape of the hull using longitudinal curves rather than sections as the curvature and shape of the hull is better controlled. The standard procedure advises that 'T' lines should be used to control the longitudinal shape of the surface. These curves are similar to hull diagonals and control the hull definition more uniformly. However, it is generally easier to use waterline or buttock data read directly from a lines plan or offset table. The definition example in Figure 22.26 uses waterlines to represent the hull shape (c). Sections are added (d) by referencing the waterlines to complete the hull mesh.

NAPA's hull definition system is very powerful, however, it requires a lot of planning to select data to position curves in the correct location. It requires a great deal of manipulation to move a curve from one location to another and maintain a good hull shape, problems also occur when the curves, using relational geometry references, are moved outside of the scope of the reference object.

At the beginning of a new design, it can be difficult to create a complete mesh without requiring a lot of accurate data on which to base the definition curves. To speed up the mesh creation process for a new hull design NAPA has a parametric hull library. The curves for these hull definitions are dependant on a table of parameters. The parameter tables specify the overall dimensions of the hull and sizes of local features such as the bilge radius. This data is modified through simple scaling functions to produce specific points or curve vertices within the definition structure. When a parameter is modified the software updates all the dependant geometry and generates a new hull.

Although this feature is still under development, it cannot be said to be a useful design tool. Currently, there are only three hull forms in the parametric library. A designer looking to create a new hull form would have to create a new parametric hull from scratch and this is much more difficult than creating a normal hull as all of the parametric scaling functions must be defined in addition. Parametric hulls are defined using NAPA's text definition technique. As there are a large number of curves for the simplest of hulls, the definition text for these hulls is quite large and there are no tools to aid the creation of parametric hulls in current versions of NAPA. Future releases of NAPA may contain a feature allowing the transformation of the hull shape, while updating the all of the definition parameters. This is a very complex operation to perform and there may be some reduced flexibility as a result. However, it would provide an improvement to the parametric hull feature.

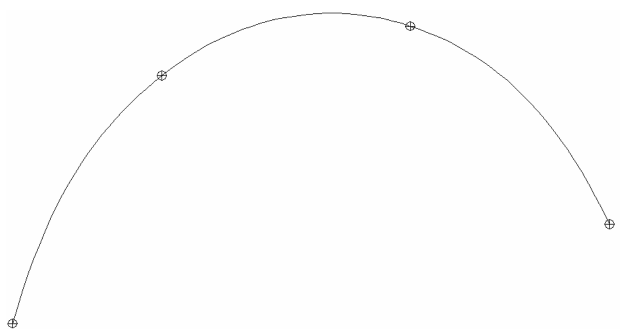
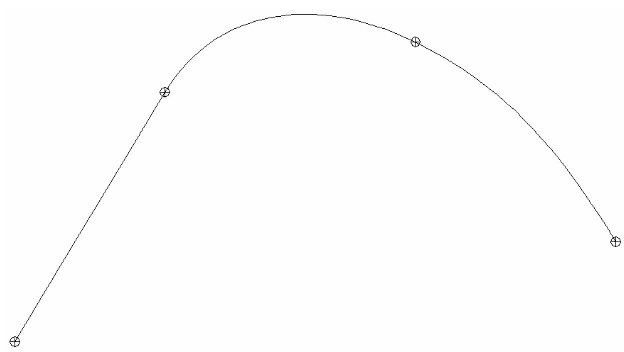
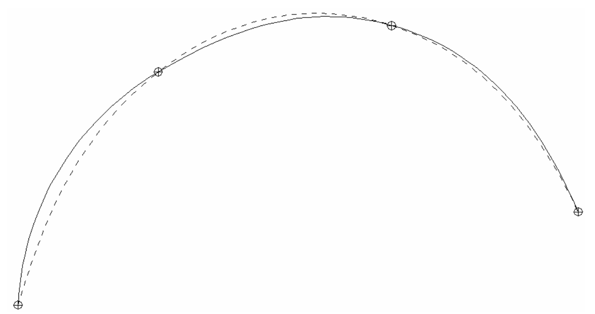
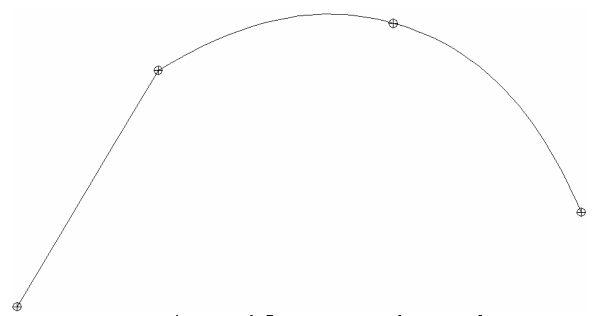
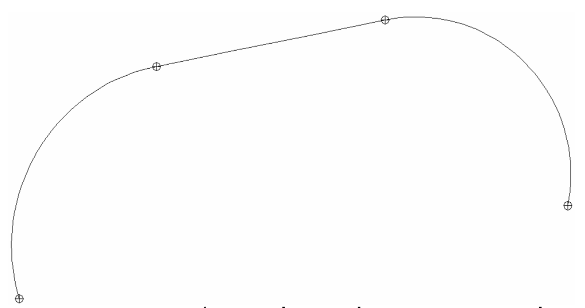
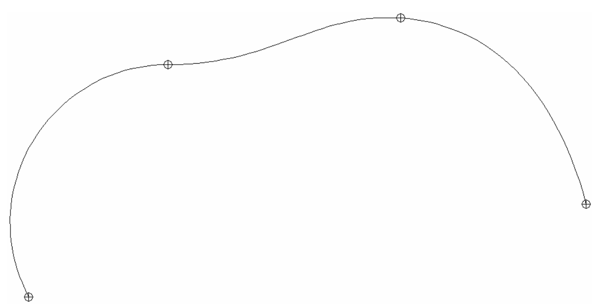
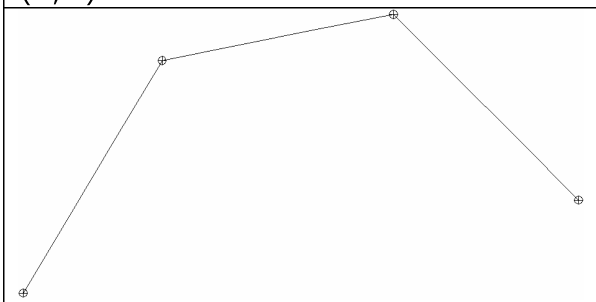
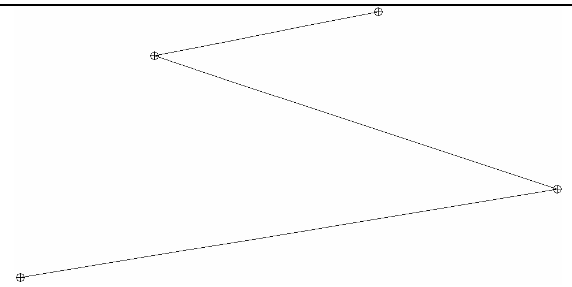
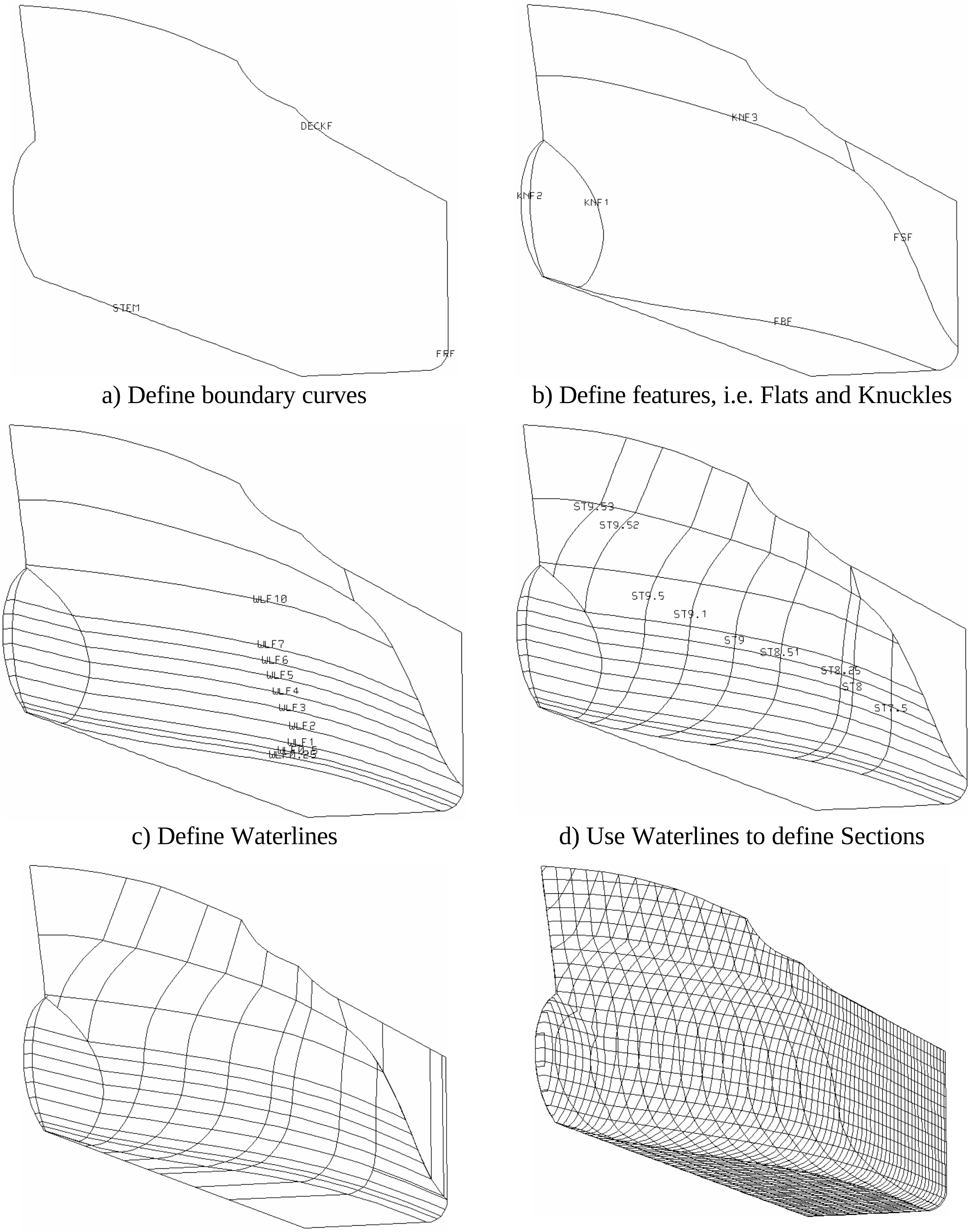
 CUR CURVE1 ‘Basic Cubic Spline’ Z 0 XY (0,0), (1.5,2.5), (4,3), (6,1)	 CUR CURVE2 ‘Relaxed into point 2’ Z 0 XY (0,0), **-/,**(1.5,2.5), (4,3), (6,1)
 CUR CURVE3 ‘Relaxed out of point 2’ Z 0 *@@ (Note: CURVE1 is show dashed)* XY (0,0), (1.5,2.5),**/-,** (4,3), (6,1)	 CUR CURVE4 ‘Knuckle at point 2’ Z 0 XY (0,0), **-/,**(1.5,2.5),**/-,** (4,3),(6,1)
 CUR CURVE5 ‘St.Line between points 2&3’ Z 0 XY (0,0), (1.5,2.5),**/-,** **-/,**(4,3), (6,1)	 CUR CURVE6 ‘0° tangent at point 2’ Z 0 XY (0,0), **0/,**(1.5,2.5), (4,3), (6,1)
 CUR CURVE7 ‘Straight segments’ Z 0 XY **<>** (0,0), (1.5,2.5), (4,3), (6,1)	 CUR CURVE8 ‘Sort points in Y direction’ Z 0 **YX** <> (0,0), (1.5,2.5), (4,3), (6,1)

Figure 22.25, examples of using text to control curve definition in NAPA.



a) Define boundary curves

b) Define features, i.e. Flats and Knuckles

c) Define Waterlines

d) Use Waterlines to define Sections

e) Resulting Patch surface

f) Hull Lines

Figure 22.26, the procedure for creating a hull surface from curves in NAPA.

NAPA uses a command line system to operate all features and calculations. The text definition system is very powerful as data and commands can be combined together to create complex operations. NAPA's command system also contains basic flow-control commands to allow rudimentary programs to be written. However, despite the powerful text interface to the software, more graphical interfaces are being developed for the program. In recent versions the Hull Editor, Figure 22.27 and the Ship Model editor, (Figure 22.28), have been included to allow the user to edit the hull definition more interactively and to visualise and edit the compartment definitions.

The Hull Editor, although a welcomed addition, does not compete well with other software providing graphical interaction with the hull surface. Unfortunately, the interface does not seem to connect well to the text definition system and it quite difficult to edit curves without unexpected events occurring such as the curve vertices being re-sorted during the drag of a vertex. Curves that use non-orthogonal location surfaces can be difficult to edit, as the vertices do not move well in orthogonal views of the hull definition.

NAPA is available for different operating systems and extra software can be required to allow it to run. On the Windows NT operating system, a product called Exceed is used to translate the internal X-Windows screen commands in to commands compatible with Windows NT. The Exceed system has control over all of the graphical elements of NAPA and consumes a reasonable amount of computer resources doing this. As a result, dynamic graphical operations on NAPA data, such as performing curve manipulation in the Hull Editor, is much slower than native Windows NT software and it requires good computer hardware for the system to function interactively. It is sometimes easier to set up a script, which allows curves to be edited numerically, updates the hull surface and redraws the screen. The user only has to edit the definition and run the script to quickly see the result of a modification.

NAPA is one of the most useful Naval Architecture packages available today. It provides all of the routines and calculations a Naval Architect requires. However, although most of the systems perform well in a design environment, the hull definition system does not. The tool is not flexible enough to allow the designer to quickly develop a hull form without the need for large amounts of data, it is better suited to hull digitisation.

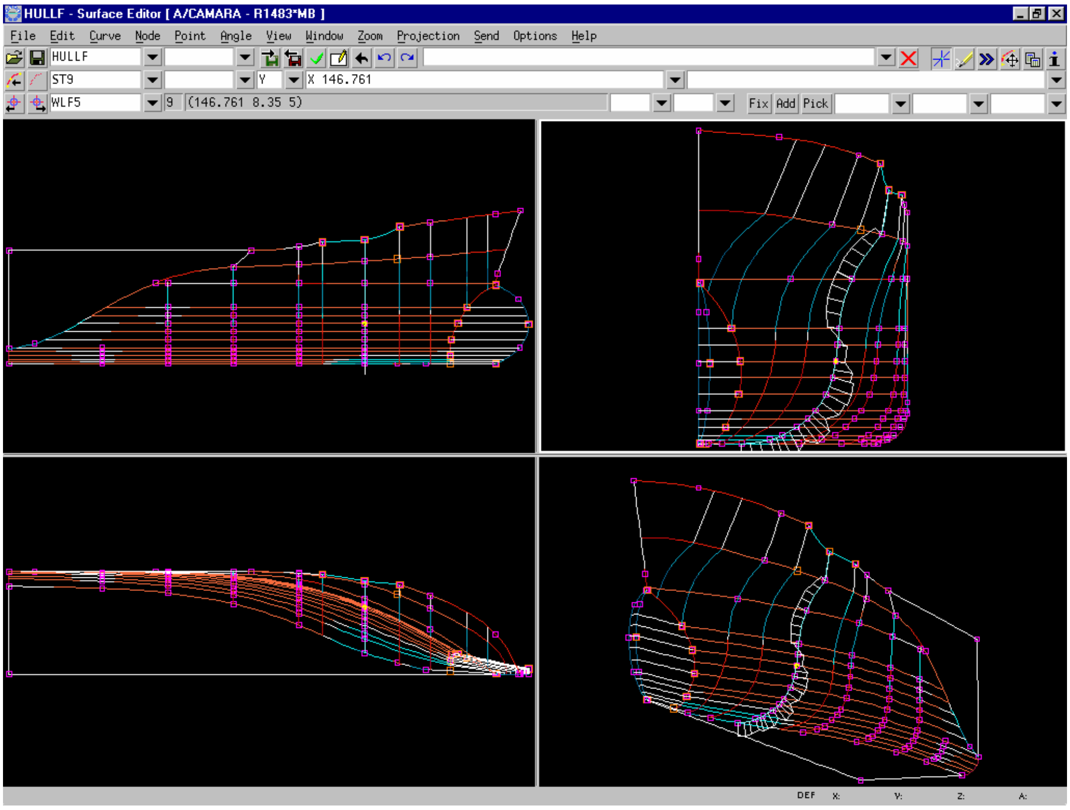


Figure 22.27, the NAPA Hull Editor.

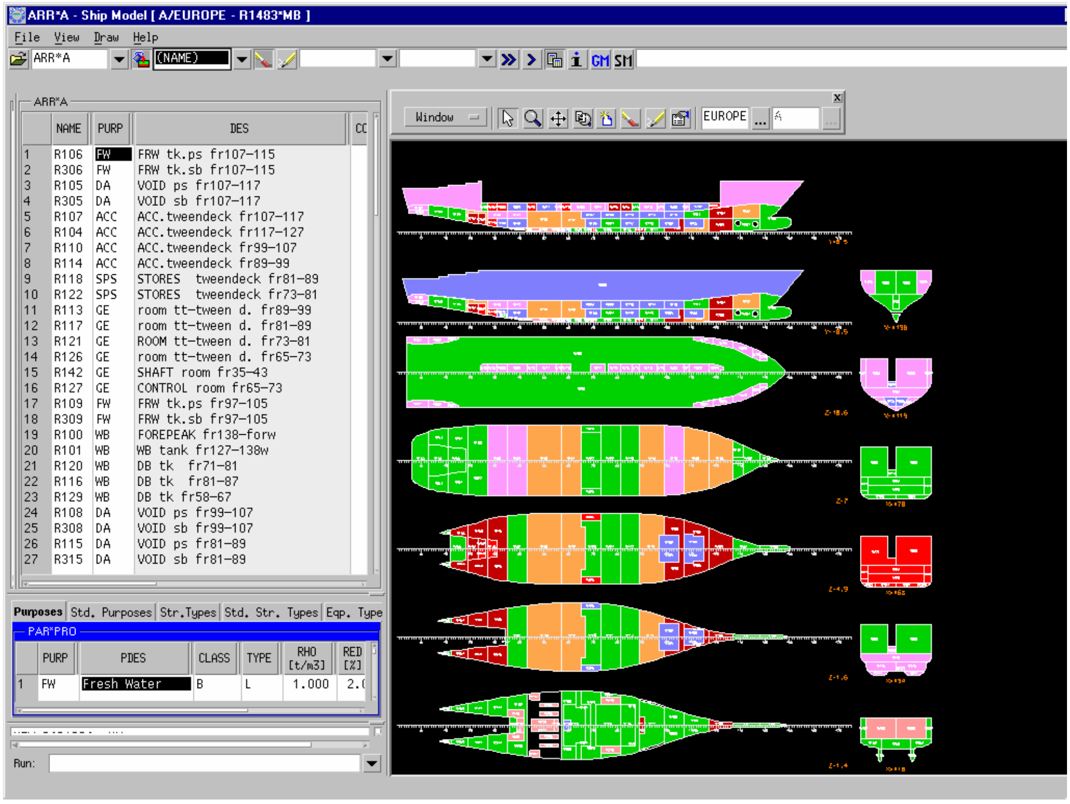


Figure 22.28, the NAPA Ship Model Editor.

22.9. Tribon M1 – Kockums Computer Systems (KCS)

Tribon M1 [30] is a suite of software developed by KCS for the initial ship design phase. Tribon M1 started life as the BLINES/SFOLDS system developed by British Maritime Technology (BMT), one of the original Naval Architectural Packages. Originally a command line orientated, the package acquired a graphical user interface and was renamed Hulltech in the early 1990's. Later KCS purchased BMT Icons, the company developing Hulltech, and renamed the system Tribon. Besides M1, a number of other Tribon software packages are developed by KCS for different phases of the ship design and construction process.

LINES and FORM are components of the M1 package used to create and define ship hull forms. Form is used to generate initial hull surfaces from form parameters or hull offsets. LINES is used to modify the hull surfaces created in FORM or loaded from file.

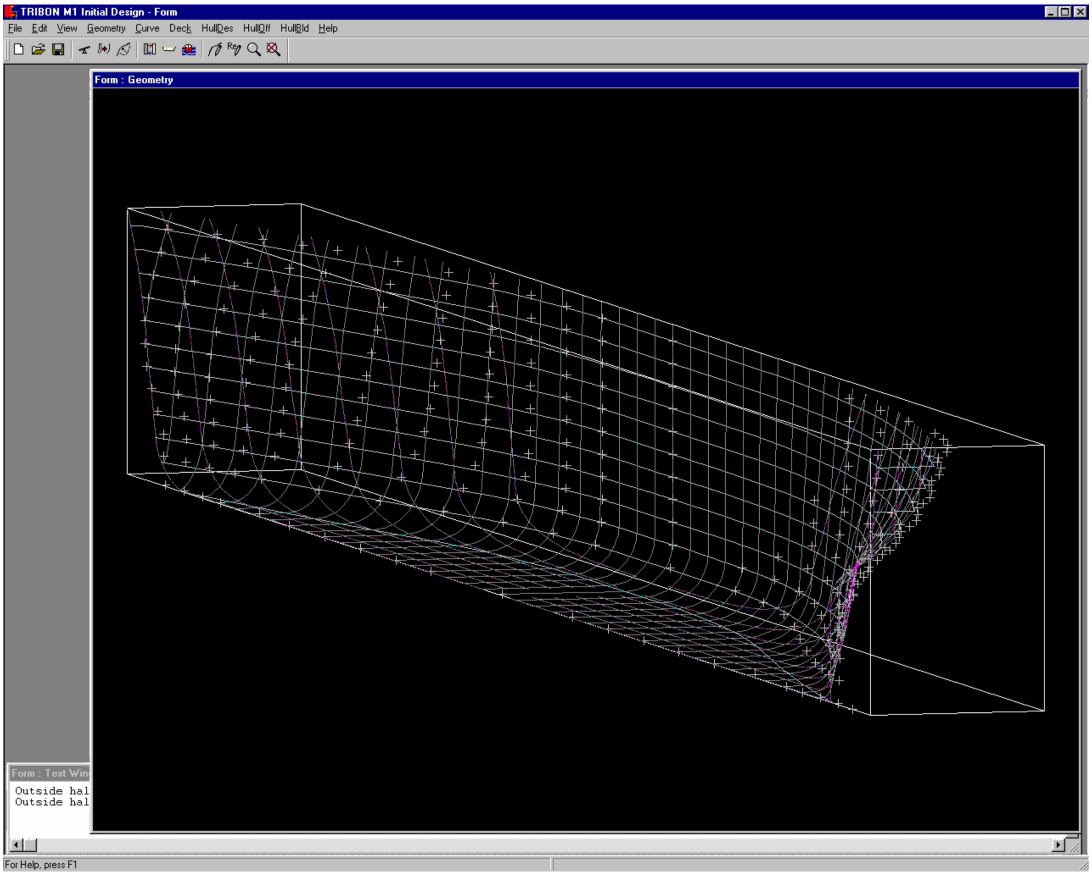


Figure 22.29, the FORM user interface.

FORM uses surface patches to define the shape of the hull, (Figure 22.30). Whole patches can be are used to define the hull flats and patch boundaries can be used to create knuckle lines in the hull. FORM has three methods of creating a hull surface definition, depending on the amount of information that is available to the designer. These are:

- HullDes is a knowledge-based method, which generates new hull forms using the minimum possible input data. Regression based algorithms advise the user on appropriate values for key dimensional parameters. These algorithms enable the designer to generate a hull surface by specifying only ship type, length, beam, draught, maximum depth and speed. The hull surface produced automatically has the required block coefficient and longitudinal centre of buoyancy.

- HullOff extends the facilities of HullDes by allowing the designer to tune the generated ship surface so that it best approximates offset data obtained from an existing ship. Least squares facilities enable the approximation to be carried out under the control of the user. The curves that HullDes uses to ensure that the ship has the correct block coefficient and longitudinal centre of buoyancy are tuned to obtain the best approximation to the offsets.

- HullBld is the software component that operates at the heart of the surface definition. It can be used to create single surfaces patches via a least squares technique and it allows the surface definition to be modified and analysed. The automatic facilities of HullDes and HullOff provides a high level of control over the shape of the fore or aft ends of the hull definition while HullBld is used to fine tune individual surface patch details.

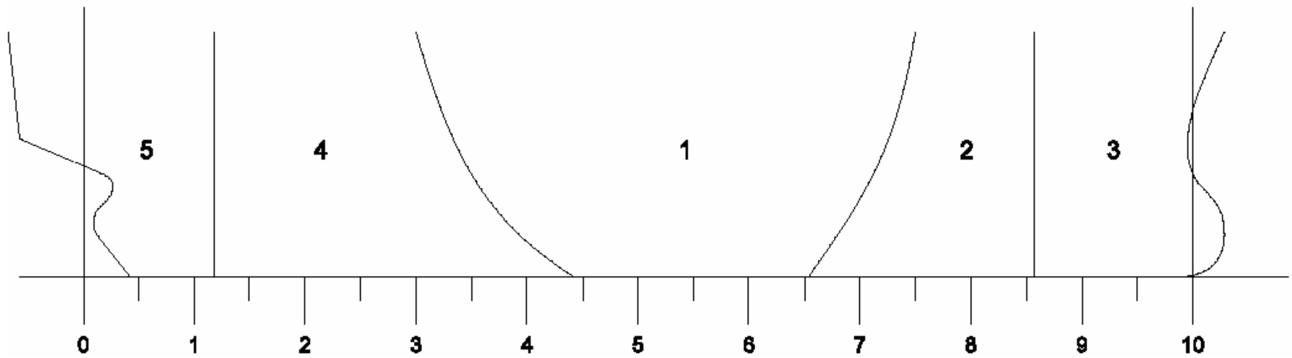


Figure 22.30, a simple arrangement of panels creating a hull surface in FORM.

HullDes leads the user through several steps leading towards the generation of a hull form. The program uses a separate dialogue box at each step to allow the user to modify each parameter. Each parameter is assigned a default value calculated from regression functions. The steps are arranged for different sets of regression functions. In the first step, the user enters the length and type of the vessel. The regression functions then use this data to produce initial values for step two, the principal dimensions, probably having a stored set of design ratios for each ship type. The complete procedure used to create a hull form in HullDes is listed below:

Step 1: Define the ship type and the Length Between Perpendiculars.

Step 2: Specify Design Speed, Design Draught, Beam and the Maximum Depth of the Hull.

Step 3: Specify Block Coefficient, or,

Prismatic Coefficient.

Step 4: Specify the Midship Section Coefficient, or,
Specify the Bilge Radius, Flat of Keel, Rise of Floor, Tumble out at bottom, Tumble out at side, Tumble in at Top and Tumble in at side.

Step 5: Specify the Entrance/Run Ratio, Longitudinal Centre of Buoyancy, Parallel Middle Body Length, Flare type, Aft body U/V and Fore body U/V.

Step 6:

Fore body Parameters:

- Bulb Volume
- Bulb Depth
- Bulb Length
- Bulb Breadth

Aft Body Parameters:

- Transom Immersion

Single Screw Parameters:

- Propeller Diameter
- Shaft Centreline Height
- Boss Diameter
- Forward Edge Position

Twin Screw Parameters:

- Skeg Half Siding

Sonar Dome Parameters:

- Depth
- Length
- Extrusion

After the procedure has been followed, FORM automatically shows the layout of the patches in a similar view to Figure 22.30. The user can review the lines of the hull by either displaying the hull bodyplan, (Figure 22.31), or an oblique view can be controlled to display the hull in three dimensions, (Figure 22.29). The hydrostatics of the hull surface can be calculated, the results are included in the system log. The user can go back and change any parameters entered during the generation procedure until the desired hull surface is reached.

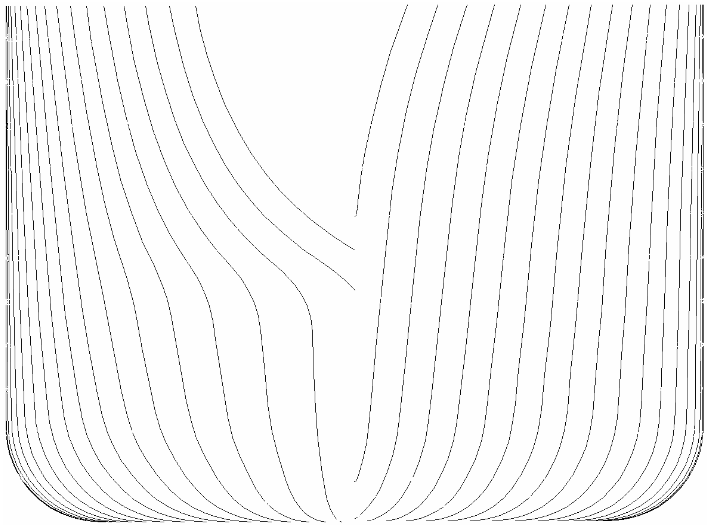


Figure 22.31, a body plan of a ship hull created using default parameters.

The definition parameters of HullDes control the global and medium level aspects of the hull shape. HullBld allows the user to edit the hull surface at a local level. HullBld give the user the ability to edit all the curves used to generate the patch hull surface. The curves are edited interactively using an unusual graphical interface shown in Figure 22.32.

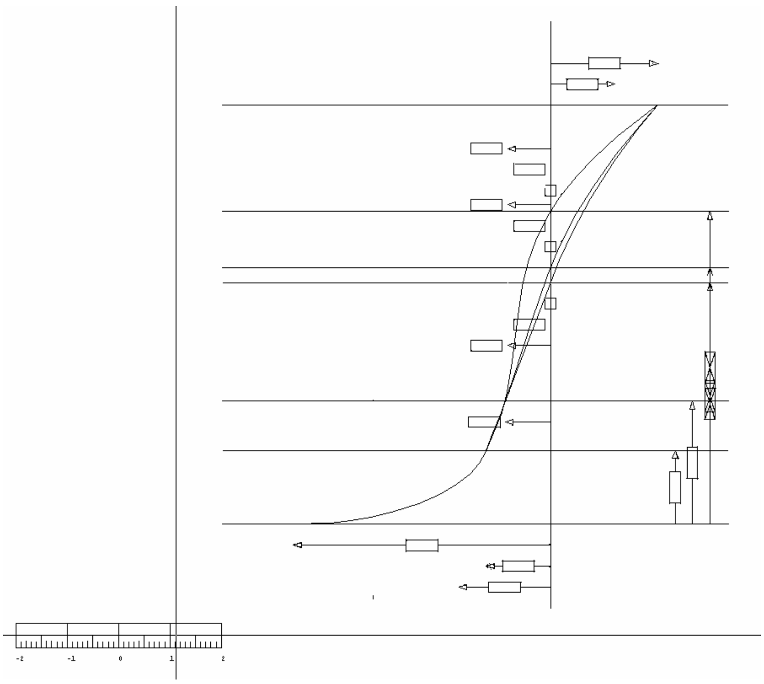


Figure 22.32, the interface to FORM hull definition curves.

Curves are edited in two dimensions. Attached to the curve are box like symbols from which arrows extend, these may be supposed to represent dimensions. To make a modification to the curve, the user must first select the dimension by clicking the mouse cursor inside one of the boxes. A selected dimension displays diagonal lines draw between opposing corners of the box.

The user modifies the dimension by clicking the mouse cursor in the scale image on the bottom left of the screen. The dimension is increased or decreased according to the location along the scale the user clicks the mouse cursor.

Editing the definition curves using this interface is very difficult. There is no indication to which part of the hull the currently edited curve belongs, the only guide is the shape of the curve. For some of the curves that define more abstract hull properties, such as tangents, the identification process is more difficult as the designer is less familiar with the curve shape. The curve shape cannot be controlled easily in the editor. It is not always clear what each dimension box controls, especially as some control curve location and some control curve tangency. It can take a considerable amount of trial and error to find the right dimension control box. However, by this time, the curve has probably changed shape significantly and there are no undo features. The user must cancel the edit and reselect the curve for editing. Once the correct dimension control box has been found, the user must use the scale feature to modify the dimension. Although numbers are displayed on the scale, the system doesn't give the user a feel of the modification process and the shape cannot be controlled accurately. Once the curve has been suitably modified, there is no definite operation for confirming or rejecting the edit. The display just clears. It is unclear why an editing technique such as this has been provided, especially when almost all modern software in which curves are edited, allow the user to modify curve vertices or tangents directly with the mouse cursor.

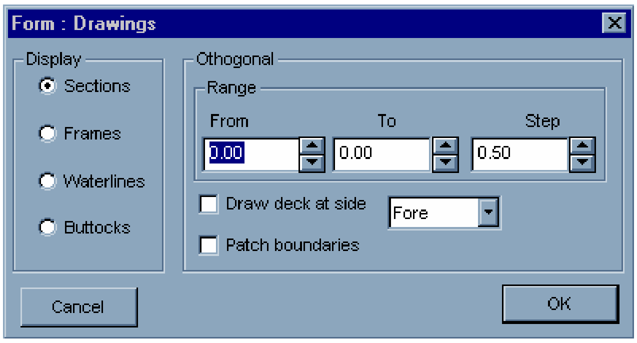


Figure 22.33, The Dialogue Box controlling contour drawing.

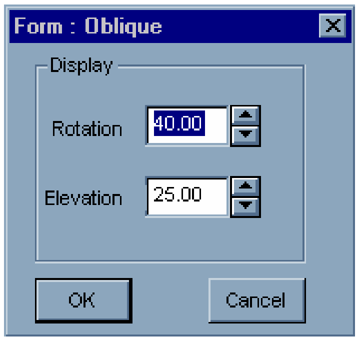


Figure 22.34, The Dialogue Box controlling the oblique display.

The user interface is not well developed. Once a hull has been generated, the system does not automatically display the hull form. To display the hull form, the user must open a dialogue box, (Figure 22.33), and specify which hull contours to draw. Section, Frame, Waterline and Buttock contours can be draw. However, only one type of contour can be drawn at any one time. The user must return to the dialog box add other contour types to the display. Sections and Frames can be drawn for the whole length of the hull. However, Waterlines and Buttocks can only be draw for the bow and stern portions separately, requiring two successive uses of the contour dialogue box to draw the waterlines or buttocks for the full hull length.

The user can display the hull as a body plan or in an oblique view. A dialogue box is used to control the oblique view display, (Figure 22.34). The user must manually edit the two angles to change the projection in which the hull is draw. Once the view is changed, the user must redraw all the contours by returning to the contours dialogue box. There is a redraw option, however, this can only display the contours specified by the last use of the contours dialogue. This is an unusual way controlling three dimensional viewing and it is not user friendly. All the other hull design packages reviewed have some interactive way of using the mouse to control three dimensional projections of the hull surface, without requiring the user to redraw the display after every change.

The LINES component of the Tribon M1 system is used to modify hull forms created by FORM. A full review of LINES was attempted on several occasions over a year. However, this software was found to so difficult to use that an objective review was impossible, even with the help of the manual. The user interface to LINES is shown in Figure 22.35. The interface has a toolbar at the top, to control drawing and selecting operations. On the left of the screen is list of all the entities that are part of the hull definition. The list allows the user to perform operations on individual entities such as editing or drawing.

The display operates slightly better than FORM, the user has good control over the entities that are displayed. Entities can be displayed singularly, by group or everything can be drawn. Unlike FORM, three dimensional viewing is control by interactively by the mouse. However, the display appears to rotate in the opposite direction to motion of the mouse.

The LINES curve editing system is very difficult to use. The vertices of a curve available for editing are displayed and the user can select a vertex to move by clicking the mouse close to the point. The user relocates the vertex by clicking again at new location of the point. During this process, the software gives no indication of which vertex is being moved or any indication that the user is performing an editing operation. A few accidental clicks on the screen can destroy the shape of the curve and the software does not have a practical ‘Undo’ command to move back in the editing process.

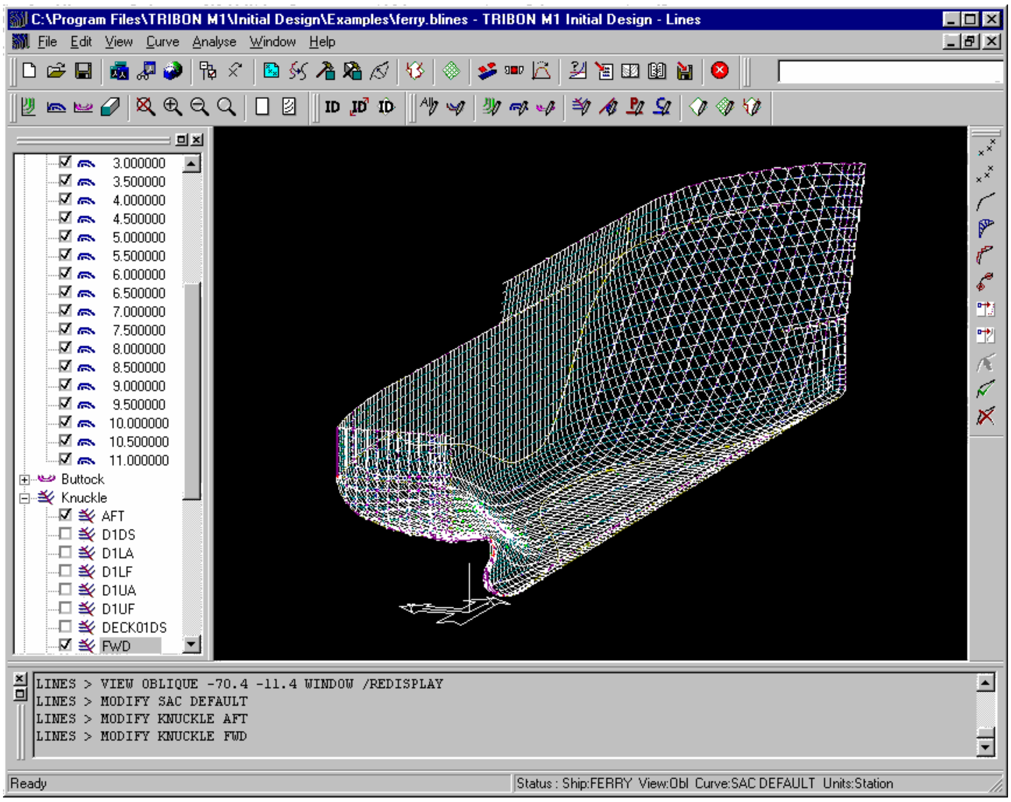


Figure 22.35, the user interface of LINES showing the hull lines of a surface definition.

One of the key concepts of the Windows interface system is to give screen controls Context Sensitivity. Any screen control that performs a currently unavailable operation should be displayed in a disabled state or not displayed at all. This informs the user control cannot be used. LINES would benefit greatly if this concept used more liberally. Both the popup menus displayed for entities in the list and the floating curve editing toolbox, (Figure 22.36), display all commands as being available, even when it obvious that some are not. For example, the popup menu gives the user the option of editing surface contours. As these curves are analytically generated from planer intersections with the patch surface, the user cannot possibly edit these curves and expect the hull form to change.

The context insensitivity in LINES suggests that the system still operates around the early BLINES command line system. Command line systems are normally context insensitive. They allow the user to type any text into the software and can only validate the command once it has been issued. The software must handle any text that the user may type in without crashing or loosing any data. Error messages are more frequent in command line systems as they are the only way of informing the user that an operation cannot be completed. LINES has very uninformative error messages, most messages give user a general reason why the programs operation has stopped. However, for efficient operation the user requires more specific information about why the operation could not be performed and if possible how to overcome the problem. This style of early program operation has given software a bad reputation, most error messages are uninformative and do not help the user in any way. Context Sensitive design in the interface can almost eliminate the use of error messages and the user gets a much better feel of the software application.

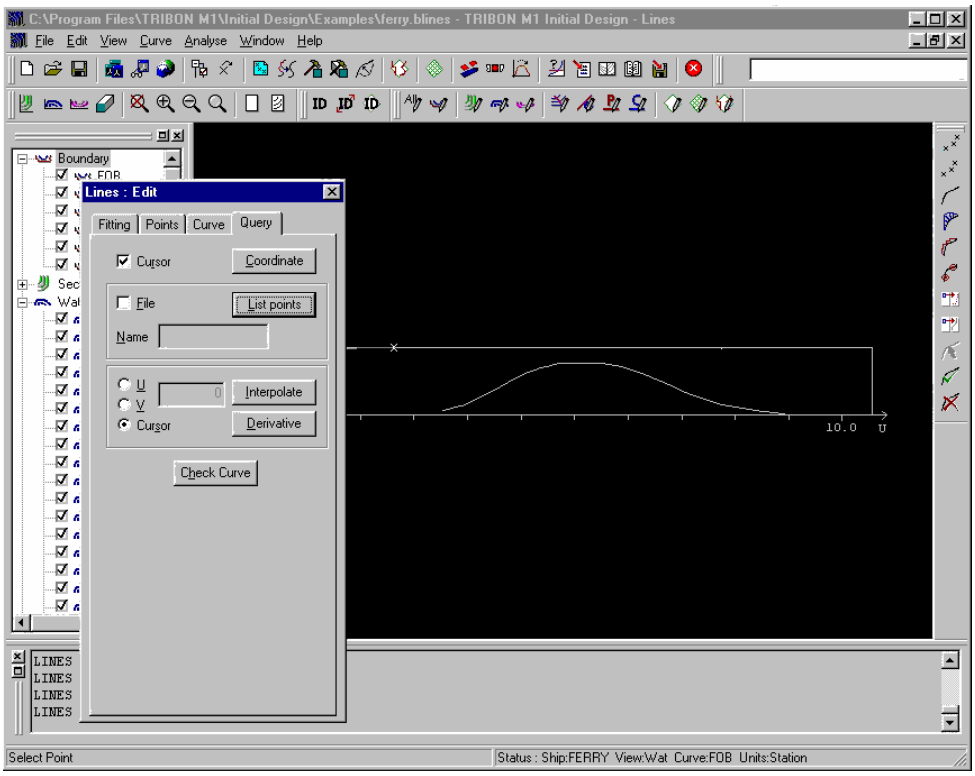


Figure 22.36, editing the flat of bottom curve.

Both LINES and FORM are very difficult programs to use and productivity using these software applications to design hull forms would be very low. This package requires much more development if it is to compete with the other hull design systems available today. Given the history behind the Tribon M1 package, it is inconceivable why this system should be so difficult to use for hull form design. The software industry has developed ideas and concepts for making program operation easy for the user. In no way does Tribon follow this line of thought.

22.10. Paramarine – Graphics Research Corporation

Paramarine [36] is marketed by Graphics Research Corporation (GRC). GRC was founded in 1989 to distribute and support Naval Architectural software originating from the MoD and its agencies such as the Defence Evaluation and Research Agency (DERA). The primary Naval Architectural product produced by the MoD was GODDESS running in a UNIX environment. In 1998, a PC version of the software suite, known as PCG, was ported to the Windows NT environment. As GODDESS is a development of many smaller programs it could be difficult to use and the system can now be considered out of date. To keep pace with the growth of Naval Architectural software technology Paramarine was developed to provide a system, which could provide flexibility in the design of Naval and Commercial vessels. Paramarine is a very new software package and at the time of writing is being validated through Beta testing.

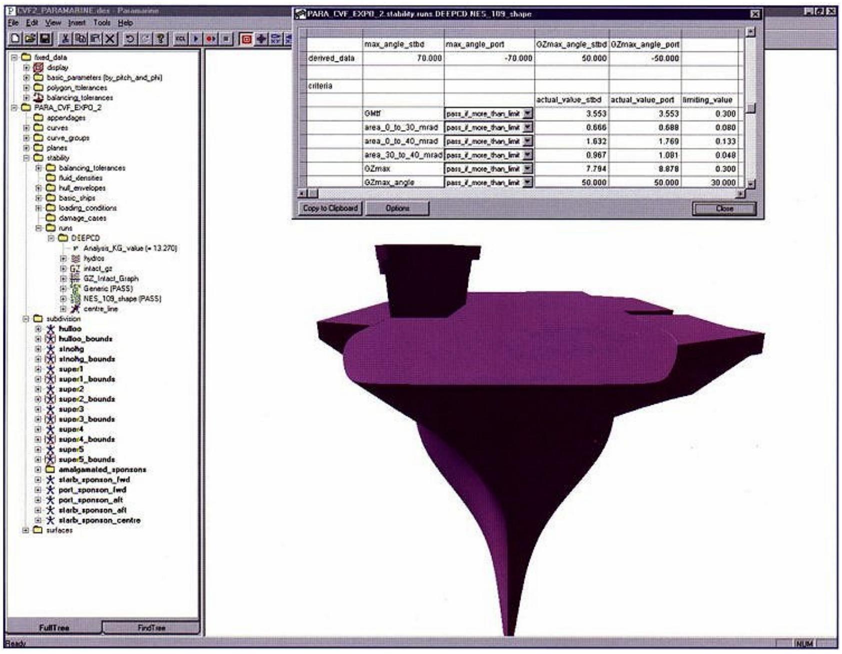


Figure 22.37, the Paramarine user interface.

Paramarine is a package for ship and submarine concept design utilising the features of a solid modelling system. Paramarine provides tools for hull design, solid modelling and stability analysis. The system uses a product modelling system, which allows the user to develop hull shapes, superstructure designs and compartment layouts. The tool uses a relational structure allowing geometry and calculations to be linked together. Thus, when a change of design parameter or geometry occurs the calculations are automatically updated. All features are controlled through the tree structure displayed on the left side of the screen giving the user information about the current relationships within the system. The structure also allows the user to view and edit all the components within the database.

Paramarine uses NURBS surfaces to design the hull form. However, unlike most other Naval Architectural packages it does not allow the user to directly modify the control polygon of the surface. Paramarine uses a parametric system to build the hull form from a hierarchical structure. The system uses two tool objects called *Quickhull0* and *Quickhull1*. *QuickHull0* contains all the boundary curves and control data that is used to mathematically create the hull surface object *Quickhull1*. To start the hull design process a *Quickhull0* must be created from a parent hull form surface, these are normally stored on disk using a file in the IGES format.

Parametric key points and guide curves are used to control the generation of the hull surface. The parametric key points control the shape of the parallel middle body, the transom and bow extents, and the aft cut-up, feature generally found in frigate hulls. B-spline guide curves are defined for the bow, transom and the midsection. The shape of the shape of the surfaces is embodied in the input boundary curves. The quality of smoothness and fairness of these curves is therefore critical to the whole shape of the hull. This information is then used to blend a hull shape from the bow to the midship section and from the midship section to the transom, (Figure 22.38).

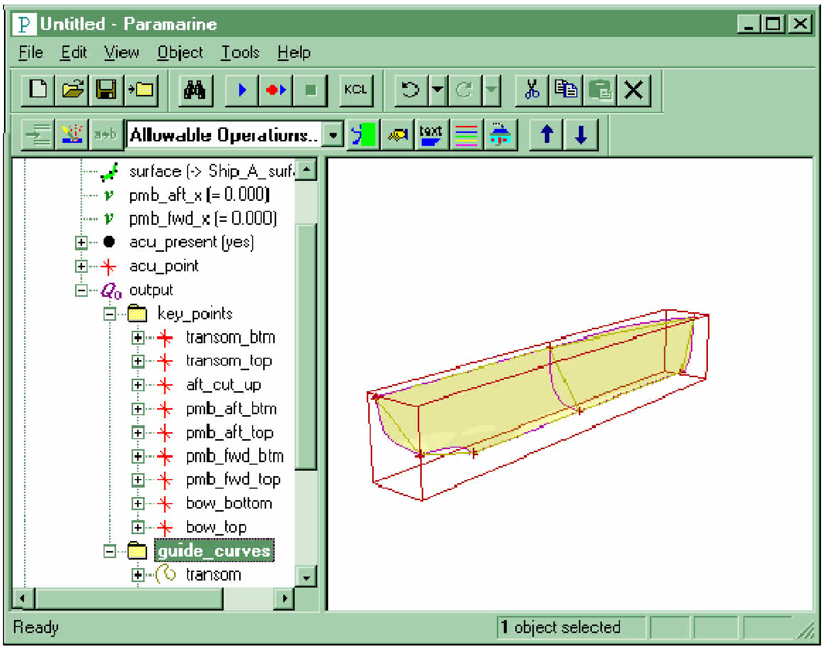


Figure 22.38, a hull surface in Paramarine with boundary curves.

Once a basic hull shape has been created, there are tools available to allow the user to modify the Cross Sectional Area (CSA) Curve and the Midship Section Coefficient. In the case of the CSA transformation, the user can change the shape of the hull to match the shape of a target curve taken from another hull or supplied data. The software can also generate a CSA curve from certain parameters, (Figure 22.39), however, it would appear that this feature is very sensitive to the input parameter as unusual CSA curves can result if values are not selected well. The success of transformation is highly dependant on the quality of the data used to define the hull. The control points of the boundary curves must be uniformly distributed as the transformation process can over emphasise any unfairness in the guide curves. For more advanced users, complete parametric design can be achieved using the scripting tool, which allows transformations to be based around user defined data and rules. However, as the script must traverse the tree structure of the database, the text references to data elements are generally long.

Paramarine appears to be good at generating initial hull forms for concept design. However, the types of hull forms that can be produced with the system seem to be orientated towards conventional military shaped vessels. The hull surface control polygon uses eight rows of vertices to influence the transverse shape of the vessel. With this low number of vertices, more complex forms such as bulbous bows and propeller hubs cannot be produced without penalising the shape of the whole surface. Knuckle lines are not directly supported, however, it is possible to create these types of features by editing the guide curves to produce discontinuities in the hull shape. As this requires at least three vertices from the eight on the guide curve, control of the hull surface will be diminished.

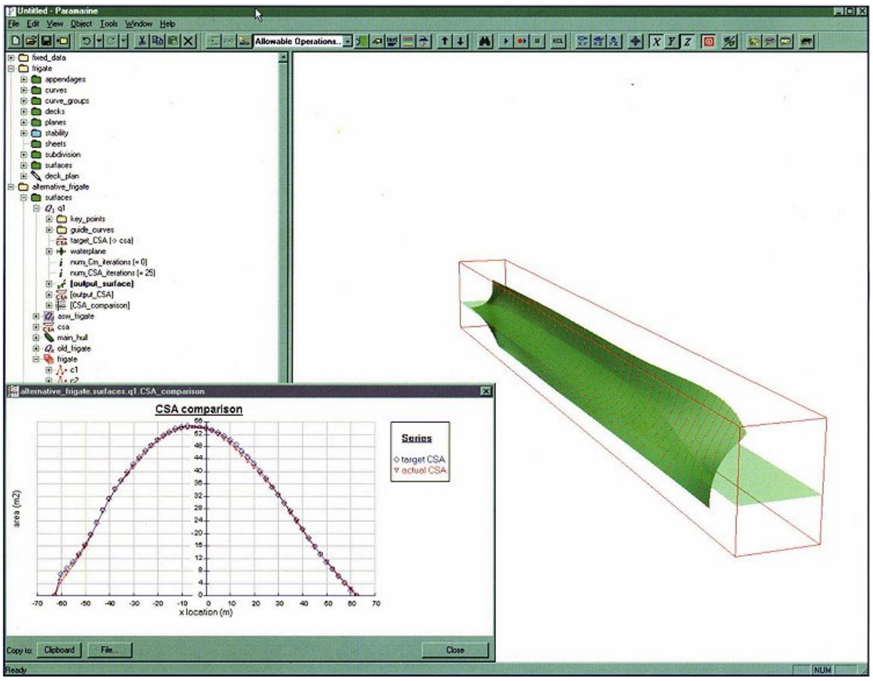


Figure 22.39, hull form visualisation with water surface and Section Area curve.

As the software has not been officially released at the time of writing, it is difficult to review the operability and interface of the software. However, the marketing material and manuals supplied by GRC have given a good insight into the system's operation. The interface to Paramarine is unusual because it does not provide many tools to manipulate the hull form. The user must edit the ship definition using the tree structure panel. For extended use, this approach is likely to become very cumbersome, especially as this Windows component was not designed for this type of data manipulation. The relational parts of the system do not seem to be very robust. The manual states that many error messages can appear during the manipulation of hull parameters when the shape becomes unsuitable for operations that the geometry is related to. Moreover, the manual advises users to 'unwire' the hull from any relationships when extensive modifications are planned. For a tool that is built on a relational database, the key to its success is the ability to handle situations when relational data becomes unsuitable without annoying the user.

Paramarine is an unusual package for a Naval Architectural design tool. The system is expensive and requires a high hardware specification on the computer executing the system. The system can only be used as part of concept design process. Beyond these levels of design, the system does not provide the tools or the flexibility to take part in later stages in design. This software is more likely to be used in the development of military ships, as these projects normally have an extended concept design period not normally found in commercial ship design.

22.11. The Foran System – Sener Ingenieria Y Sistemas, S.A.

The Foran System [9] began development in 1965 by Sener and is now one of the longest running Naval Architectural software packages. Foran was designed to take advantage of improving computer technology and provide the Naval Architect with an advanced ship design system using the concept of Engineering for Production. The Foran system provides the Naval Architect with the full range of integrated tools to cover processes from initial design right through to the end production process. Sener have developed a product modelling system allowing all the ships component parts to be stored within a database file. Using a product modelling system a designer can create a whole ship within the computer environment and any problems with location of any components can be raised before the ship begin construction, (Figure 22.40).

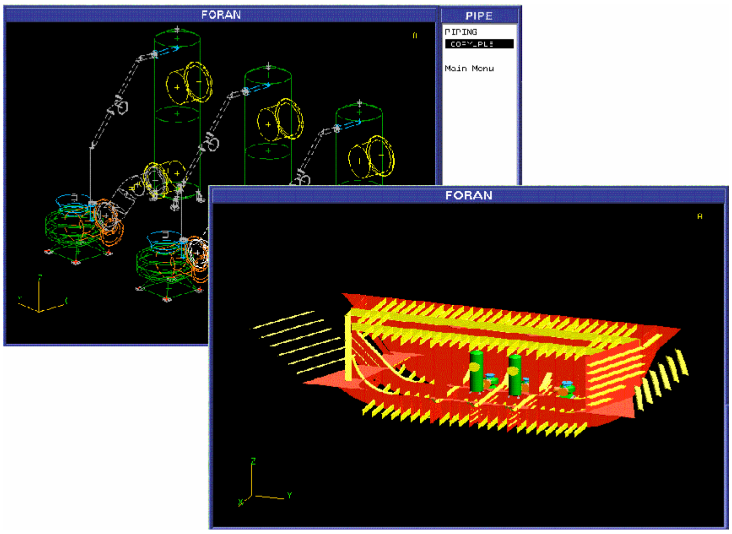
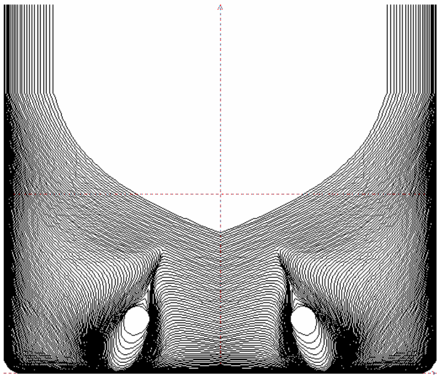


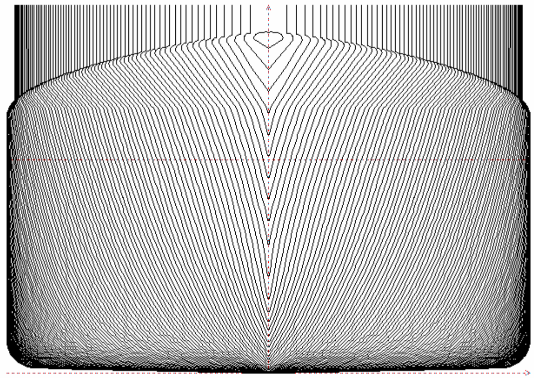
Figure 22.40, the FORAN product modelling system.

The main hull design modules within Foran are FORMF and FORMG. Predefined hull forms are manipulated within FORMF. Transformations can be used to generate new hulls from parent forms and the hull data can be fitted and faired. Since 1965, the hull representation technology does not appear to have been updated often and it is still based on section, waterlines, buttocks and boundary curve data. Despite the lack of surface technology FORAN is still able to develop complex and modern hull forms, (Figure 22.41). Recently, with version V40, Foran has embraced NURBS surfaces in the module FSURF. Future releases of Foran will further integrate NURBS surfaces into the system.

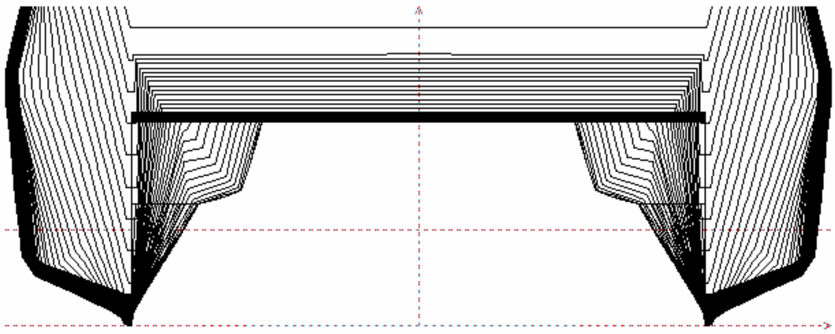
New hulls can be created with the module FORMG. This generates intrinsically faired ship hull forms from an original mathematical waterline formulation developed by Sener. The hull surface shape is created from waterlines functions, controlled by over the depth of the vessel by draught function. The basic dimensions of LBP, Beam, Design Draught, Block Coefficient and Longitudinal centre of gravity are used to define initiate the design of a hull form. Initially, most parameters are set to zero and do not affect the hull shape. The hull form parameters can be changed directly, by using the mouse to pick new locations on the graphical view of the hull. The software can display the results of a modification by overlaying the current view with the lines of the hull from the previous design iteration.



Product Carrier



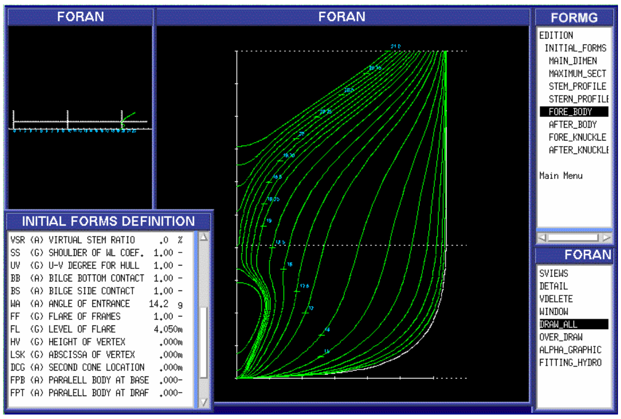
Asphalt Carrier



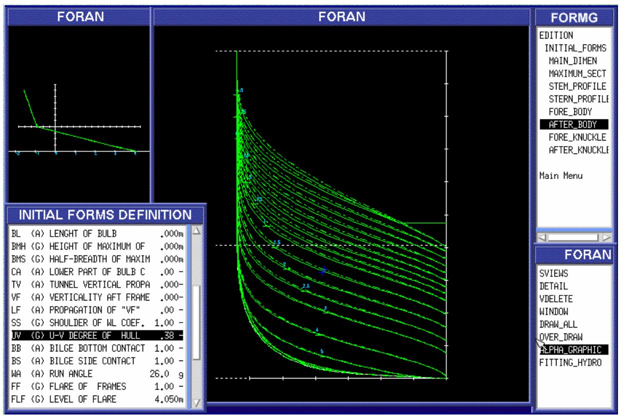
Catamaran

Figure 22.41, examples of hull defined using FORMF in FORAN.

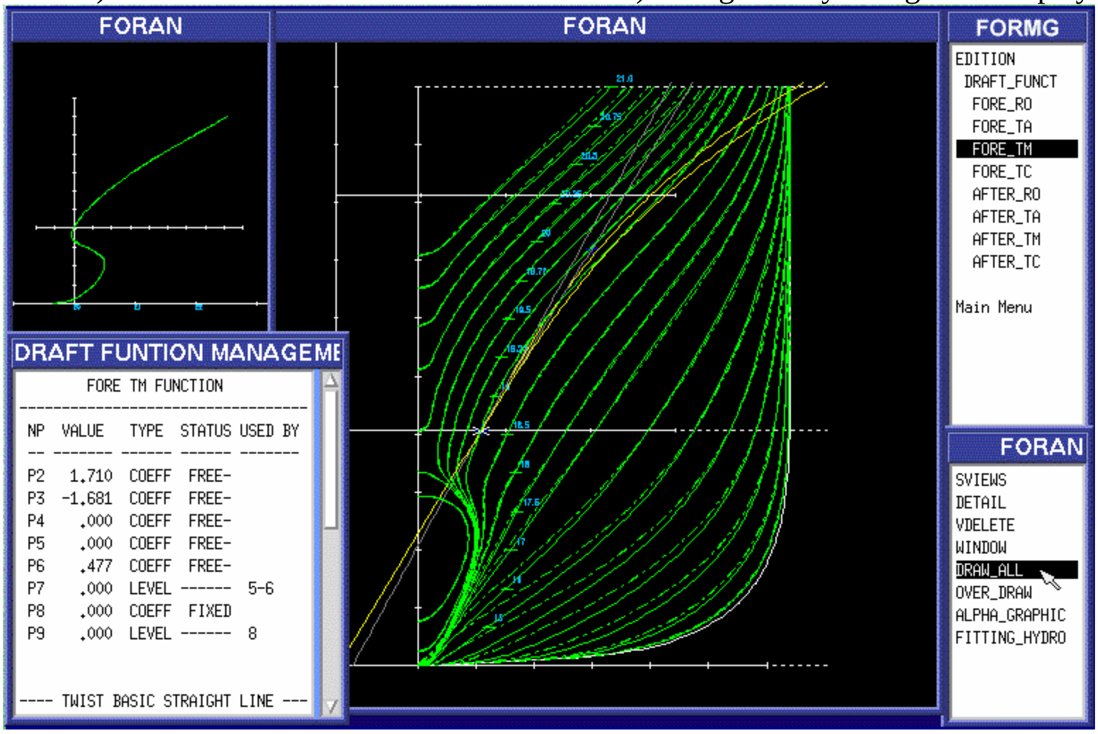
FORMG allows the user to change the shape of the hull by directly modifying the functions defining the surface. The waterline functions can be changed allowing the user to directly modify the extremities of the hull surface. Three curves are used for this task, the real profile, the tangent profile and the virtual profile. Waterlines are generated towards the virtual profile but finish at the real profile. The shape of the tangent profile controls the transition of the waterlines between the virtual and real profile. The draught functions control the halfbreadth of the waterlines over the depth of the vessel. Twist can applied at certain points along a draught function so that the halfbreadth is increase above the point and reduced below.



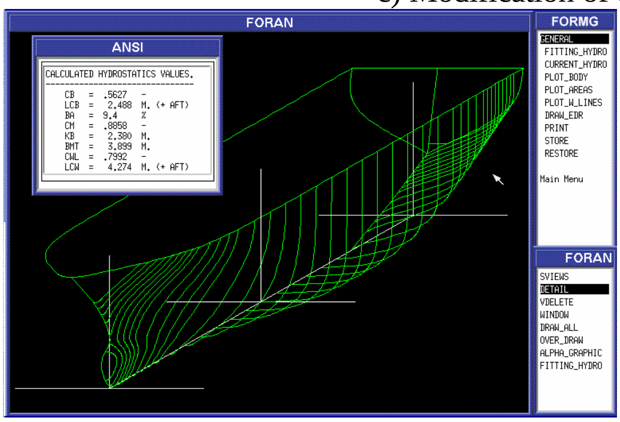
a) The initial definition



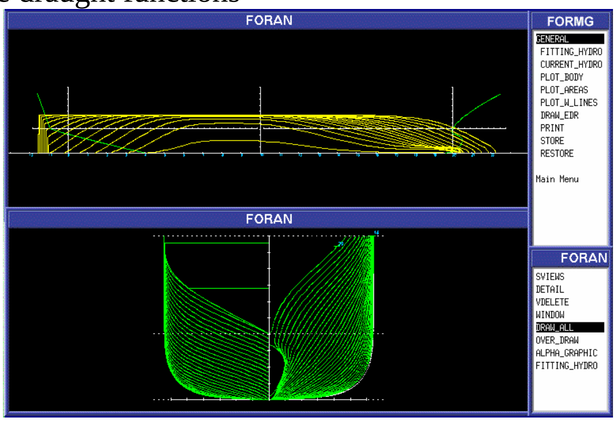
b) Hull geometry changes are displayed



c) Modification of the draught functions



d) Calculation of Hydrostatics



e) Hull Lines

Figure 22.42, Various stages of creating a hull definition using FORMG.

Foran appears to be a very capable hull design tool, however, as the software could not be reviewed directly it is difficult to judge whether the interface can be used to design hull forms efficiently, flexibly, if there are any limitations and if there is good feedback to the designer. Sener presents the system as intuitive design package which should be easy to learn. However, new users have found it difficult to use and require extensive training before the system can be used effectively.

Foran is a very large and versatile package and the concepts used to develop Foran and the design features in which it incorporates shows that the system is a highly sophisticated design tool, well worth the large expense required for purchasing and instalment.

23. APPENDIX 2 – YACHTLINES

23.1. Background

Several difficulties were encountered during the development of HullCAD, the precursor of PolyCAD. Being before the introduction of Delphi, the tool functioned in a DOS based environment and development tools of the time, excepting for Visual Basic, required too much expertise and knowledge to produce a Windows based application. The software was unable to provide the user with a capable interface for manual manipulation of NURBS surfaces due to the lack of good support for interface devices such as the mouse and keyboard and difficulties providing high quality (resolution) graphics across many machines. As manual interaction with the system was difficult, the parametric approach to hull form generation offered particular advantages as there are no requirements for the user to physically manipulate surface definition. As industry, in the development of parametric hull form generation tools, tends to concentrate on ship forms, techniques for generating yacht hull forms had not been investigated.

23.2. Approach

Experience gained with HullCAD on the use of NURBS surfaces has shown that, while the property of local modification was an advantage, the low number of definition control polygon vertices require to defined a yacht shaped hull surface required the user to spend a lot of time updating groups of points for a single geometric change. The NURBS surface representation, when applied to the definition of a yacht hull form, does not allow the user to design sectional shape independently. Jorde [20] had shown that parametric hull form generation could be implemented successfully using implicit cubic polynomials to represent hull sections in spreadsheet software. This practical approach to hull form generation was appealing because it did not require any complex mathematics and it took a much more geometric route than previous techniques. The use of curves to represent the hull form allows sectional shape to be controlled independently. Consequently, the technique does not have to dedicate significant resources into ensuring surface quality. However, implicit polynomial curves have several limitations when used to represent the sections of a hull form. Section shape is restricted to a certain form and the order of the polynomial functions are directly related to the number of constraints.

The experience gained from the development of HullCAD had shown that NURBS curves were very flexible entities and the control polygon combined with the knowledge of NURBS properties provided a very capable interface for controlling shape. Furthermore, the corresponding control vertices on the NURBS curves representing each section can be used to enable the consistent control of hull shape in the longitudinal direction. By arranging the corresponding control vertices on each of the curves to lie on inclined planes (Figure 23.3), the resulting form would be very similar to the Diagonal contours. Diagonal contours are used more significantly in yacht design as hull fairness when the vessel is heeled is important. By ensuring that the shape of "diagonal" curves through corresponding control vertices remains fair, certain hull sections can be used as control curves to the hull shape. The control sections can be manipulated independently and the other sections between will blend by the action of the diagonals. The hull form generation process becomes a structured form of lofting, (Figure 23.1).

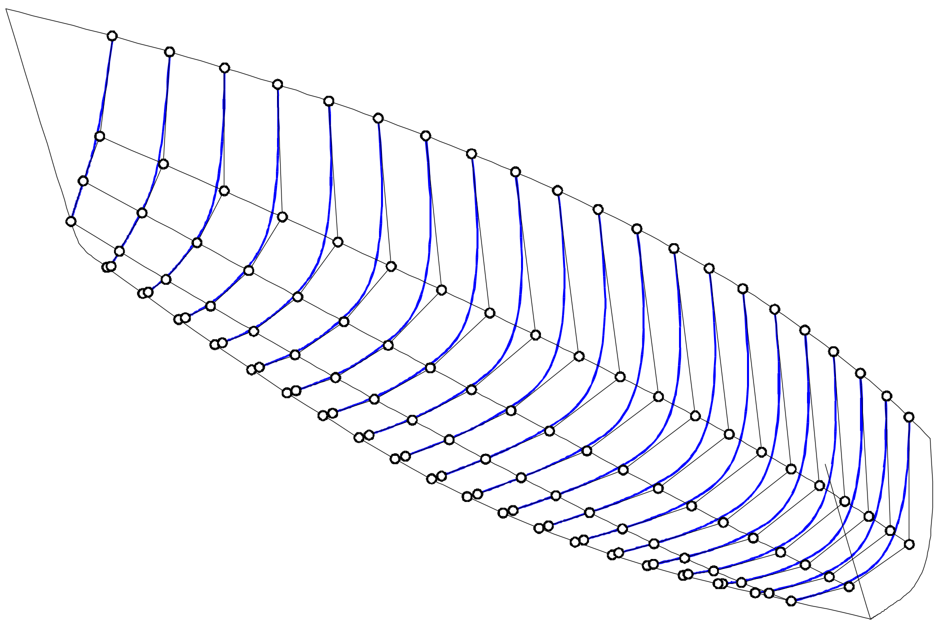


Figure 23.1, connecting the control vertices of the section curves to curves resembling diagonals develops a technique similar to lofting or skinning.

It would be possible to develop a hybrid design tool that allowed the user to manipulate the hull sections to design the hull. The resulting tool would function is a very similar manner to the Fairline [60] hull design software package. However, the technique would still require a lot of manual manipulation which could be replaced by generating the geometry parametrically. Parametrically generated longitudinal form curves, such as profile and section area curve representations for example, could be used to control the geometry of sections instead of using manual manipulation.

The control of the hull section curves to achieve the correct profile and deck shape is rather trivial. The end of a NURBS curve, used as a B-Spline with an open uniform knot vector, intersects the with the end vertices of the control polygon. However, control of the internal shape is more difficult because the curve does not go through the control vertices. Consequently, forming hull sections with the correct waterline breadth and immersed area becomes the primary task.

Previous hull surface generation techniques using NURBS representations, such as the method developed by Sanderski [32], have used mathematical solution techniques to control the shape within the boundaries of the NURBS. However, these mathematical solution techniques tend not to consider the shape requirements of the hull form very well. It becomes necessary to formulate the quality of the surface into some sort of mathematical fitness function. This increases the complexity of the solution process without ensuring that the resulting surface will be as intended. Hull surfaces generated by this process are always going to appear slightly unusual.

Rather than consider the development of a purely mathematical approach to achieving a hull form solution, a graphical and geometrical approach was desirable because the relationship between shape and how it can be controlled is stronger. Consequently, an iterative approach was selected to control the representation curves until the parametric targets are met and to form the specified geometrical shape. The main components of this technique are the hull representation structure, the parametric form definition curves and the iterative procedures control the shapes of the respective components.

23.3. Choice of Parameters

The parameters used to define the shape of the hull can be divided into two sets. There are the global parameters which can be used to define any marine vessel, and these include parameters such as waterline length (LWL) and block coefficient (C_B). In the other set there are the other parameters which are classed as local. These parameters affect the shape over small areas and are normally a function of the type of yacht being designed. As the technique will produce a modern style yacht hull form, which does not have any particular special features, the number of local parameters is minimised.

23.3.1. Overall Dimensions

The main dimensions are going to be the most important parameters. These will define the extents of the hull surface to be generated. The following parameters could be considered for inclusion:

- Length overall (L_{OA}).
- Waterline length (L_{WL} or D_{WL}).
- Maximum beam (B_{MAX}).
- Waterline beam (B_{WL}).
- Depth (D).
- Draught (T_C).
- Freeboard, at different points.

23.3.2. Form Coefficients

Once the extents of the surface have been selected, coefficients can be used to describe the effectiveness of the hull, especially with regards to controlling the generated hydrostatic qualities of the hull form. Non-dimensional coefficients can be very useful driving parameters for the technique because as the vessel is scaled these values remain unchanged allowing the displacement to remain relative. Particular form coefficients can be used in other roles. For example, in the YachtLINES software application, the Delft Series [61] is used to develop an optimum form shape. The Delft Series takes the main form coefficients as parameters, so the use of the same form parameters within the generation technique presents a great advantage.

The basic form coefficients that can be used to develop a yacht hull form:

- Block coefficient (C_B).
- Prismatic coefficient (C_P).
- Midship section coefficient (C_M).
- Waterplane coefficient (C_{WP}).
- Vertical prismatic coefficient (C_{VP}).

23.3.3. Centroids

The volumes and areas considered in the hydrostatics are associated with centroids. The locations of these centroids are an important consideration in the design, affecting stability, balance and the overall performance of the vessel. Usually, these quantities are considered as linear distances from the origin. However, it will be necessary to scale these values accordingly when the size of the vessel is parametrically changed. By considering these values as percentages of a reference distance, such as the waterline length, there is no longer any need to scale these values. Furthermore, the Delft Series also considers the position of centroids using non-dimensional percentages.

The centroids available are:

- Longitudinal centre of buoyancy (LCB), as a percentage of LWL.
- Longitudinal centre of flotation or longitudinal centroid of the waterplane (LCF), as a percentage of LWL.
- Vertical centre of buoyancy (VCB), as a percentage of the hull draught T_C.
- Centre of Lateral Resistance (CLR), as a percentage of LWL.

23.3.4. Local Parameters

While the global parameters are capable of controlling the major factors affecting the design of the hull, they do not consider the subtleties of the design like the control of shape. The following parameters were selected to control the shape characteristics of the stem, deck and transom:

- Bow profile - angle at deck.
- Bow profile - deck tangent.
- Bow profile - angle at WL.
- Bow profile - tangent above WL.
- Bow profile - tangent below WL, (adjusts forefoot depth).
- Transom angle.
- Aft Extreme - above WL.

- Freeboard, at different points.

It was felt that there were too many parameters defining the shape of the bow profile. Consequently, the number of bow profile parameters was reduced to two; one to vary the curvature in the bow by adjusting the angle of the tangents and the other parameter to control the tangent of the bow profile below the waterline.

As some of the available parameters are co-dependent on other parameters listed and to reduce the number of local parameters, the set was reduced to independent parameters and a minimum number of local parameters. The following parameters are used to generate the hull form within YachtLINES.

- Length overall (L_{OA})
- Waterline Length (L_{WL})
- Maximum Breadth (B_{MAX})
- Waterline Breadth (B_{WL})
- Breadth at Transom ($B_{TRANSOM}$)
- Draught (T_C)
- Forward Overhand (F_{OVH})
- Freeboard at Bow ($Free_{BOW}$)
- Freeboard at Midship ($Free_{CENTRE}$)
- Freeboard at the Aft Perpendicular ($Free_{AFT}$)
- Stem tangent angle (BPDA)
- Forefoot tangent length (BPLWLT)
- Angle of the transom plane (TransomAngle)
- Prismatic Coefficient (C_P)
- Midship Coefficient (C_M)
- Longitudinal Centre of Buoyancy (LCB)
- Angle of the Deck Tangent at the Aft Perpendicular (ApAngle)

23.4. Hull Form Section Representations

A yacht section has quite a characteristic arc shape. The lack of any particular shaped features in a yacht hull surface makes it a great deal more complex to form the correct shape. However, with considerations for the number of numeric parameters that are available, the overall shape of the section and the particular control structure formed by the diagonals, it is possible to develop an effective solution. By reviewing the number of constraints, (Figure 23.2), on the shape of a hull section, the number of control vertices and the location for the diagonals can be developed (Figure 23.3):

- Control at the deck line (1 Vertex)
- Control at the centre line (1 Vertex)
- Control of the curve to form the waterline breadth (1 Vertex)
- Control of the curve to form the specified immersed area and control shape (2 Vertices)
- Control of tangency across the centreline (1 vertex)

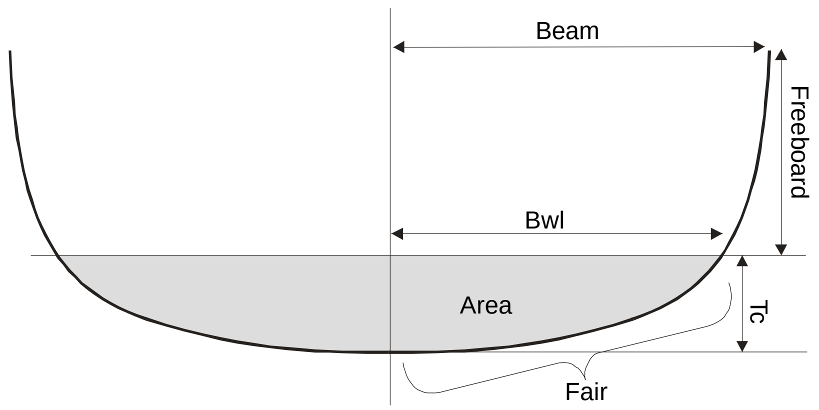


Figure 23.2, the shape of a section is based on various parameters and must have a fair shape,

The control polygon arrangement for the definition of a hull section is constructed using the following rationale: Two vertices are required to form the ends of the section. These will be located on the deck and the profile form curves. One vertex is to control the tangent of the curve at the profile, to ensure that the section curve intersect the centre plane perpendicularly. Sections in the bow of the hull will need to be quite a sharp ‘V’ shape, but with a small radius into the profile plane. Consequently, this vertex is located on a diagonal that will keep the tangent length small. The remaining vertices control the shape of the curve within the mid part of the section.

The most effective control of section shape is achieved if the vertices move approximately normal to the curve. As a result, the angles of the diagonal planes are chosen to best maximise the arrangement to achieve this. To control the waterline breadth, the diagonal is located to intersect with the waterplane at the half waterline breadth of the midship section. The inclination of the diagonal plane is based on an angle chosen to ensure that the control vertex will be approximately normal to the section curve. The plane intersects with a point one quarter of the waterline breadth above the waterline on the centre plane, resulting in an inclination of 63.4 degrees. To control the immersed area of the section, one vertex is located on 45 degree diagonal plane intersecting with the waterline at the centre plane. This angle was chosen based on the frequency of its use within yacht hull design. The final vertex is located on a diagonal plane which forms the locus between the 45 and 63.4 degree diagonals. It has an angle of 54.2 degrees intersecting at a point an eighth of the waterline breadth of the midsection above the waterline, at the centre plane. This arrangement is illustrated graphically in Figure 23.3.

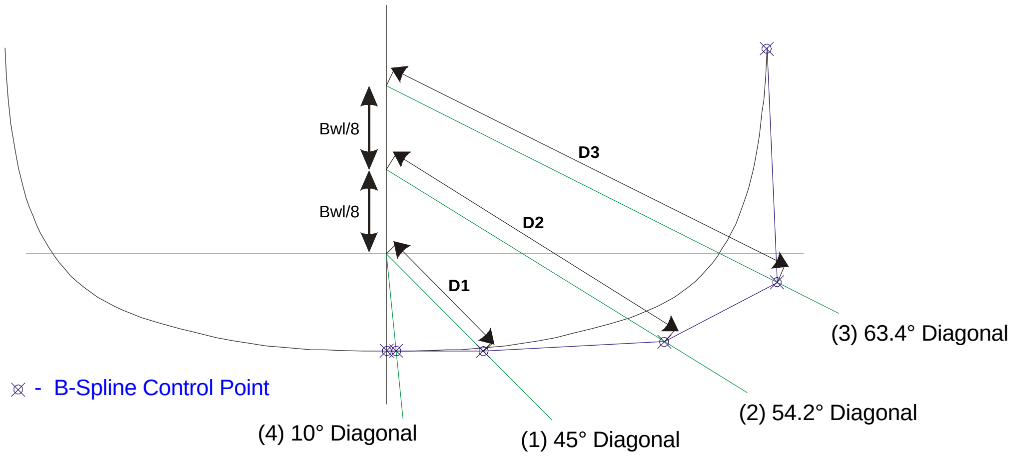


Figure 23.3, the arrangement of the control vertices lying on the inclined (diagonal) planes

The shape of the hull form is built up in stages. The diagonals are used to minimise the number of sections that must be generated purely from parameters alone. Consequently, only the midship, aft perpendicular and quarter sections are parametrically generated. A variety of different approaches are used to generate these sections.

The midship section is generated first. The control vertex on diagonal three is used to control the waterline breadth and the vertices on diagonals one and two are used to control the immerse area. To reduce the number of control parameters down to two, the gradient of the segment line between diagonals one and two is controlled on the basis of the midsection area coefficient. A simple rule for the gradient is developed: If the value of the coefficient is one then the shape of the section should be rectangular and the line segment should be horizontal. If the coefficient is zero then the gradient should be vertical. For a value of 0.5, the gradient should be the same as the line through baseline to the waterline at the waterline breadth. The shape of the section is generated by iterating the parameters controlling the waterline breadth and the immersed area sequentially, until a solution is reached.

The aft perpendicular section does not require any iterative procedure to generate the shape. By definition, as diagonals one and four cross the centre plane at the waterline, the control vertices for these diagonals are located on the centre plane. Consequently, to ensure that the section leaves the centre plane perpendicularly, the control vertex on diagonal three is positioned to develop the appropriate tangent. The control vertex on diagonal three is controlled directly by a parameter specifying the flare angle at the deck.

Generating the shape of the quarter sections was found to be quite a complex task. However, a basic shape of the section can be found by developing an initial set of diagonal curves. Based on the shape of these curves, it was found that is best not to manipulate the control vertex on diagonal three, as a good waterline shape was produced by the fitted diagonal. The control of the immersed area uses a similar technique to the one used to control the midship section curve. However, if the position of the control vertex on diagonal one was found to be below the base of the section at the end of the iteration, the position of the vertex is relocated to the height of the base of the section and the section is regenerated using the location of the control vertex on diagonal two only. The arrangement ensures that the section remains concave, (Figure 23.4).

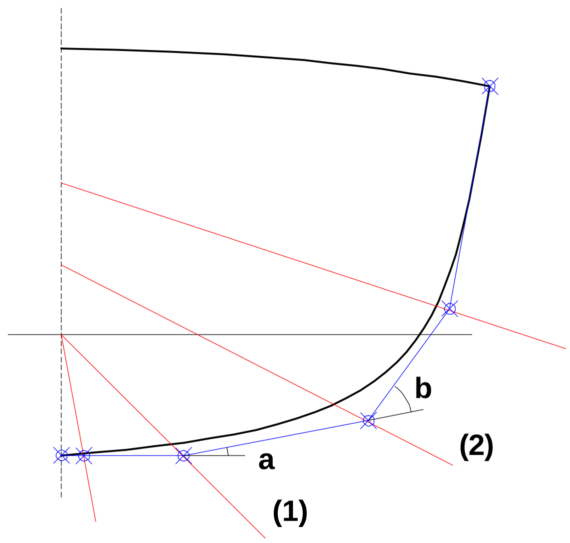


Figure 23.4, quarter sections are maintained in a concave shape by considering the external angles between line segments.

23.5. Longitudinal Curves of Form

The longitudinal curves of form control the shape of the hull surface along the length. Each curve controls the value of one of the parameters affecting section shape. The form curves are completely defined using numerical parameters removing the need to manually manipulate any of the hull form definition. The following form curves are used to control hull shape:

- Deck and Sheer Profile Curves.
- Profile Curve.

The waterline and a representation of the hull section area were originally used as curves of form in earlier tools used to research this technique. However, it was found that to force the hull to have a waterline or section area shape defined by an independent curve generation procedure resulted in very unsuitable shapes. The geometry structure used to build up the hull shape, by definition, already has a section area and waterline curve shape and the separate waterline and section area curves of form are unable to take account of the shape of the hull surface and unable to consider the fairness of the hull form produced. Consequently, the separate curves of form representing the section area and waterline shape were removed in favour of more direct analysis of the shapes resulting from the hull form generation procedure.

23.5.1. The Deck and Sheer Curves

While the shape of the deck line curve does not affect the hydrodynamic performance of the yacht, it probably has the greatest control over the formation of a pleasing shape. In yacht design, this is a very important characteristic. Although the physical shape is one line, the representation is three dimensional, varying in both the water plane and centre plane directions. Consequently, two curves are used, one to model the shape of the sheer and the other the shape of the deck.

There are various methods of producing the shape of the sheer for a hull. A B-Spline fit, (Figure 23.5), through three points of freeboard was selected because the resulting curve had the most pleasing shape and the technique allowed for a good deal of flexibility as it is capable of developing straight and inverse sheer in addition to the standard curved shape.

The development of the deck form curve requires a more involved procedure. The two end points of the curve are known, being located at the bow and at the half breadth of the transom. The remaining specification for the shape of the curve is that it has a maximum offset from the centre plane of half the maximum breadth. A curve fitting procedure could form the curve so that it intersects with a point on the maximum half breadth. However, this approach would not ensure that the deck curve would not be wider. Consequently, an iterative approach was selected which transversely varies a control point, located at amidships, until the maximum offset from the centreline is the half the specified maximum breadth.

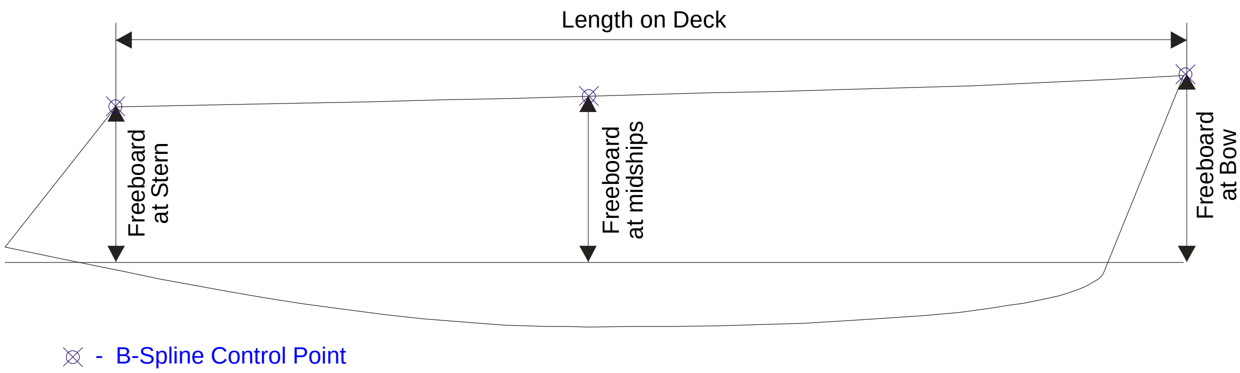


Figure 23.5, the sheer curve is formed by fitting a B-Spline curve through three parametrically controlled points of freeboard.

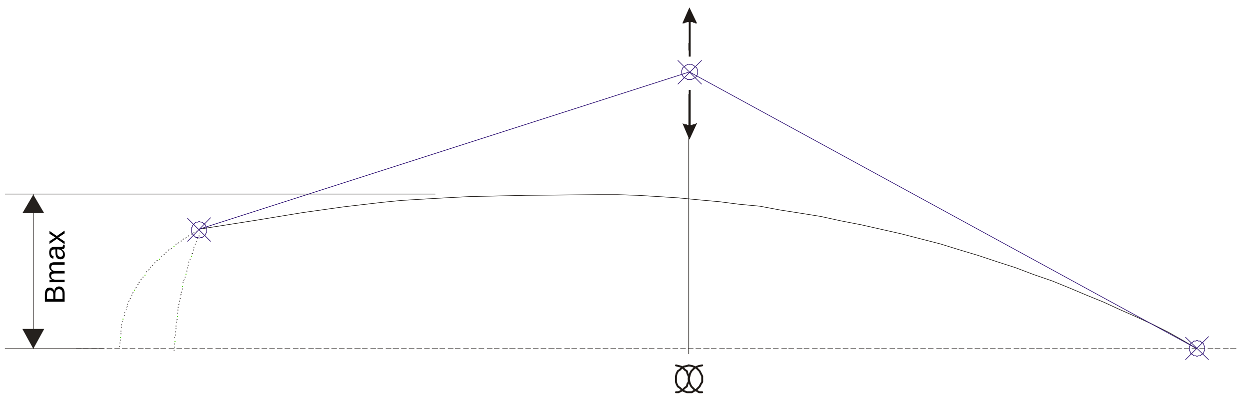


Figure 23.6, limiting the deck curve to the maximum half breadth is achieved by using an iterative procedure to transversely manipulate a control vertex located amidships.

23.5.2. The Profile

In modern yacht hull forms the shape of the profile is a great deal simpler than the classic yachts with integrated keels. Even so, the profile shape is made up many different components. A modern yacht will usually have a straight, inclined stem down to a sharp radius at the forefoot. As the profile moves toward the stern, a gentle curve shape is used to increate the draught of the hull form up to the midship section and then reduce back to the waterline at the aft perpendicular.

Depending on the arrangement, the curve should be continued on in a similar style to the bottom of the transom. The global parameters affecting the shape of the profile are shown in Figure 23.7. However, more parameters are used to control the shape of the profile at the stem.

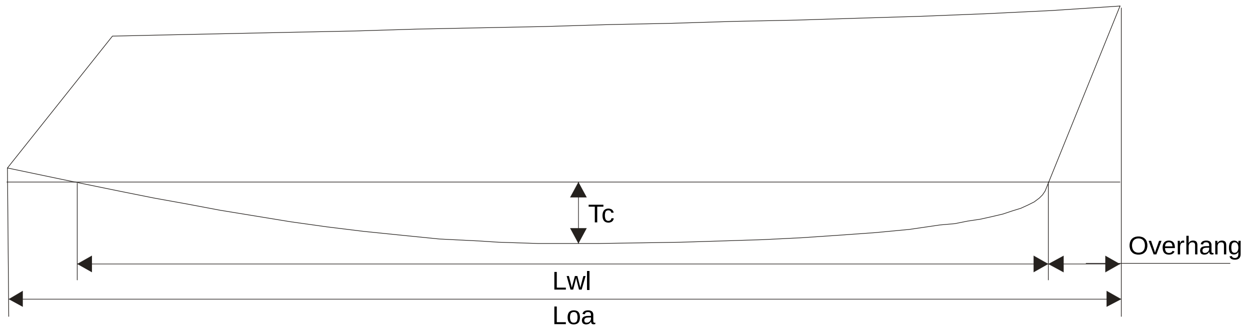


Figure 23.7, the global parameters affecting the shape of the profile.

A single cubic B-Spline curve is used to model the shape of the profile from the stem to the base of the transom. Cusps are formed to ensure that the curve intersects with certain points along the length without the need to use iterative procedures and to ensure that the profile keep a characteristic shape. The tangents are kept linear across the cusp so that a knuckle point is not formed. The arrangement of the control polygon for the profile form curve is show in Figure 23.8.

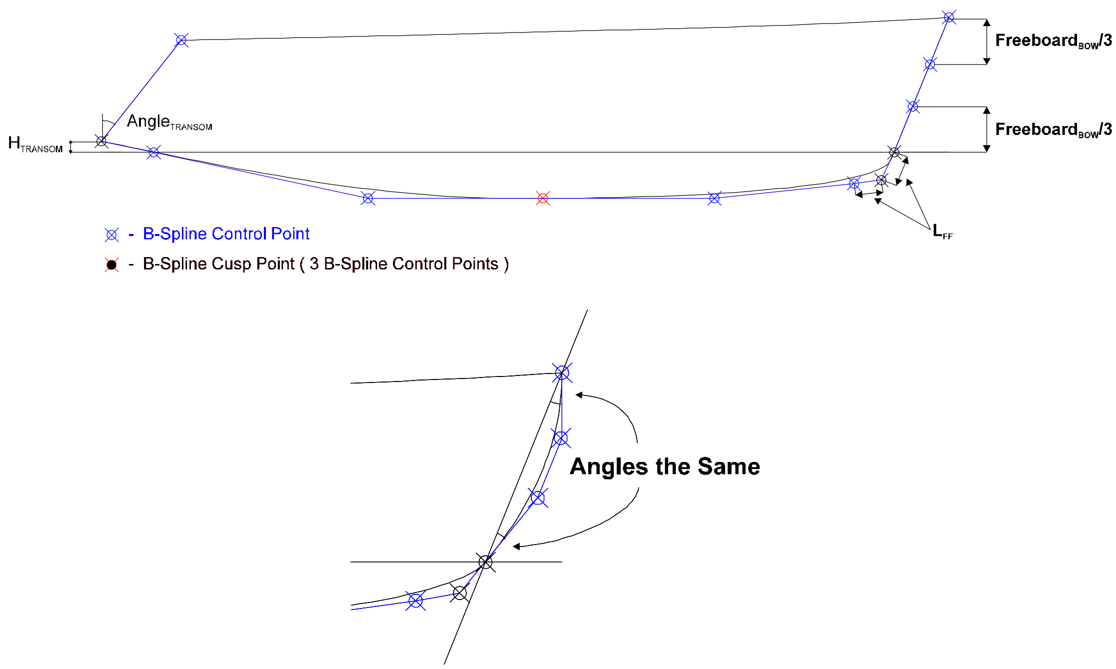


Figure 23.8, the B-Spline control polygon defining the profile shape.

23.6. Diagonals and Transom

Once individual sections can be generated, it necessary to use some means to transfer sections shape information along the length of the hull. Twenty-one sections are used to represent the shape of the hull form, located at station positions. A curve through consecutive control polygon vertices on each section can be used as a representative diagonal on the hull form shape. If this curve shape remains fair, then it follows that the hull sections developed by the procedure will also be fair and the sections will form a "family" of curves. Fairness in the shape of the diagonal curves is more easily achieved if the number of points forming the curves is kept small. Hence, only four stations are controlled to form the shape of the hull form using the section generation procedure previously discussed. Once these control sections have been generated, the remaining sections can be developed by considering vertex locations formed by the intersection of the station plane with the diagonal curves, (Figure 23.9). A B-Spline fit procedure is used to form the fair diagonal curve shapes. Only the first three diagonals are considered in the fitting process. The control vertex on diagonal 4 is such that the tangent of section curve is normal to the centre plane and does not, therefore, require a curve to be generated.

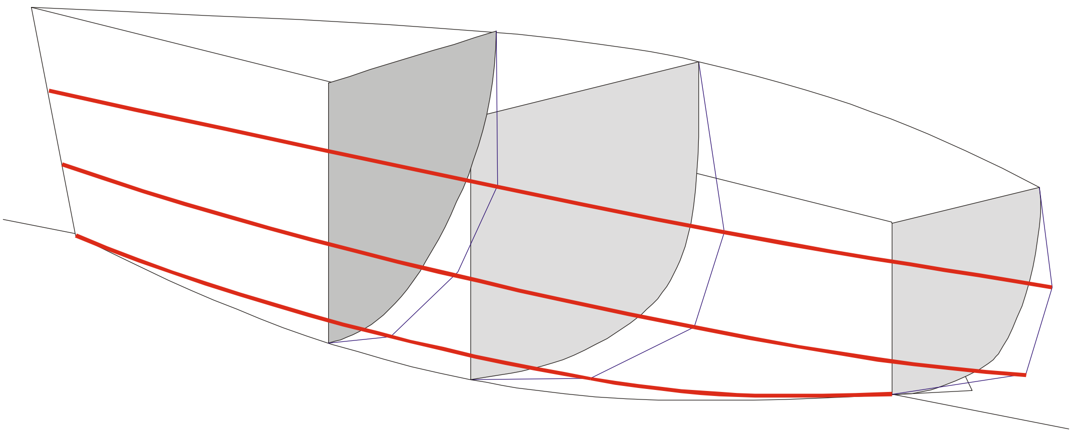


Figure 23.9, the formation of a section control polygon based on points found from a planer intersection with the diagonal curves.

While the midship and the aft perpendicular sections are generated entirely from parameters, the quarter sections are developed using the initial set of diagonals formed through using midship and aft perpendicular sections and the locations where the diagonal planes intersect with the profile curve, (Figure 23.10).

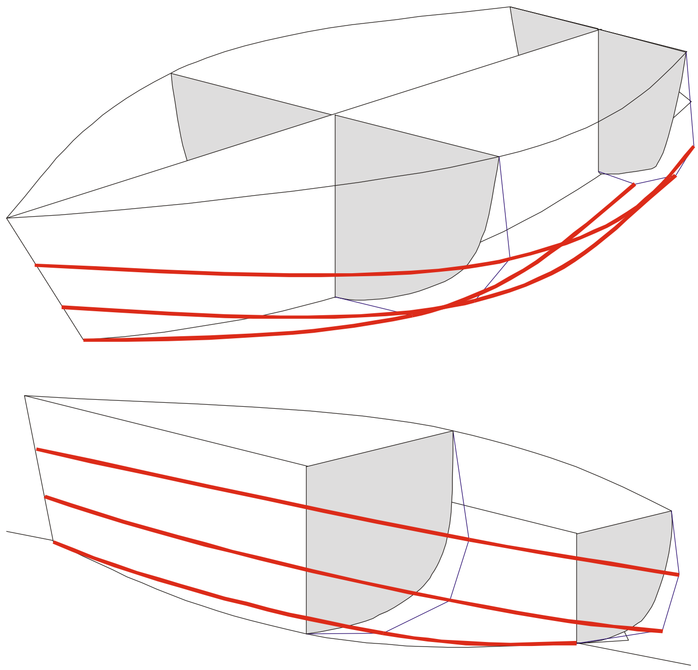


Figure 23.10, the initial arrangement of the diagonal curves used to form the quarter sections.

After the quarter sections have been formed, a final set of diagonals can be developed and the remaining stations of the hull form generated, (Figure 23.11).

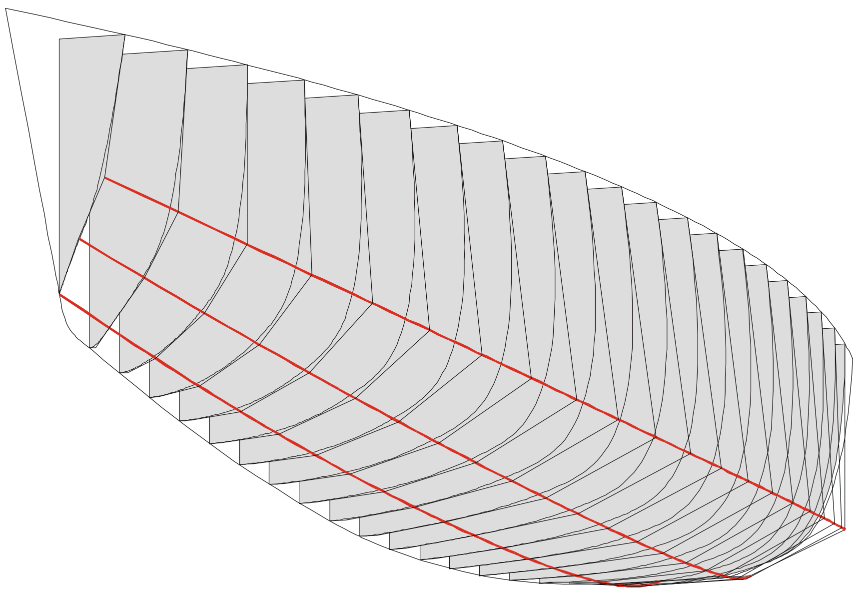


Figure 23.11, the complete set of sections developed with corresponding control vertices attached to the diagonal curves.

While the diagonals formed between the control sections allow all station curves to be generated, the B-Spline fit technique does not have the functionality to enable the curve to be extrapolated aft to form the transom. As the shape of the transom is an extension of the shape throughout the station curves, it is appropriate to use an alternative method to extend the diagonal curves. A Least-Squares regression curve, of quadratic order, can be fitted to the control vertices of all stations between the aft perpendicular and the midship section. The shape of the diagonal curves can then be extrapolated back to form the transom shape. If the transom surface is assumed to be a plane, the intersection points between the extrapolated diagonals and the transom can be found. The points can then be used to form the vertices of the control polygon, (Figure 23.12).

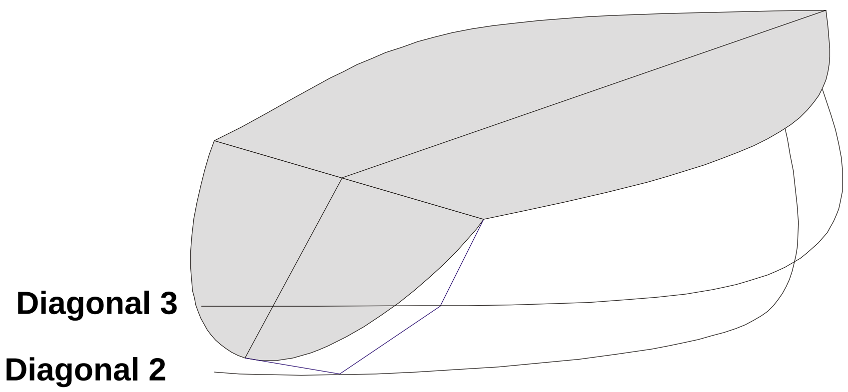


Figure 23.12, A Least-Squares technique is used to extrapolate the shape of the diagonal back to the transom plane.

Control vertices on diagonals 1 and 4 will not intersect with the transom. Consequently, the vertices are located at the end of the curve at the base of the transom and the shape of the curve will not be affected. This is also the case when an intersection cannot be found for the control vertex on diagonal 2. Under the specification of certain parameters it is possible to get unusually shaped transoms. This should not be regarded as a failure, as in these conditions, it is likely that the transom has been projected from a hull that was specified with unsuitable parameters.

23.7. Producing the Hull Form

23.7.1. The Hull form Generation Procedure

The primary aim of the generation technique is to develop a hull form with the desired hydrostatic qualities. Most of the processing is dedicated to the development of the correct displacement and longitudinal centre of buoyancy (LCB). However, before this processing begins, the form curves must be developed. Once this information is available, the midship and aft perpendicular sections can be developed and an initial set of diagonal curves fitted. With this information the quarter sections can be developed and modified to enable the target displacement and LCB to be achieved.

Section area coefficient is used to control the shape of the quarter sections. The iteration procedure modifies the value of the section area coefficient parameters until the desired displacement and LCB is met. Once this has been completed, the final task is to develop the transom shape. The generation procedure is illustrated using a flow chart in Figure 23.13. The diagram also shows the development of the quarter sections as a subtask.

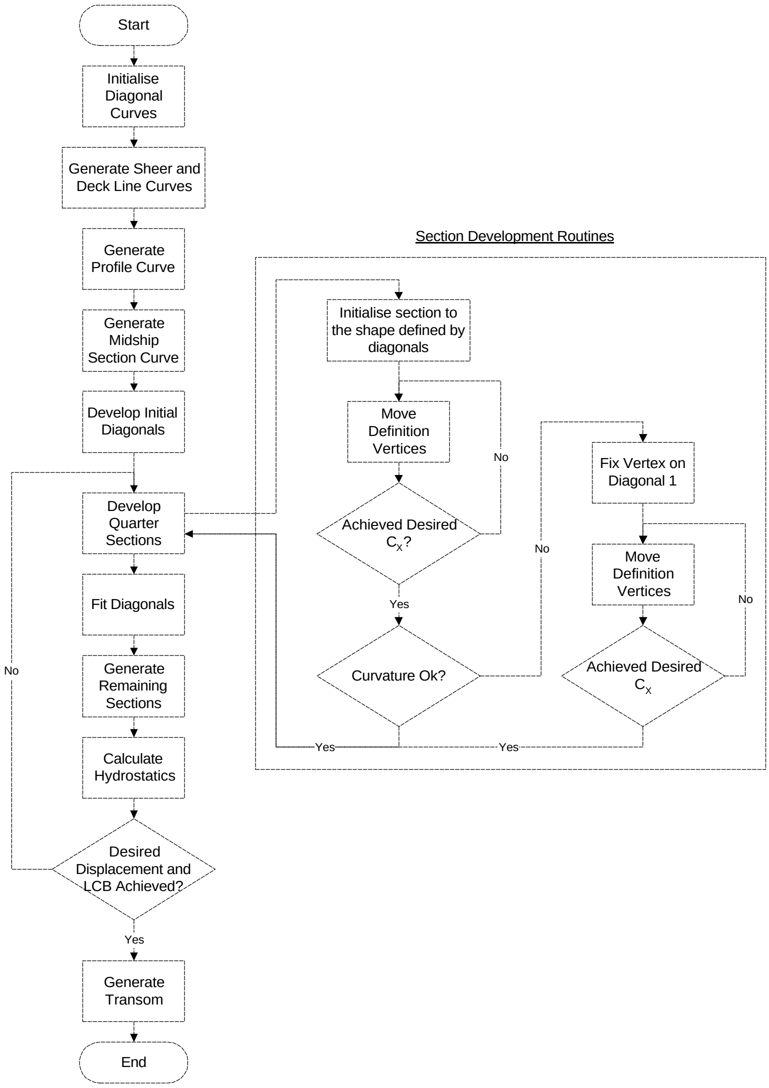


Figure 23.13, a flow chart overview of the YachtLINES hull generation technique.

23.7.2. The Iteration procedure

The design of the process of attaining the goals is very important. Previous hull generation systems have used the Newton-Raphson technique to obtain solutions. The Newton-Raphson is considered one of the most efficient techniques for obtaining, numerically, a solution to a function. However, to use the Newton-Raphson technique with any success one requires a mathematical function which can be differentiated. A differential function for a B-Spline function can be obtained, but it is much more complex to calculate than the original B-spline function. So, to attain goals in the current hull generation method, a practical engineering approach was taken, by using simple standard linear interpolation within the iteration technique over previous results.

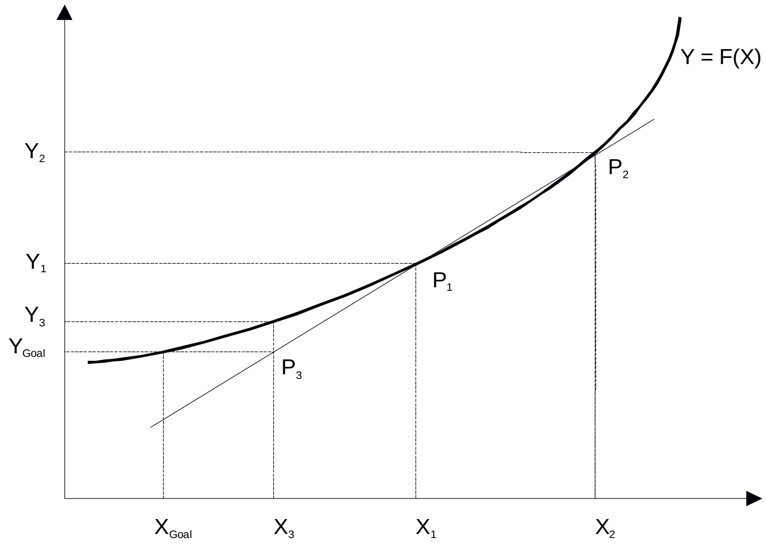


Figure 23.14, the structure used in linear interpolation.

Considering Figure 23.14, the value of X_1 is sent to the iteration procedure when it is initially called. X_1 is a variable which can be used to vary, for example, the prismatic coefficient C_P. The iteration process calls the modification function with X_1 as a parameter. The modification function takes the form of Y_n=F(X_n). The result of the modification function, Y_1, in this case will be the returned value of C_P. Next, as part of the initialisation process, the iteration procedure increases X_1 by 5% to obtain a value X_2. X_2 is passed to the modification function which returns a C_P in Y_2. Now a line can be placed through the points P_1 and P_2 with co-ordinates (X_1, Y_1) and (X_2, Y_2). The line can be extended so that the required prismatic coefficient, Y_{GOAL}, can be used to find the value X_3. X_3 is now passed to the modification function. If Y_3 is close enough to Y_{GOAL}, then the required C_P has been found and the value X_3 can be used to create the hull. If the value of Y_3 is not close enough to Y_{GOAL}, a line is drawn though P_2 and P3 and the process repeats. The linear interpolation technique may fail if applied to a curve with complex shape. However, it is assumed that most of the curves iterated in the hull generation procedure are fairly simple. The iteration procedure is illustrated as a flow chart in Figure 23.15.

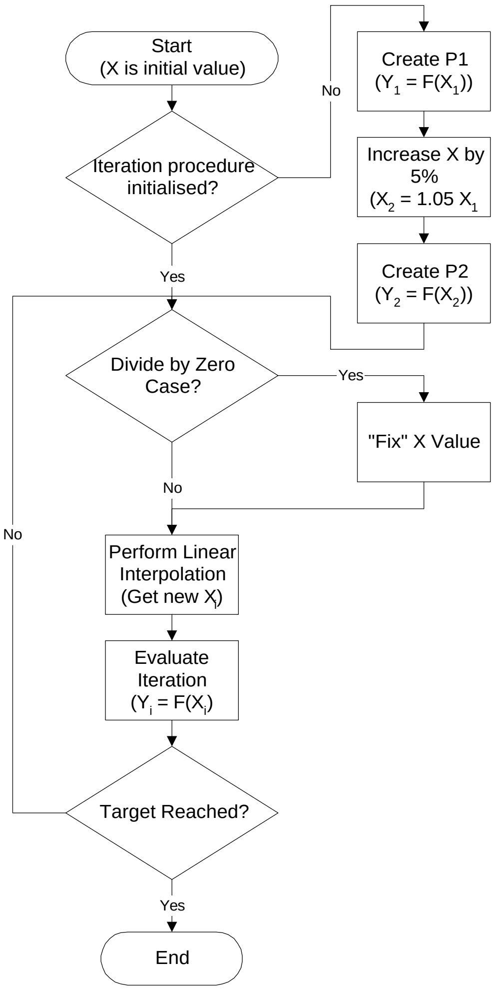


Figure 23.15, a flow chart of the iteration procedure.

23.7.3. Development of a Surface Representation

A further development of the hull generation procedure is the creation of a surface representation. The generation of a surface from the type of offset data that this technique produces can be a very complex and involved task. Developers have attempted to develop NURBS surface representations using Least-Squares techniques [43] and recently using Geometric Algorithms [27]. However, it is not necessary to consider these approaches as the structured geometry that represents the hull form sections already uses the NURBS representation and can be easily transformed into a surface.

To develop the surface representation, the fact that all sections are represented by uniform B-Spline curves with six control vertices means that the same knot vector is used throughout the longitudinal shape. The B-Spline fit technique that is used to link the corresponding control vertices together on each section to form the diagonal can be extended to fit through all rows of corresponding vertices. If the fit is performed to result in uniform B-Spline curves, the resulting control vertices on each longitudinal curve can be combined to form a mesh. By the tensor product nature of NURBS surfaces, the representation produced by this surface will closely match the original sections generated by the technique. It is not necessary to get a definite accurate match as the technique is a hull generation procedure. If a surface can be generated, there is no need to consider the shape of the sections. However, one area that cannot be represented well by this approach is the stem. This part of the hull is generated with more degrees of freedom that can be represented by the six control vertices that span the transverse direction of the surface. It should be possible to develop the correct stem shape by considering techniques to insert more control vertices. However, as the technique would remain a parametric hull generation tool, affected by the shortcomings highlighted by the work on TSCAHDE, the resulting tool would still be largely ineffective for design purposes.

23.8. Implementation

The YachtLINES technique was developed using Turbo Pascal 7.0. Due to the lack of software that would allow a good review of each generated hull form to be made, the development tool used a graphical implementation to enable the iteration process to be demonstrated and the final results analysed, (Figure 23.17). The iteration process of each generated form and section curve is displayed allowing the progress to be reviewed and to allow the execution to be stopped if the results have become unstable. Parameters can be interactively modified, (Figure 23.16), and the generation process can be reinstated.

Once satisfactory hull forms could be developed by the technique, the introduction of Borland Delphi allowed the code to be implemented within a Windows environment without the need for significant change, (Figure 23.18). With multiple windows and the graphical nature of the environment, modification of parameters and the generated hull form in two or three dimensions can be displayed at the same time. Furthermore, the software allows hydrostatic calculations to be performed on the hull representation and the geometry can be exported to various exchange file formats. To demonstrate the effectiveness of the technique when used for optimisation, a utility to analyse the hull form with respect to the Delft series is included. The tool can be used to search for optimum values of prismatic coefficient and longitudinal centre of buoyancy with respect to the other parameters used to define the hull form.

The development of PolyCAD as an environment that allows geometry to be transferred between representations provides an ideal platform for the YachtLINES technique to be implemented, (Figure 23.19). The implementation code was upgraded to use the standard PolyCAD library of geometry tools and the technique is presented using the surface version. While the surface format cannot represent the exact shape of the hull represented by the sections because of the additional complexity in the stem shape, the surface representation is more useful as the contour lines can be generated for display and for hydrostatic calculations. Furthermore, the surface can be rendered graphically as is or coloured with respect to the mean or Gaussian curvature. As the hull surface is developed using the standard library, the generation geometry can be extracted and independently reviewed and manipulated as any entity can. While it is possible, the generated hull surface is not updated by these changes. However, a fully functioning implementation of the TSCAHDE approach could, of course, accommodate this arrangement.

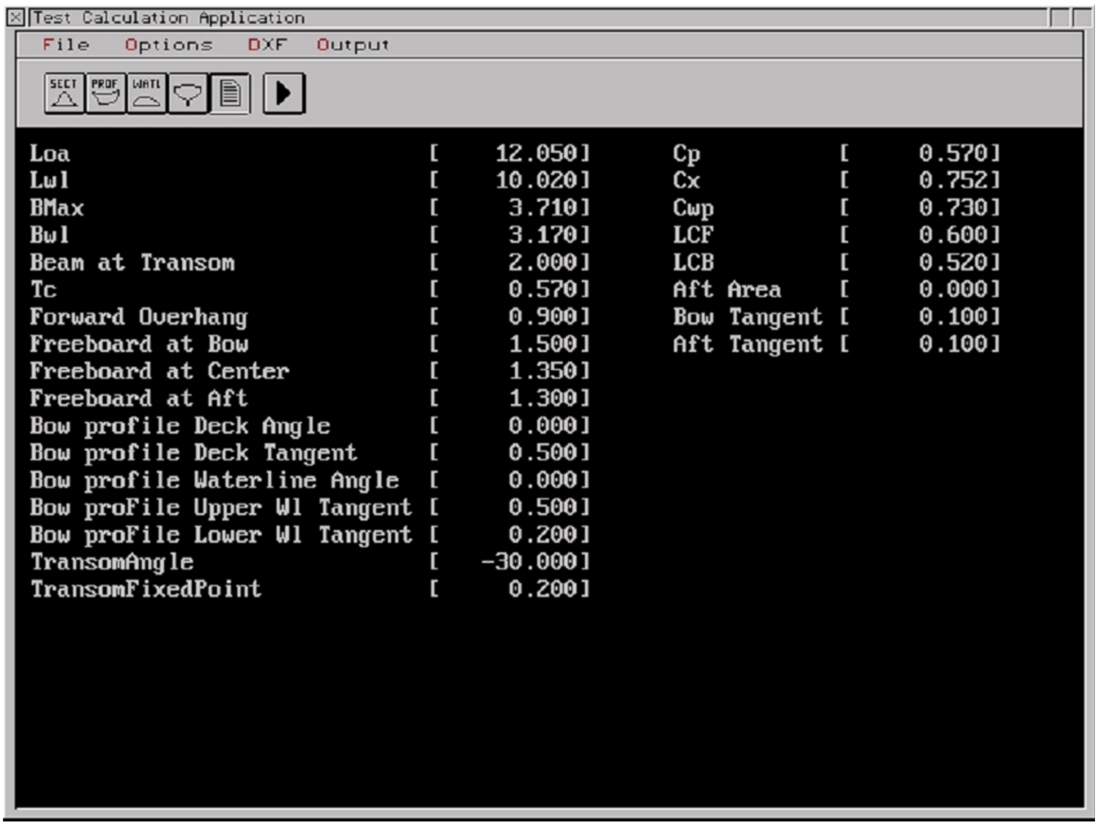


Figure 23.16, within the development, software parameters can be interactively changed and the hull form updated.

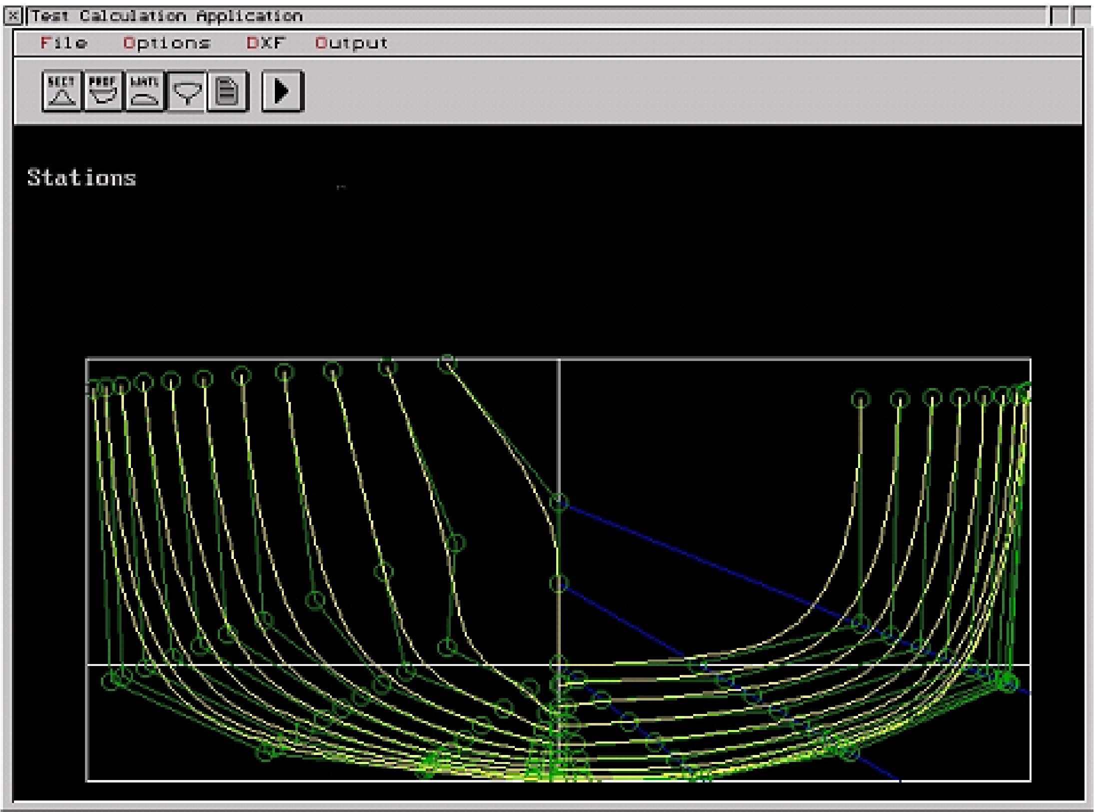


Figure 23.17, the software graphically illustrates each of the iterations as they are generated.

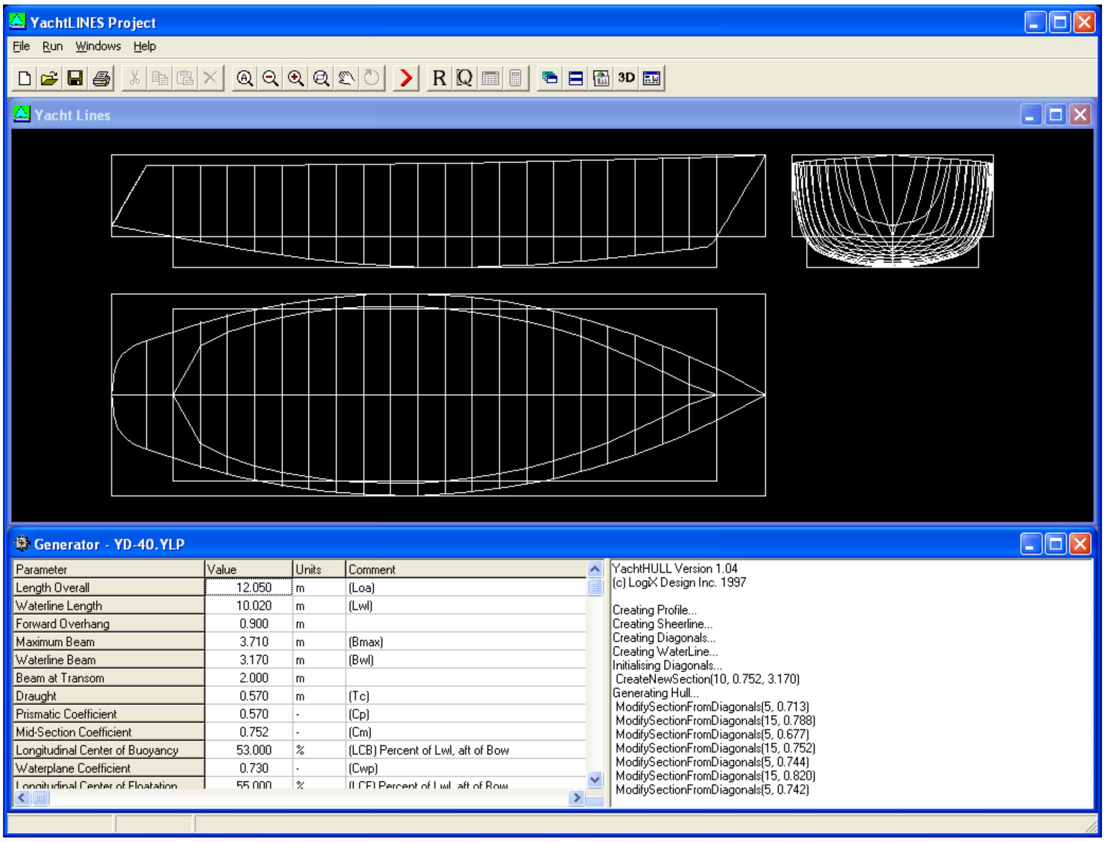


Figure 23.18, the YachtLINES software demonstrating a generated hull form.

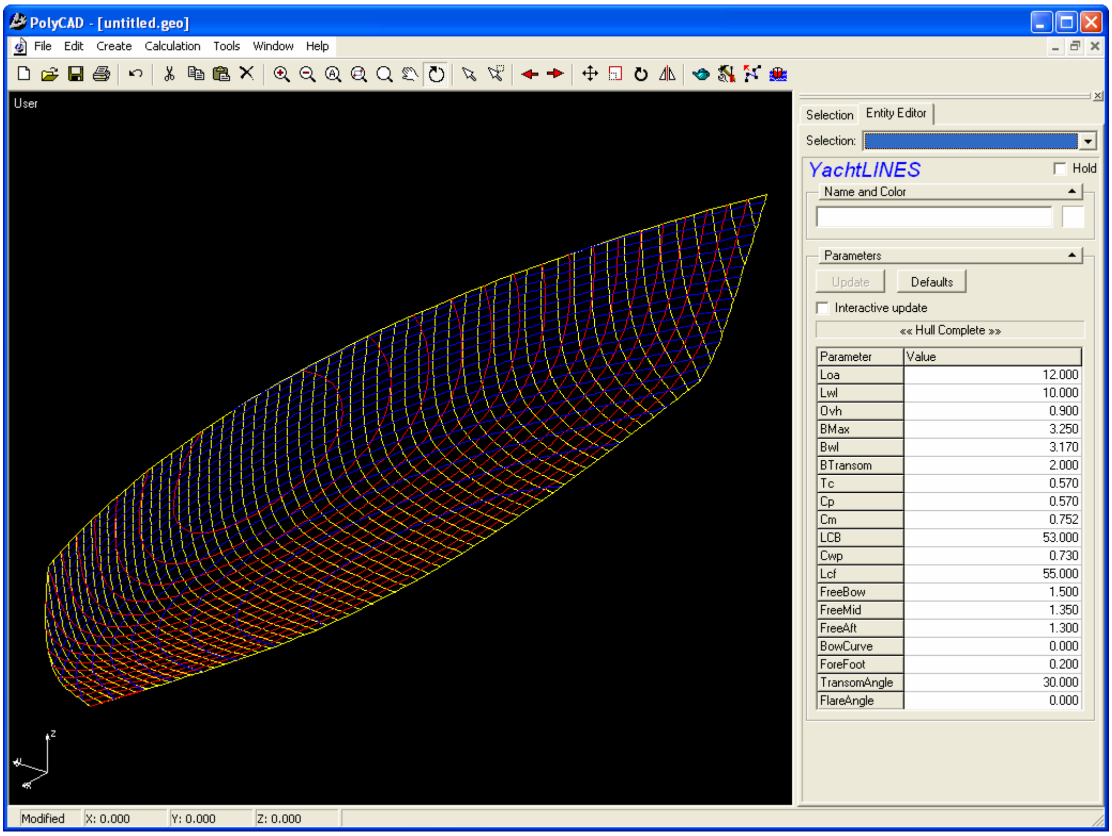
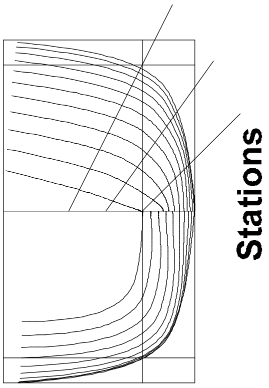
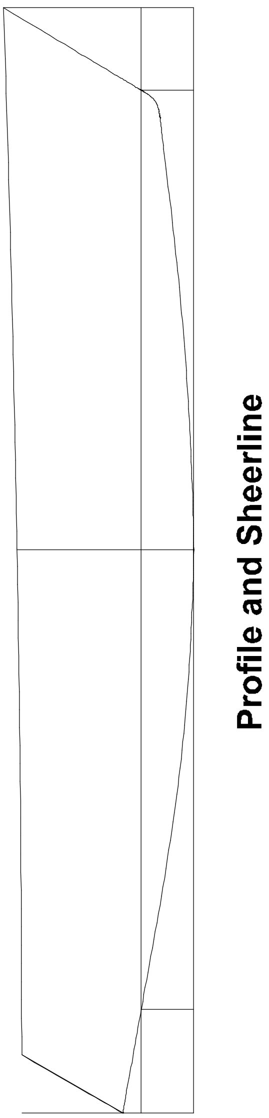


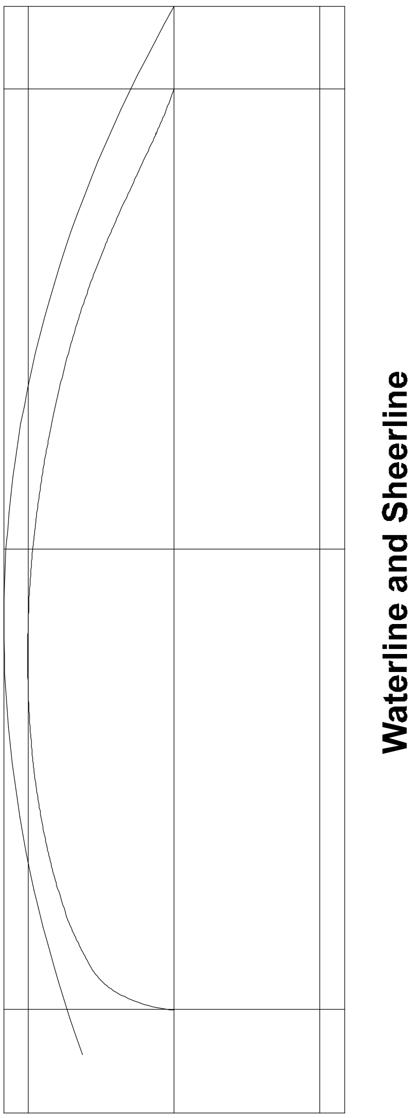
Figure 23.19, the surface form of the YachtLINES hull generation technique, upgraded and implemented within PolyCAD.

23.9. Examples



Output From Hull Generation Method		
Method	N° 3.	
Parameter Basis	YD-40 Hull	
Parameters		
Loa	12.050	m
Lwl	10.020	m
Bmax	3.710	m
Bwl	3.170	m
Beam at Transom	2.000	m
Tc	0.570	m
Forward Overhang	0.900	m
Freeboard at Bow	1.500	m
Freeboard at Centre	1.350	m
Freeboard at Aft	1.300	m
Cp	**0.570**	-
Cm	**0.752**	-
Cwp	Not Used	-
LCF	Not Used	% of Lwl
LCB	**53.000**	% of Lwl
Bow Profile - Deck Angle	0.000	°
Bow Profile - Deck Tangent	0.500	m
Bow Profile - Waterline Angle	0.000	°
Bow Profile - Upper WL Tangent	0.500	m
Bow Profile - Lower WL Tangent	0.200	m
Transom Angle	-30.000	°
Transom Fixed Point	0.200	m on WL





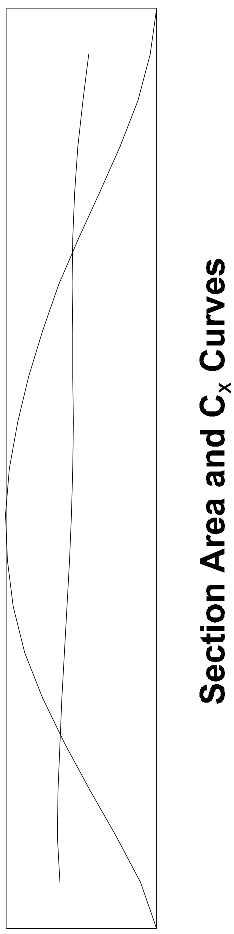
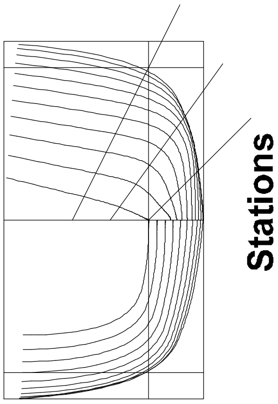
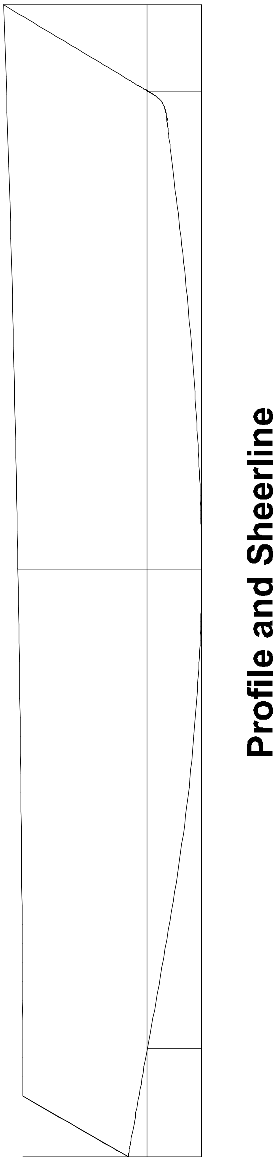
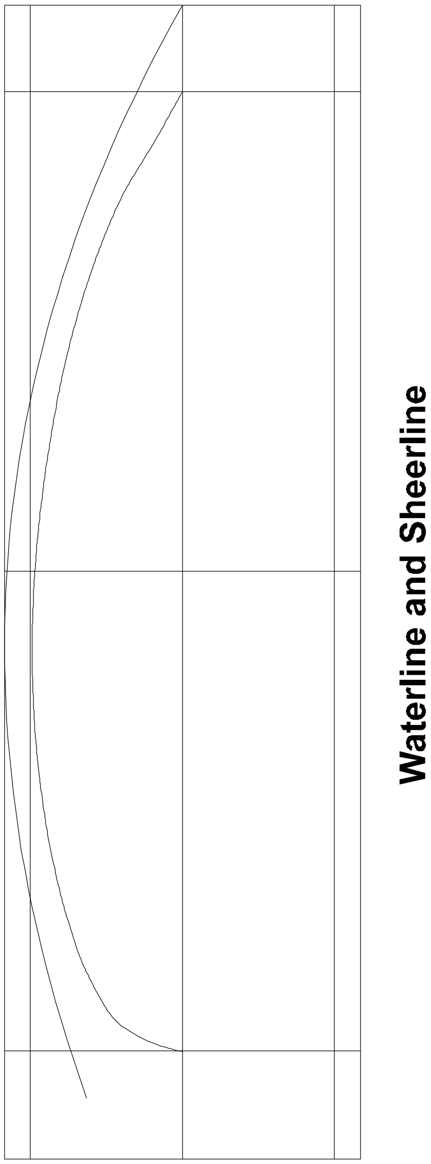


Figure 23.20, C_P = 0.57, LCB = 53%.



Output From Hull Generation Method	
Method	Nº 3.
Parameter Basis	YD-40 Hull
Parameters	
Loa	12.050 m
Lwl	10.020 m
Bmax	3.710 m
Bwl	3.170 m
Beam at Transom	2.000 m
Tc	0.570 m
Forward Overhang	0.900 m
Freeboard at Bow	1.500 m
Freeboard at Centre	1.350 m
Freeboard at Aft	1.300 m
Cp	**0.570** -
Cm	**0.752** -
Cwp	Not Used -
LCF	Not Used % of Lwl
LCB	**50.000** % of Lwl
Bow Profile - Deck Angle	0.000 °
Bow Profile - Deck Tangent	0.500 m
Bow Profile - Waterline Angle	0.000 °
Bow Profile - Upper WL Tangent	0.500 m
Bow Profile - Lower WL Tangent	0.200 m
Transom Angle	-30.000 °
Transom Fixed Point	0.200 m on WL





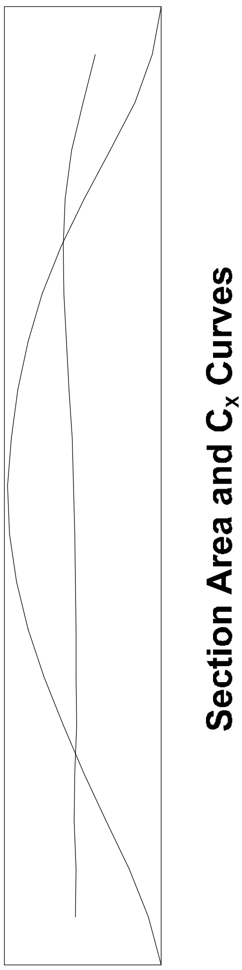
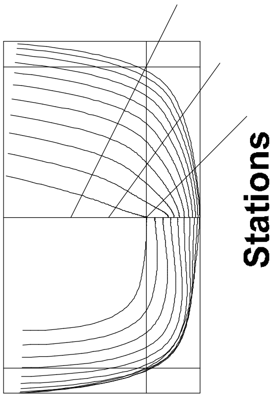
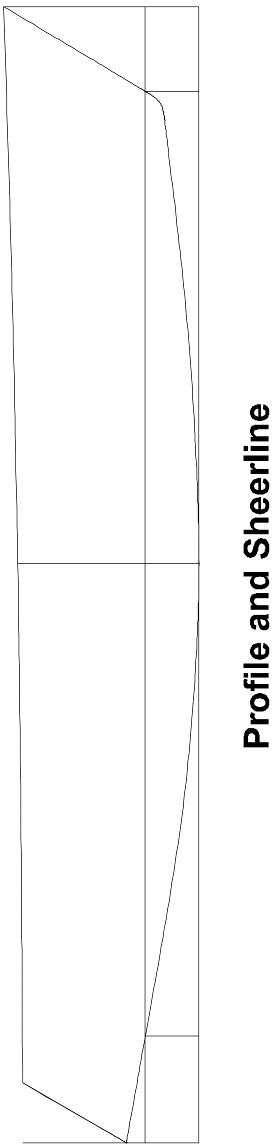


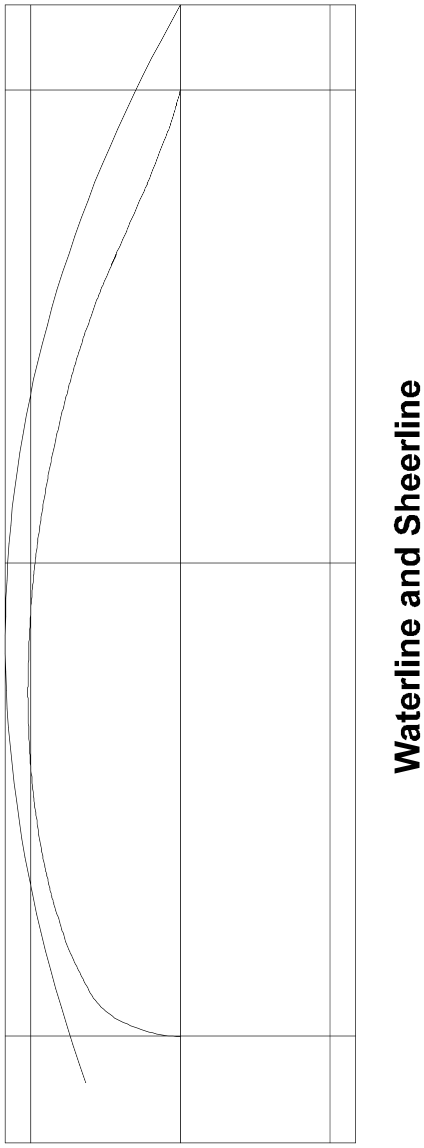
Figure 23.21, C_P = 0.57, LCB = 50%.



Output From Hull Generation Method

Method		Nº 3.
Parameter Basis		YD-40 Hull
Parameters		
Loa	12.050	m
Lwl	10.020	m
Bmax	3.710	m
Bwl	3.170	m
Beam at Transom	2.000	m
Tc	0.570	m
Forward Overhang	0.900	m
Freeboard at Bow	1.500	m
Freeboard at Centre	1.350	m
Freeboard at Aft	1.300	m
Cp	**0.570**	-
Cm	**0.752**	-
Cwp	Not Used	-
LCF	Not Used	% of Lwl
LCB	**55.000**	% of Lwl
Bow Profile - Deck Angle	0.000	°
Bow Profile - Deck Tangent	0.500	m
Bow Profile - Waterline Angle	0.000	°
Bow Profile - Upper WL Tangent	0.500	m
Bow Profile - Lower WL Tangent	0.200	m
Transom Angle	-30.000	°
Transom Fixed Point	0.200	m on WL





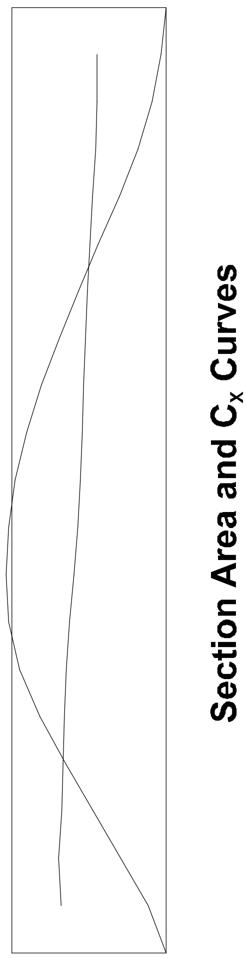
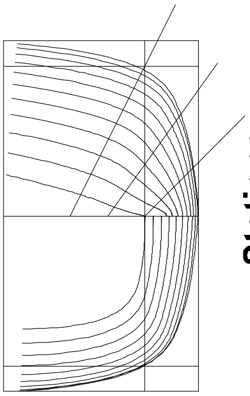
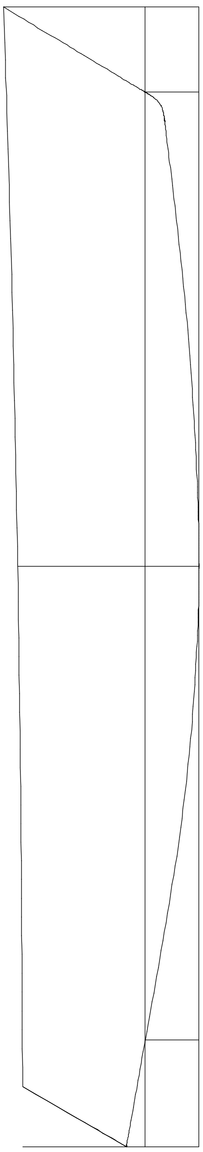


Figure 23.22, C_P = 0.57, LCB = 55%.

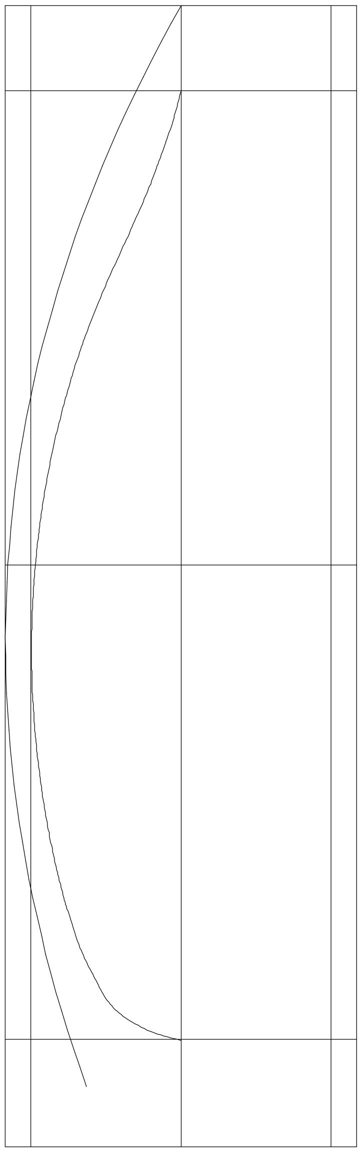


Stations

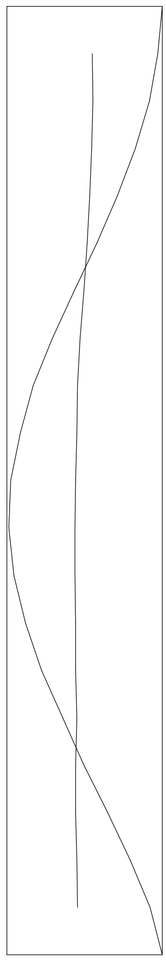
Output From Hull Generation Method		
Method	N° 3.	
Parameter Basis	YD-40 Hull	
Parameters		
Loa	12.050	m
Lwl	10.020	m
Bmax	3.710	m
Bwl	3.170	m
Beam at Transom	2.000	m
Tc	0.570	m
Forward Overhang	0.900	m
Freeboard at Bow	1.500	m
Freeboard at Centre	1.350	m
Freeboard at Aft	1.300	m
Cp	**0.520**	-
Cm	**0.752**	-
Cwp	Not Used	-
LCF	Not Used	% of Lwl
LCB	**53.000**	% of Lwl
Bow Profile - Deck Angle	0.000	°
Bow Profile - Deck Tangent	0.500	m
Bow Profile - Waterline Angle	0.000	°
Bow Profile - Upper WL Tangent	0.500	m
Bow Profile - Lower WL Tangent	0.200	m
Transom Angle	-30.000	°
Transom Fixed Point	0.200	m on WL



Profile and Sheerline

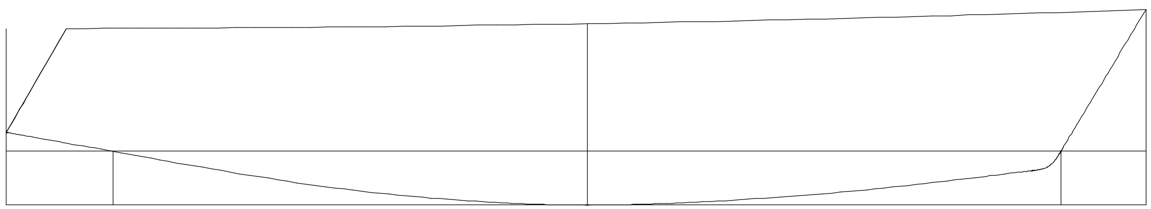


Waterline and Sheerline

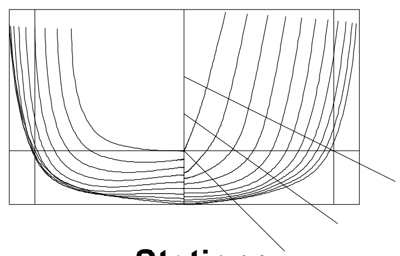


Section Area and C_x Curves

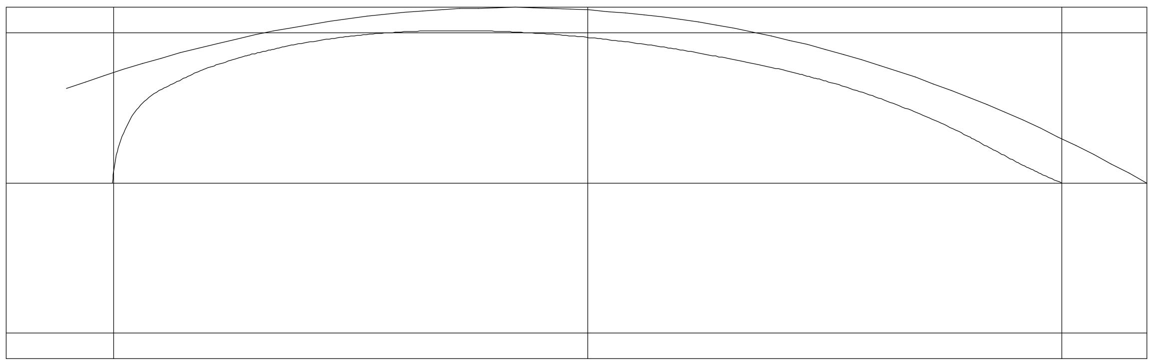
Figure 23.23, C_P = 0.52, LCB = 53%.



Profile and Sheerline

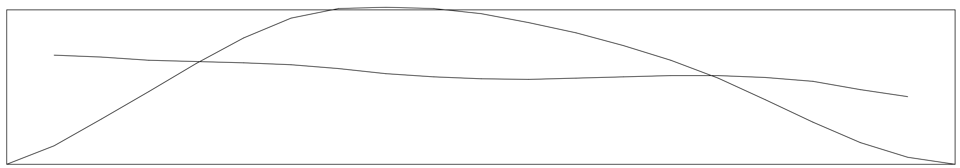


Stations



Waterline and Sheerline

Output From Hull Generation Method		
Method		N° 3.
Parameter Basis		YD-40 Hull
Parameters		
Loa	12.050	m
Lwl	10.020	m
Bmax	3.710	m
Bwl	3.170	m
Beam at Transom	2.000	m
Tc	0.570	m
Forward Overhang	0.900	m
Freeboard at Bow	1.500	m
Freeboard at Centre	1.350	m
Freeboard at Aft	1.300	m
Cp	**0.600**	-
Cm	**0.752**	-
Cwp	Not Used	-
LCF	Not Used	% of Lwl
LCB	**53.000**	% of Lwl
Bow Profile - Deck Angle	0.000	°
Bow Profile - Deck Tangent	0.500	m
Bow Profile - Waterline Angle	0.000	°
Bow Profile - Upper WL Tangent	0.500	m
Bow Profile - Lower WL Tangent	0.200	m
Transom Angle	-30.000	°
Transom Fixed Point	0.200	m on WL



Section Area and C_x Curves

Figure 23.24, $C_P = 0.60$, LCB = 53%.

23.10. Discussion

The YachtLINES hull generation technique is reasonably successful. The high similarity between modern yacht hull form shapes means that the technique can always generate a very close approximation. However, the technique does not always generate a form that can be considered completely fair.

The project was developed in a fixed amount of time and there was little allowance to fine tune the system. The lack of experience with geometry, NURBS and numerical programming tools put the development of a fully capable tool at a disadvantage. As the development of TSCAHDE demonstrates, and even with the approach taken with ShipLINES, a better appreciation of how the technology functions can result in very successful and relatively simple solutions. YachtLINES did not attempt to make the best use of NURBS with regards to taking advantage of the properties or looking for ways of improving the use of the iterative approach. It would be certainly possible to take this technique and by using some of the experience gained in the development of ShipLINES and TSCAHDE to develop a more effective parametric yacht hull generation tool.

24. APPENDIX 3 – SHIPLINES

24.1. Background

While the development of a yacht orientated parametric hull generation tool was an interesting project, there are very few occasions when a parametric hull design tool is going to be effective during the yacht design process. Very few yachts are designed considering systematic variation and most yacht designers would prefer to manually manipulate the hull form rather than have no control over the shape of the surface whatsoever. However, the design of ship hull forms is a different matter. While ship hull surfaces are primary designed using manual approaches, there is a greater need to look at systematic variation to optimise for performance. Furthermore, there is a great deal more research within the maritime industry that requires a ship hull form surface as a basis and that does not have the time available to use tools relying on manual manipulation.

24.2. Introduction

The parametric generation of ship hull form surfaces presents more of a challenge than the development yacht hull forms. Ship hull forms contain a wide variety of different shapes within the surface, which makes the generation process much more intricate. Furthermore, the existence of comparatively large appendage features that are faired into the main hull surface can complicate matters. The majority of hull form generation techniques have concentrated on the development of ship forms and a wide variety of methods have been created. However, few have considered NURBS as the primary representation technique for the hull surface and the supporting geometry used within the generation technique.

The methods developed under the supervision of Nowacki [10], [14], since the early 1970's, have concentrated on using NURBS representations in conjunction with the generic solution processes used by many other techniques. The advantage of this approach is that the hull form can be easily transferred to mainstream hull design packages for manipulation using standard manual techniques. However, the hull forms produced by the technique are rigidly linked to the mathematical framework used to produce the surface. Mathematical frameworks are less open to customisation due to the hard coded nature of the technique within the software. A more flexible approach may be achieved by considering a framework that uses different technology. A geometric approach, for example, would more rely more heavily on shape and relations with the framework rather than the representation mathematics. Consequently, opportunities can be made to customise the framework arrangement.

The development of YachtLINES demonstrated that basic geometric techniques could be used to generate hull forms. However, if the particular properties of the representation medium are not strongly considered in the development of the technique, it can be difficult to achieve a satisfactorily fair hull surface. Ship hull forms can incorporate planar areas of the surface and corners between regions. NURBS technology provides the means to implement these shapes within a single representation surface. A generation technique could be developed by taking full advantage of these properties to develop a single NURBS surface which can incorporate all the features generally found in ship hull forms.

24.3. The Conceptual Approach

A NURBS surface is defined by a control polygon mesh, i.e. grid of vertices. A fair and smooth surface has to be developed using a discrete set of definition points. Consequently, the most important aspect of the technique is to develop the means to accurately control the surface to make it form the desired shapes. The human user performs this task using an iterative approach, which though continual manipulation of the control vertices, achieves a visibly fair surface after some time. The hull generation technique, with access to accurate mathematical tools, should be able to define a fair, smooth and correctly shaped surface at the first attempt. The best approach is to make use of the representations capabilities, particularly the properties, to develop a geometric structure that will control the definition vertices to achieve the desired surface shape.

A NURBS surface designed to represent a shape as complex as a ship hull form has many control vertices and hence many degrees of freedom, three for each vertex. To develop a ship surface from numerical parameters, the means must be developed to take the highly compressed information in the parameters and expand the information to form the hull surface. The approach taken to develop this process will ultimately control the overall flexibility and success of the technique when developing hull surfaces.

There are two approaches that can be taken to build up the surface definition. The method employed by Sanderski [32] takes the constraint approach. Because the success of the method is dependant on the actual number of degrees of freedom there are in the surface definition, some parameters, coordinate components of the control vertices, are constrained to fixed values so that they no longer have to be considered in the solution process. The constraints used in the technique developed by Sanderski, fix all vertices to longitudinal and vertical positions to form a uniform grid, (Figure 24.1). Vertices on the boundary of the surface may be further constrained to lie on the centre plane. The remaining vertices are position by the solution process in the transverse direction only. The solution is produced considering various parametric aspects of the surface dimensions and qualities. However, while this approach is very successful in reducing the number of parameters for the solution process, the hull surfaces produced by the method are always governed by the arrangement of constraints. A constrain arrangement that is fixed is unlikely to be able to provide the flexibility required for practical hull form design.

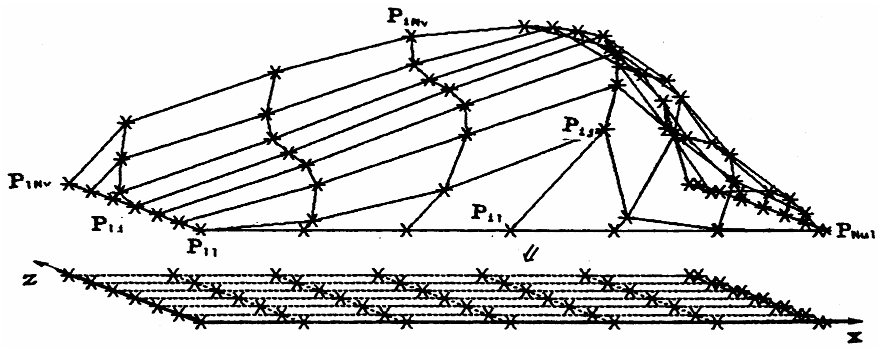


Figure 24.1, the control mesh arrangement used by Sanderski [32].

The second approach to develop all the definition points by considering form characteristics that can be used to form the hull surface. As previously discussed, these approaches are mainly based heavily on mathematical rules that do not allow for any customisation. Furthermore, the basic range of design parameters does not provided enough satisfactory information to form the surface. Consequently, more abstract parameters such as various moments of volume must be considered. Ultimately, only the developers of the technique can use the tools because they have investigated the range of appropriate values for these unusual parameters.

Instead of relying on form functions devised through fixed mathematical procedures to create the right shapes, it is possible to develop a framework that can be used to develop all the surface definition by using a geometric approach. The geometric approach has the capabilities to form the correct surface shape by considering the topology of the form and can be used to develop the particular arrangements of control vertices required to induce NURBS properties where desired. Furthermore, the geometric framework, being formed from many visible entities, aids by also providing the means to allow for manual manipulation and input during development for reviewing new constructions and is capable of customisation to allow for more flexibility in generation technique when provided as a tool.

In the consideration of how a hull surface, defined using NURBS, can be generated, the geometric approach offers quite a few advantages over the other approaches. The fact that the output of the technique is a geometric entity means that the approach can always find a way to adapt to the shape. For example, to develop a fair hull surface, previous techniques have had to consider routines that will analyse the surface. These checks introduce further complications because solution has to be implemented to change the surface when it is found to be invalid. However, in a geometric approach, using a surface defined by a geometric entity, it is obvious that if the definition control polygon is fair, the resulting surface will also be fair. Consequently, if an approach to the development of the geometric framework prevents an unfair control polygon, the whole issue no longer needs to be considered.

There are a limitless number of characteristics that can be found in hull forms. An initial hull with particular characteristics was chosen to keep the range of features small for the initial stages of development. However, it was desirable to produce practical hull form shapes to demonstrate that the approach was feasible and that ship hull form generation could play a role in the design of vessels. A hull consistent with a single screw general cargo ship was chosen. The vessel would have stern propeller bossing and would be generated with or without a bulbous bow. Unlike previous techniques, the parameters controlling the shape of the vessel would be decided on the basis of those required to create geometric framework. It is obvious that the main dimensions are going to be included. However, many additional parameters may be required to control the shape of the geometry, allowing some flexibility in the shape of the forms produced by the technique. To aid development, some basic size dimensions were considered to produce a trial surface as follows:

- LBP: 100 m
- Half Breadth: 20 m
- Depth: 9.5 m
- Draught: 4 – 5 m

24.4. Investigation into the representation Ship Hull Forms using Single NURBS Surface Patches

Compared with the development of other types of hull form, the development of ship hull forms which incorporate appendages is quite a complex and time consuming task. As all previous experience was mainly based around the development of small craft hull forms, an investigation was necessary to find out and understand how ship hulls can be represented using a single NURBS surface and to develop ideas around how such a hull form can be created through a parametric generation process. Due to the lack of readily available and effective hull design tools, the curve and surface manipulation capabilities of PolyCAD [50] were developed to the extent that would allow hull form representations to be freely manipulated and analysed effectively. To aid the investigation process, the bow and the stern portion of the hull form were developed separately. A split at the midsection allowed the entrance or run to be investigated with the inclusion of reviewing the transition to the parallel segment of the hull form. The important aspects that were to be achieved by the investigation was to realise the number and respective arrangement of the definition vertices that would be required to develop a generated hull form representation.

The bow portion of the hull form was developed relatively rapidly. Considering the lack of experience, the front portion of the hull form took a week of development time using PolyCAD. The development of the surface shapes was found to be fairly straight forward. To begin with, as the boundaries of the NURBS surface behave as curves, the midsection, stem and bow were developed. The midship section was developed using two groups of linearly arranged control vertices at the baseline and at the half breadth. A satisfactory bilge radius shape was formed by the intersection of the two groups. The stem was initially formed in a similar way to the midsection curve shape. However, to introduce a bulb shape, a “box” like arrangement was formed to bulge the curve into a desirable shape and the number of control points was increased at the transition towards the top of the bulb to form a tight curve. Using the NURBS property that the next control vertex from an end vertex controls the tangent direction, the next column of vertices from the stem was formed by transversely projecting the position of the vertices on the stem boundary. In a similar approach, to form the section of the parallel middle body, the position of vertices were longitudinally projected from the location of the control vertices on the midship section boundary. This relationship was made until the row of vertices crossed the boundary defined by the forward flat of side curve. Between the stem and the flat of side curve, the vertices were first position by row and by column to form a smooth shape between. The columns of vertices were maintained in transverse planes where possible. To form the bulb shape, the box-like nature of the control vertices on the stem was continued, aft, into the surface. The control vertices were faired into this shape. Throughout the development of the surface, maintaining the surface with a fair and uniform control polygon mesh was a high priority.

The final version of the manually formed forward surface consisted of a mesh of twelve rows and nine columns, (Figure 24.2). Fairing the shape was found to be very easy, it was only necessary to blend the shape of the rows of control vertices from the parallel section to the stem. The forward surface could be formed adequately with a fair and uniform control polygon mesh. However, while the shape of the surface appeared to be adequate and reasonably fair, the contours indicated unfair shapes in the surface when transitioning away from flat of side and flat of bottom curves. At this point in time, this was put down to the level of accuracy that could be achieved by manually interacting with the control points.

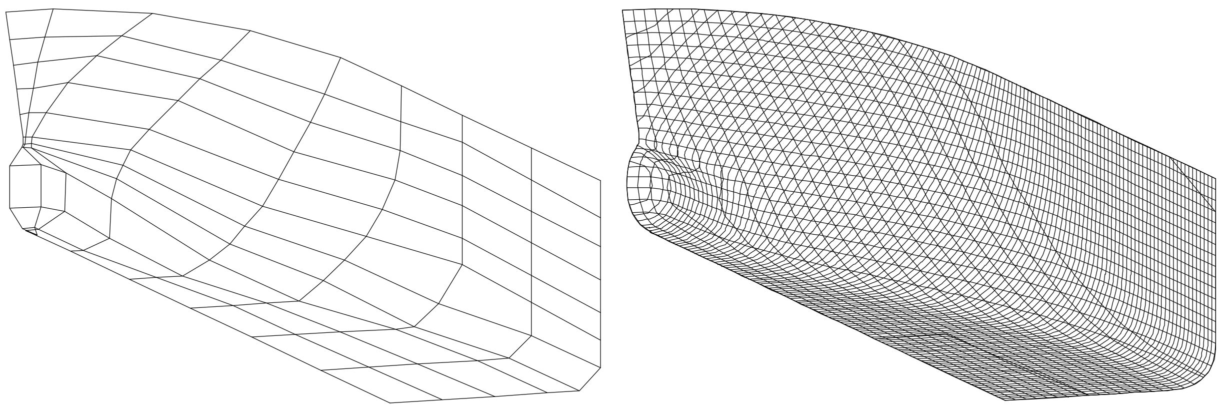


Figure 24.2, the forward hull form control polygon and surface produced by the manual investigation.

The development of the stern shape presented much more of a challenge because the arrangement of the surface does not match closely shape of the surface when it includes propeller bossing. The surface was arranged in a similar fashion to the bow surface, with the forward boundary representing the midsection. The aft boundary represented the transom with the remaining boundaries forming the deck at side and the centre line. This arrangement developed a reasonable uniformly shape surface. However, to form the propeller bossing, the centreline boundary was deformed and dragged aft to from the characteristic shape, (Figure 24.3). This resulted in the control polygon no longer representing a uniform mesh. The shape of the propeller bossing was developed using a similar approach to that used to develop the bulbous bow, by developing a “box” like shape in the control polygon mesh.

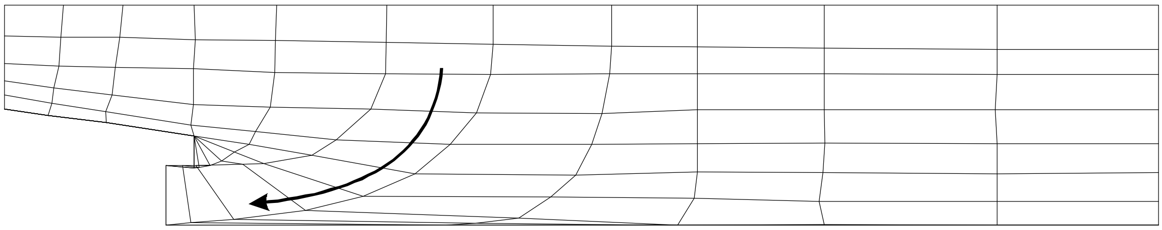


Figure 24.3, to form the propeller bossing appendage, significant deformation is required in the surface control polygon.

The process to fair the aft surface was a little more complex than the bow. The resulting shape of the deformed control polygon meant that the rows and columns no longer represented longitudinal and transverse curves respectively, and therefore could not be considered analogous to waterlines and stations. For the area of the surface above the waterline, the control polygon mesh retained a uniform shape and was therefore treated rather easily. However, below the waterline, the deformed shape resulted in row and columns of the control polygon that switch between longitudinal and transverse shape and back. Consequently, the only practical approach that could be used was to ensure that the rows and columns of the surface were reasonably smooth shape in all three dimensions at once. In contrast to the bow surface, the aft surface took around two to three weeks to develop.

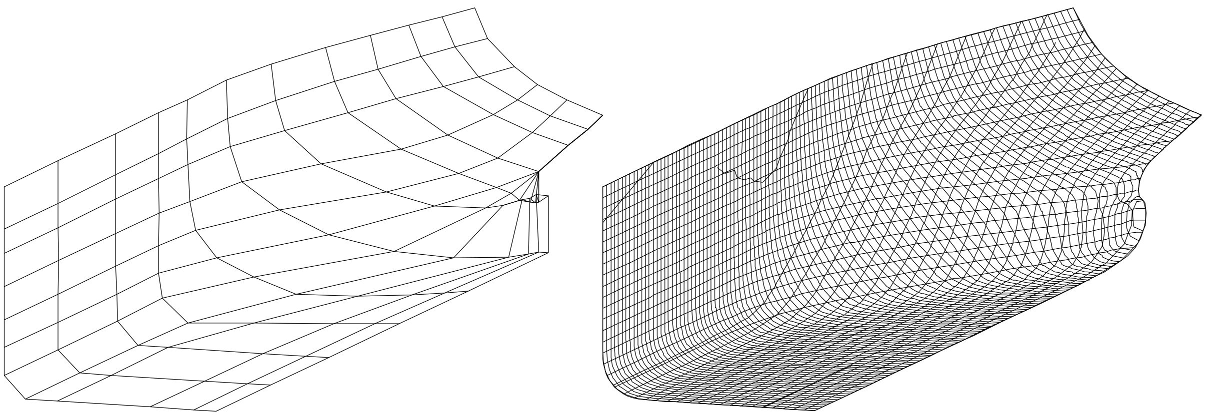


Figure 24.4, the aft hull form control polygon and surface produced by the manual investigation.

The uniform control polygon mesh shape above the waterline resulted in an almost one-to-one relationship between the control vertices on the transom and the midship section. As a result, a position to terminate all the definition rows below the waterline had to be found somewhere on the transom boundary. As the propeller bossing is somewhat a protuberance, the solution was obtained by making the definition rows of the control polygon converge at the point where the propeller bossing meets the hull at the aft. Consequently, the stern boundary continued aft to the transom defined by multiple rows. Once complete, the stern surface consisted of a control polygon of twelve rows by twelve columns, (Figure 24.4).

The investigation was found to be extremely useful for understanding how to build up hull surface using NURBS. In fact, once this investigation had been performed, it became necessary to quickly develop a visually accurate surface of a large bulk carrier. Apart from the bulbous bow, the hull did not have any appendage features. The hull form surface was digitised to a reasonable visual accuracy over two days. For generating hull forms from numeric parameters, the investigation highlighted that the best approach would be one which uses the parameter to defined the characteristic features and then form the remaining surface by maintaining a smooth control polygon mesh between. However, the arrangement of the stern surface control polygon illustrated that it was not always possible to achieve a uniform mesh shape. Consequently, there was a great possibility that the resulting tool might not be able to achieve the level of flexibility in the development of hull form shapes that was initially envisaged.

24.5. The Hull Form Generation Process

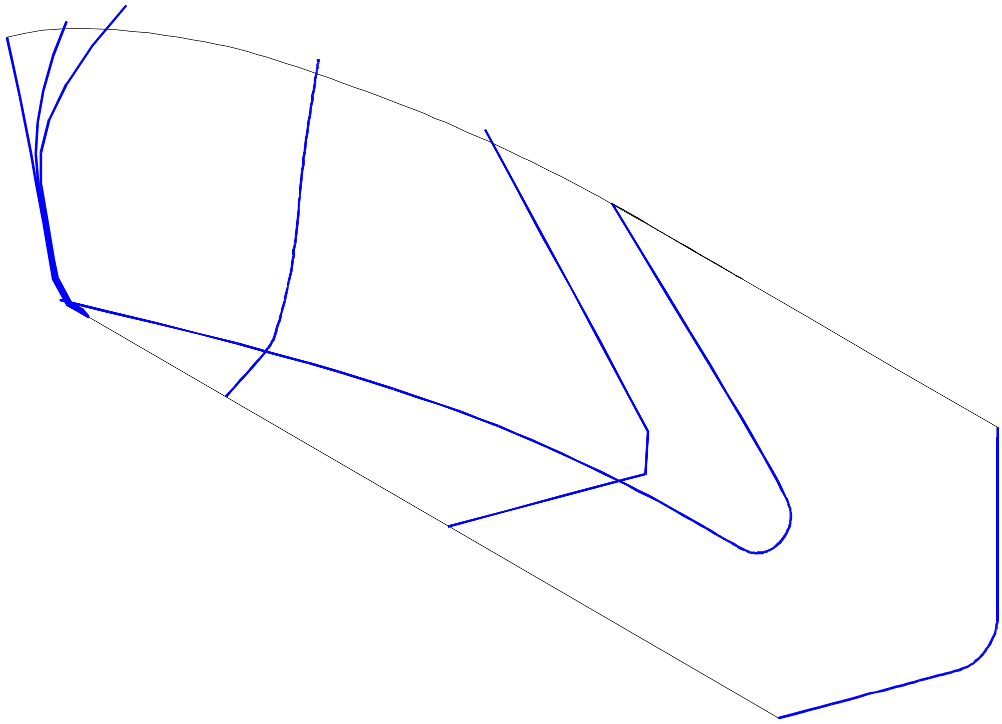
The manual development of representative hull surfaces used the longitudinal rows of the surface to form a fair shape. The columns were considered, where possible, as planar stations. The hull generation procedure could use a similar approach, by first developing curves to represent the locations where the rows of control vertices would lie and the final vertex positions would be calculated by intersecting the longitudinal curves with the section planes of the columns.

The investigation illustrated that the parallel middle body could be formed very easily. A prismatic surface could be developed by forming control polygon columns by longitudinally offsetting the shape of the control polygon forming the midship section curve. Consequently, the curves forming the rows of the longitudinal control polygon would be lines parallel to the x-axis located at the control vertices. An advantage of the arrangement of the control points in the parallel middle body is that by the longitudinal alignment, a discontinuity is formed in the x direction. Consequently, the entrance and run regions of the hull surface can be treated independently of each other.

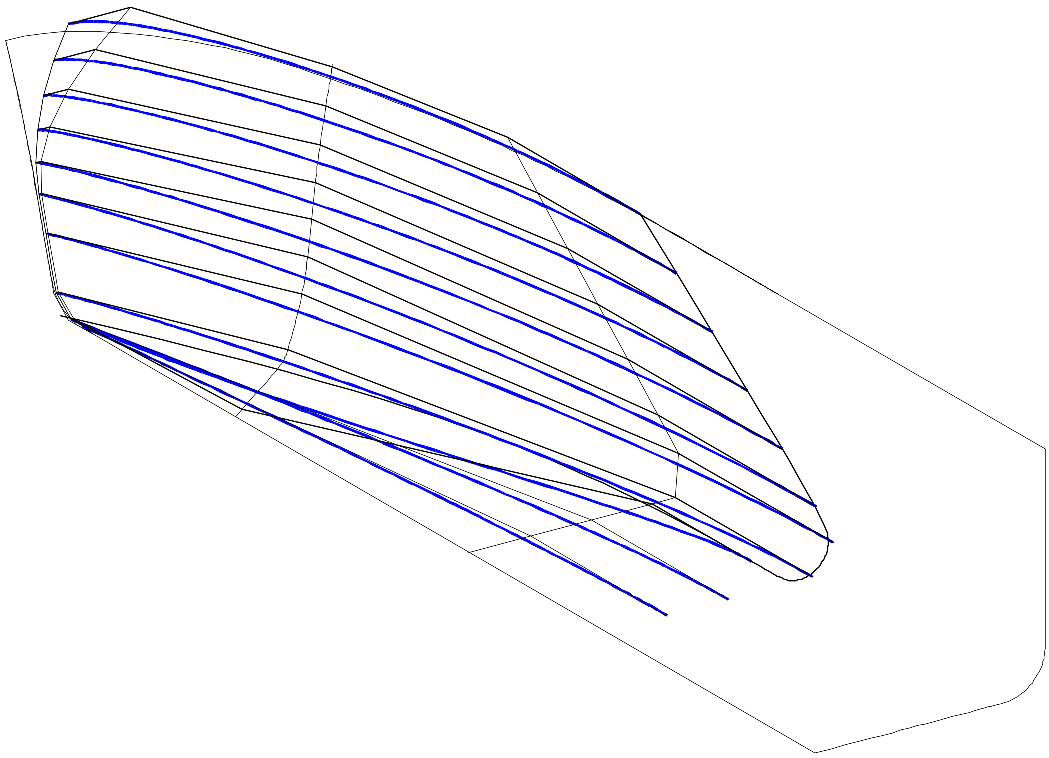
On the basis of the approach taken by the manual investigation, the generation procedure develops the shape of the entrance by first considering the shape of the features using individual parametrically generated form curves. Because the surface is generated around longitudinal curves, the number of control polygon columns defining the surface is relatively flexible. Furthermore, as the curves representing the longitudinal rows of control vertices have to intersect with the particular feature or form curves defining the characteristics of the hull form, these curves could be used as formers to control the shape of the longitudinal curves.

The development of the forward section is relatively straight forward, it follows the approach of the basic generation procedure closely, (Figure 24.5). First, the form feature curves are generated, (Figure 24.5a). From these curves the longitudinal blending curves are formed, (Figure 24.5b). The longitudinal curves are intersected at various transverse sections to produced the control polygon columns, (Figure 24.5c), which in conjunction with the longitudinal curves form the control polygon mesh. Because the blending curves must form a family of shapes, the local bulb feature cannot be considered at this stage. The bulb shape is developed in the surface by directly modifying the control polygon using various local control parameters, (Figure 24.5d).

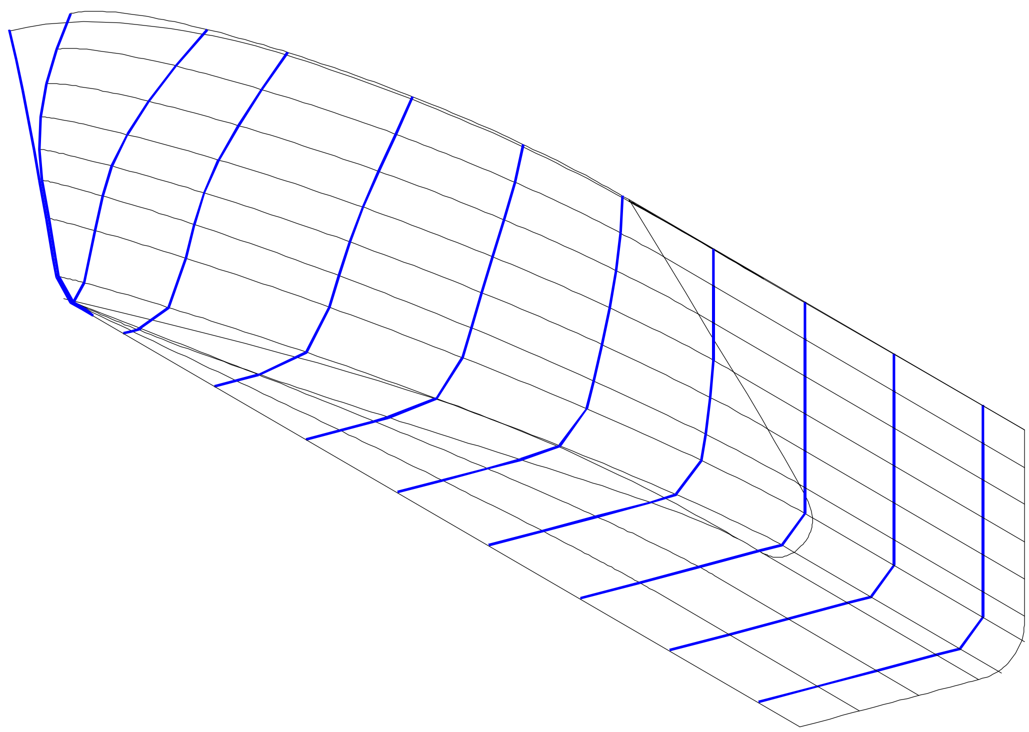
The development of the stern shape does not follow the basic generation procedure as closely, (Figure 24.6). The distorted control polygon forming the propeller bossing appendage requires a more elaborate technique to develop a fair control polygon mesh. As with the development of the bow surface, the form feature curves are generated first, (Figure 24.6a). However, the blending function development process is split into parts to handle the deformed shape more easily. Blending curves are formed only in two dimensions. To develop a fair shape in the rows and columns of the control polygon, blender curves are developed considering the sections and appendage shape on the centre plane, (Figure 24.6b). These locate the position of the control vertices, the offset of which is formed by a third blender curve developed around the form curves in the horizontal plane, (Figure 24.6c). The procedure is repeated for all vertices to generate the complete control polygon, (Figure 24.6d).



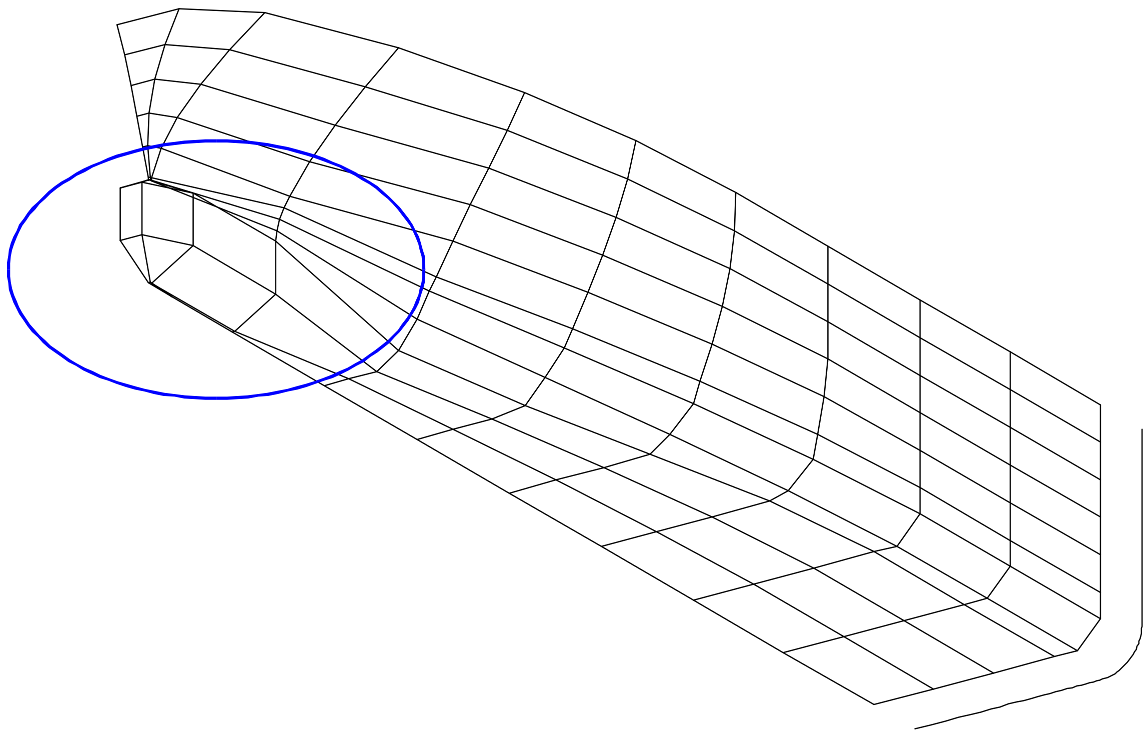
a) Construction of the form curves



b) construction of the blending functions covering the curved part of the entrance.

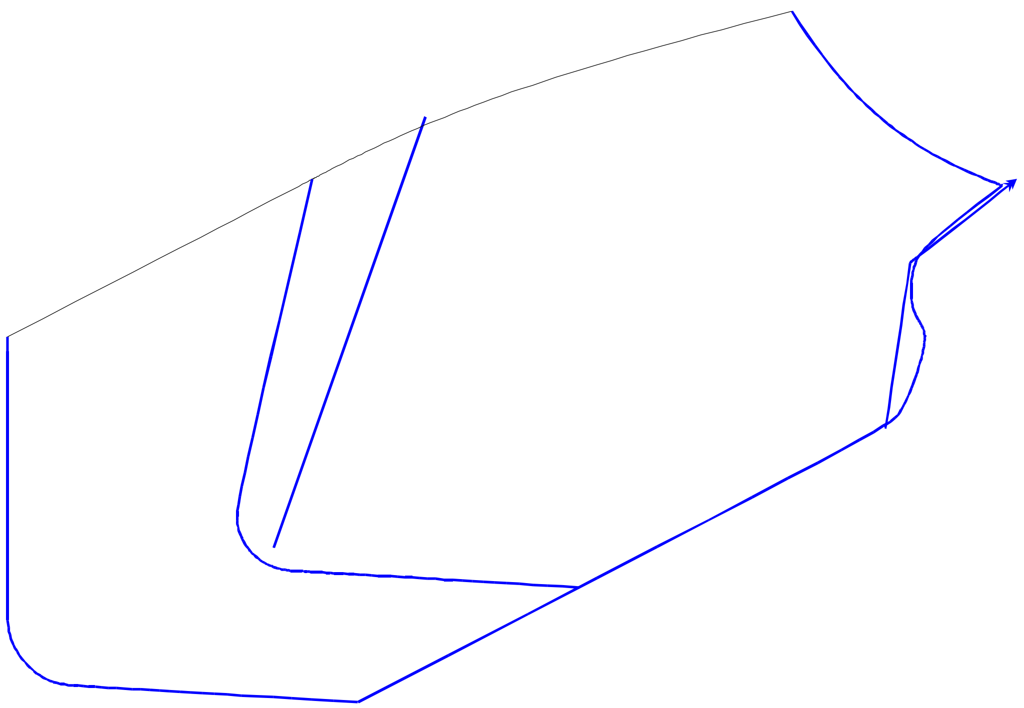


c) development of the control polygon columns from the blending curves and respective extensions

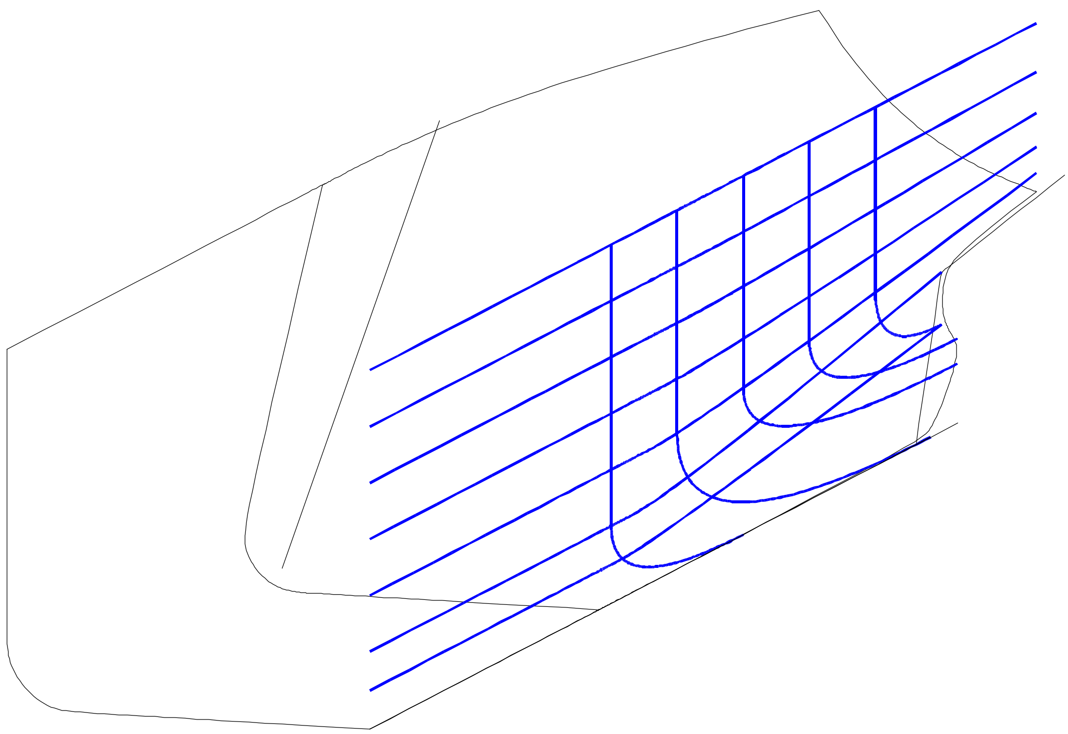


d) application of the bulb to the control polygon after generation by direct manipulation of the mesh

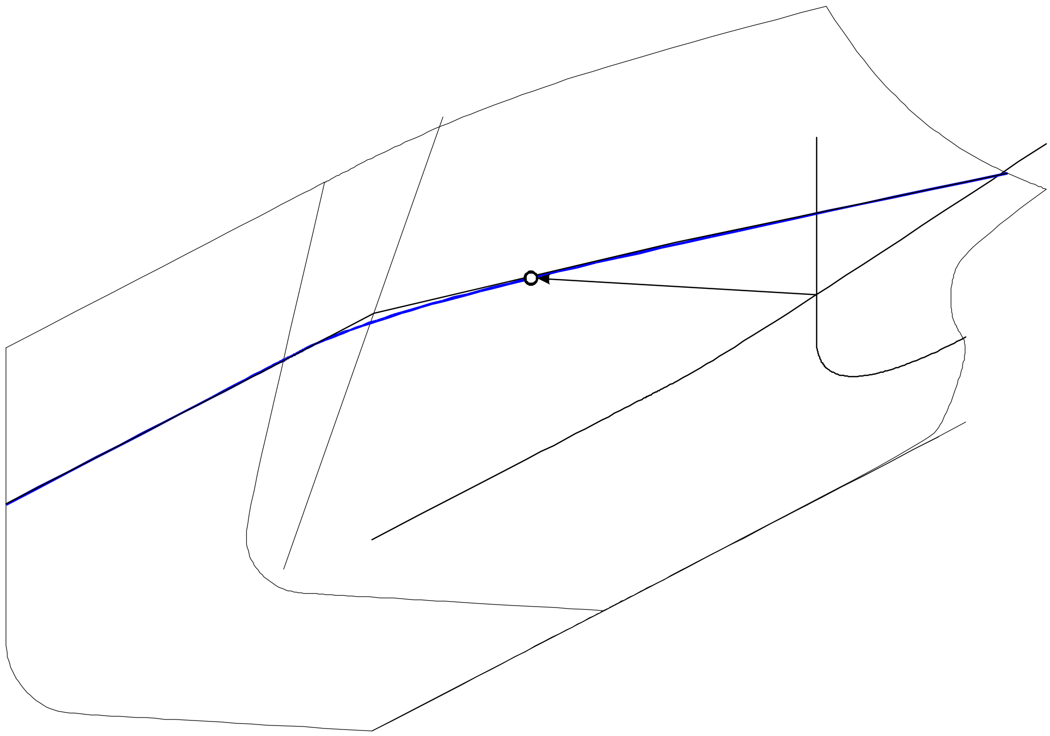
Figure 24.5, the process used to generate the hull forward of the midship section.



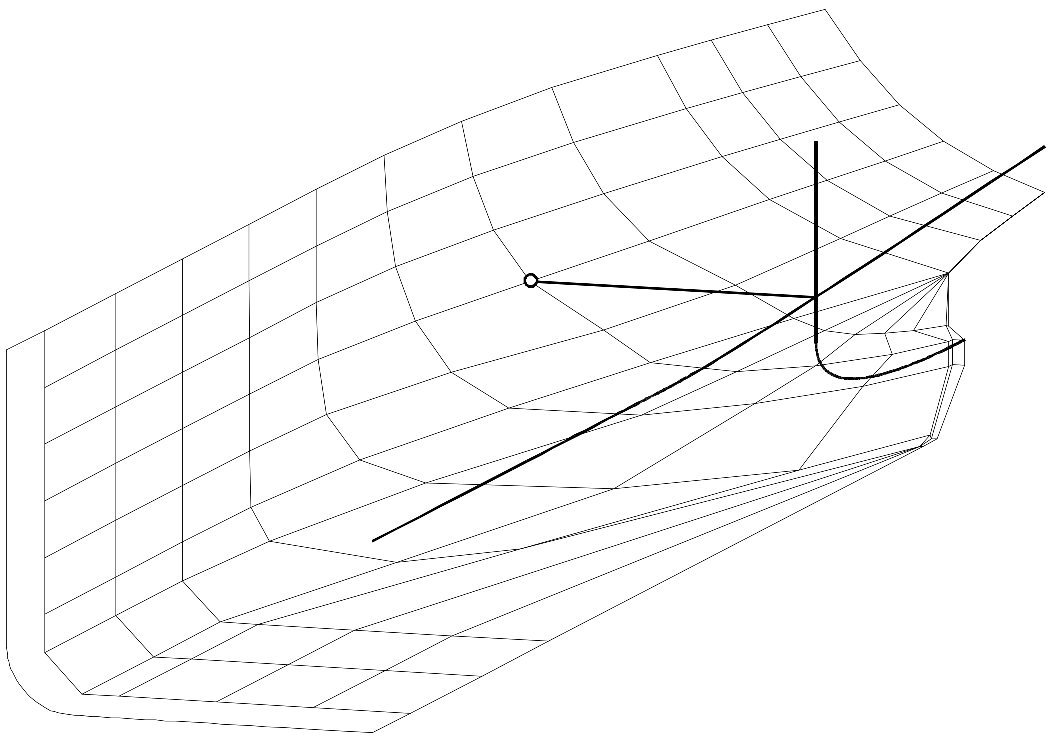
a) Development of the form curves.



b) Development of the row and column blending functions, identifying offset location by intersection.



c) Offset calculated by considering a third set of horizontal blender based on the form curves



d) the finalised control polygon mesh.

Figure 24.6, the process used to generate the hull aft of the midship section.

24.5.1. Form Curves

24.5.1.1. Constructions for Developing Particular Shapes in the NURBS Form Curves

There is a great level of flexibility in the NURBS representation used to define the form curves. To control these curves with numerical parameters it is necessary to develop methods that will construct arrangements of control vertices to develop appropriate shapes. NURBS properties aid the development of exact representations such as straight lines and cusps. Furthermore, exact representation of conics can be developed by utilising the homogeneous coordinates. However, as a single surface is generated, the rational part of the NURBS representation cannot be used in local areas without having an effect on the whole shape of the surface. To develop circular shapes, other techniques must be considered. As the hull form is developed with one surface, there are no practical possibilities for imposing accurate curved shapes such as circular arcs. Consequently, forms that roughly represent circular shapes are considered adequate for this technique. The constructions detailed here are used throughout the development of the form curves to impose a means of developing and controlling practical hull form shapes.

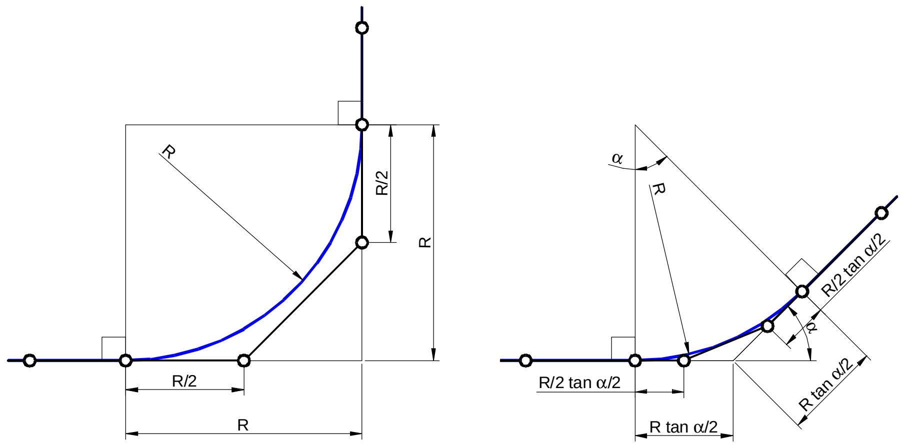


Figure 24.7, construction of a fillet shape between two linear segments.

Ship hull forms have a great deal of flat planar areas. However, these areas must be connected together by smooth curves for structural and hydrodynamic reasons. The bilge radius is a prime example of this. A basic fillet shape between two linear segments can be formed using the arrangement shown in Figure 24.7. To develop this arrangement, two control vertices are place at the end points of the fillet. A further two vertices are located on the projected lines of the linear segments halfway between the intersection and end points of the fillet. This arrangement should create a reasonably circular shape for angles less than ninety degrees. For larger angles, the shape will become more distorted.

The construction of small sharp curved shapes is quite easy with NURBS. However, it is much more difficult to develop shallow curved shapes, particularly when the number of control vertices that must be used to form the shape is large. Various part of the hull form require a gentle curved shape and must used a pre-specified number of vertices because each form curve is required to have the same number of vertices. To develop a curved arc shape, a "spanning" construction is used, (Figure 24.8). It uses the symmetrical nature of the framework and resultantly, the fact that the angle of the curve segment can be divided by two for each additional span.

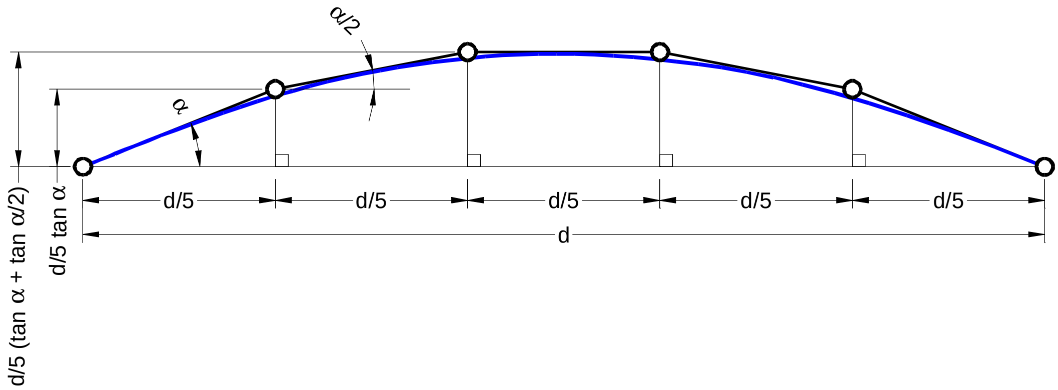


Figure 24.8, the construction used to develop a shallow circular arc, resembling a bridge span.

Due to the relative complexity of the shape, the construction frame work is developed horizontally and transformed into location using translation and rotational techniques. The arrangement is controlled with a parameter representing a percentage of the maximum angle that can be achieved across the diagonal in the rectangular arrangement, (Figure 24.9). The non-dimensional parameter means that the user will understand the expected shape and range of shapes that can be produced by the construction. Furthermore, the control parameter always remains valid if the hull form is scaled.

Construction of basic curved shapes can be very easy. However, when it comes to the development of particular shapes that are not quantifiably definable, an arrangement that develops pleasing shape with some flexibility for modification is welcomed. Longitudinal "waterline" shape, particularly at the deck was found to be quite a difficult area to find an appropriate shape for. Furthermore, it is undesirable to search for additional parameters that could be used to control deck shape. Control of the tangents at the bow or stern and side flats should be adequate. However, these control vertex arrangements alone make a curved but fairly flat shape, a further vertex between the tangents is required to impart some fullness. A simple geometric ratio, entitled the "half-third" rule, (Figure 24.10), was developed. This positions the centre vertex halfway between the two tangent vertices in the *x* direction, two thirds of the way toward the outer tangent vertex in the *y* direction. With modification, this technique is capable of developing a greater range of shapes if a parametric technique is used to geometrically locate the centre vertex. Consequently, this arrangement became the inspiration for the diamond control structures used in TSCAHDE.

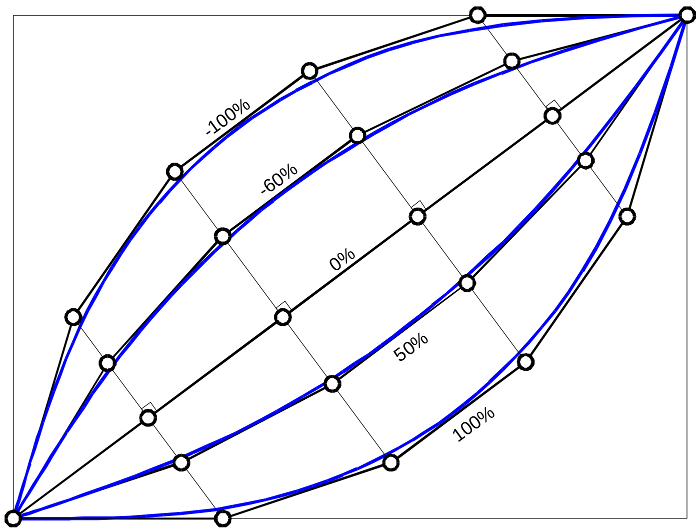


Figure 24.9, a percentage parameter is used to control the framework based on the maximum angle that can be achieved.

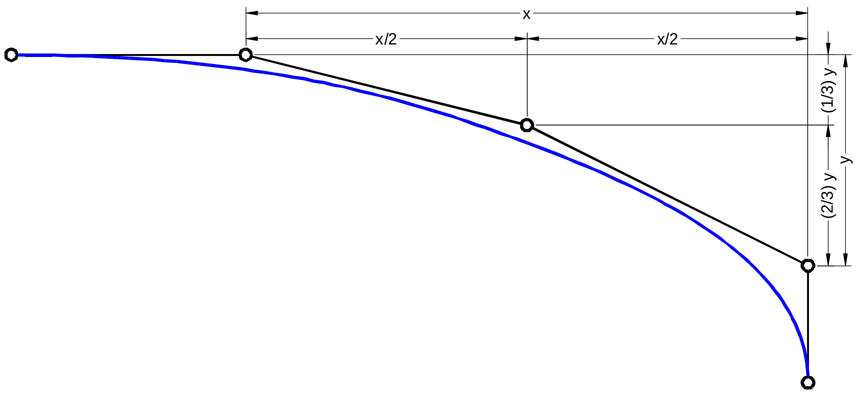


Figure 24.10, a half-third arrangement for locating the centre vertex between two tangents was found to produce a please shape particularly for decks.

24.5.1.2. The Midship Section Curve

The midship section is developed using two sets of linearly orientated control vertices, (Figure 24.11). The end points are located on the base line and at the half breadth at the deck. A fillet is implemented at the bilge radius. The remaining vertices are distributed at equal interval. More control vertices are required on the side of the section to help control the shape of the hull. Fewer vertices are required on the horizontal part of the section as they are only used to control the bottom flat and therefore not required to develop detailed shape control.

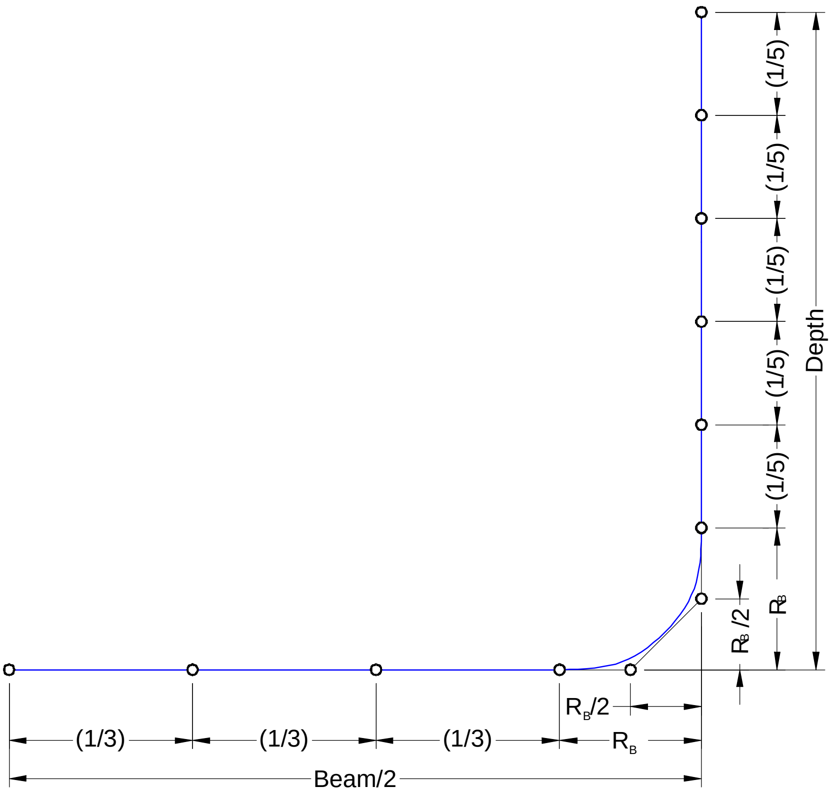


Figure 24.11, the parametric construction of the midship section control polygon.

24.5.1.3. The Stem Curve

The stem control polygon is a very good example of how the attempt to develop practical control of form shapes, the geometric constructions becomes quickly complex, (Figure 24.12). The construction uses a similar approach to the development of the midship section, with minor modification by inclining the side of the shape and applying the “arc” construction to curve the upper part. It should be noted that the distribution of vertices is different between the stem and the midship section on the horizontal and vertical portions of the control polygon. This was later found, in the development of TSCAHDE, to be one of the major reasons for the inability to develop accurate flat boundary shapes in the surface. The number of points on the vertical and horizontal portions of sections should be the same wherever possible throughout the hull definition.

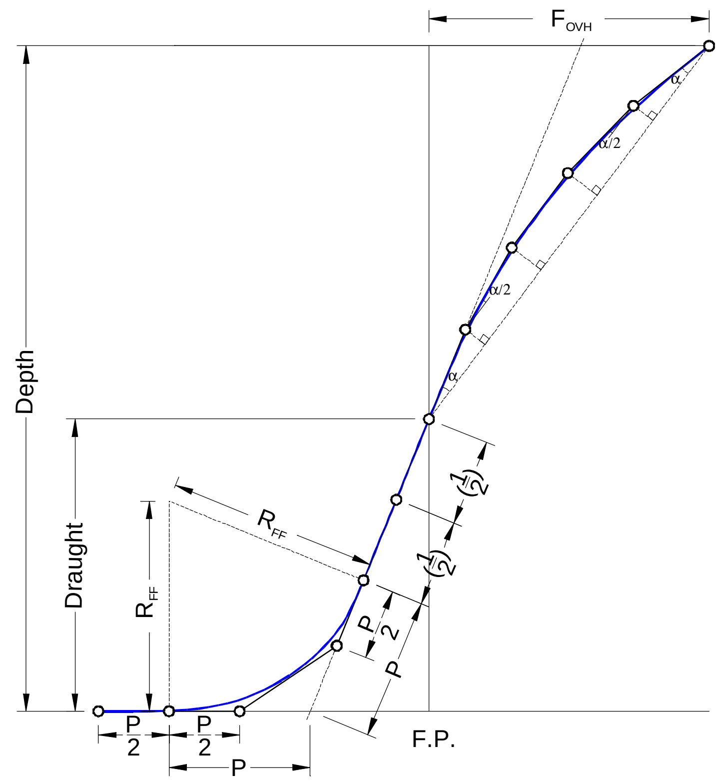
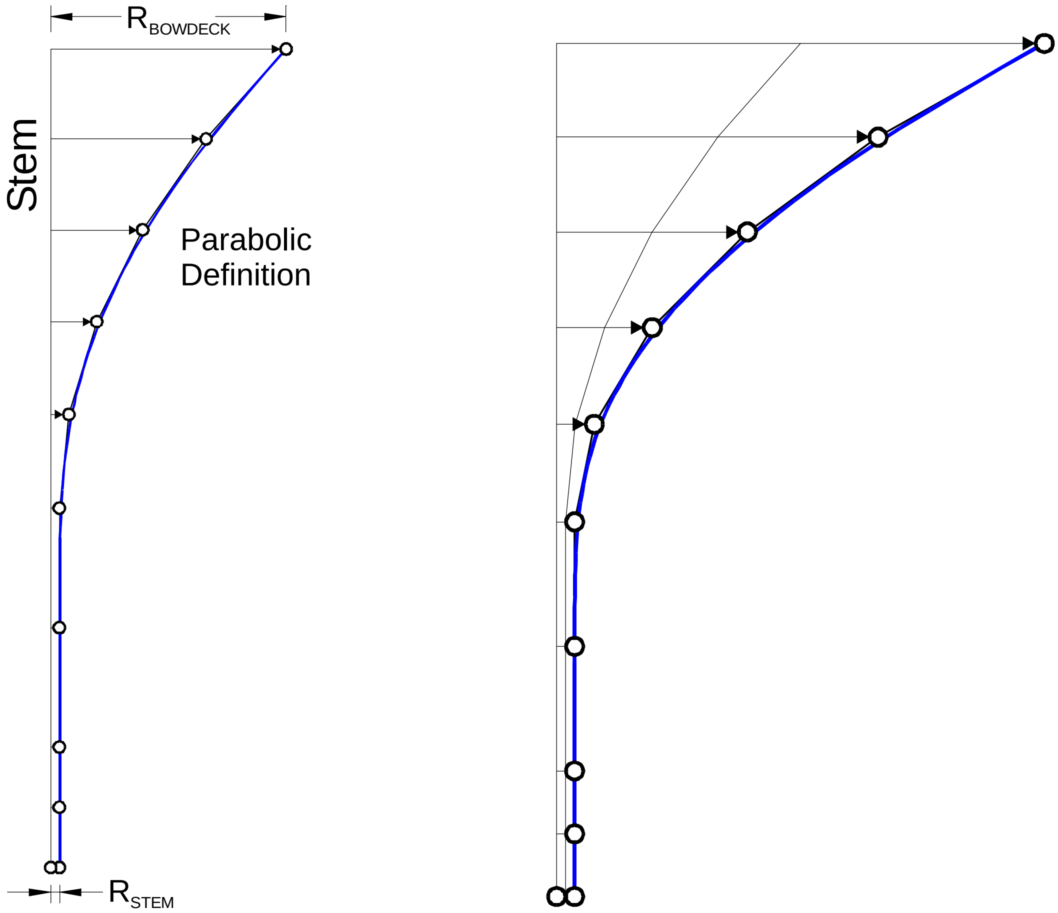


Figure 24.12, the parametric construction of the stem control polygon.

24.5.1.4. Stem Surface and Blender Tangent Control Curves

To develop the appropriate tangents in the surface, the first segments of the control polygon needs to be normal to the centre plane. This can be achieved offsetting the stem curve control polygon to form another curve, (Figure 24.13a). Flare can be controlled by increasing the amount of the offset and a sharp entrance can be developed by keeping the offset small. A parabola is used to blend the control vertices from the small offset to the large offset at the deck. The shape of the blending curve must reflect the arrangement of the explicit tangent control in the surface. Consequently, a further curve is generated using the stem control polygon and offsetting by double the distance between the stem and the stem tangent control curves, (Figure 24.13b).



a) Surface Tangent Control b) Blender Tangent Control

Figure 24.13, the parametric and geometric construction of the a) Surface and b) Blender Tangent Control curve control polygons.

24.5.1.5. The Forward Flat of Side Control Curve

The forward flat of side curve is formed using a similar method to approach used to develop the stem curves, by offsetting control vertices from a basis shape, in this case the midship section, (Figure 24.14). However, unlike the procedure adopted for TSCAHDE, the midship section curve does not match the control vertices on a one-to-one basis. Consequently, the transverse shape of the curve does not match the shape of the midsection. This would be a problem if this approach were used within TSCAHDE as it relies more heavily on geometric construction. As ShipLINES uses a more parametric approach to develop the hull surface, it ignores the consequences of the transverse shape by only considering the longitudinal position when developing the blender curve control polygons.

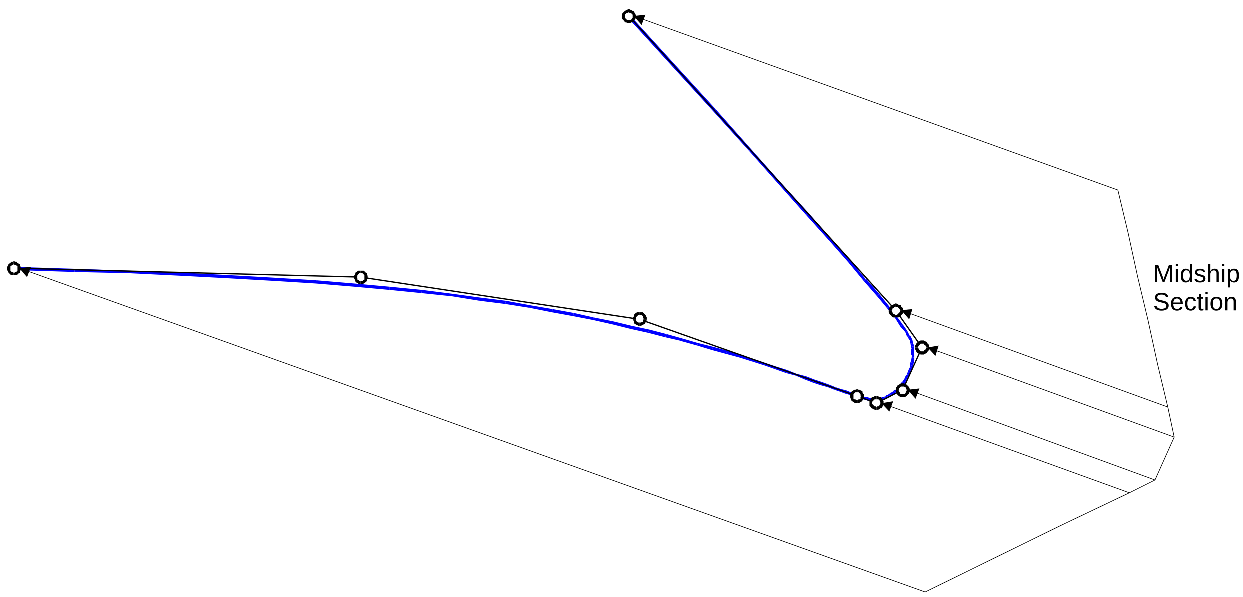


Figure 24.14, in a similar approach to the development of the stem control curves, the forward flat of side is developed by offsetting the basic shape of the control polygon from the midship section curve.

The control polygon shape is constructed using a combination of geometric, parametric and proportional positioning, (Figure 24.15). Using the major shape of the midship section curve, the shape is offset using parameters defining the extents of the parallel middle body and the parallel deck. The shape of the flat of bottom is developed by positioning the forward vertex at the location of the first vertex on the stem and using the location of control vertex at the deck and the half-third rule to position the remaining vertices. The shape of the curve used to represent the forward flat of bottom shape is not used to control the shape of the hull surface and is a legacy from the initial stage of development of the technique.

The size of the blender tangents to the flat of side curve is based on a constant distance in relation to the size of the surface, a third of the distance between the forward extent of the ship and the position of the forward parallel deck point.

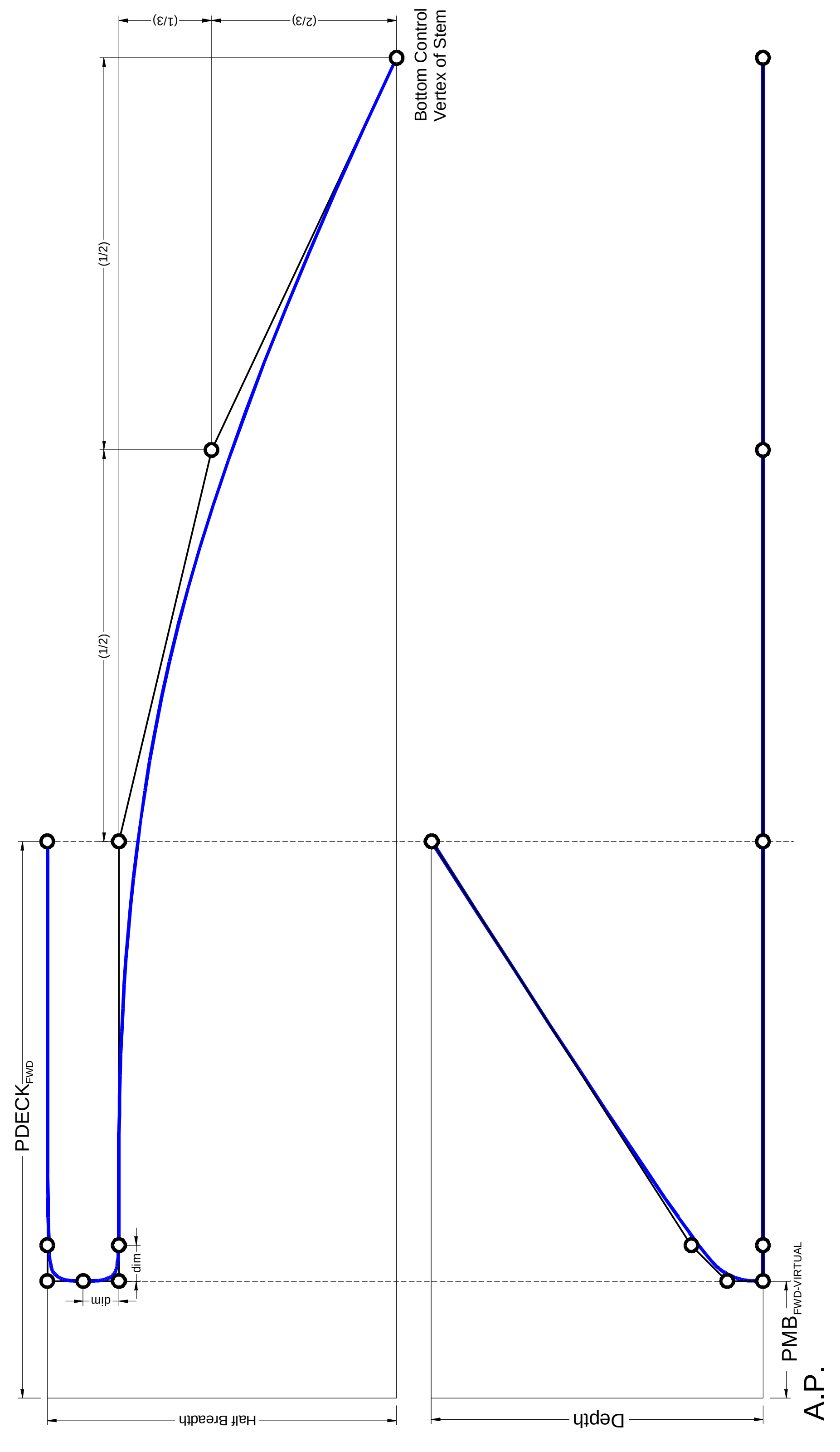


Figure 24.15, the forward flat of side curve is developed using a combination of geometric, parametric and proportional techniques.

24.5.1.6. The Bow Fullness Control Curve

The bow fullness control curve is used to follow exactly the same role as the volume control curves used within the TSCAHDE technique. However, ShipLINES does not implement a geometric process to control the shape of the fullness based on the geometric potential of the definition, (diamond control structures). ShipLINES uses a purely parametric approach by developing a curve from which a position is found for the blending curve definition based on intersection. However, as the fullness curve is generated purely parametrically, it has no basis to the shape of the rest of the hull form definition geometry. Consequently, an effective procedure for controlling the shape of the hull form with respect to the volumetric characteristics was not found because the shape curve cannot be modified satisfactory. The control polygon was developed using a basic structure, (Figure 24.16), and a set of parameters that were found to develop a reasonable shape. As a satisfactory solution was never found for this curve, a full parametric description has been omitted.

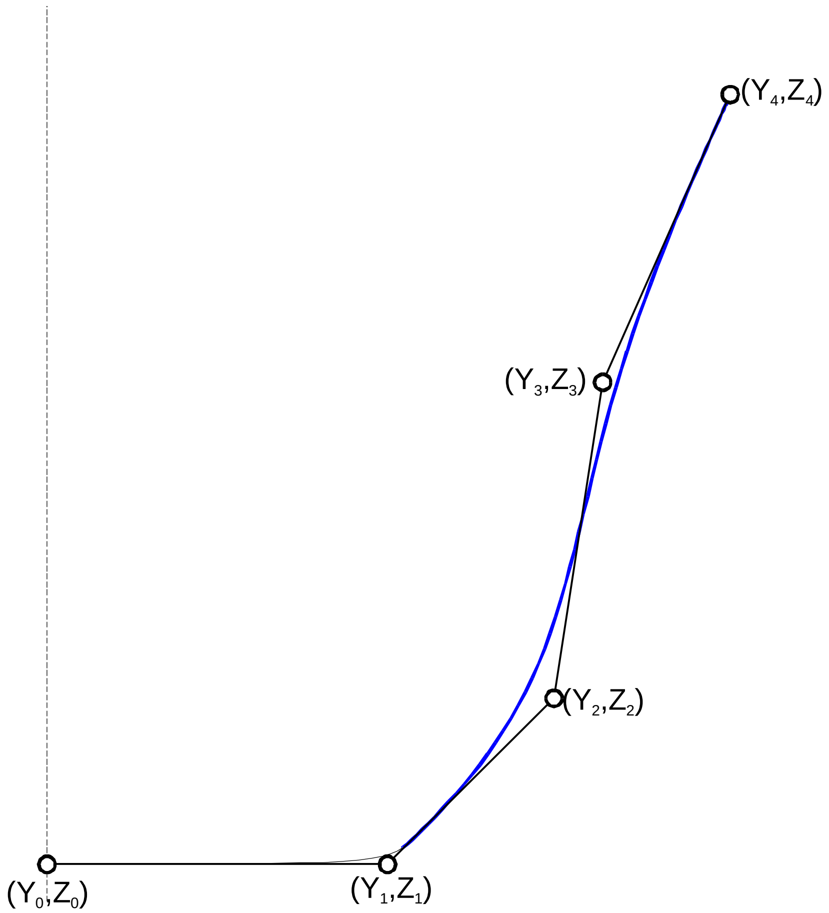


Figure 24.16, the control polygon constructing the bow fullness control curve.

24.5.1.7. The Stern Curve

The stern curve represents the surface control polygon at the centre line, from the position of the midship section curve aft. The stern curve is the only longitudinal part of the surface definition that is controlled directly, as a result of the deformation process used to form the propeller bossing appendage. The shape of the curve is formed using a very similar approach to the development of the control polygon for the bulbous bow. Local parameters are used to control the shape of the control polygon by considering the geometric characteristics of the appendage shape, (Figure 24.17). The figure also shows, as a dashed line, the shape of the "Projection" line in this region of the surface.

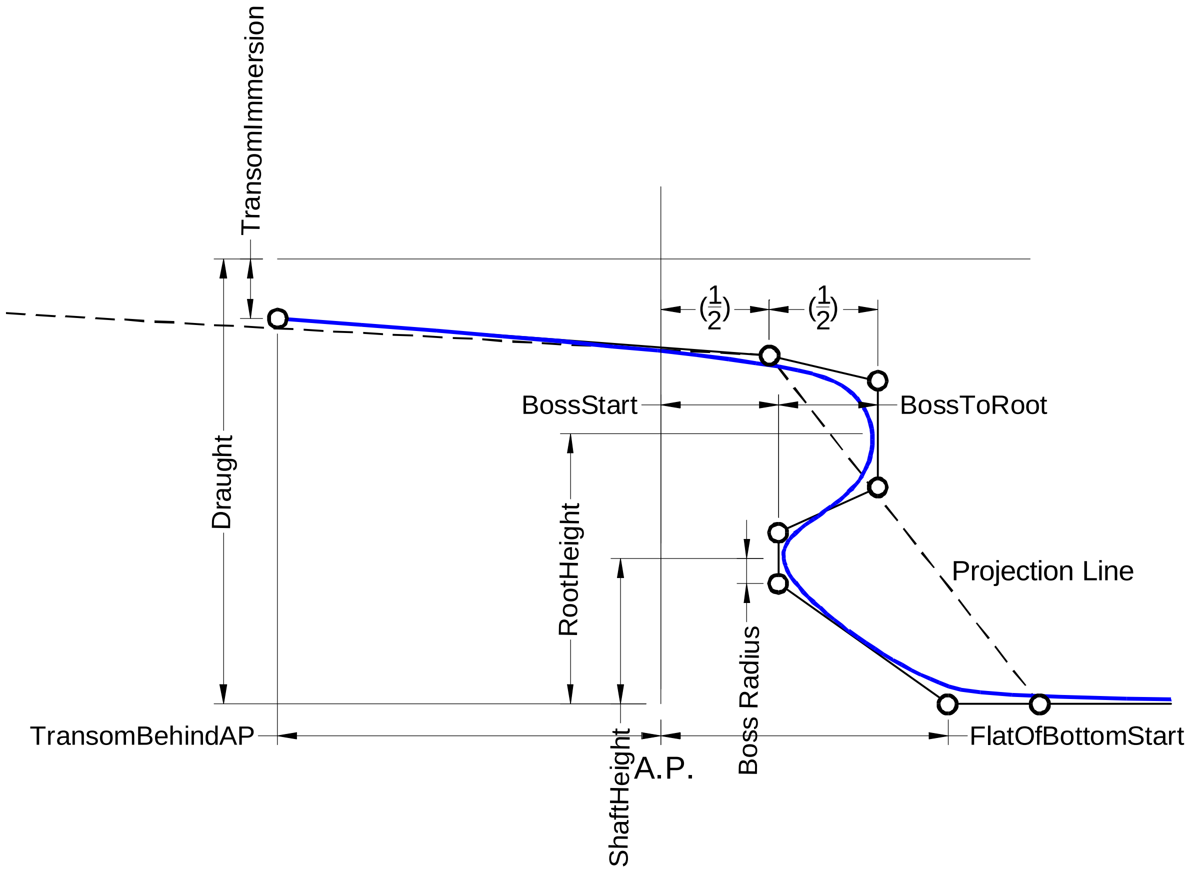


Figure 24.17, construction of the propeller bossing arrangement in the Stern curve control polygon

24.5.1.8. The Aft Stern Projection Line

Many techniques have used various projection lines to control the shape of the hull form. Both the FORAN [9] and Tribon LINES [30] use these types of construction and class them as virtual curves. In ShipLINES, the projection line is used to control the shape of the set of blender curves that are used to calculate the control vertex offsets. An aft fullness control curve cannot be implemented for the run of the hull form due to the level of complexity in the surface control polygon. Consequently, a projection line is used to locate the blender control vertex lying between the flat of side and stern/transom constructions. The projection line represents a reasonable shape of the aft boundary if the propeller bossing appendage was not considered and the surface ran aft to the centre line. The projection line is formed by two line segments. The first is between the effective aft extent of the flat of bottom and the control vertex at the top of the propeller bossing appendage construction, (Figure 24.17). The second segment is developed by continuing on to a point at the height of the deck line, where a line projected back from the forward part of the run deck line, constructed by the half-third rule, meets the centre plan, (Figure 24.18)

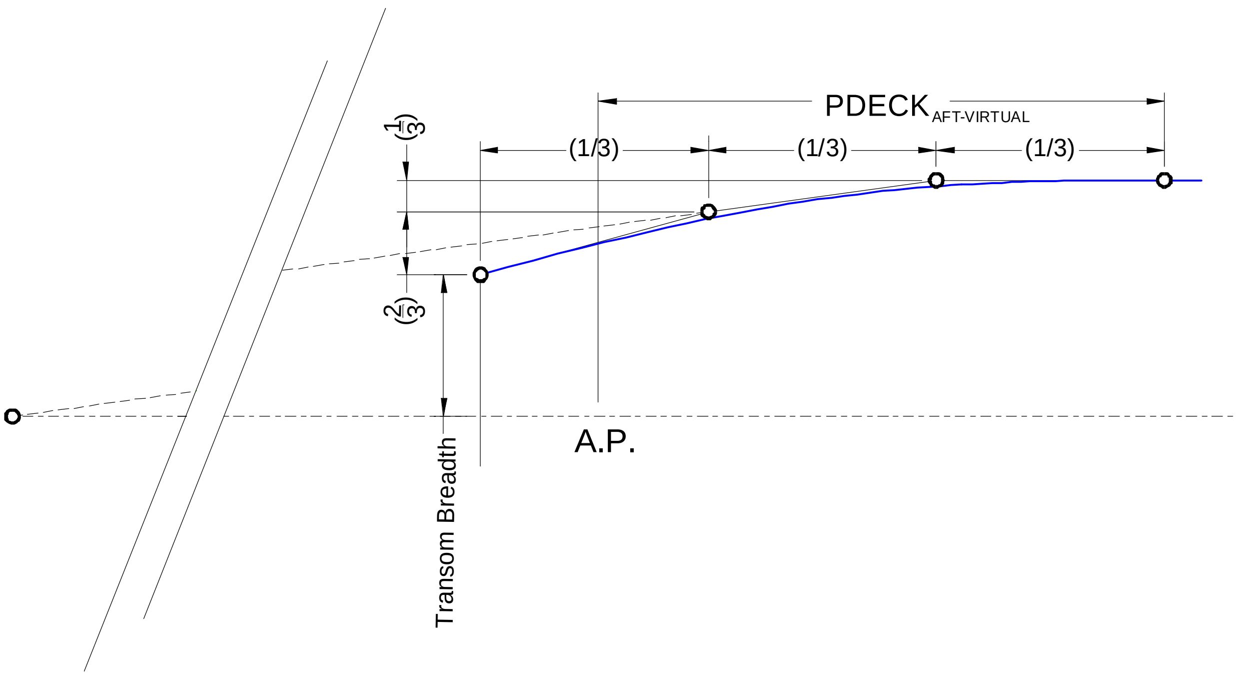


Figure 24.18, the aft most point of the projection line is formed by projecting, aft, the line from between the control vertices of the run deck line to the centre plan.

24.5.1.9. The Transom Curve

The transom curve control polygon is formed using the "arc" construction to form a pleasing shape, (Figure 24.19). The curve is constructed between points representing the transom half breadth at the deck and the point of transom immersion at the centreline. The lower control vertex of the transom consists of seven control vertices resulting from the surface control polygon rows used to form the propeller bossing appendage.

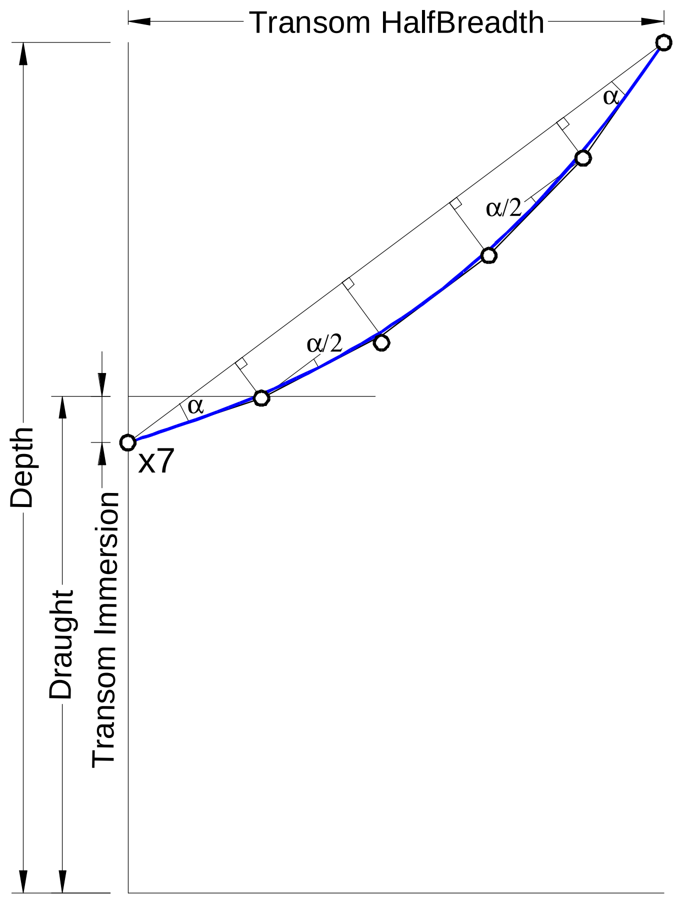


Figure 24.19, the construction used for form the transom curve control polygon.

24.5.1.10. The Aft Flat of Side Control Curve

The aft flat of side control curve is formed using a similar procedure to the development of the forward flat of side control curve, through the offset of the shape of the midship section curve control polygon, (Figure 24.20). However, unlike the forward version of the curve, the flat of bottom segment is taken directly to the centre line. The surface control polygon is deformed for the propeller bossing appendage such that there are no longer any vertices on the flat of bottom that can be used to control the shape.

The simple shape of the control polygon is formed by considering the longitudinal extents of the parallel middle body and parallel deck, (Figure 24.21). The control polygon is formed linearly between the two points considering further details around the bilge radius area.

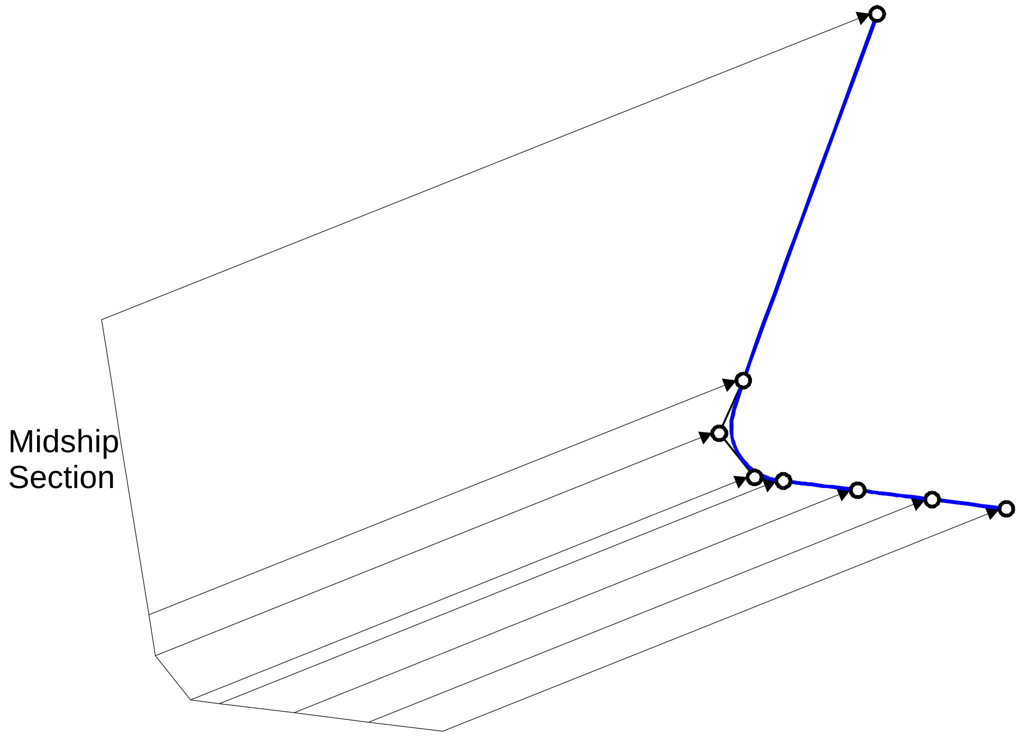


Figure 24.20, like the forward flat of side control curve, the shape of the control polygon is developed by offsetting control polygon vertex locations from the corresponding vertices on the midship section.

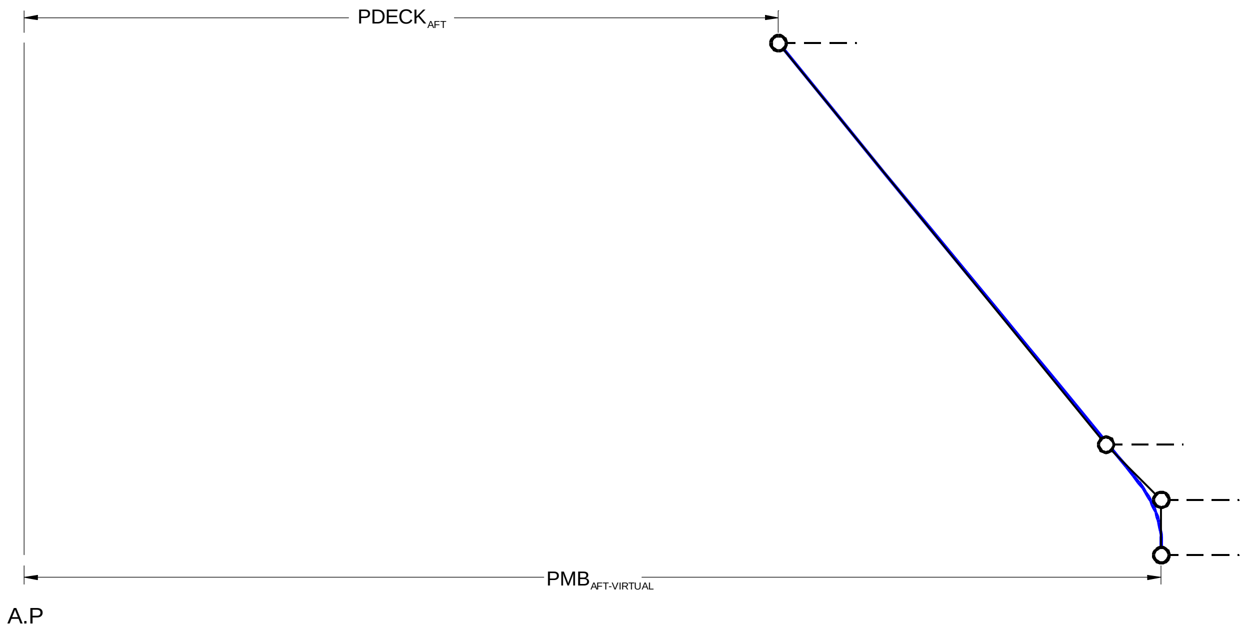


Figure 24.21, the shape of the control polygon is developed by considering the longitudinal extents of the parallel deck and parallel middle body.

24.5.2. Construction of the Blending Curves

The conceptual approach of the blending curves is to fair the shape of the surface control polygon rows from the tangential arrangement at one point in the surface to another with perhaps some control of the fullness of the shape between. This procedure was developed into a more generic approach in TSCAHDE, where the approach is to develop a family of curves. ShipLINES was used to initiate this approach. However, due to the consideration of including practical appendage shape into the surface, the simplicities and effectiveness of the blending curve approach was not maximised.

24.5.2.1. Development of the Bow Blenders Curves

The development of the curve for blending the entrance of the hull form surface laid down the foundations for the approach used within TSCAHDE. However, the approach used within ShipLINES does not consider the wealth of information that can be obtained from the topology of the hull form definition. Consequently, the development process is just one of considering the intersections between form curves.

The blending curves arrangement used to form the bow use a five vertex control polygon of Bezier order. The tangent at the stem is formed between the stem tangent and stem blender tangent control curves, (24.5.1.4). At the flat of side, the tangent is formed between the flat of side curve, at the transverse location corresponding to the position of the vertex on the midship section curve. The length of the tangent is based on the procedure discussed in 24.5.1.5.

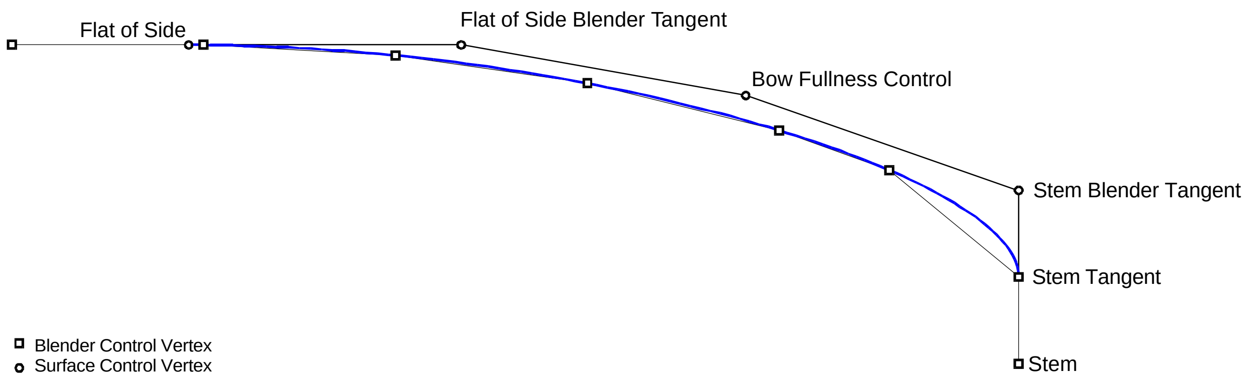


Figure 24.22, construction of the bow blender curves.

The location for the fullness control vertex is calculated from the bow fullness control curve, (24.5.1.6), by considering the height halfway between the *z* components of the two tangent control vertices.

24.5.2.2. Development of the Aft Blending Curves

Due to the nature of the complexity of the surface control polygon in the stern of the vessel, a process similar to the approach used for the bow cannot be used. A variety of techniques were attempted to find the ideal way of developing the definition control vertices. However, the existence of the appendage shape has the overriding control to how the control polygon mesh should be developed. A procedure developed around the approach of ensuring a fair control polygon mesh was finally decided upon. As the control polygon has quite a complex shape, blenders curves are developed two dimensionally to make the process easier. However, three sets of curves are now required to develop the location of the control vertices. These sets control the shape of the control polygon from the point of view of the rows, the columns and a final set to develop the offset locations of the vertices.

The row blender curves are formed using the aft flat of side and aft flat tangent control curves, (24.5.1.10), and connect to corresponding points on the transom, (24.5.1.9), or the stern curve, (24.5.1.7), depending on which control vertex is the first to be controlled by form curves or for the development of the bossing appendage.

The column blender curves are formed by developing curves at the basic longitudinal locations of where the columns are located. The blending only takes place for the part of the control polygon that will be deformed, the remaining part of the curve is aligned with the location of the section.

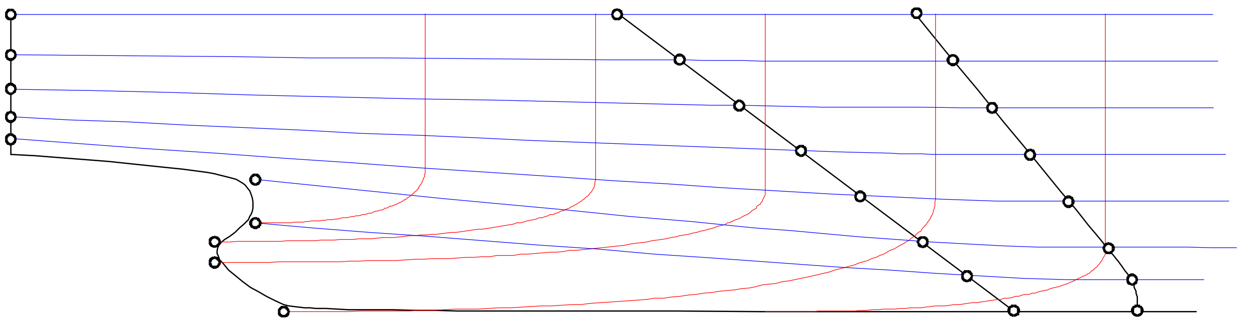


Figure 24.23, blender curves are formed for the rows (blue) and columns (red) of the parts of surface control polygon forming the propeller bossing appendage.

Both the row and column blending curves are developed on the centre plane. The intersections between the two curves form the *x*-*z* locations for the control polygon vertices for the surface mesh. The offset for the control polygon is found by developing a third set of blending curves in the horizontal plane. There is correspondingly one curve for each control polygon vertex. The curve is constructed from the flat of side curve, the flat of side curve tangent and the transom. A fourth intermediate point is located halfway between the transom and the flat tangent arrangement. The location of this vertex is found from the projection curve, (24.5.1.8).

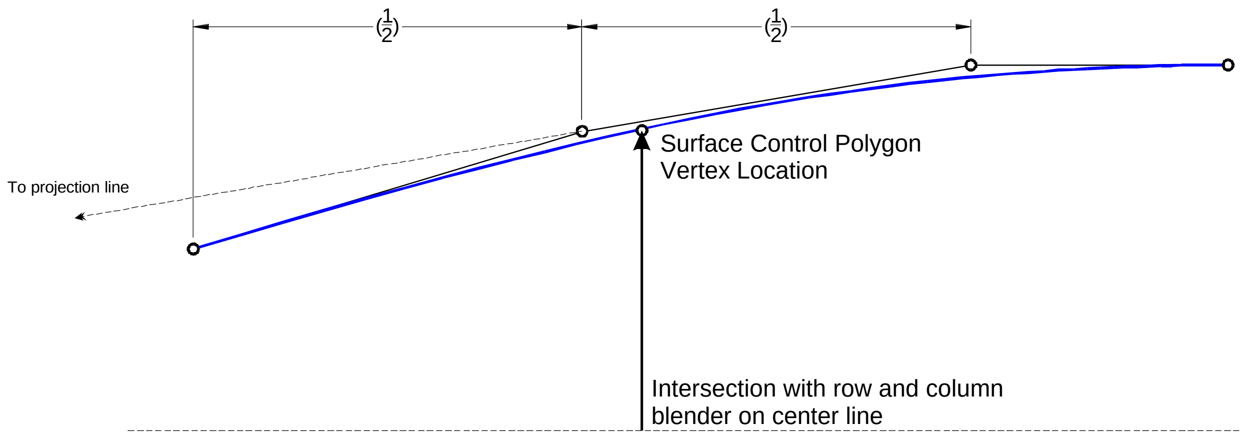


Figure 24.24, the third blending is used to find the offset of the control polygon vertices within the area around the propeller bossing appendage using the intersections between the row and column blender curves on the centre plane.

24.5.3. Construction of the Surface Control Polygon

Once the blending curves have been constructed, the development of the control polygon for the surface representation is quite straight forward. Excepting for the complexity that is developed around the propeller bossing appendage, control polygon columns are generated by intersecting with the blending curve with a planar section. The locations of the control vertices result from this intersection, (Figure 24.25).

The sectional cuts are determined by considering the number and separation of sections in each part of the hull form. There are five sections between the bow arrangement and the forward flat of side curve and the number of section behind the aft flat of side curve is fixed due to the nature of the bossing appendage. Between the flat of side curves the number of sections is adjusted, depending on the length of the parallel middle body. In ShipLINES, due to the construction of the hull form, the actual extent of the parallel middle body was found to be from the next set of control polygon columns in from the flat of side curves within the parallel middle body. This was found to be one of the consequences of using section based control polygon columns. In TSCAHDE, the arrangement of the columns is adjusted so that the surface is constructed more accurately, accounting for the nature of NURBS surfaces. To compensate for the fact that the actual extent of the parallel middle body does not meet the flat of side curves, the parameters forming the curves are adjusted from the value entered by the user, based on the section interval between the columns.

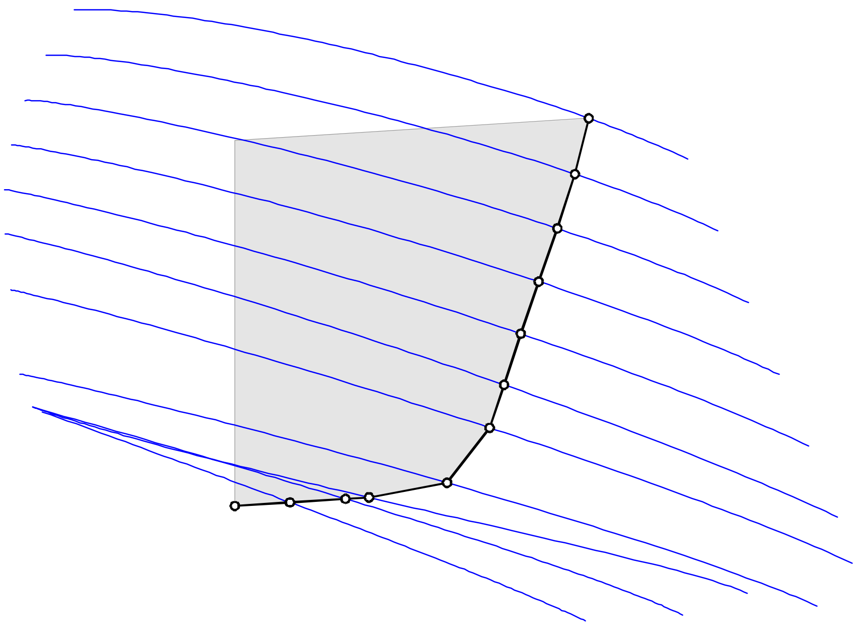
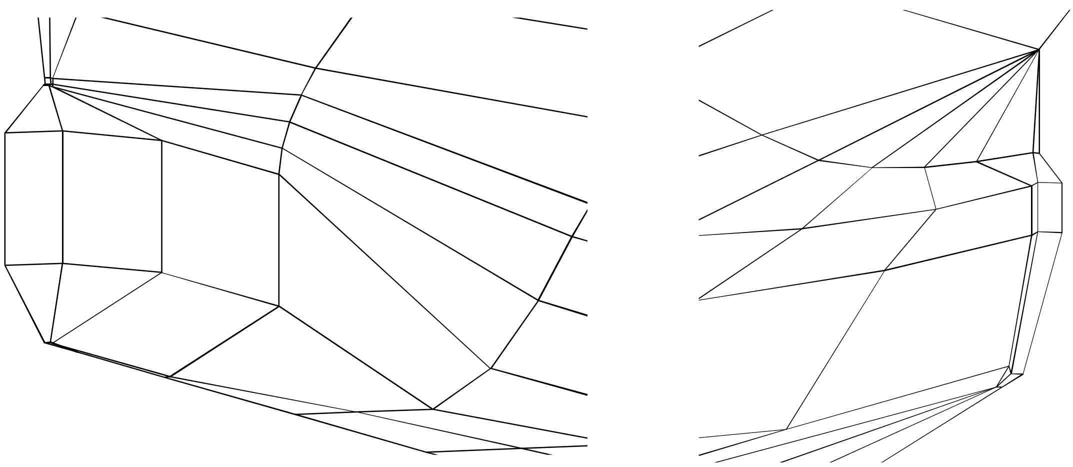


Figure 24.25, control polygon columns are developed by intersecting the blending curves with a section plane.

24.5.4. Appendage Control Polygon Constructions



a) b)

Figure 24.26, the control polygon mesh arrangements for a) the bulbous bow and b) the propeller bossing appendage.

Local appendages have quite a tight and therefore, easily developed shape, (Figure 24.26). As a result, it is possible to develop the control polygons for these shapes directly into the surface representation. In ShipLINES, two different approaches are used to develop these shapes. In the case of the bulbous bow, the shape is developed after the main surface control polygon has been form because it is easier to develop a family of smooth blending curves without considering the bulb. For the propeller bossing appendage, the development of the shape has to be considered during the development of the main surface because the control polygon is required to be heavily deformed from a uniform mesh shape.

24.5.4.1. Forming the Bulbous Bow Appendage

To form the bulbous bow, a box shape arrangement is developed in the surface control polygon, (Figure 24.27). Three parameters are used to control the shape. To accommodate the bulb definition, during the development of the main control polygon, some of the affected columns are generated in different location to the bulb-less hull form. Once the box arrangement has been formed, some of the columns are smoothed to ensure that there are no unfair areas in the surface transitioning into the bulb shape.

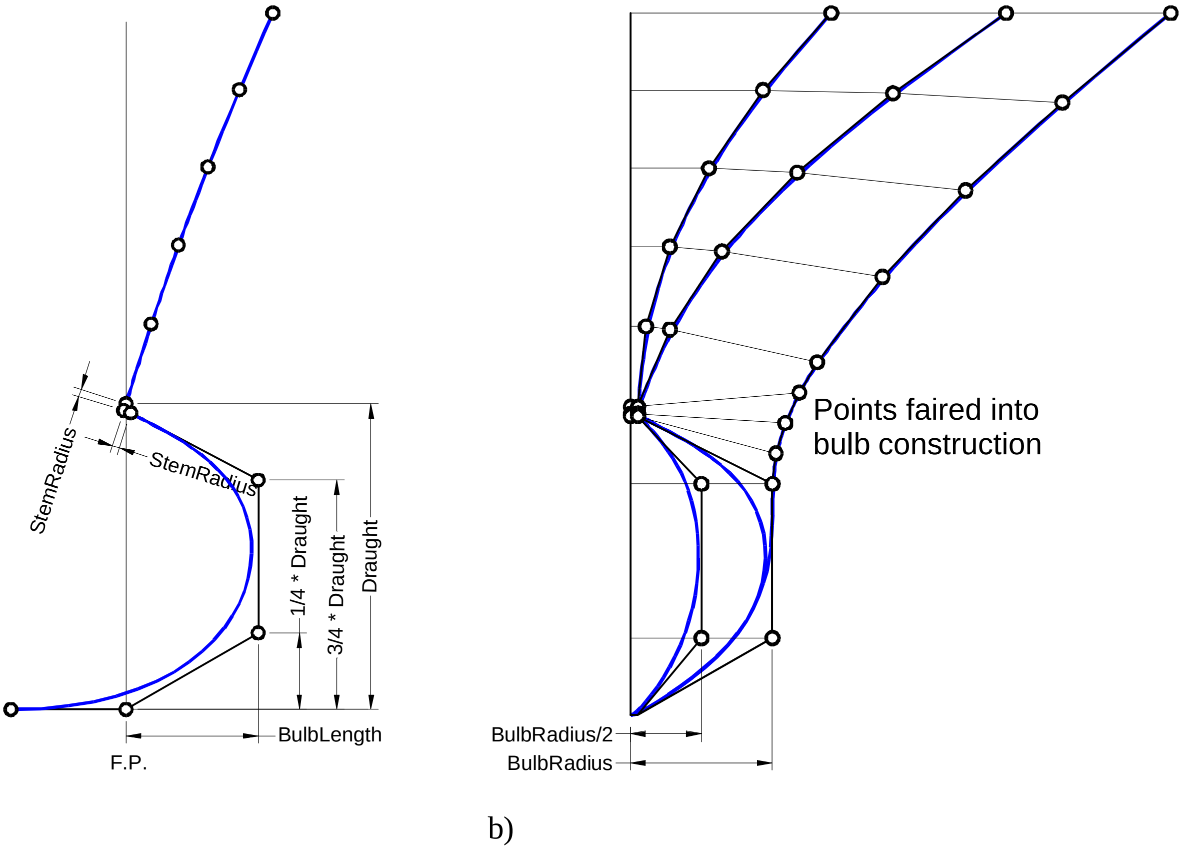
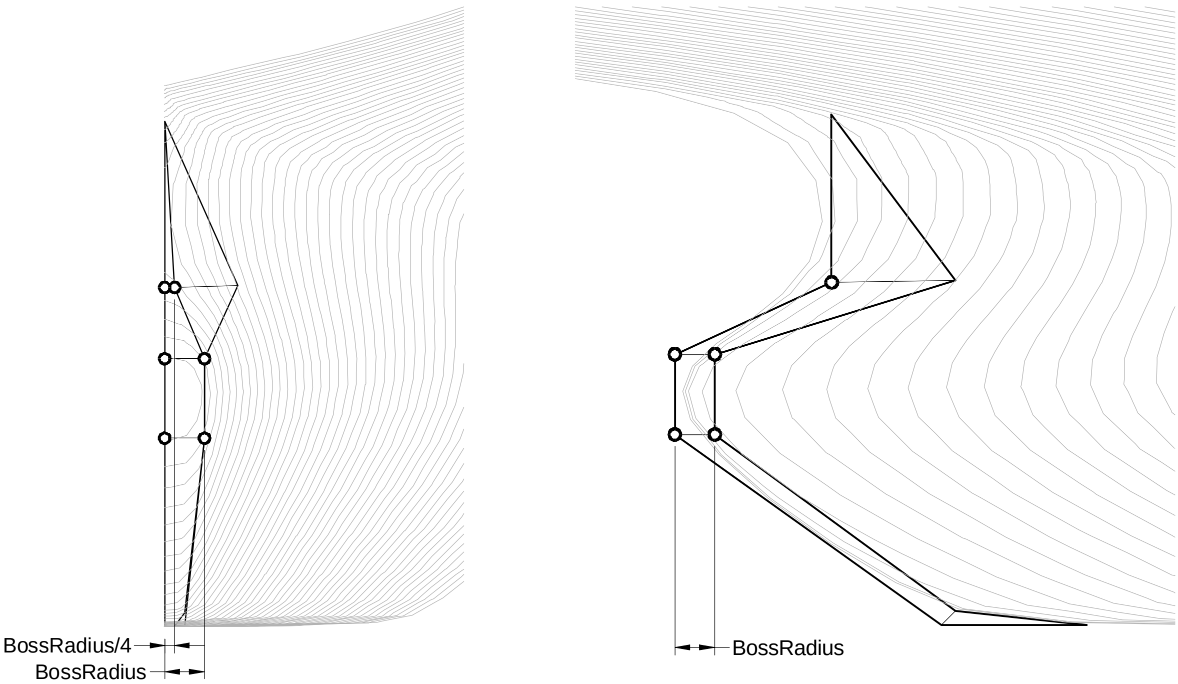


Figure 24.27, the direct modification developed in the forward four control polygon columns implementing the bulbous bow shape.

24.5.4.2. Forming the Propeller Bossing Appendage

The propeller bossing definition is formed before the blending curves are generated, to ensure that there will be a smooth transitioning shape. Consequently, there does not actually need to be so much direct manipulation of the surface control polygon. Based on the shape of the stern curve, tangents are setup the corresponding control polygon rows to develop the sharp rounding, (Figure 24.28). Subsequently, the blending curves use the tangent control vertices as termination points.



a) b)

Figure 24.28, to develop the sharp rounding of the propeller bossing, tangents are first developed a) transversely from the stern curve definition and then b) forward.

24.5.5. Parameters

With the parametric and geometric generation procedure in place, the full range of parameters is detailed below.

- BowShape – Quantifies whether the hull is developed with a bulb (values: bsBulb, bsNoBulb).
- LBP – Length between perpendiculars.
- Beam
- Depth
- Draught

- ParallelDeckF – Distance to the forward point of the parallel deck from A.P. (Figure 24.15).
- ParallelDeckA – Distance to the aft point of the parallel deck from A.P. (Figure 24.20).
- FOSBow – Distance to the forward point of the parallel middle body from A.P.
- FOSAft – Distance to the aft point of the parallel deck from A.P.
- BowRadiusAtDeck – *(Local)* Controls of the amount of flare in the bow, (Figure 24.13).
- BilgeRadius – *(Local)* (Figure 24.11).
- TransomBehindAP – *(Local)* The location of the transom behind A.P. (Figure 24.17).
- TransomImmersion – *(Local)* The location of the bottom point of the transom below the waterline (draught), (Figure 24.17).
- TransomBeam – *(Local)* Breadth of the transom at the deck, (Figure 24.19).
- TransomCurvature – *(Local)* Percentage factor used to control the ‘arc’ construction forming the transom, (Figure 24.19).
- BowCurvature – *(Local)* Percentage factor used to control the ‘arc’ construction forming the curved stem shape (Figure 24.12).
- BulbRadius – *(Local)* controls the transverse extent of the bulb, (Figure 24.27).
- ForeOverHang – *(Local)* The distance between the forward extents of the stem curve and F.P. (Figure 24.12).
- BulbLength – *(Local)* forward extent of the bulb control polygon structure in front of A.P. (Figure 24.27).
- ForeFootRadius – *(Local)* radius of the fillet at the foot of the stem curve (R_{FF}), (Figure 24.12).
- StemRadius – (Local) size of the tangents affecting the surface at the stem below the waterline, (Figure 24.13,Figure 24.27).
- RootHeight – (Local) (Figure 24.17).
- ShaftHeight – (Local) (Figure 24.17).
- BossToRoot – (Local) (Figure 24.17).

- BossRadius – (Local) (Figure 24.17).
- BossStart – (Local) (Figure 24.17).
- FlatOfBottomStart – (Local) (Figure 24.17).

24.6. Implementation

The implementation of the generation procedures into a useable tool was fairly straight forward. ShipLINES was developed using PolyCAD and the associated geometry library. Consequently, a graphical user interface to the procedure could be developed very easily and information on intermediate steps of the generation process could be transferred to PolyCAD for review seamlessly. The majority of the code is dedicated to the positioning of the form curves. The remaining code developed the hull form considering the intersections between form curves and the blenders, and the blender and the columns.

The technique was wrapped into a standalone application, (Figure 24.29), which provides the user with parametric control over the hull form using an unsuccessful graphical approach to the problem of educating the user to the function of the parameters. Furthermore, as with YachtLINES, the technique is implemented within PolyCAD, (Figure 24.30), which also provides the user with the ability to extract generation geometry.

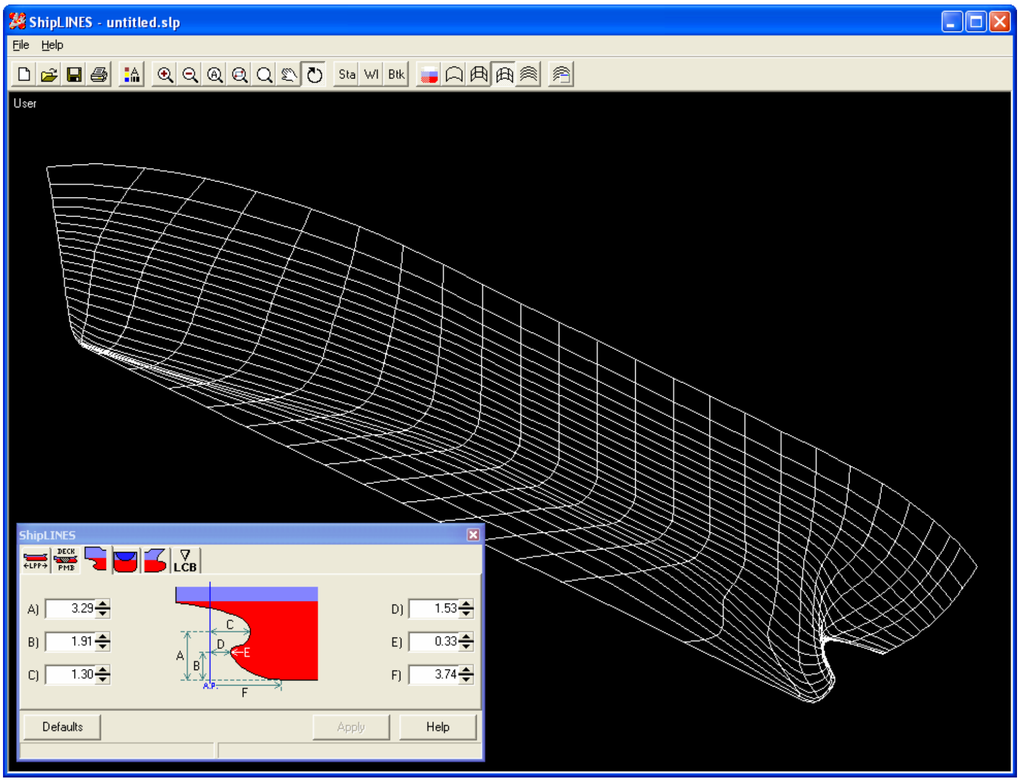


Figure 24.29, the ShipLINES standalone software implementation.

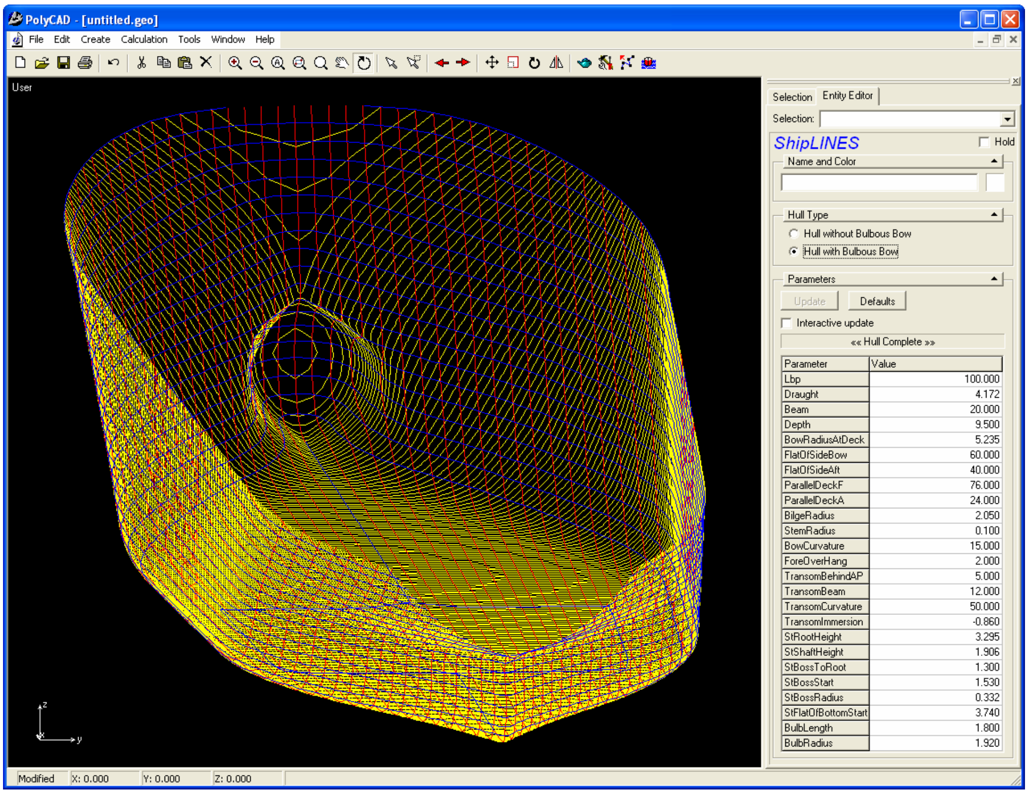


Figure 24.30, ShipLINES implemented within PolyCAD.

24.7. Examples

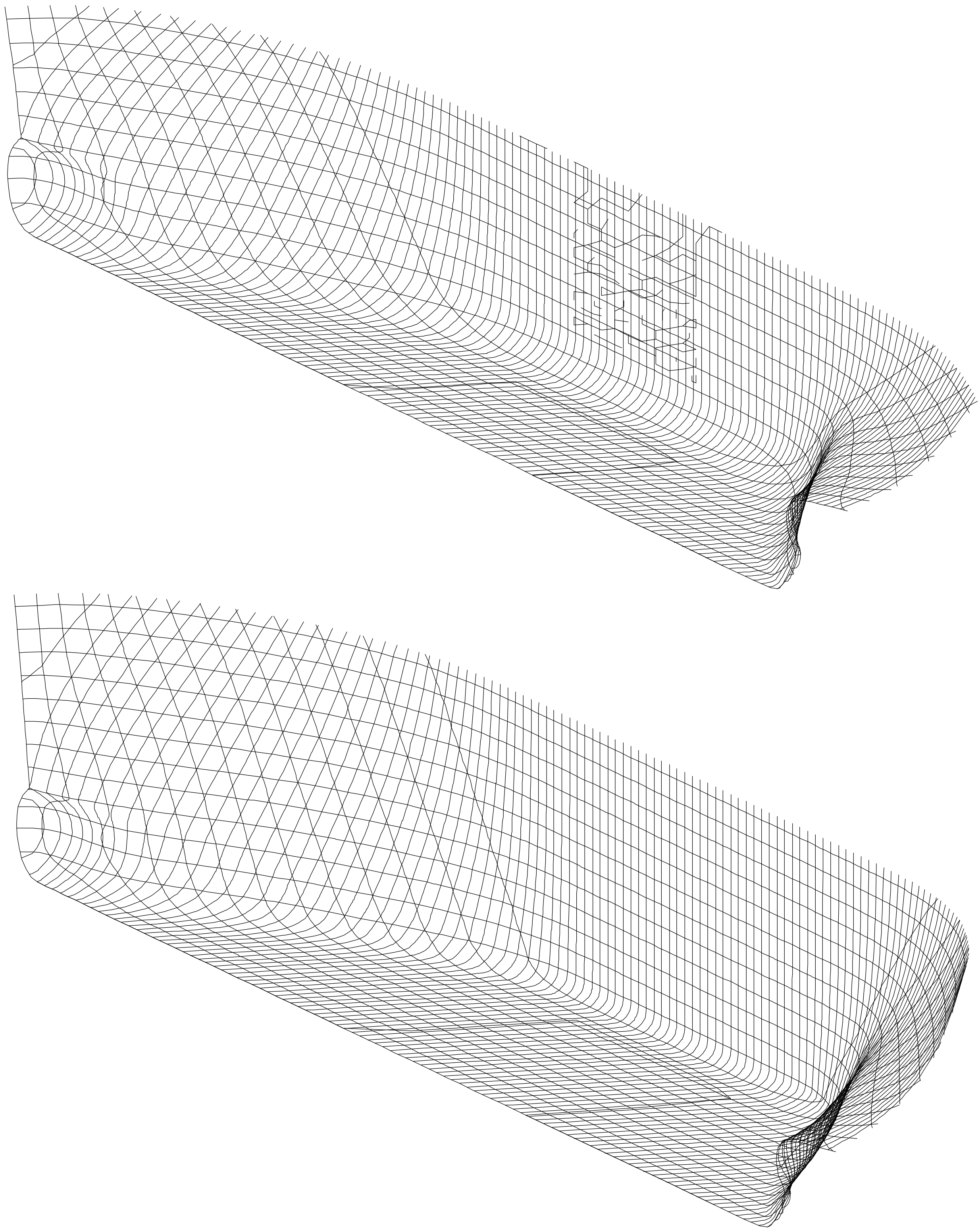


Figure 24.31, Modification of Principle Dimensions, Beam (20.0 → 25.0m), Depth (9.5 → 12.5) and BilgeRadius (2.05 → 1.0m)

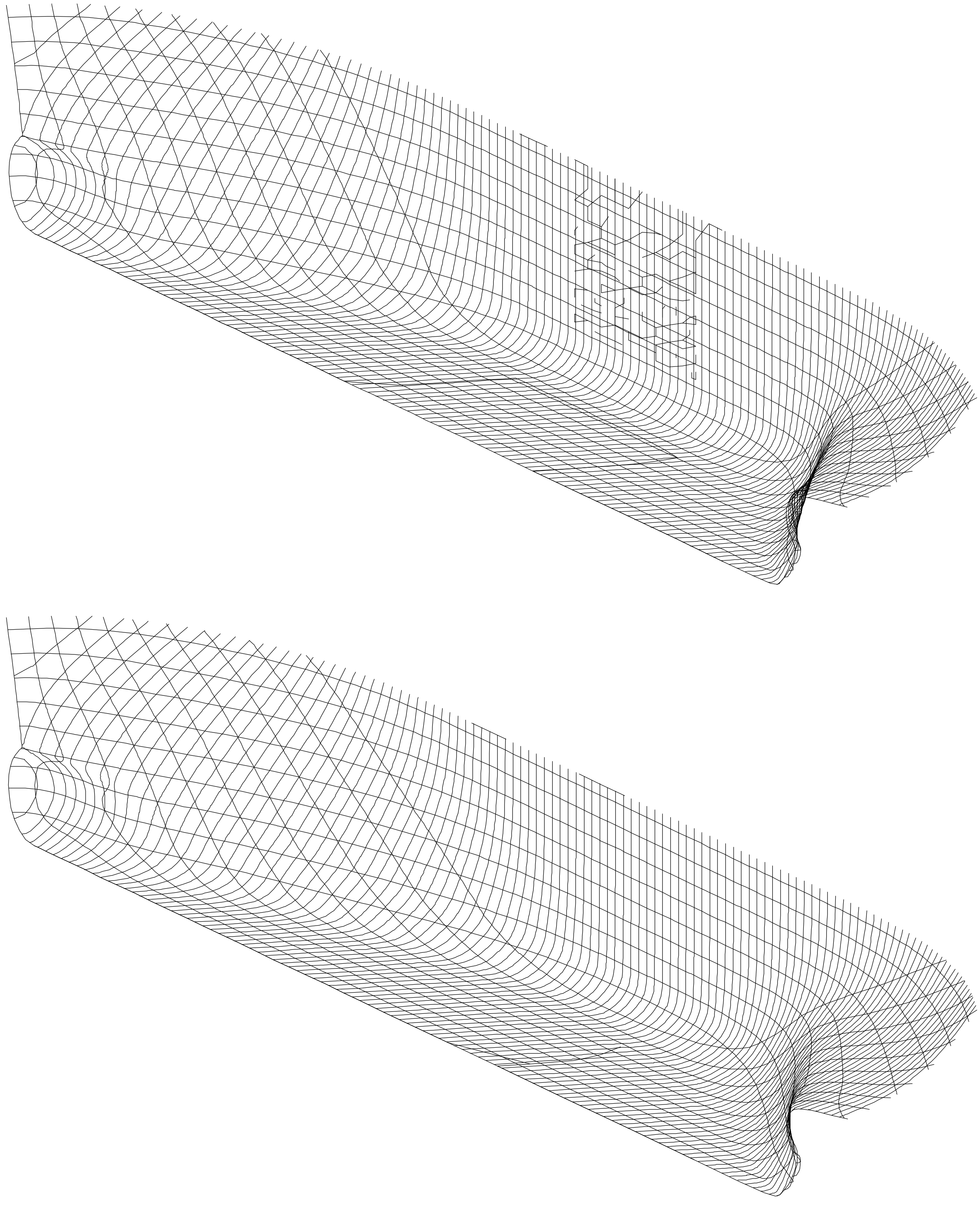


Figure 24.32, Modification of the Parallel Middle Body Parameters, FlatOfSideBow (60.0 → 50.1m), FlatOfSideAft (40.0 → 49.9m), ParallelDeckF (76.0 → 80.0m), ParallelDeckA (24.0 → 20.0m).

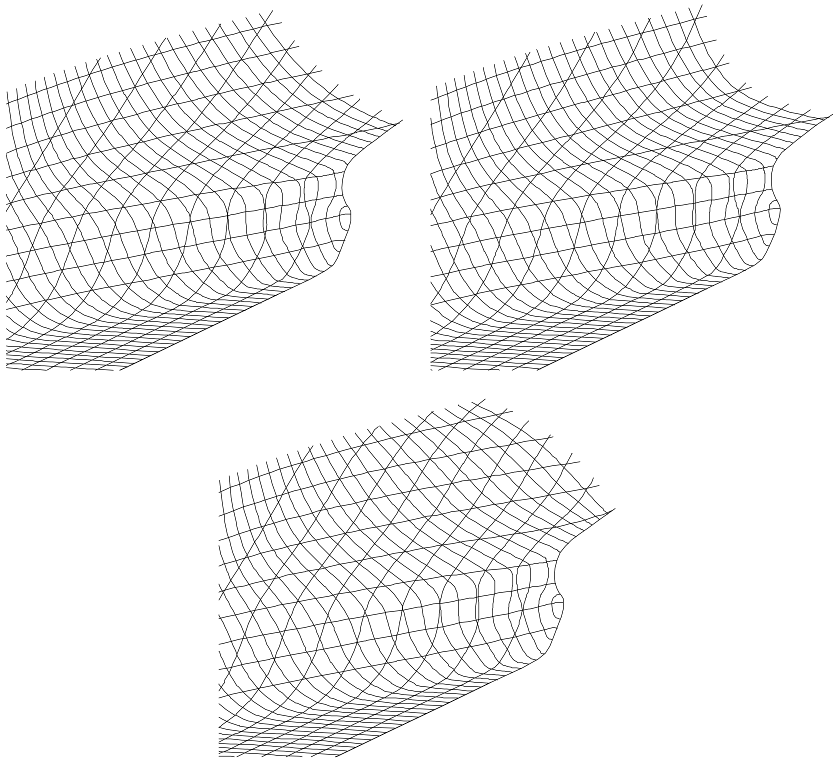


Figure 24.33, Modification of the transom curvature, 50%, 100%, -50% respectively.

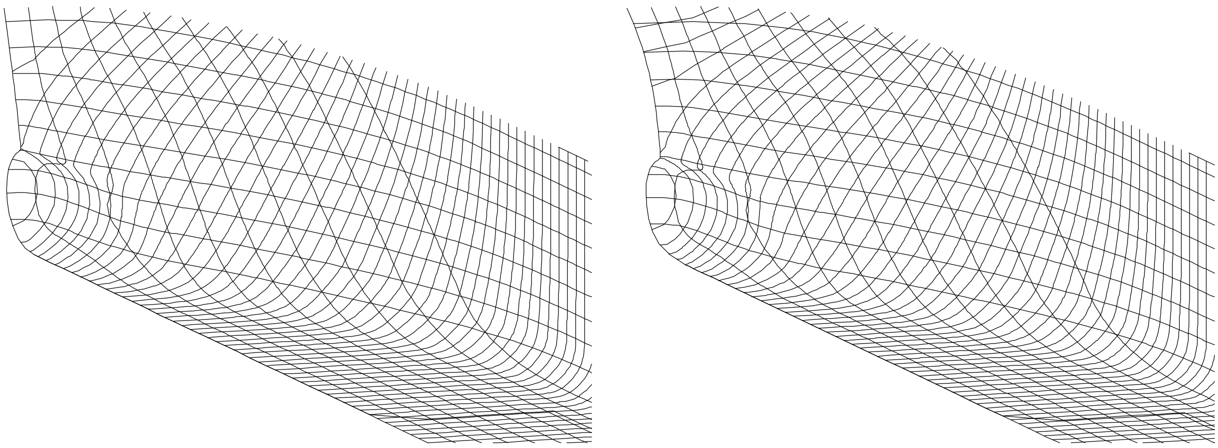


Figure 24.34, Modification of the StemRadius (0.1→ 0.25m), BowCurvature (15% → 60%) and BowRadiusAtDeck (5.235 → 7.0m)

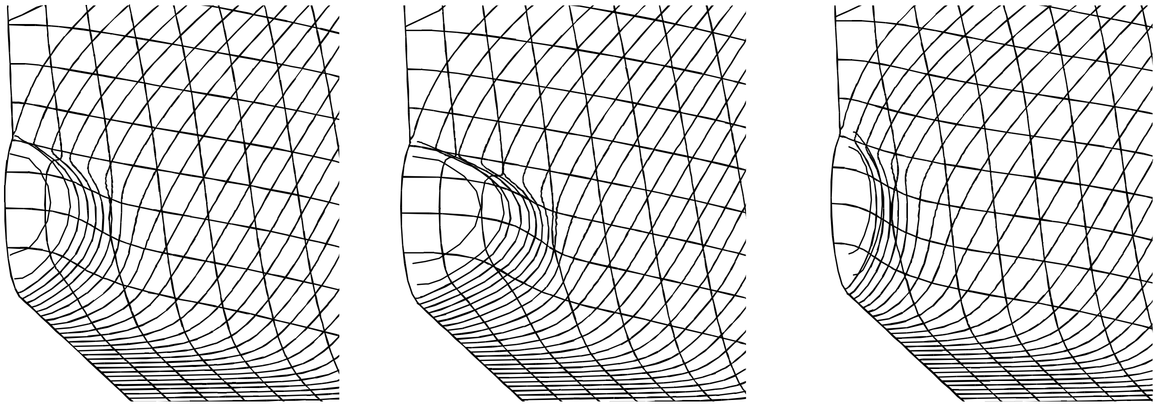


Figure 24.35, Modification of the BulbRadius through 1.92m, 3.0m and 1.0m.

24.8. Discussion

While it can be demonstrated that the generation procedure is capable of developing hull form surfaces of reasonable quality, the tool itself cannot be considered a great success. The development was approached from the idea that it is possible to develop a practical hull design tool using parametric generation procedures. Unfortunately, it was found that the inclusion of many practical features such as the bulbous bow and the propeller bossing appendage forced a level of complexity into the procedure to the extent that is just not possible to consider all this information in one step of processing and still provide a flexible tool to the designer. The result of these practical considerations is that the procedure becomes so rigidly based around the development of the local shapes that there is no longer any scope to deal with more important effects such as hydrostatics.

A lack of experience with NURBS, generation procedures and CAD can be seen as one of the major reasons why the results of the development were not as successful as hoped. Unfortunately, there are many areas in the process that are not dealt with properly. One of the most significant issues that is highlighted in reviewing the results of the technique is that to achieve particular shapes in the surface, there are specific requirements that need to be followed in the generation of the control polygon arrangement. Making sure that control vertices on the side of the vessel stay in that region is important to prevent accurate shapes like the flat of side being deformed. This deformation is not detectable visually. However, it is visible when reviewing accurate contours calculated from the surface. Other features of the technique which results in a performance loss is the use of section orientated columns. Again, visually the surface shape appears to be accurate. However, this approach is not capable of properly controlling the shape of the surface across the flat boundaries unless the columns of the control polygon follow the shape of the flat boundaries.

There are a considerable number of disadvantages that a hull generation technique imposes on the practical design process. To develop a practical hull form surface parametrically, it is necessary to obtain all the information to generate the shape from the supplied numerical parameters. As the main particulars have no reference to local shapes such as the shape of the bulb, new parameters must be introduced. A consequence of this, as ShipLINES demonstrates, is that there are many strangely named parameters that are not only difficult to understand, but it is also difficult to gauge how modification to these parameters will affect the shape of the hull form. Controlling aspects such as the main particulars of the hull is reasonably easy. However, development of local shapes can be quite complex. An example of the level of complexity is demonstrated in the development of the stem shape, (Figure 24.12). To allow for some variation in the shape, the curve is constructed allowing the user the ability to control the radius of the forefoot and the curvature of the stem. The construction of these features in combination can be quite awkward. However, even though a solution can be found, the result is not able to match the level of flexibility that could be obtained if the user directly manipulated the control vertices of the particular feature shape. One can conclude that, based on the results of the ShipLINES development, the idea of *designing* hull form surface using numerical parameters alone is impractical. A hybrid tool that uses an appropriate approach for controlling each aspect of the hull form will provide a better solution that a tool based around the religious implementation of a certain interaction strategy.

While ShipLINES cannot be considered as a successful practical tool, the development has yielded a great deal of knowledge about how to develop hull form surfaces, particularly, in the identification of successful approaches. In the development of ShipLINES and subsequently TSCAHDE, it can be seen that the development of hull form surfaces requires a great deal of skill. Most developers creating these types of tools are unlikely to have obtained these skills. However, during development, there has been a great deal of observation to how hull surfaces are formed. One important observation is that hull form surfaces, from the ideal of the parametric hull generation tools, are not as independent as the parameters suggest. There is actually quite a great deal of dependence across surface shape. For example, the shape of the stem has influence on the arrangement of the form flare. Consequently, if the shape of the forward flat of side curve is not compatible, the flare will be deformed. A successful hull generation tool must consider these dependences in the approach it takes to form the shape of the hull surface from one feature to the next.

One of the most important ideas that have been born out by this research is that if the technique uses a relatively simple concept and approach, the more flexible and hence, successful the technique appears to be. This can be demonstrated by comparing both the ShipLINES and TSCAHDE techniques. ShipLINES uses a variety of different methods to develop the geometry, to the extent that the process has become difficult to follow and the procedures used to develop geometric components are incomprehensible. TSCAHDE, developed as the hybrid solution to the hull design problem, uses many of the same components from ShipLINES, such as the form curves and the blending functions. However, because the technique did not require so much geometric complexity as a result of the number of parametric controls, the technique is more consistent and easier to understand.

To be considered as a practical hull generation tool, a technique must provide the user with the ability to achieve the same type of shapes that can be developed with manual surface development tools. ShipLINES attempted to provide these features by including the bulbous bow and the propeller boss appendage. In the case of the latter, the complexity required to develop this shape was the undoing of the technique. It demonstrates that it is not practical to consider the development of the hull form, with all its features in one generation step. Local features have there own characteristic shape, which is not influenced by the overall shape of the hull form. Furthermore, the overall shape of the hull form should not be influenced by the shape of local features. This independence promotes the approach of developing each part separately. In the case of the bulb feature, independence between the development of differently shaped parts was achieved, although this was mainly to simplify the implementation code. As hull form generation is a purely automated process, it is possible to extend the idea to allow the hull form to be developed as a series of iterative construction steps. For example, first the basic hull form shape could be formed. Then it could be successively modified to include the appendage features, faired into the shape of the main surface.

ShipLINES, as a tool, was never finalised to a completed state. Once a solution to create the stern portion of the surface was achieved, the complexity of the development process was too great and the fact that properties, such as the hydrostatics, can not be controlled is a significant drawback. However, the experience gained with ShipLINES fostered many ideas based toward making more effective use of tools and to how information that can be extracted from the hull definition structure. TSCAHDE was developed on the basis of these ideas.

25. APPENDIX 4 – TSCAHDE PARAMETERS: DEFINITION AND TRANSFORMATIONS

25.1. LBP (Length between Perpendiculars)

25.1.1. Definition

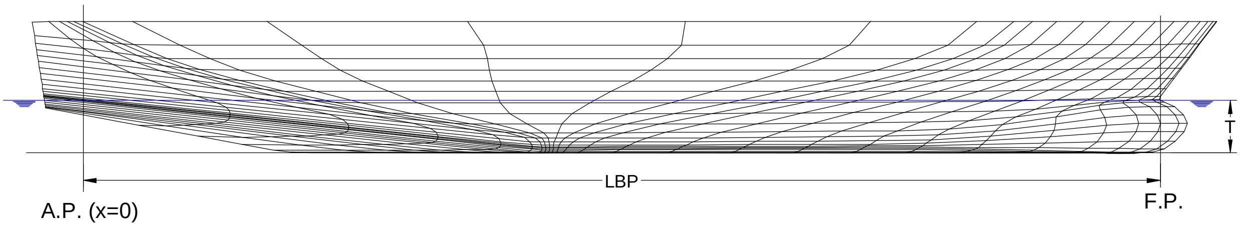


Figure 25.1, definition of LBP in TSCAHDE.

LBP is defined as the distance between A.P., which is defined to lie at x = 0, and the location where the stem intersects with the water plane at the defined draught T, (Figure 25.1). In the implementation, LBP is calculated, during hydrostatics, by considering the maximum forward extent of the underwater body. Being originally based on calculations for yacht hull forms, this approach has yet to be adapted to consider the possibility of the existence of bulb appendages.

25.1.2. Transformation

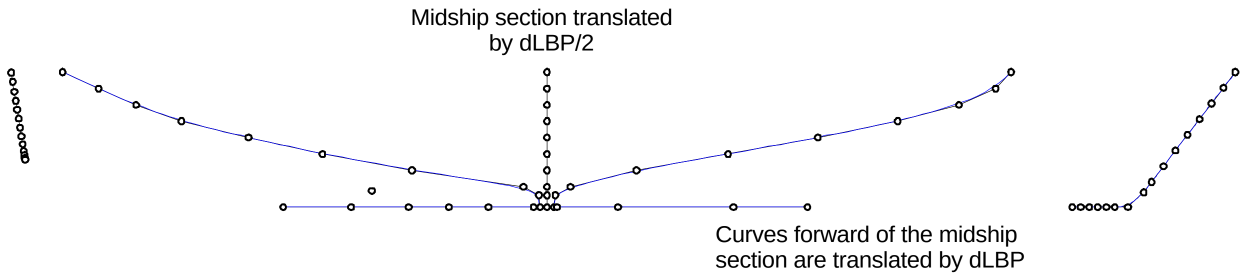


Figure 25.2, length can be transformed through the translation of definition curves alone.

Unlike most other hull definition techniques, the discrete nature of the definition curves used in TSCAHDE allows the hull to be lengthened by translating the curves alone. Lengthening the ship by increasing the parallel middle body only requires curves forward of the midship section curve to be translated by the change in LBP and the midship section curve by half the change. If the definition does not have a representation of the parallel middle body then it is possible to perform the operation using standard affine scaling techniques. In the reduction of LBP, the transformation is implemented so that the parallel middle body is first reduced. If this is not sufficient then a scaling transformation is used to take change the definition over the remaining distance.

25.2. BWL (Beam at Water Line)

25.2.1. Definition

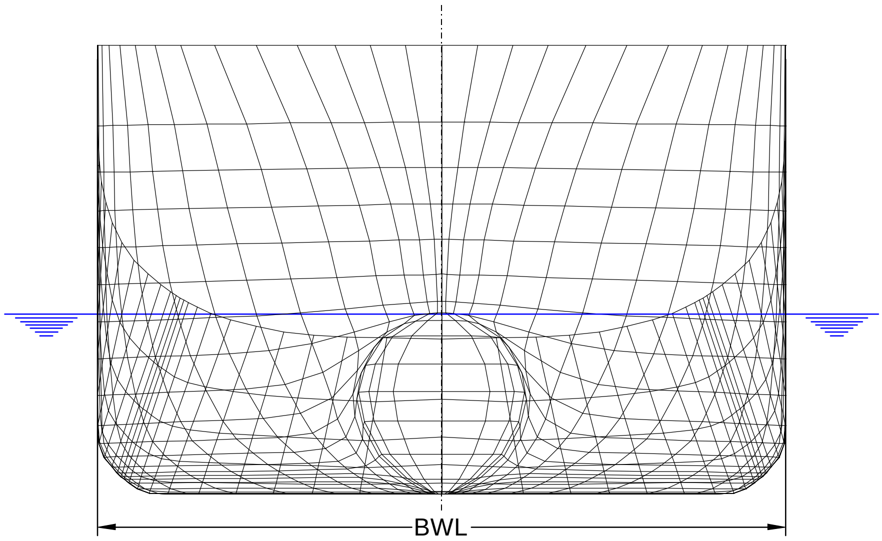


Figure 25.3, definition of BWL in TSCAHDE.

The waterline breadth is calculated as the maximum transverse extents of the underwater body intersected at the water plane defined by draught T, (Figure 25.3).

25.2.2. Transformation

One of the major advantages of the form topology approach is that it is possible to use local transformations to change the definition. A certain part of the definition can be modified and the form topology structure takes care of modifying any related geometry by the nature of the hull shape. This creates the opportunity to develop transformations for changing the hull form that minimise unwanted distortion to the hull form shape. The BWL transformations implements this approach by preventing the bilge radius shape from being distorted under the usual affine scaling procedure. A point is selected, (Figure 25.4), based on a discrete analysis of the curvature of the control polygon, (26.4.2), all the points with indices less than the selected point are scaled and points greater are translated. This procedure for implementing this operation is as follows:

```
// select the scale/move “pivot” point
Iselect = (found from procedure detailed in 26.4.3)
```

```
// calculate scale and move factors
C_scale = (BWL_new – (BWL_current – P[I_select].y × 2)) / (P[I_select].y × 2)
C_move = (BWL_new – BWL_current) / 2

// scale control vertices before the "pivot" point
for i = 0 to I_select
      P[i].y = P[i].y × C_scale

// move control vertices after the "pivot" point
for i = I_select-1 to P_numpoints-1
      P[i].y = P[i].y + C_move
```

Only curves that are manually defined, i.e. not automatically generated by the technique, are transformed. Furthermore, the procedure adapts the point selection process if the curve represents a tangent or a flat. In these cases, the point selection procedure uses the curve from which the tangent definition is directed. If a "pivot" point cannot be selected, in the situation that the curve does not have a region of high curvature, the selected point defaults to a location allowing the whole curve to be processed under scaling.

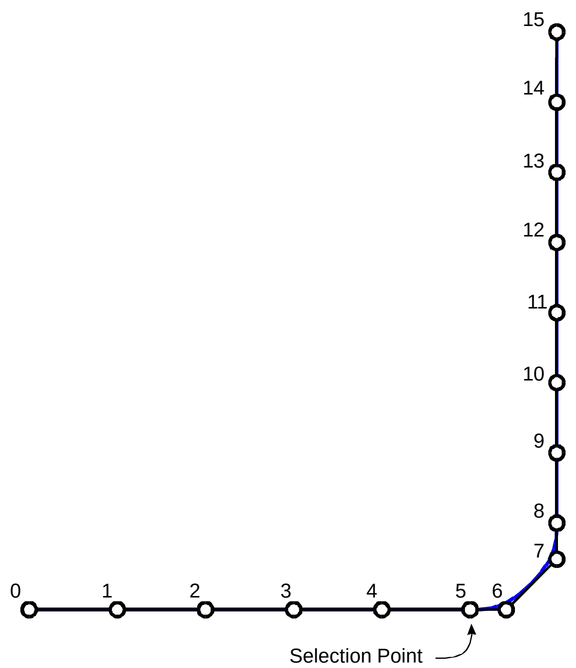


Figure 25.4, to minimise distortion of shape a "pivot" point is selected to locally transform about.

25.3. T (Design Draught)

25.3.1. Definition and Transformation

The draught parameter is considered to be a reference value used during design. Consequently, it defines the locations for which many of the other parameters are calculated. A change of the parameter does not result in any physical changes in the hull form. However, all the parameters are updated to the dimensions defined by the new reference draught.

25.4. D (Depth)

25.4.1. Definition

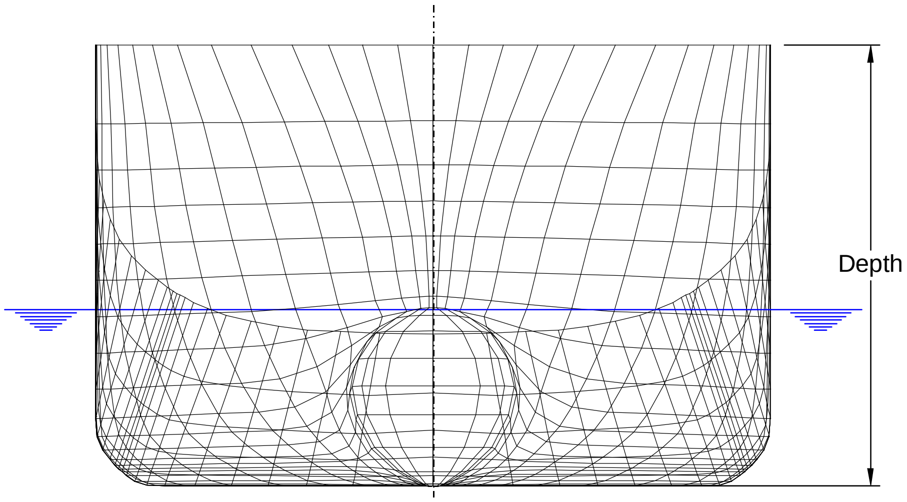


Figure 25.5, definition of Depth in TSCAHDE

The depth is calculated as the maximum vertical extents of the entire surface, (Figure 25.5).

25.4.2. Transformation

Transformation to change the depth is implemented in almost the same way as for BWL. The differences are that different point is selected and for depth, only vertices beyond the selected points are scaled, leaving the remaining point unchanged:

```
// select the leave/scale “pivot” point
I_select = (found from procedure detailed in 26.4.3)

// calculate scale factor
C_scale = (Depth_new – P[I_select].z) / (Depth_Current – P[I_select].z)
```

```
// scale control vertices after the "pivot" point only
for i = Iselect+1 to Pnumpoints-1
      P[i].z = P[i].z × Cscale
```

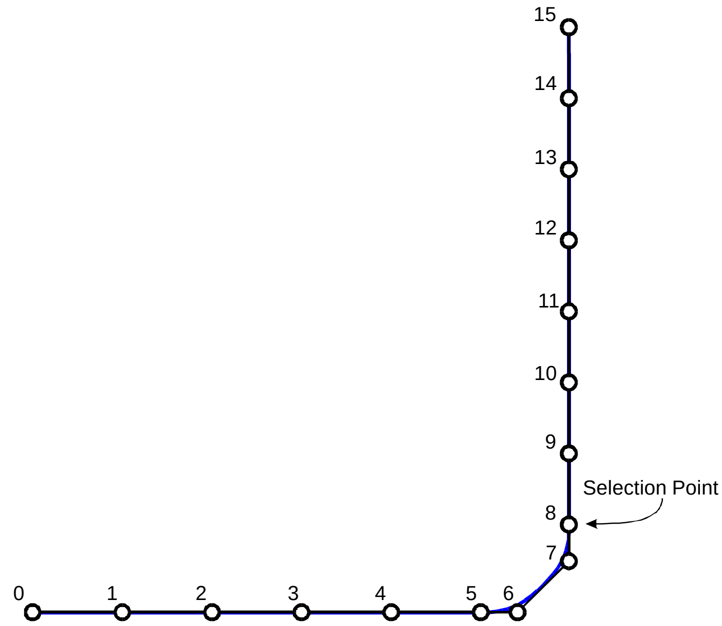


Figure 25.6, to transform for depth, points beyond the selection are scaled.

25.5. PMB (Length of Parallel Middle Body)

25.5.1. Definition

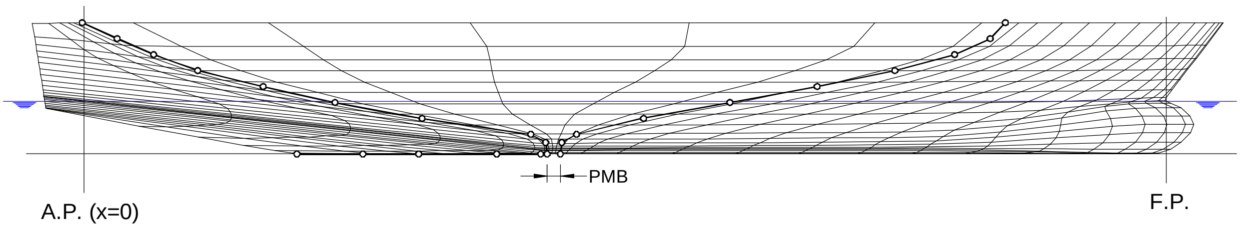


Figure 25.7, definition of PMB in TSCAHDE

The extents of the parallel middle body is defined as the distance between the aft most control vertex on the forward flat definition curve and the most forward control vertex on the aft flat definition curve, (Figure 25.7). It is calculated as follows: PMB = PMBF – PMBA.

25.5.2. Tranformation

The change in the extents of the parallel middle body (? PMB) is calculated as the difference between the present extents ($PMB_{current}$) and the new extents as desired by the designer, (PMB_{new}).

$$\Delta PMB = PMB_{new} - PMB_{current}$$
$$C_{PMBF} = \frac{PMBF - X_{MIDSECTION}}{PMB_{current}}$$
$$C_{PMBA} = \frac{PMBA - X_{MIDSECTION}}{PMB_{current}}$$
$$PMBF = PMBF + C_{PMBF} \times \Delta PMB$$
$$PMBA = PMBF + C_{PMBA} \times \Delta PMB$$

As PMB is defined from PMBF and PMBA, these parameters are modified on the basis of the ratio of the relative distances between the extents of the parallel middle body and the longitudinal location of the midship section, ($X_{MIDSECTION}$), to account for any differences in the longitudinal location of these curves from the midship section. Note that C_{PMBA} will calculate as a negative value.

25.6. PMBF (Forward Extent of Parallel Middle Body)

25.6.1. Definition

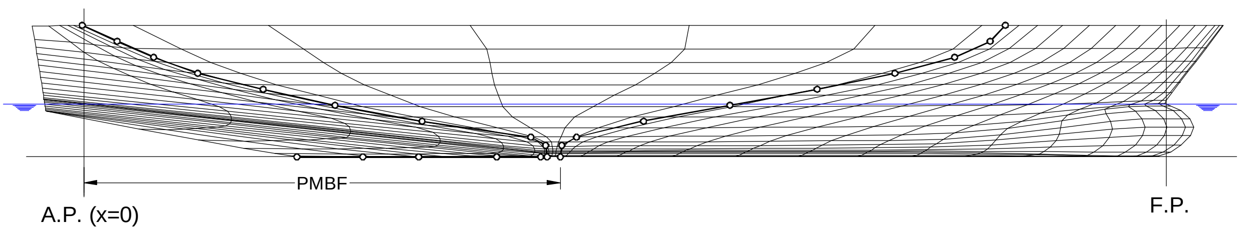


Figure 25.8, definition of PMBF in TSCAHDE

The forward extents of the parallel middle body is calculated as being the location of the aft most vertex of the control polygon defining the forward flat curve (FSF), (Figure 25.8).

25.6.2. Transformation

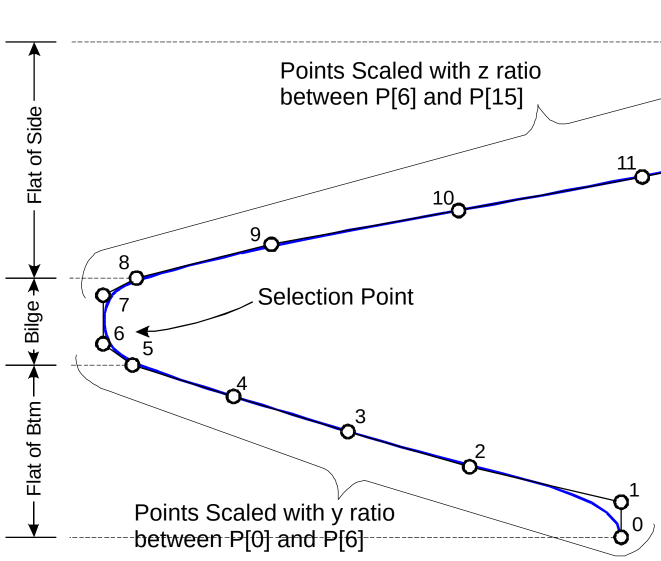


Figure 25.9, to change the PMBF, points on the flat of side and bottom are scaled on the basis of a z or y ratio, respectively.

To change the forward extent of the parallel middle body, a point is first selected on the basis of the minimum longitudinal position, (Figure 25.9). Control vertices beyond this selected point are linearly position in x on the basis of the *z* ratio between the deck and selected points. Control vertices below the selected point are similarly transformed using a ratio in *y*. This operation is only applied to the first curve forward of the midship section curve. The operation is implemented as follows:

```
dPMBF = PMBF_new - PMBF_current

// select the control vertex index with the smallest x
x = ABigNumber // (10e32)
I_select = -1
for i = P_numpoints-1 downto 0
    if P[i].x < x then begin
       x = P[i].x
       I_select = i
       end

// scale control vertices after the “pivot” point only (flat of
side)
for i = I_select to P_numpoints-2
    C_scale = (P[i].z – P[P_numpoints-1].z) / (P[I_select].z - P[P_numpoints-
1].Z)
    P[i].x = P[i].x + dPMBF × C_scale

// scale control vertices after the “pivot” point only
(flat of bottom)
```

```
for i = I_select to P_numpoints-2
    C_scale = (P[i].y – P[0].y) / (P[I_select].y - P[0].y)
    P[i].x = P[i].x + dPMBF × C_scale
          // check that points remain forward of PMBF
    If P[i].x < x + dPMBF Then P[i].x = x + dPMBF
```

25.7. PMBA (Aft Extent of Parallel Middle Body)

25.7.1. Definition

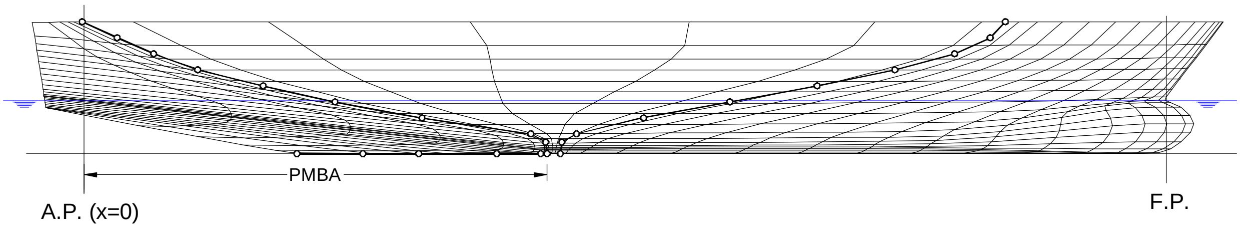


Figure 25.10, definition of PMBA in TSCAHDE

The aft extents of the parallel middle body is calculated as being the location of the forward most vertex of the control polygon defining the aft flat curve (FSA), (Figure 25.10).

25.7.2. Transformation

Change in the aft extent of the parallel middle body is implemented using exactly the same approach as for changing PMBF. The exception are that the point selection is made on the basis of the maximum longitudinal extent and only the first curve aft of the midship section is modified by this procedure.

25.8. PD (Length of Parallel Deck)

25.8.1. Definition

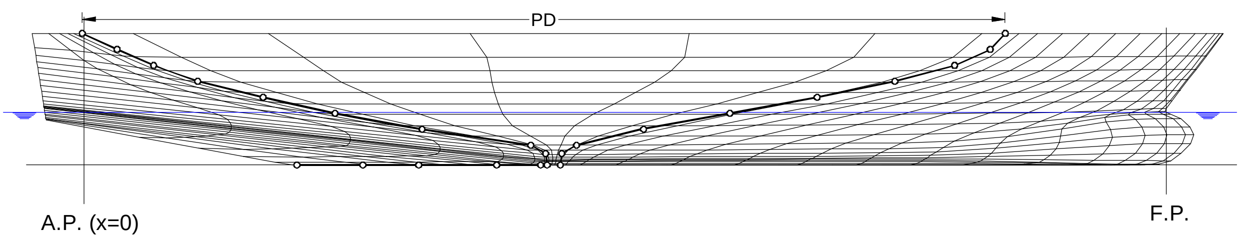


Figure 25.11, definition of PD in TSCAHDE

The extents of the length parallel deck is considered as the longitudinal distance between the last vertices on the forward and aft flat curves, FSF and FSA respectively, (Figure 25.11). If the last vertex on the transom has the same transverse location as the last vertex on the midship section curve, the aft extent of the parallel deck is taken to be the longitudinal position of the last vertex in the control polygon of the transom curve and consequently, the PDA parameter is not defined. The extents of the parallel deck are calculated as PD = PDF – PDA.

25.8.2. Transformation

The change in the extents of the parallel deck (? PD) is calculated as the difference between the present extents ($PD_{current}$) and the new extents as desired by the designer, (PD_{new}).

$$\Delta PD = PD_{new} - PD_{current}$$

As PD is defined from PDF and PDA, these parameters are modified on the basis of the ratio of the relative distances between the extents of the parallel deck and the longitudinal location of the midship section, ($X_{MIDSECTION}$), to account for differences in the longitudinal location of these curves from the midship section. Note that C_{PDA} will calculate as a negative value.

$$C_{PDF} = \frac{PDF - X_{MIDSECTION}}{PD_{current}}$$
$$C_{PDA} = \frac{PDA - X_{MIDSECTION}}{PD_{current}}$$
$$PDF = PDF + C_{PDF} \times \Delta PD$$
$$PDA = PDA + C_{PDA} \times \Delta PD$$

However, if PDA is not defined, PDF is modified directly:

$$PDF = PDF + \Delta PD$$

25.9. PDF (Forward Extent of Parallel Deck)

25.9.1. Definition

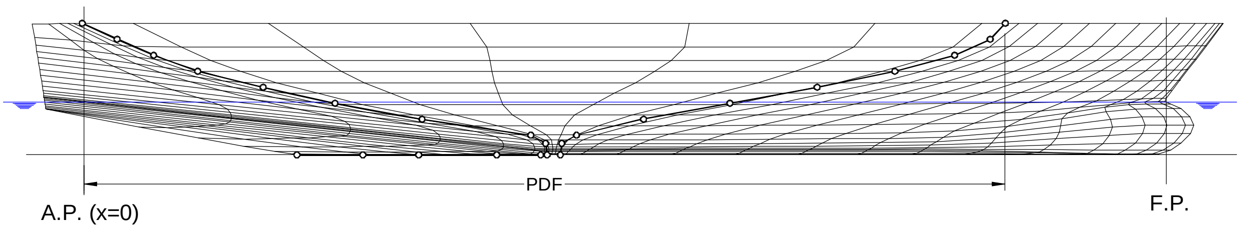


Figure 25.12, definition of PDF in TSCAHDE

The forward location of the parallel deck is defined as the longitudinal position of the last vertex on the control polygon of the forward flat curve, (Figure 25.12).

25.9.2. Transformation

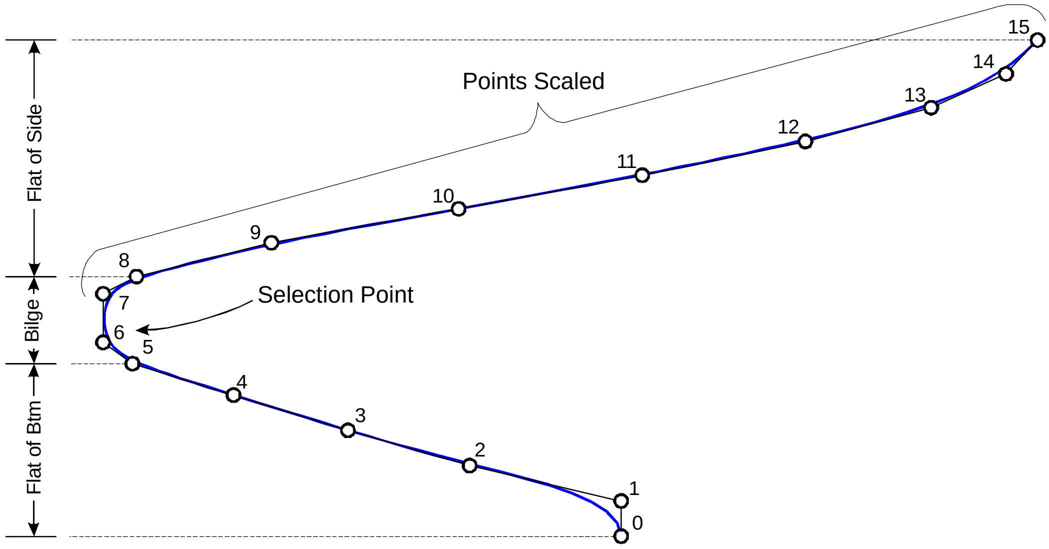


Figure 25.13, to change the forward extent of the parallel deck, the last vertex (15) is to the new location and the remaining vertices are scaled on the basis of the ratio in z.

To change the forward extent of the parallel deck, the last vertex is moved to the desired location, (Figure 25.13). The remaining vertices between the deck and the bilge radius are scaled according to the ratio in depth from a selected vertex on the bilge radius definition. This vertex is selected on the basis of minimum longitudinal location. This operation is only applied to the first curve forward of the midship section curve. The operation is implemented as follows:

```
dPDF = PDF_new - PDF_current

// select the control vertex index with the smallest x
x = ABigNumber (10e32)
I_select = -1
for i = P_numpoints-1 downto 0
    if P[i].x < x then begin
       x = P[i].x
       I_select = i
       end

// scale control vertices after the “pivot” point only
for i = I_select+1 to P_numpoints-1
    C_scale = (P[i].z – P[I_select].z) / (P[P_numpoints-1].Z - P[I_select].z)
    P[i].x = P[i].x + dPDF × C_scale
```

25.10. PDA (Aft Extent of Parallel Deck)

25.10.1. Definition

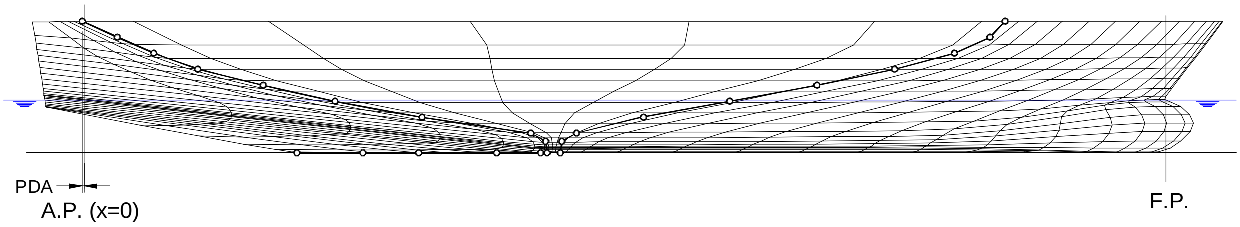


Figure 25.14, definition of PDA in TSCAHDE.

The aft location of the parallel deck is defined as the longitudinal position of the last vertex on the control polygon of the aft flat curve, (Figure 25.14). The PDA parameter is not defined if the last vertex on the transom has the same transverse location as the last vertex on the midship section curve.

25.10.2. Transformation

Changes in the aft extent of the parallel deck are implemented using exactly the same approach as for changing PDF. The exception are that the point selection is made on the basis of the maximum longitudinal extent and only the first curve aft of the midship section is modified by this procedure.

25.11. DISP (Displacement)

25.11.1. Definition

Displacement is found from the volume of the under water body at draught T.

25.11.2. Transformation

Modifications to the DISP parameter result in changes to the hull form surface using the procedure developed in Chapter 14.

25.12. CB (Block Coefficient)

25.12.1. Definition

Block Coefficient (CB) is defined from other the parameters:

$$CB = \frac{DISP}{DWL \times BWL \times T \times r}$$

25.12.2. Transformation

Hull form transformations as a result of changes in the Block Coefficient are implemented through DISP:

$$DISP = (DWL \times BWL \times T \times r) \times CB$$

25.13. CP (Prismatic Coefficient)

25.13.1. Definition

Prismatic Coefficient (CP) is defined from other the parameters and the midship section coefficient c_m resulting from the hydrostatic calculations:

$$CP = \frac{DISP}{DWL \times c_m \times r}$$

25.13.2. Transformation

Hull form transformations as a result of changes in the Prismatic Coefficient are implemented through DISP:

$$DISP = (DWL \times c_m \times r) \times CP$$

25.14. LCB (Longitudinal Centre of Buoyancy)

25.14.1. Definition

The longitudinal centre of buoyancy (LCB) is found from the volume of the under water body at draught T.

25.14.2. Transformation

Modifications to the LCB parameter result in changes to the hull form surface using the procedure developed in Chapter 14.

26. APPENDIX 5 – VARIOUS FUNCTIONS AND PROCEDURES

26.1. Floating Point Tolerance and Other Global Constant Definitions

While humans have a very good understanding of the accuracy of numbers, the accuracy of numbers within the floating point processor of a computer are not necessarily as accurate as would be expected, particularly in the identification of zero. The problem is not a result of inaccuracies in calculation, but usually in the conversion from another representation. The conversion of a floating point number represented by string, or an integer into a floating point number does not always create a number of the desired accuracy. For example, a conversion of zero represented as a string to floating point results in a value that is usually within $\pm 1\times10^{-8}$. Consequently, checking that something is zero using the "= 0" is no longer possible. The accuracy of the floating point processor can be adjusted. However, graphical libraries such as OpenGL and DirectX require diminished floating point accuracy and regularly crash if changed.

To overcome the problem of testing for zero or floating point equality, a tolerance is defined as a global constant throughout the software (Float_Tolerance). Consequently, the checking for zero and floating point equality is implemented as follows:

```
Function IsZero(ANumber) : Boolean
Begin
     Result = Abs(ANumber) < Float_Tolerance
End

Function IsEqual(ANumber1, ANumber2) : Boolean
Begin
     Result = Abs(ANumber1 – ANumber2) < Float_Tolerance
End
```

For the purposes of general use within PolyCAD the floating point tolerance is set to value of 0.0001, representing a tenth of a millimetre. Other global constants defined throughout the software represent large and small values. These are usually used in search procedures.

```
Const
     Float_Tolerance = 0.0001
     ABigNumber      = 1×10^33
     ASmallNumber    = 1×10^-30
```

26.2. Hydrostatics

The calculation of hydrostatics in TSCAHDE is an important task in the development of the tool. When changing the form using a parameter that is calculated from the hydrostatics, an iterative procedure is used to change the shape of the form on the basis of desired hydrostatic properties. Consequently, the speed at which these iterations can be accomplished is dependant on the efficiency of the procedure used to calculate the hydrostatic information from the surface representation. As it is desirable to for the tool to function interactively, the iteration procedure and hence hydrostatic calculation must execute rapidly.

The calculation of hydrostatics is considered, today, a reasonably trivial task in comparison to the other much more complex analysis such as CFD. Naval architects have been implementing numerical integration techniques since it became necessary to calculate the displacement from accurate curves. While the integration processes are well known, it is possible to significantly optimise the calculation procedure. Furthermore, rather than relying on standard techniques, such as Simpson's rules, it is possible to take advantage of more advanced approaches, particularly when the entity on which the calculations are to be performed is known to be an accurate representation.

Sanderski's [32] hull generation technique implemented integration techniques that calculate on the surface representation directly. As the surface only represents the underwater body, the implementation of an analytical technique is easier. However, the surface generated by TSCAHDE also represents the topsides of the vessel. Consequently, as a significant proportion is not below the waterline, implementation of an analytical technique, considering the fact that draught is a user controlled parameter, is more complex. Calculation of the hydrostatic properties using numerical techniques still appears to be more practical.

Traditionally, hydrostatic calculations are conducted using transverse section representations. These are well suited to task, particularly if the calculation is being performed manually. Simpson's rules or the other multiplier techniques are considered to be the best technique for integrating the area. However, unlike humans, software programs find it more difficult to calculate accurately if the arrangement of the shape being integrated changes. In the multiplier techniques, this occurs if there is a discontinuity in the shape. The calculation must be adjusted so that an ordinate lies on the discontinuity and for the computer this means additional calculation. Considering the accuracy of entities represented in modern CAD systems, there are more computational techniques that can be use by software much more effectively than if implemented manually. An example of this is Green's Theorem which is used to calculate the area and centroid of a closed, non-overlapping polygon using the concept of overlapping triangles. Depending on the winding orientation of the triangle, area is added or removed from the sum total. Green's Theorem implements this process by considering the vector form of calculating the area of a triangle. The calculation for the area and centroids of a closed planar polygon defined in the *x*-*y* plane is as follows:

$$A = \frac{1}{2}\sum_{i=0}^{n-1} a_i$$

where:

$$a_i = x_i y_{i+1} - x_{i+1} y_i$$

The centroid of the shape $(\bar{x}, \bar{y})$ is found as follows:

$$\bar{x} = \frac{\iint_R x dx dy}{A} = \frac{m_x}{A}$$
$$\bar{y} = \frac{\iint_R y dx dy}{A} = \frac{m_y}{A}$$

where:

$$m_x = \frac{1}{6}\sum_{i=0}^{n-1}(x_{i+1} + x_i)a_i$$
$$m_y = \frac{1}{6}\sum_{i=0}^{n-1}(y_{i+1} + y_i)a_i$$

(source: http://www.efg2.com/Lab/Graphics/PolygonArea.htm)

For an accurately generated section shape, from a surface for example, this approach to the calculation of area is not affected by discontinuities in the shape. Furthermore, this method is significantly easier and smaller to implement and executes more rapidly than the multiplier approach.

Improved techniques are available to calculate hydrostatics from sections over traditional manual numerical integration procedures. However, all these approaches assume that sections can be obtained from the surface.

Contouring NURBS surfaces, on the basis of the availability of research into different procedures and the fact that CAD vendors are not happy to indicate how their systems calculate these curves, is not a trivial process. Since NURBS were first implemented in HullCAD, various contour generating techniques have been implemented to find the fastest and most appropriate method. The contouring procedure was first implemented using a surface tracing approach. However, this was heavily dependent on the arrangement of the surface in space and it had no methods for finding loops. As the tracing procedure was not very flexible, a more generic technique was selected. This converts the surface into polygons and analyses each to identify the plane intersections. A similar approach is detailed in BYTE [52]. It is a very effective technique as it functions on any boundary representation, not only NURBS. Furthermore, it requires no additional procedures to identify loops. The only disadvantage is that the quality of the contours is dependant on the number of polygons generated from the surface. For NURBS surfaces defined with distorted control polygons, it can be difficult to get good quality contours without using a significantly large number of generated polygons.

The development of sections using a contouring approach that constructs the curves from the geometrical analysis of polygons is a two stage procedure. First, from the surface representation, (Figure 26.1), the polygon representation of the surface is generated as a grid from the parameter lines, (Figure 26.2). Using the polygon representation, sections can be generated by analysing each polygon, in turn, for planar intersection, (Figure 26.3). Furthermore, the procedure locating the sections functions by checking the accuracy of the volumetric representation. This processing takes a reasonable amount of time. A much more effective route is to make a direct calculation by using the finite elements of the polygon representation. This approach follows an analytical approach to the calculation of the hydrostatics by considering the surface integral of the hull form.

By definition, the displacement of the surface is the integration of the hydrostatic pressure over the underwater surface, S:

$$Displacement = \int_S pds$$

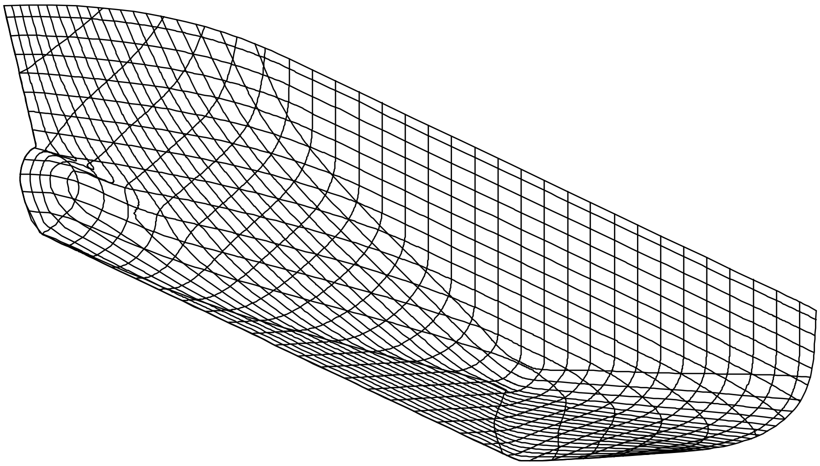


Figure 26.1, surface representation of the Hull Form.

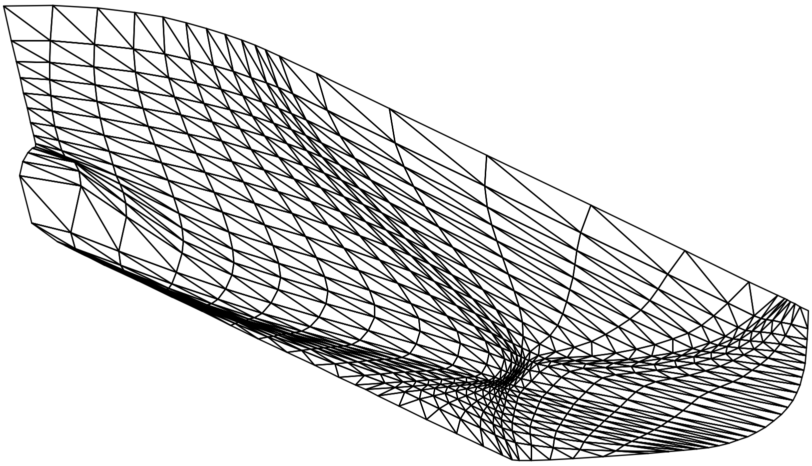


Figure 26.2, Polygon representation defined along the parameter lines.

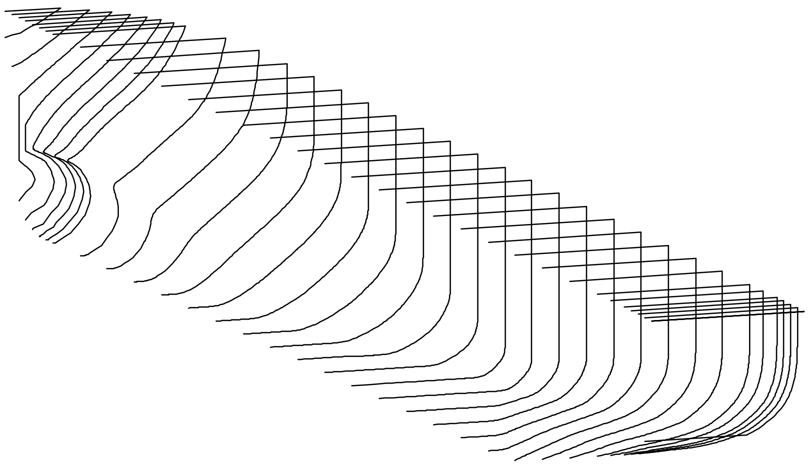


Figure 26.3, calculation sections of the hull form developed from the polygon representation

In finite element terms, this becomes the summation for each polygon, P_i of the product of the pressure, and area in the z direction:

$$Displacement = \sum_{i=0}^{n-1} p_i \cdot a_i \cdot (n_i.z)$$

where p_i is the pressure at the centre of the *i*th polygon

a_i is the area of the *i*th polygon

n_i is the normal vector of the *i*th polygon (pointing into the hull)

z is the z vector [0 0 1]

This approach can be used to calculate the entire range of hydrostatic components, such as displacement, volume and wetted surface area:

$$Displacement = 2rg\sum_{i=0}^{n-1}(-cz_i)\cdot a_i \cdot (n_i.z)$$

$$Volume = 2\sum_{i=0}^{n-1}(-cz_i)\cdot a_i \cdot (n_i.z)$$

$$WettedSurfaceArea = 2\sum_{i=0}^{n-1} a_i$$

Where cz_i is the z coordinate centre of the *i*th polygon, similarly for cx_i and cy_i

Centre of Buoyancy:

$$LCB = \frac{2\sum_{i=0}^{n-1} cx_i \cdot (-cz_i)\cdot a_i \cdot (n_i.z)}{Displacement}$$

$$TCB = \frac{2\sum_{i=0}^{n-1} cy_i \cdot (-cz_i)\cdot a_i \cdot (n_i.z)}{Displacement} = 0\,(by\,definition\,of\,the\,symmetry)$$

$$VCB = \frac{\sum_{i=0}^{n-1} cz_i \cdot (-cz_i)\cdot a_i \cdot (n_i.z)}{Displacement}$$

Waterplane Area and associated quantities:

$$WaterplaneArea = 2\sum_{i=0}^{n-1} a_i \cdot (n_i.z)$$

$$LCF = \frac{2\sum_{i=0}^{n-1} cx_i \cdot a_i \cdot (n_i.z)}{WaterPlaneArea}$$

$$2ndwaterplaneAreaMmtX = \sum_{i=0}^{n-1}(cx_i)^2 \cdot a_i \cdot (n_i.z)$$

$$2ndwaterplaneAreaMmtY = \sum_{i=0}^{n-1}(cy_i)^2 \cdot a_i \cdot (n_i.z)$$

To calculate the hydrostatics using the polygon representation, a densely populated surface is first generated, (Figure 26.4). This surface is then clipped to the waterline to remove all topside polygons and translated so that the water plane is represented at z = 0. Using this representation alone, the hydrostatics can be calculated by considering the area and arrangement of each polygon as detailed above.

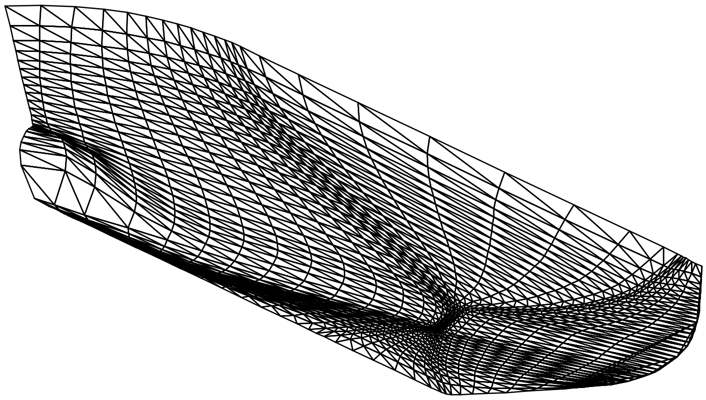


Figure 26.4, a dense polygon representation is generated for hydrostatic purposes.

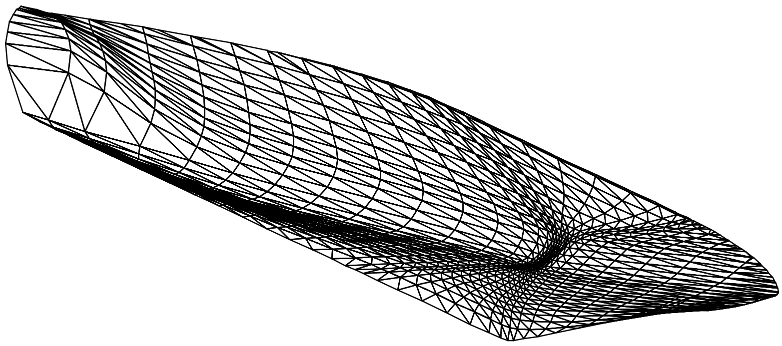
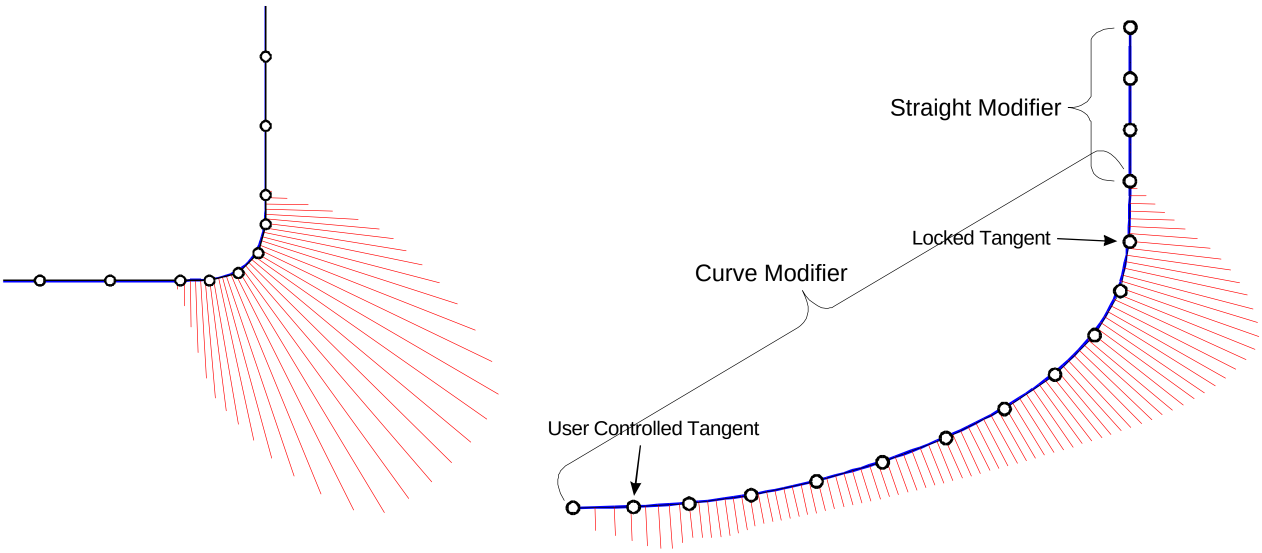


Figure 26.5, the calculation is preformed after the representation is clipped to the waterline.

26.3. Implementation of the Curve Modifier

There are several uses of the curve modifier constraint. It can be used to develop a blended curve shape between one or two straight modifiers, (Figure 26.6a), or it can be used to implement an arc shape over a number of control polygon points, (Figure 26.6b), with the knowledge that the shape produced by the arrangement is fair.



a) b)

Figure 26.6, the curve modifier, a) creating a blend between straight segments and b) creating a smooth arc shape over a number of control polygon vertices.

The approach to develop the curve modifier is consistent with the idea that if the control polygon is fair then the generated curve will also be fair. In this case, a fair control polygon can be developed by finding a function that can blend the shape of the control polygons over the range using the two end segments as tangents to the curve. If the end vertex is also part of a straight modifier, then the tangent direction is already predefined. Otherwise, the tangent is controlled by the user. Rogers and Adams [37] have some very good examples on the developments of blending curves. In this case, a single cubic spline can suffice.

The approach is to use a single cubic spline section to form a curve which all the internal points of the modified section will lie on, (Figure 26.7). The curve will start and end at the first internal points from the ends of the segment, between M_1 and M_{n-2}. The location of these points will lie on the tangent vector from the end points, M_0M_1 and $M_{n-1}M_{n-2}$. The length of the segments will be such that:

$$\left|M_0M_1\right|,\left|M_{n-1}M_{n-2}\right| = \frac{\sum_{i=1}^{n-3}\left|M_iM_{i+1}\right|}{n-2}$$

with

$$\left|Ts\right| = \left|Te\right| = 0.8\left|M_0M_{n-1}\right|$$

This is achieved through an iterative procedure that changes the distance between M_0M_1 and $M_{n-1}M_{n-2}$ until this is achieved.

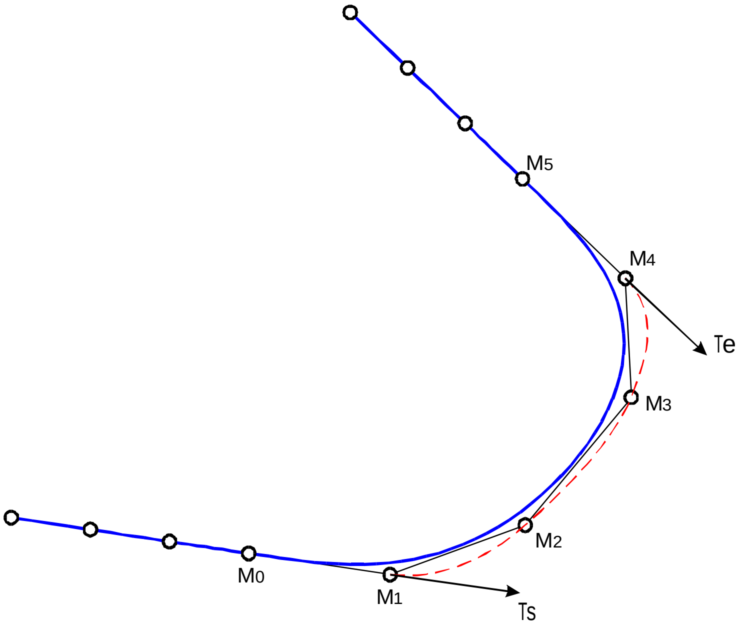


Figure 26.7, the internal points within the modifier range are controlled by a single cubic spline segment.

If only four points make up the curve modifier, a cubic spline curve cannot be formed. This is the usual arrangement that can be used to develop the bilge radius on the midship section. Consequently, the tangent lengths are specified based fixed values which were found to produce the closest match to a circular arc, (Figure 26.8).

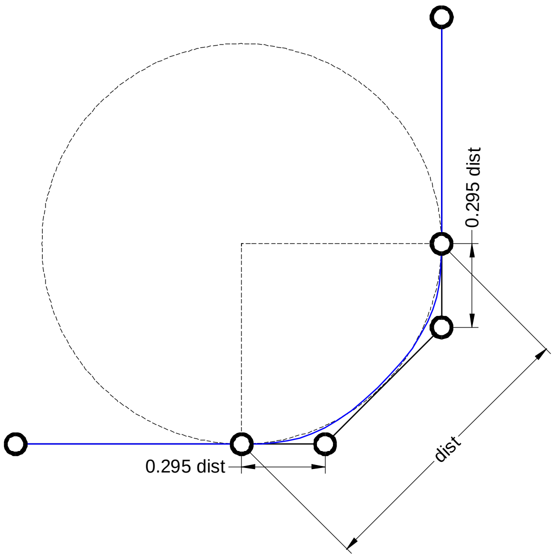


Figure 26.8, if only four points make up the modifier, tangent lengths are specified based on values that develop the closest match to a circle for the quadrant arrangement shown.

26.4. Selecting the Control Vertex to Transform the Beam (BWL) and Depth (D) about

26.4.1. Identifying the Shape of a Ship Sectioned Shaped Form Curve

Designing a transformation to vary the shape of the midship section to minimise distortion to a standard ship type section is not conceptually difficult. The transformation is based around a choice of which control vertices to scale and which to translate. While the transformation calculations are trivial, the selection of a control vertex which subdivides the points that will under go different transformations is more difficult.

If the problem of identifying this point is approached from a totally conceptual view, the problem is one that involves recognising the shape as a ship section and then identifying the extent of the bilge radius. Technical disciplines for performing tasks such as shape identification and matching do exist. However, a search of the material on the Internet in this area revealed that the technology is quite complex and intricate. The use of these approaches to identify the shape of the midship section would require more development and technology than necessary to implement the remaining project. The use of specialist technology such as shape identification approach is impractical for this minor part of the technique.

As the particular technology to identify shapes was not going to applicable, an alternative and considerable more practical solution was required. If a heuristic approach was taken to developing a practical solution, the function should not require a great deal of processing. While the shape of a form curve can be quite detailed, the representation is directly controlled by the arrangement of the control polygon. Consequently, an approach that analyses the shape of the control polygon would only have to consider a practical maximum of thirty vertices. (There is no limit on the number of control vertices that can be used to define a form curve. However, there has never been a need to use more than twenty to twenty-five vertices to define the transverse shape of a ship hull form). Based on the highly geometric approach taken to develop this hull design technique, the approach of analysing the shape along the length of the curve is one that could be developed into a practical solution. If the curvature at each vertex is calculated, (Figure 26.9), a search based around the identification of the highly curved bilge radius area of rectangular shaped ship sections can be used to find the control vertices which bound the definition of this shape. These vertices are relevant locations for subdividing the scaling and translation transformations.

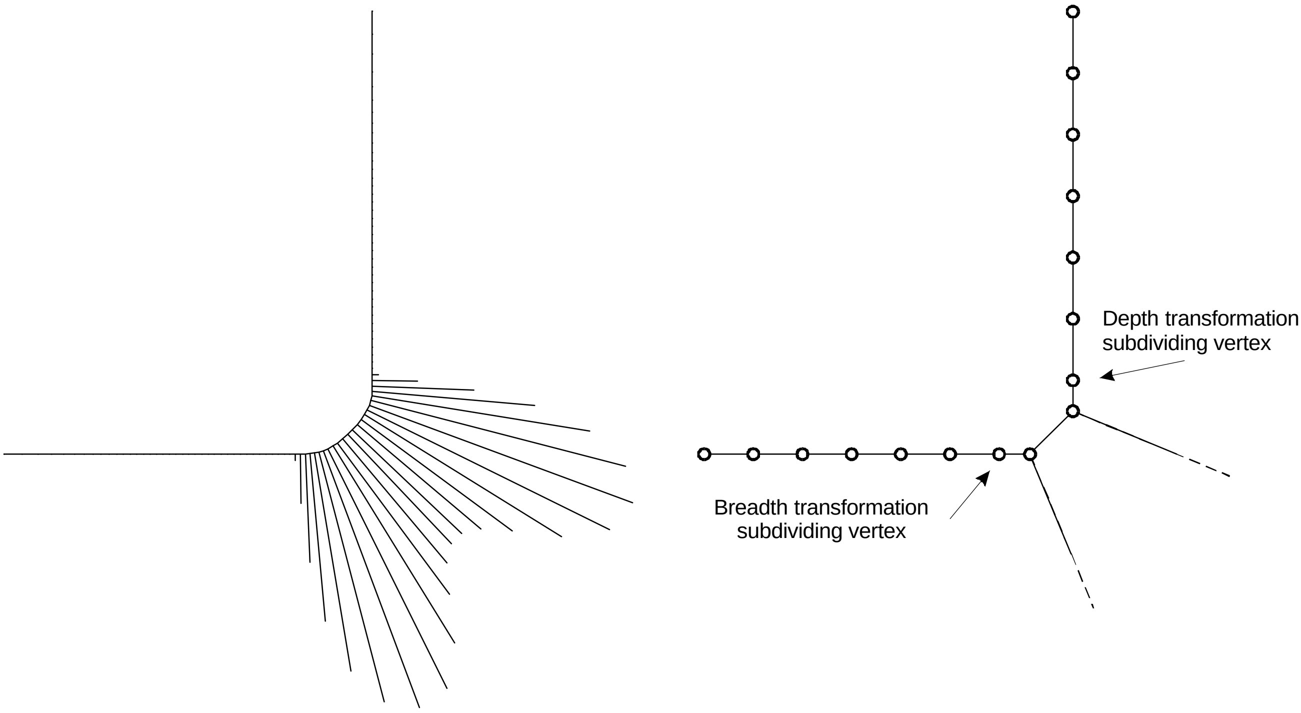


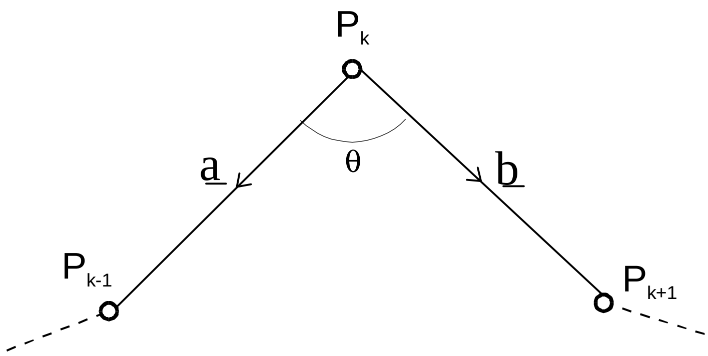
Figure 26.9, analysing the shape of the section using the control vertex representation rather than the curve reduces the number of calculations and makes the problem more concise. The vertices that will subdivide the transformations will be the two before the curvature regions starts, from the property of collinear vertices and the fact that form curves are cubic.

26.4.2. Calculating the Curvature of a Control Polygon

The formal definition of curvature is:

$$K = \left| \frac{dT}{ds} \right|$$

However, when the curve is defined using discrete points, the differential becomes less well defined. A search of techniques for calculating the curvature of discrete curves reveals that there appear to be a variety of methods. It is possible to consider the differences between the values of the line segment vectors each side of a vertex. However, the majority of techniques approach the problem by considering the idea of the radius of curvature.



$$K = \frac{1}{2}\left(1 + \frac{\underline{a}.\underline{b}}{|\underline{a}||\underline{b}|}\right)$$
$$|\underline{a}| = |\underline{b}|$$

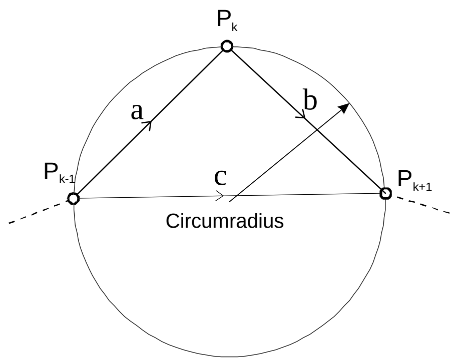
Figure 26.10, the arrangement used by Rosenfeld [62] to calculate the curvature *K* of a digital curve

Rosenfeld [62] developed a technique for use on digital curves, i.e. defined with pixels. The function is based on the cosine rule, (Figure 26.10). The range of *K* produced by this function is between zero and one. Vectors $\underline{a}$ and $\underline{b}$ should be equal lengths for the approach to function effectively. In an implementation of this method, this can be achieved by calculating the vectors to the adjacent points and normalising the vectors.

An alternative and more frequently used approach is to consider the curvature corresponding to the radius of the circle formed through the three adjacent vertices. There are various ways of finding this radius. This operation is frequently used in CAD to define an arc through three points. However, this approach requires additional calculation to identify the centre of the circle. A more efficient approach is to use the Circumradius identity [63]. The Circumradius is the radius of the circle (Circumcircle) which intersects all three points of a triangle. The radius of this circle is defined with respect to the triangle as follows [64]:

$$Circumradius = \frac{Perimeter}{4 \times Area}$$

If this approach is used to calculate the radius of the circle using vectors, the function is as follows:



$$R = \frac{|\underline{a}| + |\underline{b}| + |\underline{c}|}{4(\frac{1}{2}|\underline{a} \times \underline{b}|)}$$

Figure 26.11, the Circumradius of three adjacent vertices of the control polygon

However, the distances between the vertices are a factor in this approach to the analysis. As the angle between the line segments is the feature of interest, the length of vectors $\underline{a}$ and $\underline{b}$ can be normalised. Consequently, the only variable factor is the vector $\underline{a} + \underline{b}$. By simplifying the function, the Circumradius can be calculated as follows:

$$R = \frac{1}{\sqrt{4 - |\underline{c}|^2}}$$

Where: $\underline{c} = \underline{a} + \underline{b}$

As curvature is the inverse of the radius, curvature is analysed as:

$$K = \sqrt{4 - |\underline{c}|^2}$$

When considered with normalised vectors $\underline{a}$ and $\underline{b}$, the range of K lines between zero and two. The two techniques for calculating curvature can be compared by dividing the curvature of the Circumradius approach by two, (Figure 26.12). It can been seen that the Circumradius has a much better resolution for the smaller external angles and this level of resolution (curve gradient) is continued until the curve flattens out at 180°. The curve produced by Rosenfeld's function being tangentially to the *x*-axis results only small changes at the small external angles. Figure 26.13 to Figure 26.20 illustrate a variety of form curve control polygons analysed using the Circumradius function.

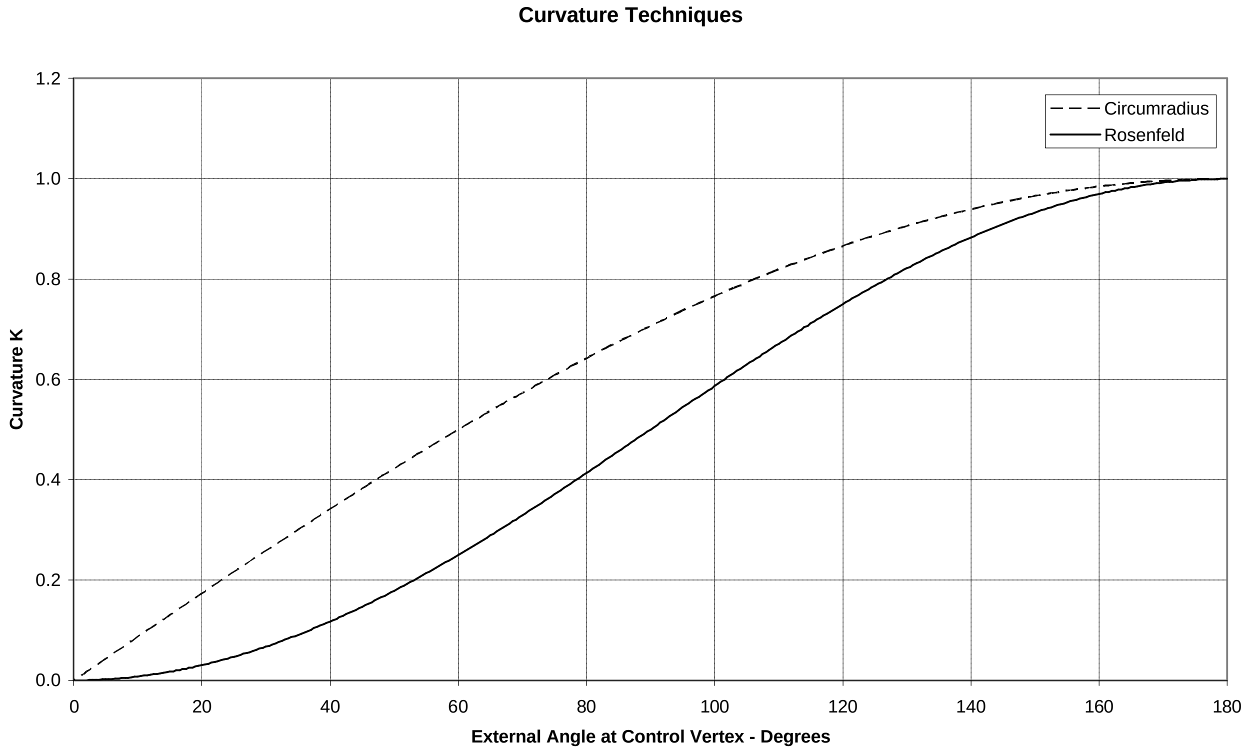


Figure 26.12, a comparison of curvature calculation on a control vertex using normalised line segment vectors. An angle of 0° represents a straight line. When the results of the Circumradius function are scaled to match Rosenfeld's technique, the Circumradius function has much better resolution at small external angles.

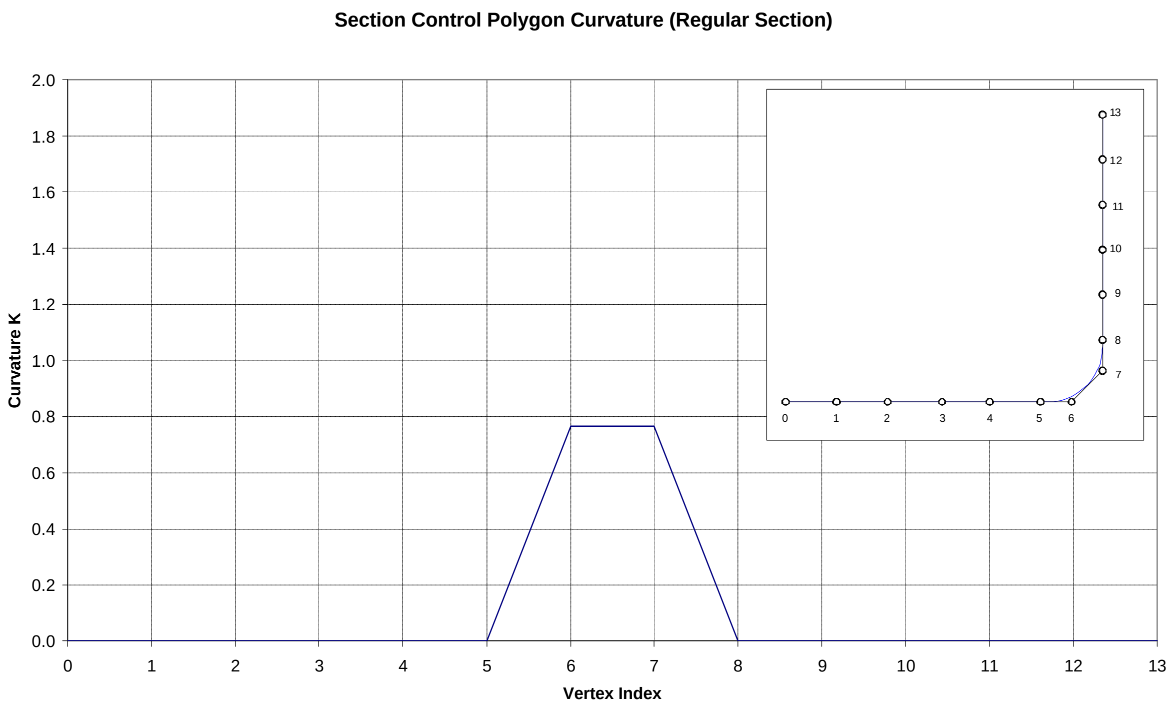


Figure 26.13, curvature for a regular shaped ship section control polygon.

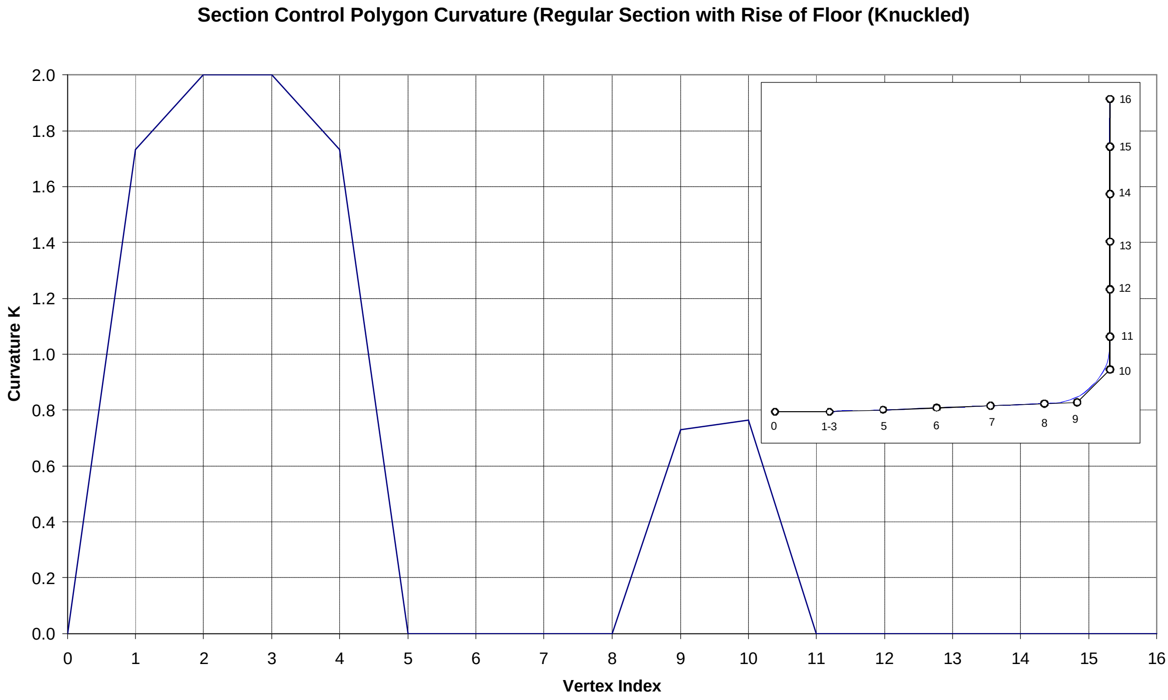


Figure 26.14, curvature for a regular shaped ship section control polygon with knuckled rise of floor.

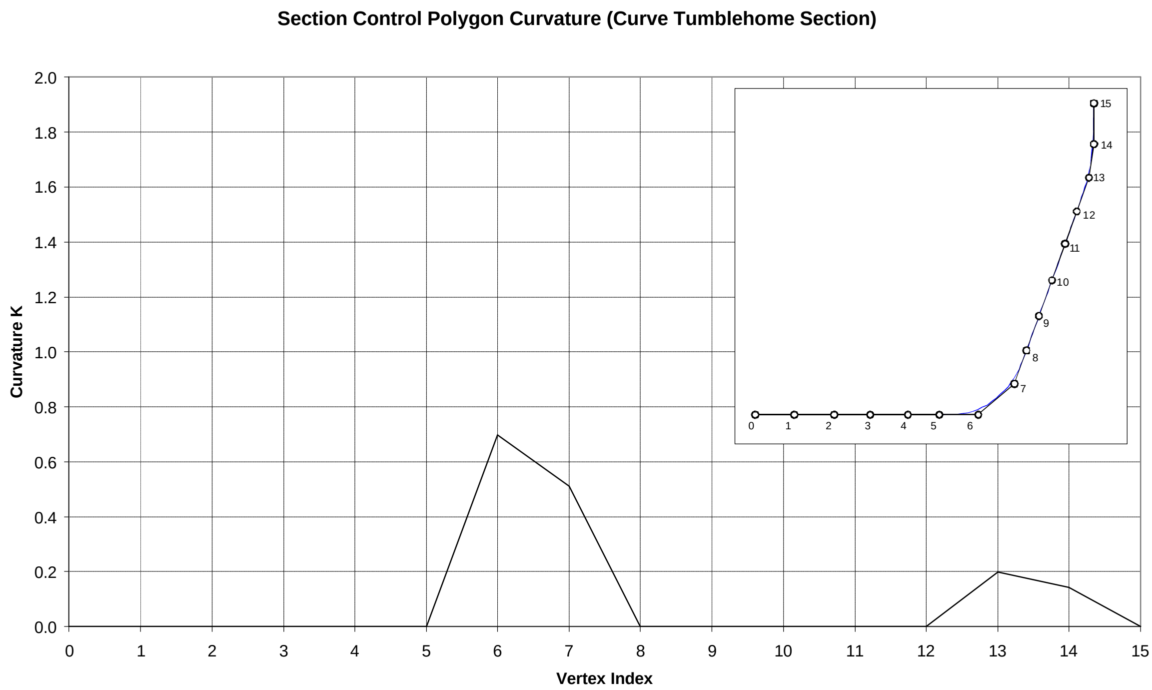


Figure 26.15, curvature for a regular shaped ship section control polygon with curved tumblehome.

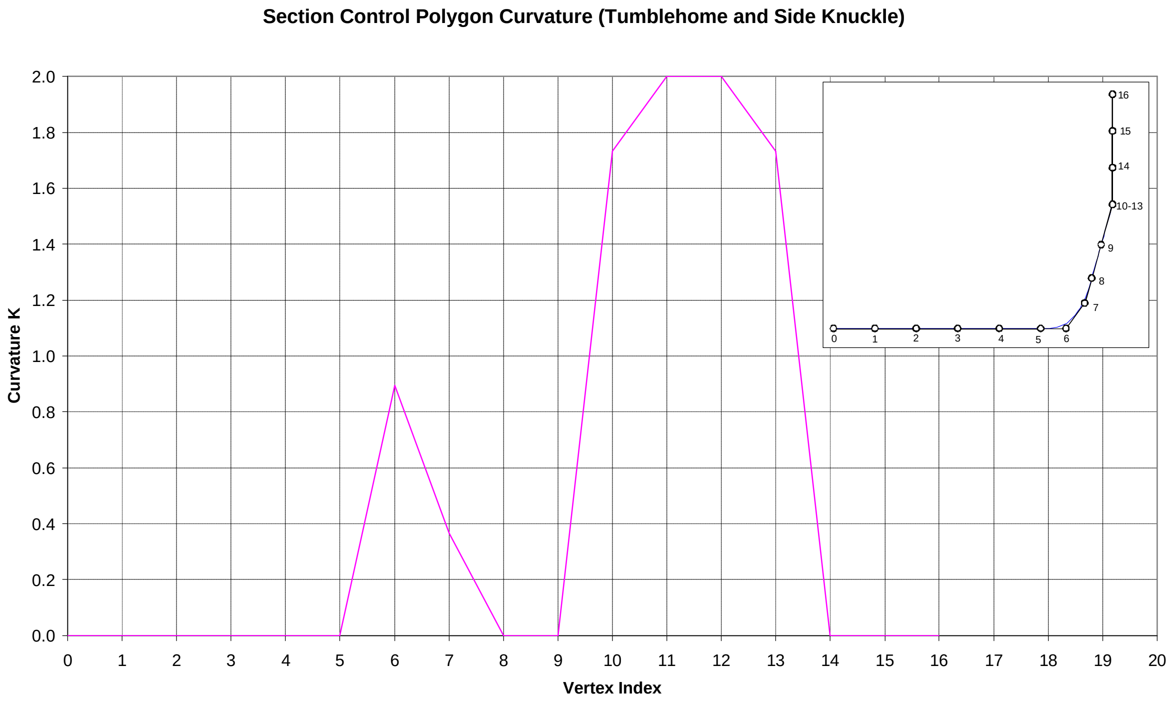


Figure 26.16, curvature for a regular shaped ship section control polygon with knuckled tumblehome.

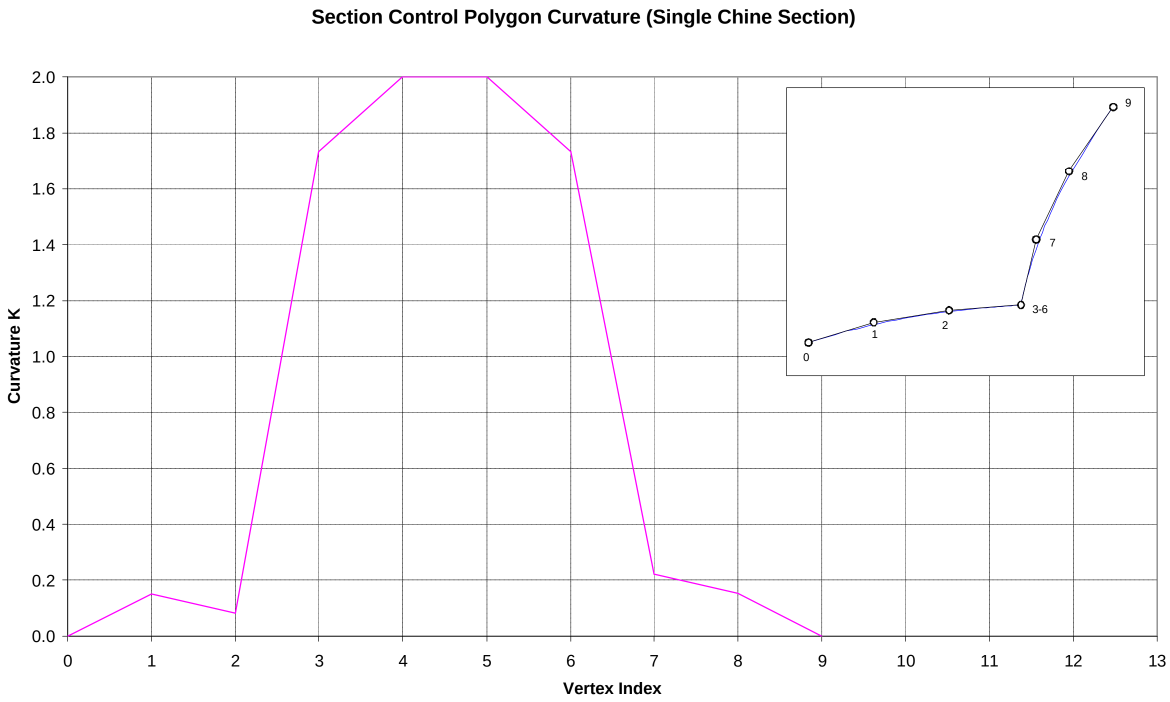


Figure 26.17, curvature for a single chine section control polygon.

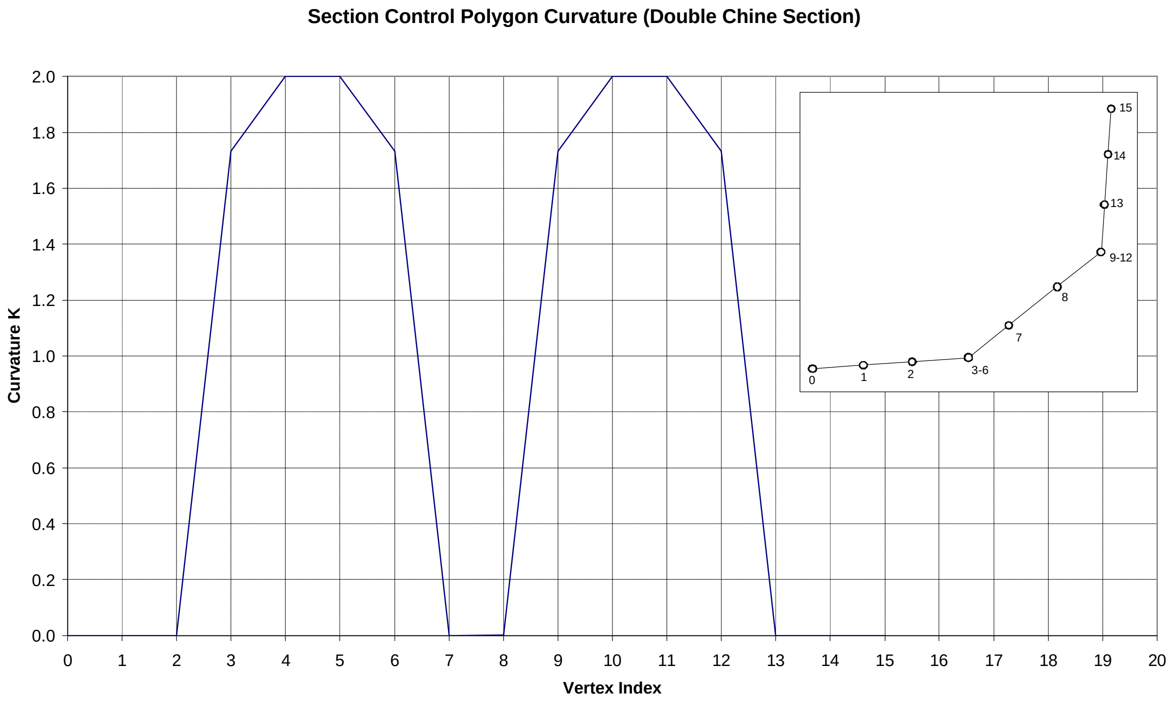


Figure 26.18, curvature for a double chine section control polygon.

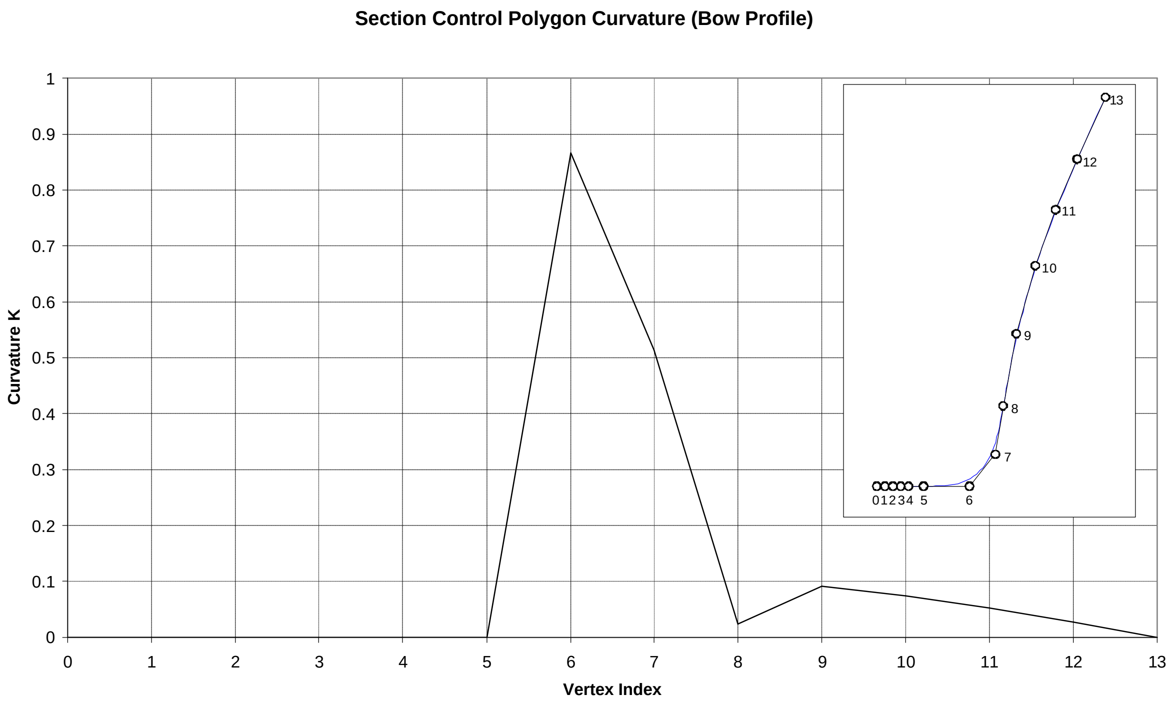


Figure 26.19, curvature for a bow profile control polygon.

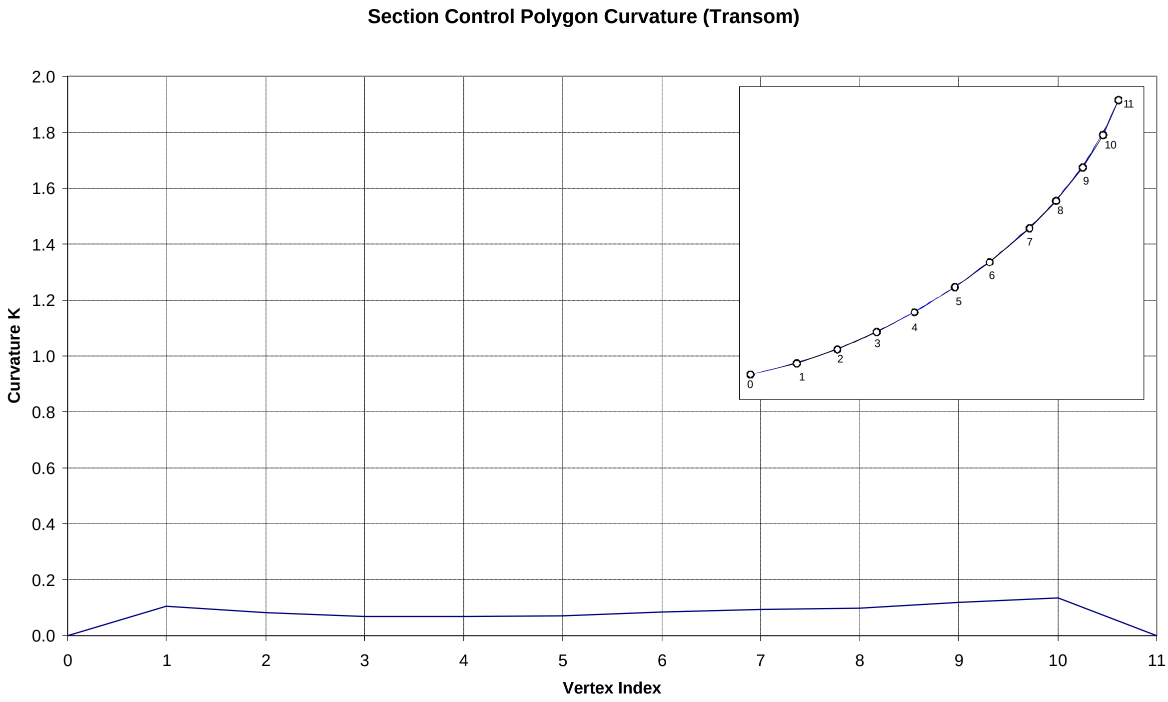


Figure 26.20, curvature for a transom section control polygon.

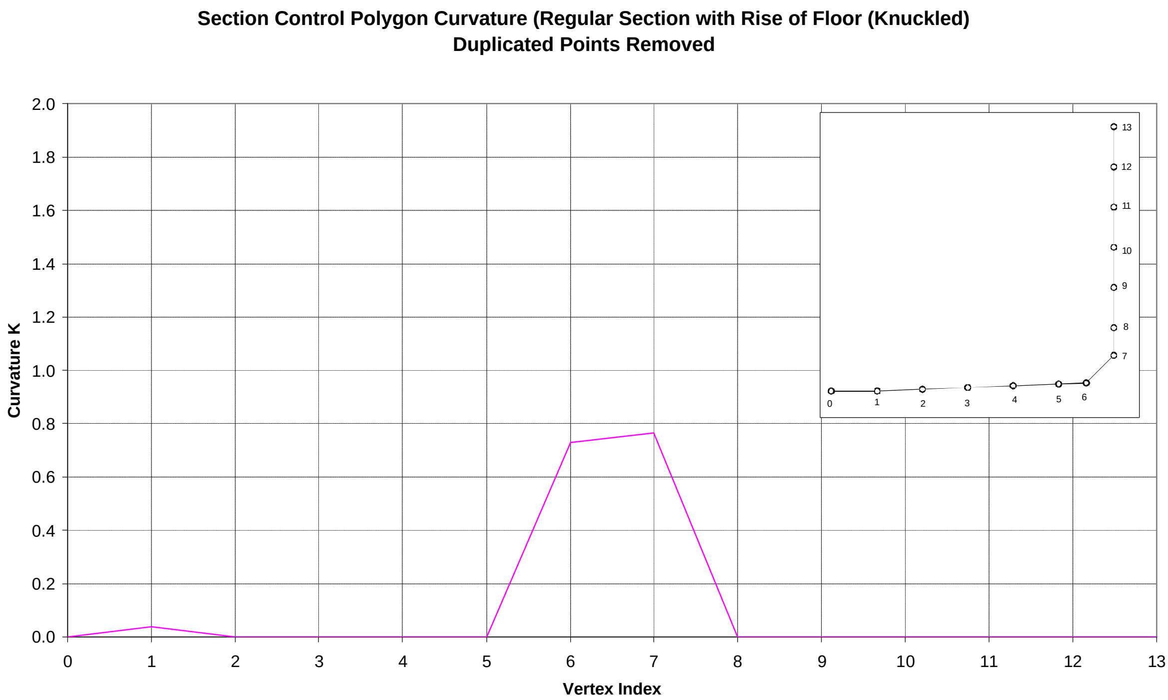


Figure 26.21, the curvature with duplicate points removed corresponding to the section with rise of floor, (Figure 26.14).

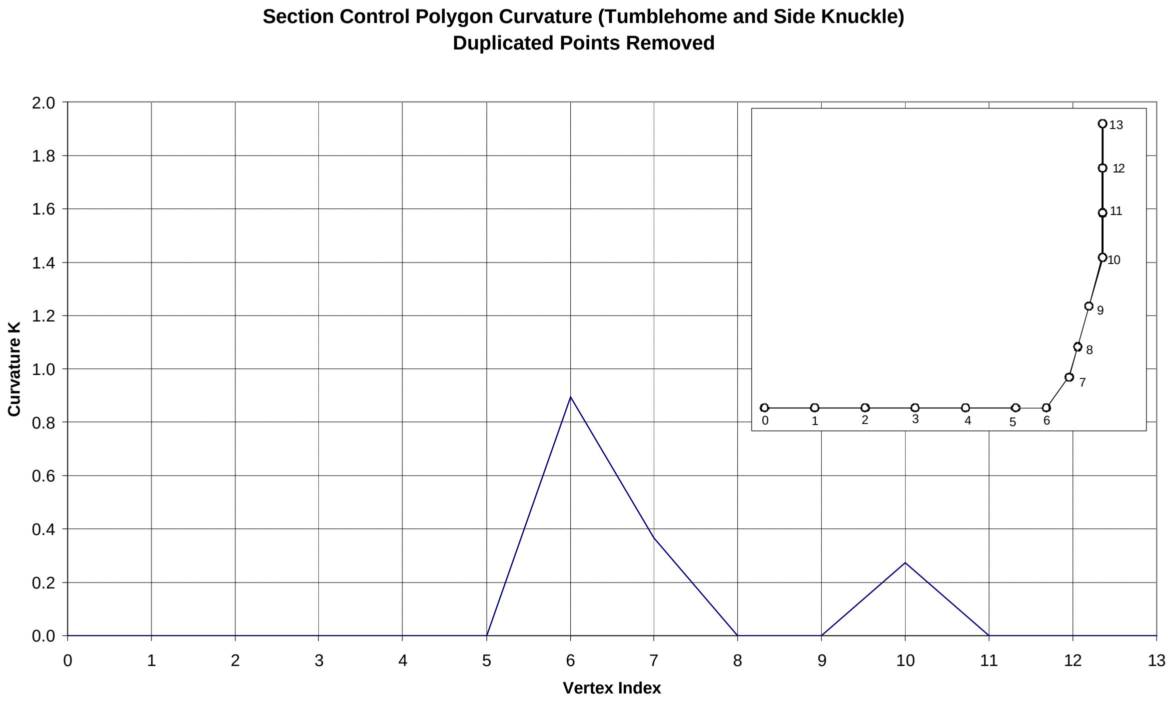


Figure 26.22, the curvature with duplicate points removed corresponding to the section with tumblehome and a side knuckle, (Figure 26.16).

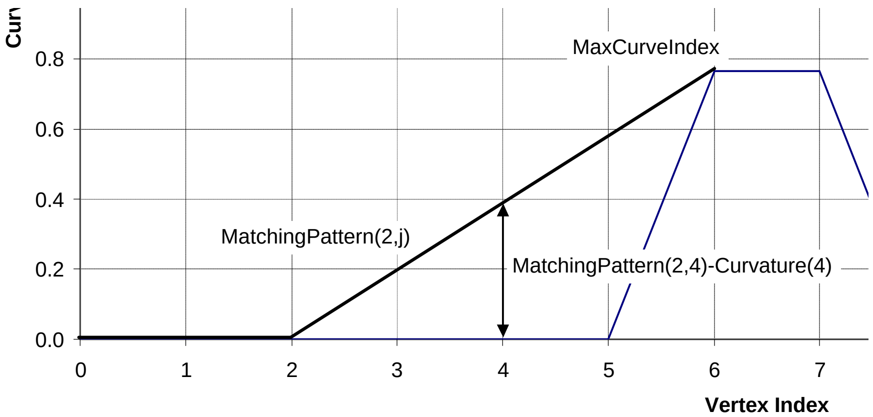
26.4.3. Identifying the transformation subdivision vertex

Once a function for calculating the curvature at each vertex has been selected, a search technique is required to identify the control vertex which will subdivide the scaling and translation transformations. By using NURBS properties, the vertices that will be used to subdivide the different transforms are the two vertices each side of the highly curved region, (Figure 26.9). This is selection is based on the property of collinear vertices that is present between the linear and curved shapes and the fact that form curves are defined using cubic B-Spline curves.

The curvature analysis shows that ship sections have a particular shape characteristic. The straight segments result in very low curvature increasing to a peak around the bilge radius. However, while the peak is recognisable on all ship shaped sections, control polygons with knuckle point definitions (duplicate control vertices), create locations with maximum curvature, which is higher than the area at the bilge radius, (Figure 26.14 and Figure 26.16). These more prominent regions of curvature increase the complexity of the search and a selection based on curvature magnitude alone cannot be used directly on this data. The knuckle point definitions are constructs applied to affect the shape of the curve. They do not affect the shape of the control polygon. Consequently, the duplicated points can be removed reducing the magnitude of the local curvature and enabling simplification in the search algorithm. Furthermore, when the duplicate points are removed, the characteristic ship shape curvature distribution becomes more recognisable on sections with knuckle points, (Figure 26.21 and Figure 26.22).

Now that there is a pattern of characteristic shapes, a search procedure can be developed to identify the control vertices at the foot of the central peak of curvature. A variety of means can be used to identify this point. It would be quite easy to develop a geometric procedure to search for the right-bottom most vertex on the left side of the peak to identify the beam subdivision point, similarly the left-bottom most vertex on the other side for the depth subdivision point. However, in practical situations where data provided by the user is to be analysed, it is always necessary to consider the data with some suspicion. Search techniques employing assumptions on the geometric distribution of the data have the least resilience to errors, as the geometry has a large number of degrees of freedom. Consequently, due to these limitations, a purely geometric search approach was rejected.

As the distribution of curvature shows a very characteristic shape, a simple pattern matching approach could be used to identify the subdivision points. If the identification of the subdivision points is considered separately, the characteristic shape for the beam is a two piece linear function, shallow in the first section and then steep to the peak. For the depth, the reverse approach is the case. An "ideal" pattern can be created by fitting this function to the maximum point, the end point and the centre point to a vertex *i*. By analysing the least squares difference between the ideal function and the actual data over all the possible vertex locations between, the best match will identify the correct subdivision vertex *i* producing the minimum error. A further adaptation is to consider situations where the actual curvature is greater than the ideal curvature, i.e. when there are local knuckle point locations, as perfect matches to prevent any bias due to local features.



For Beam: $$Error(i) = \sum_{j=1}^{MaxCurveIndex-1} \left\langle MatchingDistribution(i,j) - Curvature(j) \right\rangle^2$$

For Depth: $$Error(i) = \sum_{j=MaxCurveIndex+1}^{n-1} \left\langle MatchingDistribution(i,j) - Curvature(j) \right\rangle^2$$

Where

- MaxCurveIndex is the index of the vertex with maximum curvature
- Curvature(j) is the curvature calculated at control vertex index j
- MatchingDistribution(i,j) return the a value for curvature based on a function through the end index (0 or n), vertex i and the vertex MaxCurveIndex
- <x> return x if x > 0 else 0

Figure 26.23, the subdivision index is found by searching for a MatchingDistribution function with the minimum error to the calculated curvature.

This approach should be able to deal with ship section shaped control polygons. However, when the section does not have this characteristic shape, such as the transom in Figure 26.20, the procedure cannot return an appropriate index. In these cases, the mean value of curvature is compared with the maximum value to see if the section predominantly consists of straight segments. To identify a ship section, the following threshold is used based on checks with a variety of section shapes:

$$\frac{AverageCurvature}{MaxmumCurvature} > 0.25$$

A further practical optimisation can be made to the procedure in that if the maximum point of curvature is a knuckle point, then as this consists of duplicate points, the location can be considered as both the beam and depth transformation subdividing point.

26.5. Computing the Offset Representation of Polyline (Control Polygon)

A detailed study was performed in Chapter 15, to identify the best approach to generating the tangent definitions for the blending curves from the shape of the flat curves. The study found that the best results were produced by a tangent curve positioned a uniform distance, normal to the flat curve. The tangent curve is uniformly offset from the flat curve. The offset operation is frequently used in CAD systems to create a curve a certain distance away from another. The development of the tangent curve using an offset procedure involves additional complications as both flat and tangent curves are non-planar. Curve offsetting procedures mainly function within a plane unless additional information can be provided. As the definition curves are three dimensional, there were some initial problems until it became clear that it was possible to re-parameterise the curve definitions by un-wrapping the curves off the transverse shape of the parallel middle body onto a plane. As a result, the task of calculating the offset shape becomes significantly easier.

A search of the literature regarding the calculation of representations offset from another reveals a great wealth of information. Development of procedures based on NURBS curves and surfaces [65],[66] feature highly due to present level of use of these representations throughout CAD tools. However, in the case of flat tangent curves, the offset procedure need only consider the shape of the control polygon and not the curve representation itself, as the parameters affecting curve shape are fixed for all form curves. The control polygon can be considered to be a Polyline, a representation which consists of vertices connected by linear segments. This representation is used most frequently particularly in the more generic CAD tools such as AutoCAD. In fact, the offset procedure in AutoCAD is quite advanced, being able to consider situations when the result of the operation produces separate entities. Such an advanced procedure is not required to offset the flat definition curves and it should be noted that transverse location of the vertices are constrained. Consequently, the variety of shapes that can be produced is limited. It is only necessary to develop a procedure that can handle at least a case when an offset result self intersects. Most techniques for offsetting Polyline curves are developed for CAD. Consequently, the quality of the result produced by the operation is required to be of a high level. The technique developed by Choi and Park [67] creates very neat radius around vertices reducing the amount of protuberance for sharp points. As the result of the offset calculation of the flat definition curve is an intermediate representation used to find the location of the vertices extended longitudinally from the flat curve itself, the quality of the shape does not have to be very high. A basic procedure can be developed by considering vectors and their intersections, (Figure 26.24).

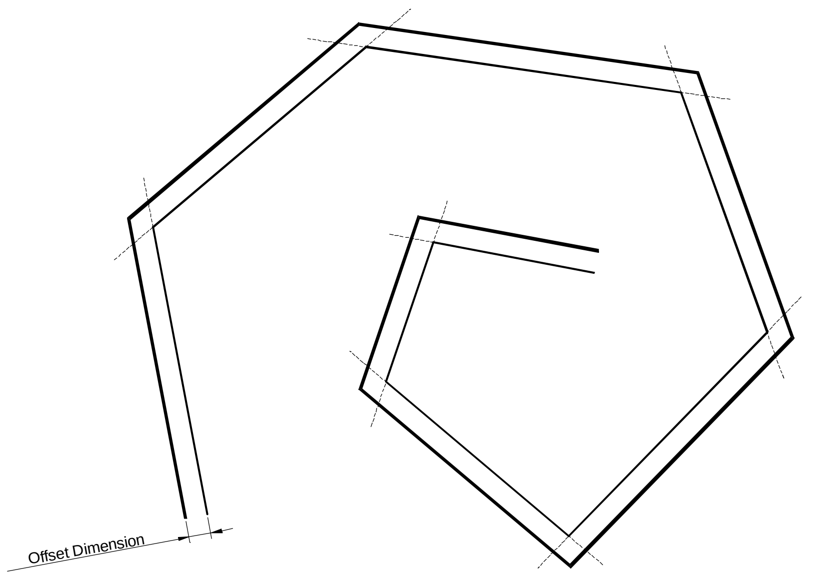


Figure 26.24, an offsetting procedure can be developed by considering the intersections of vector lines offset from the original representation by the relevant distance.

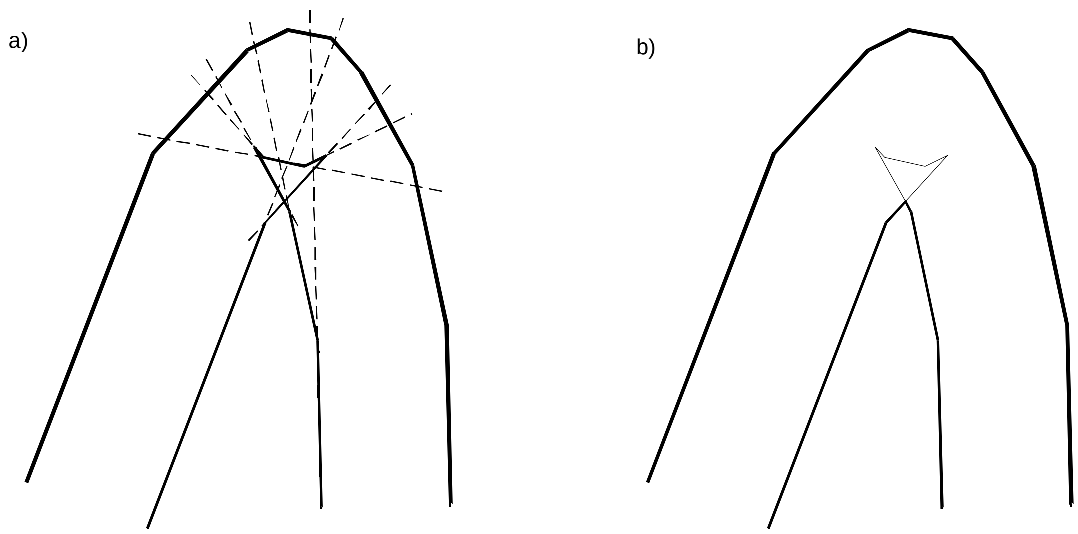


Figure 26.25, self intersections due to well populated sharply curved areas, a), can be dealt with by trimming off the additional loop, b).

The only additional complication that must be considered when using this approach is that areas that are sharply curved and have many segments cause the offset curve result to self-intersect and create a loop in the reverse direction, Figure 26.25a. As the loop is additional to the shape of the offset curve, it can be trimmed off at the location when the offset curve intersects with itself, Figure 26.25b.

27. APPENDIX 6 – RAPID INTERACTIVE DEVELOPMENT OF A HULL FORM

Quick, accurate and practical development of an initial hull surface is one of the primary problems with present hull development tools that this project has tried to address. The following screen shots, (Figure 27.1 to Figure 27.24), show the process of a hull form being defined. The hull is developed using the following specification of form characteristics:

1. LBP = 120.0m
2. BWL = 26.0m
3. Depth = 12.0m
4. Draught = 5.7m
5. Block Coefficient C_B = 0.60
6. LCB = 57.0m, to be coincidental with the midship section definition curve
7. Bilge Radius = 1.8m
8. No parallel middle body, PMB = 0.
9. Deck runs parallel to the transom. The forward extent of parallel deck, (PDF), starts at 95.0m
10. Inclined Pram type transom

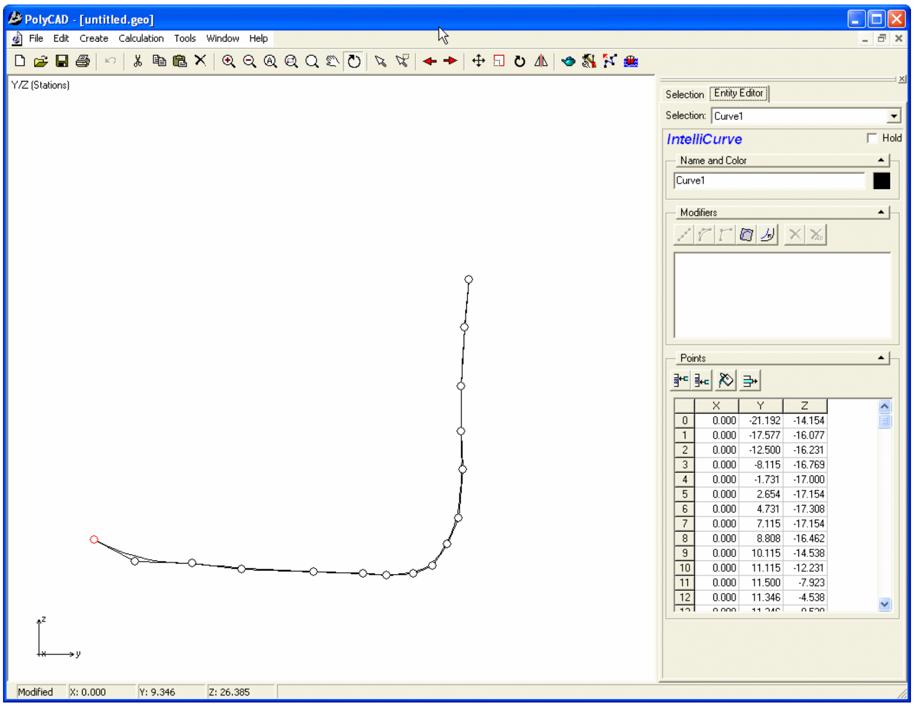


Figure 27.1, sixteen points of a curve roughly arranged into the shape of the midship section

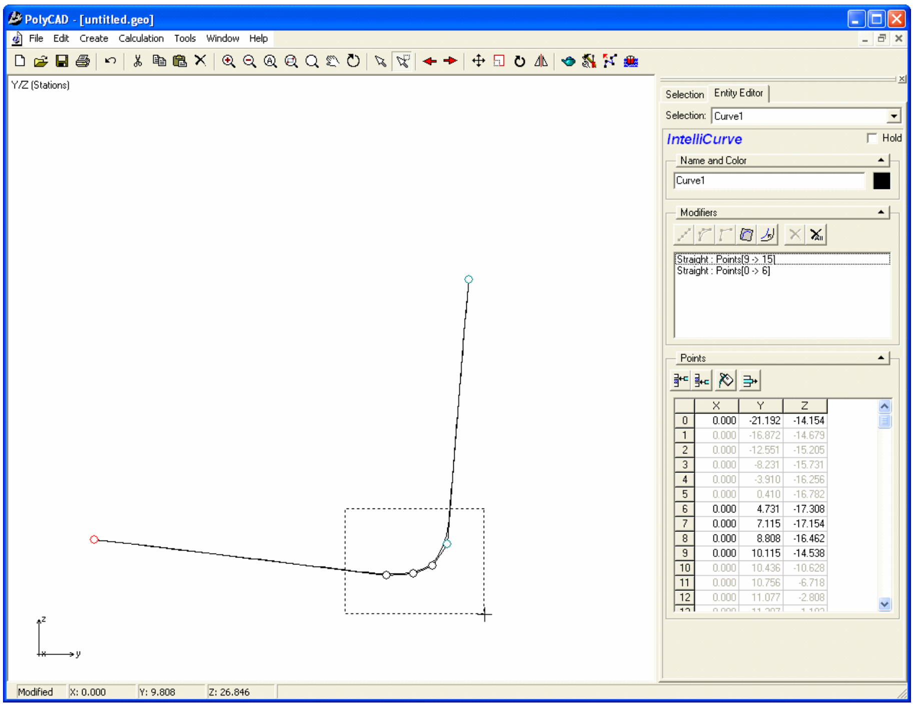


Figure 27.2, modifiers are applied to the curve to constrain the straight and curves segments reducing the number of vertex locations that have to be manually defined.

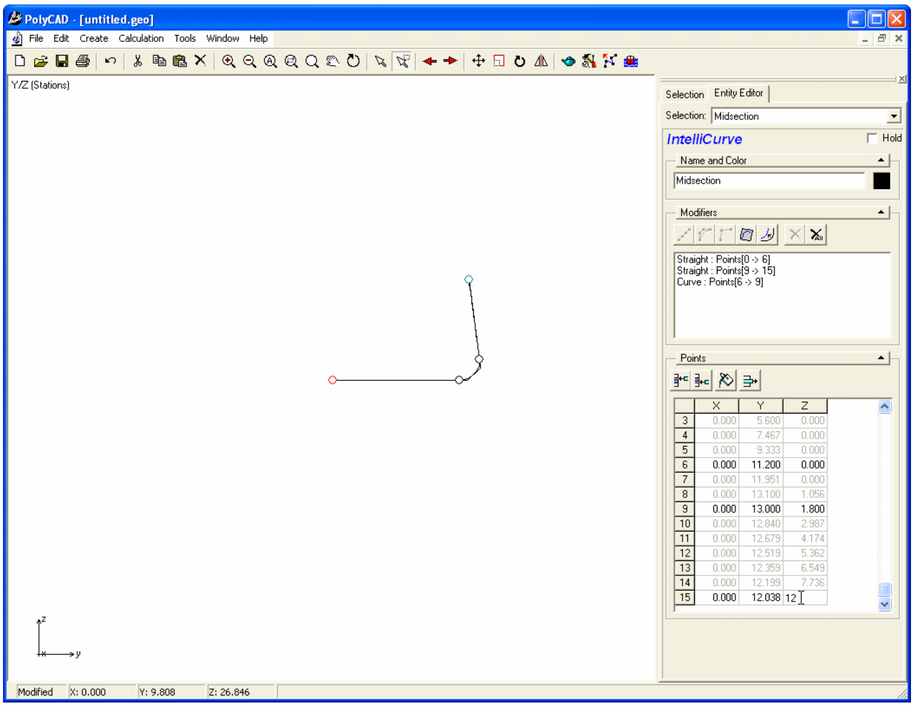


Figure 27.3, using the active vertices (only 4), the transverse dimensions of the section are accurately specified

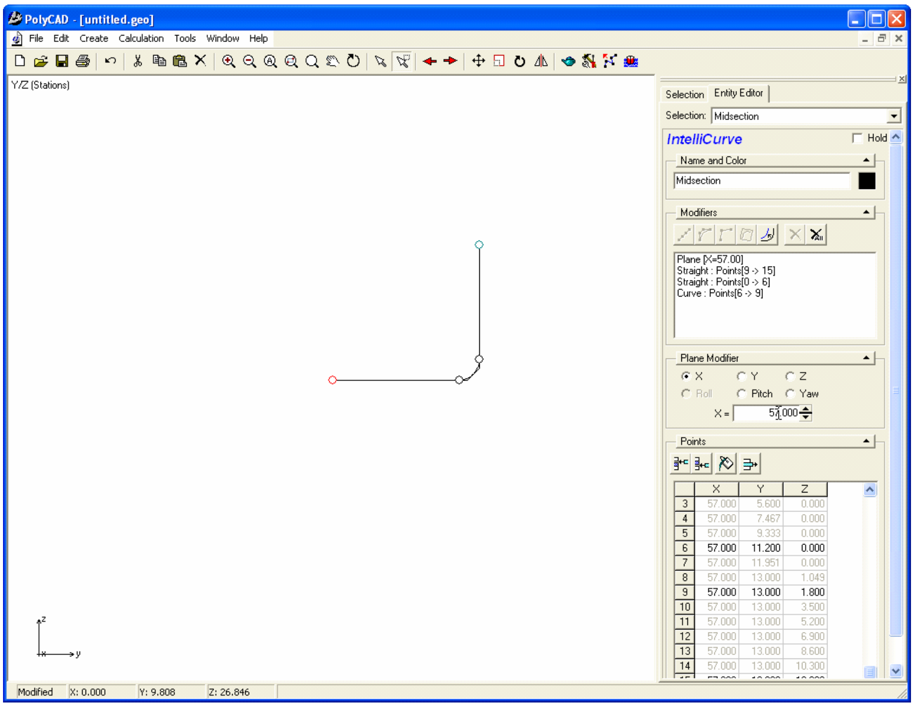


Figure 27.4, a plane modifier is used to locate all points at the midsection plane (x=57m)

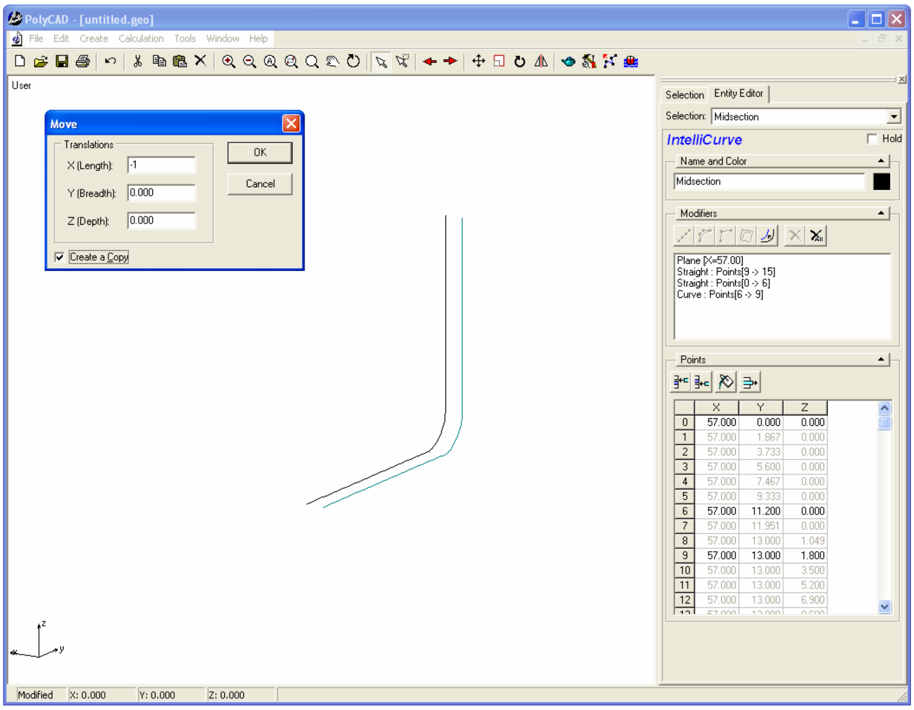


Figure 27.5, the midsection is copied to produce the forward surface flat (FSF) and aft surface flat (FSA) curves

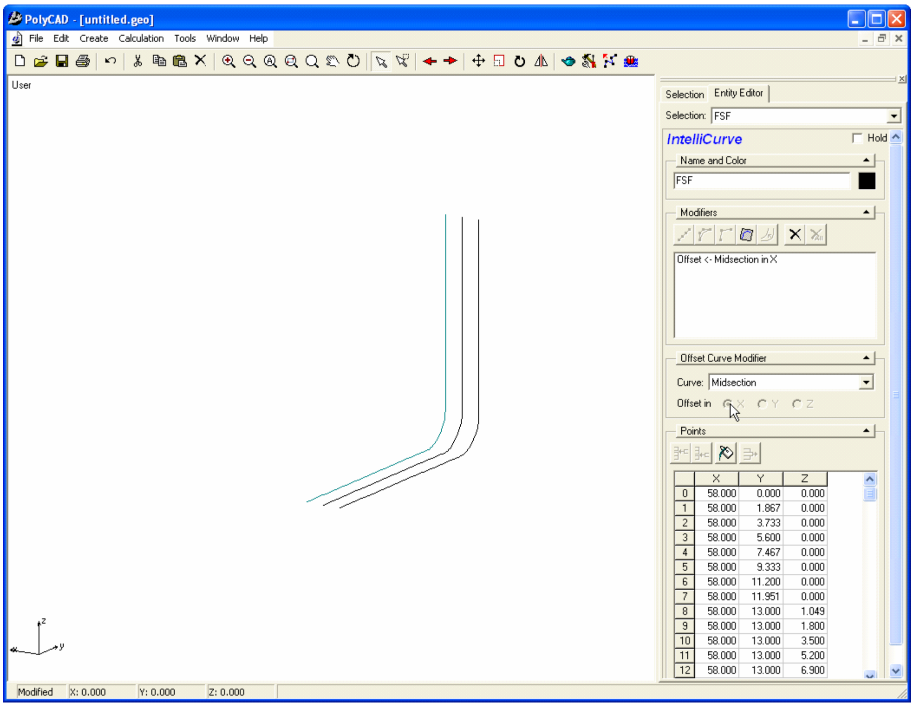


Figure 27.6, the offset modifier is applied to both flat definition curves in the *x* direction and linked to the midsection curve.

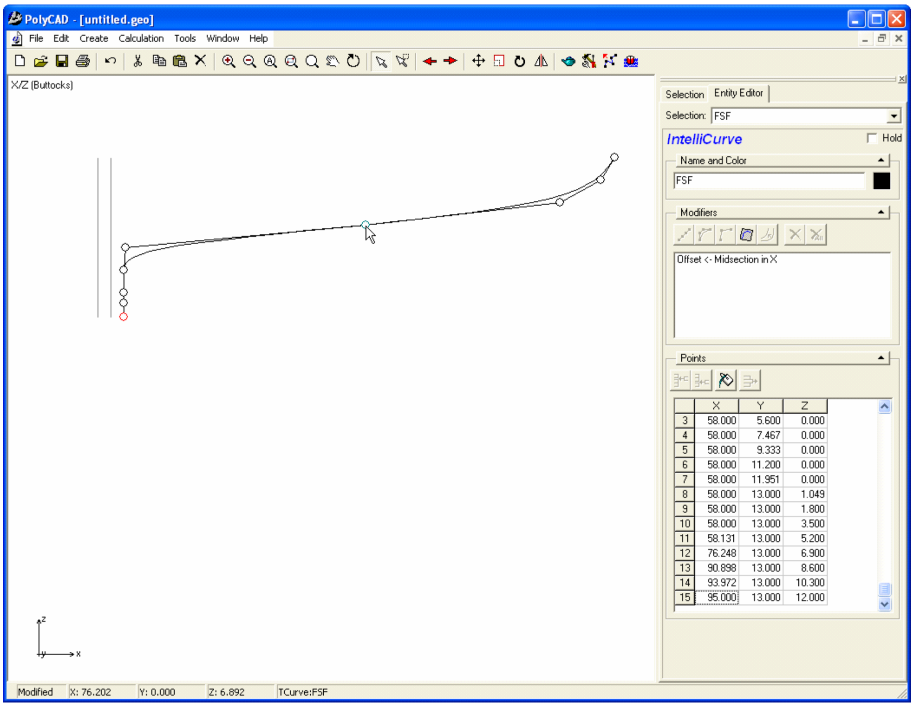


Figure 27.7, the FSF is shaped by drawing out the vertices. The forward extent of the parallel deck is set by position the last vertex of the curve accordingly. This dimension can be modified parametrically once the hull surface is available.

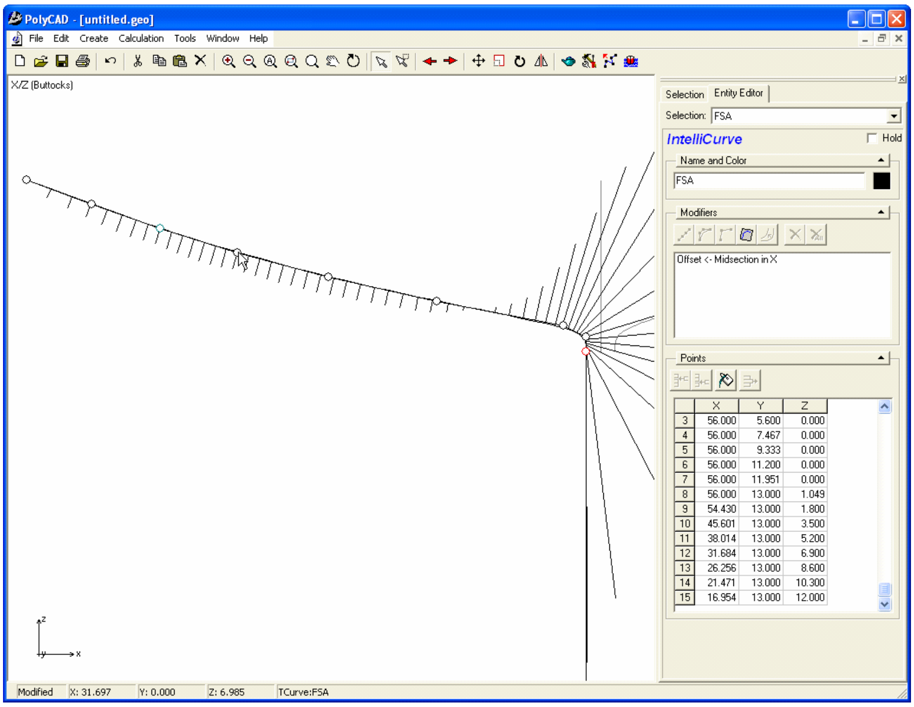


Figure 27.8, FSA is modified into shape on the flat of side, the curvature display is used to control the fairness.

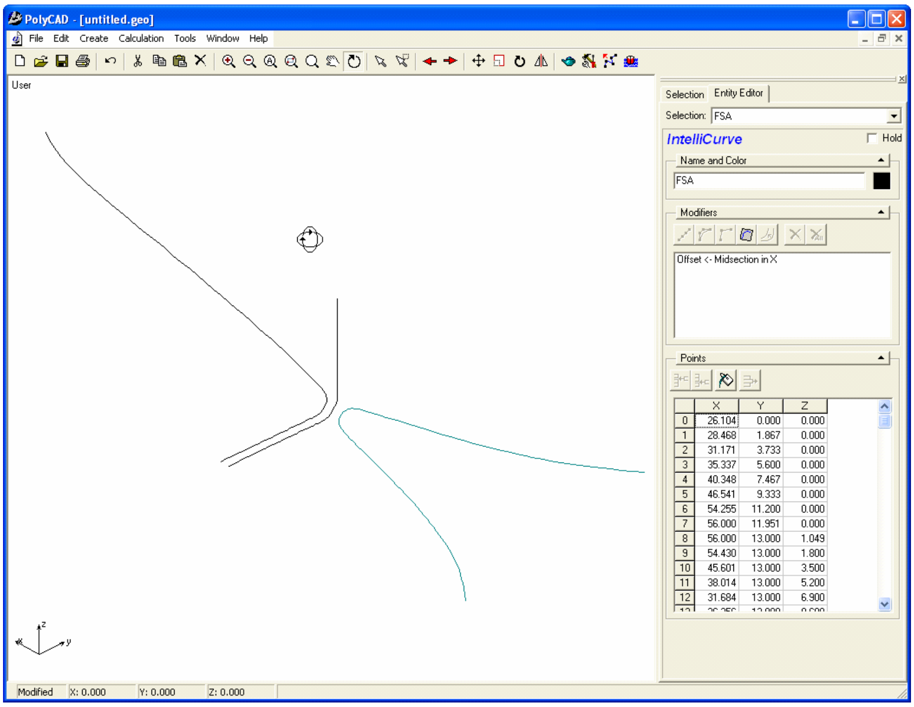


Figure 27.9, a view of the three curves forming the parallel middle body. From left to right, the FSF, Midsection and FSA curves, respectively.

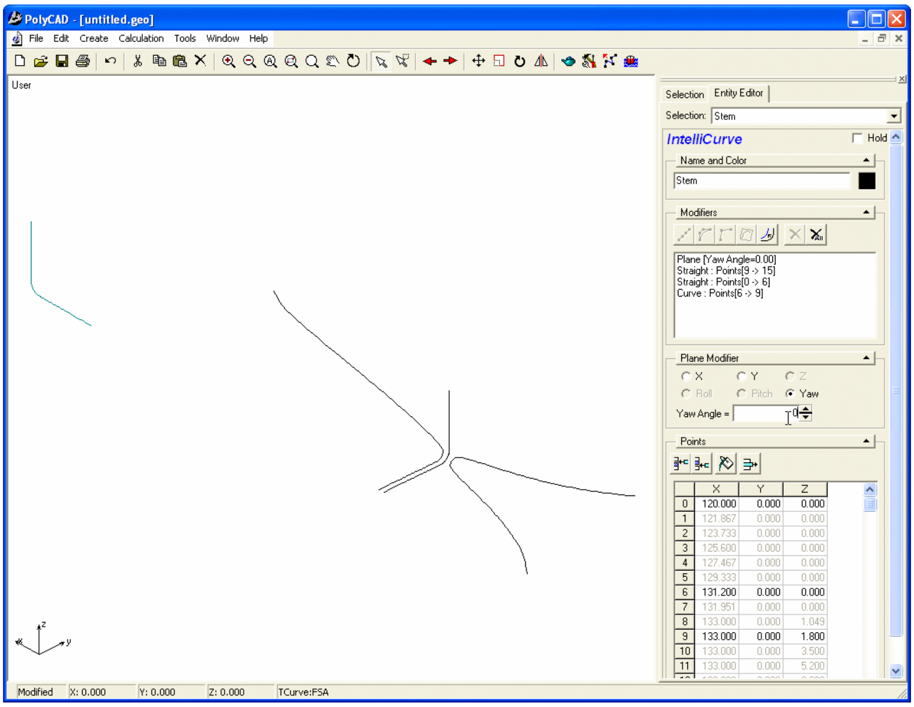


Figure 27.10, the stem curve is created by copying the midsection forward and rotating the curve round to the centre plane using the yaw angle from the plane modifier.

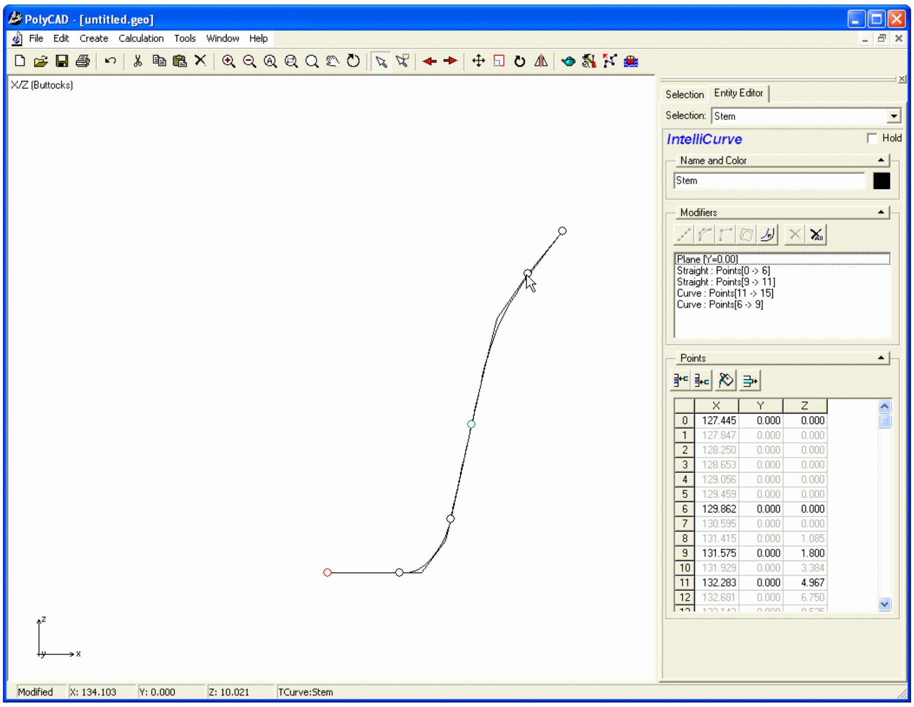


Figure 27.11, the modifiers are updated to allow for a curved stem shape.

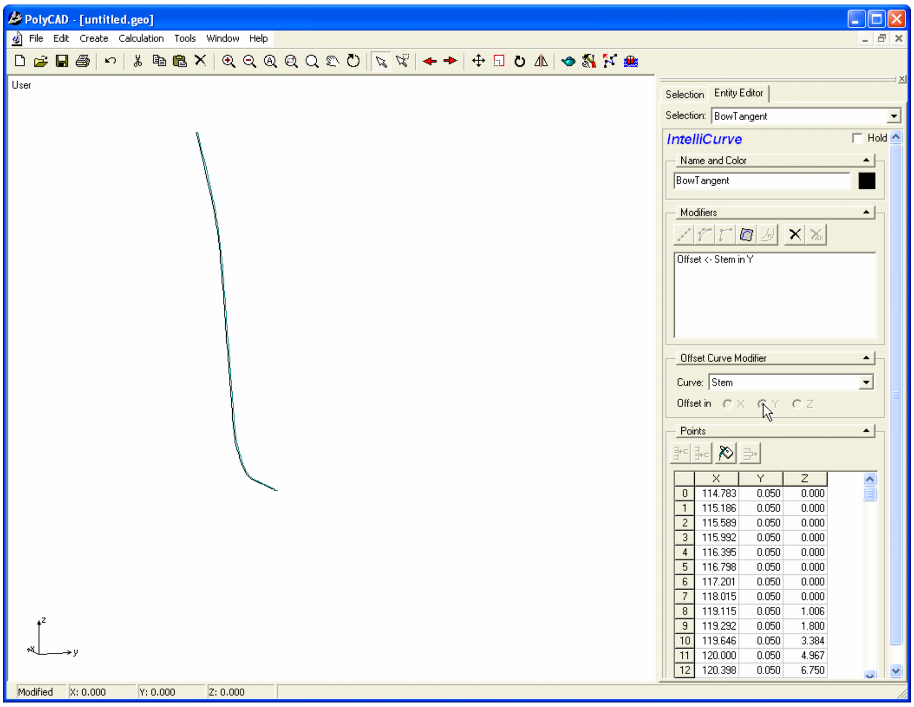


Figure 27.12, the stem is copied transversely to form the BowTangent curve. An offset modifier is used in the *y* direction to link this curve's shape to the stem curve's definition.

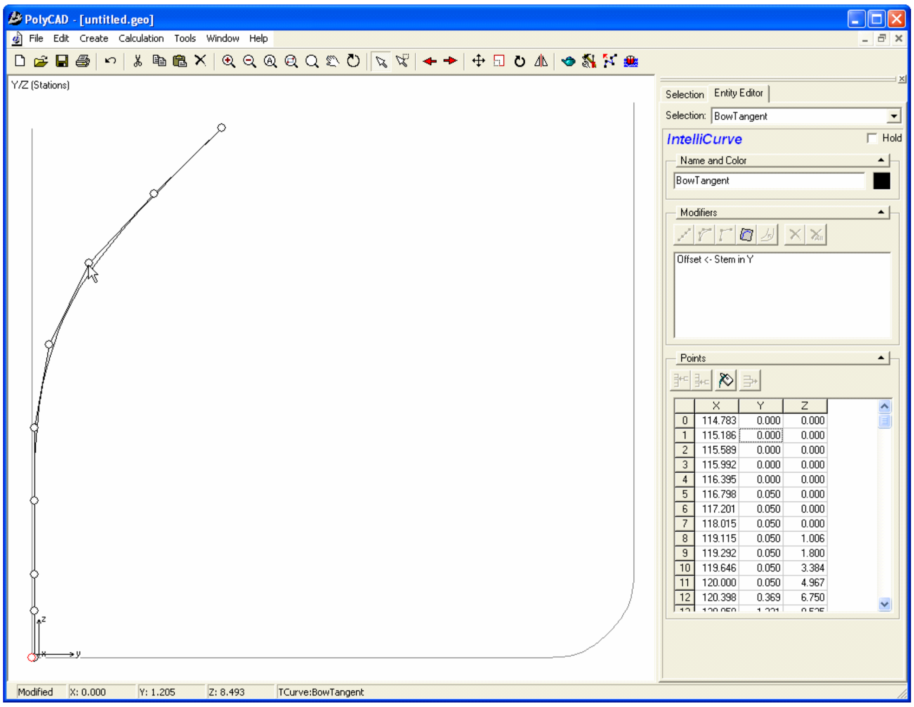


Figure 27.13, the BowTangent is shaped to produce flare at the deck.

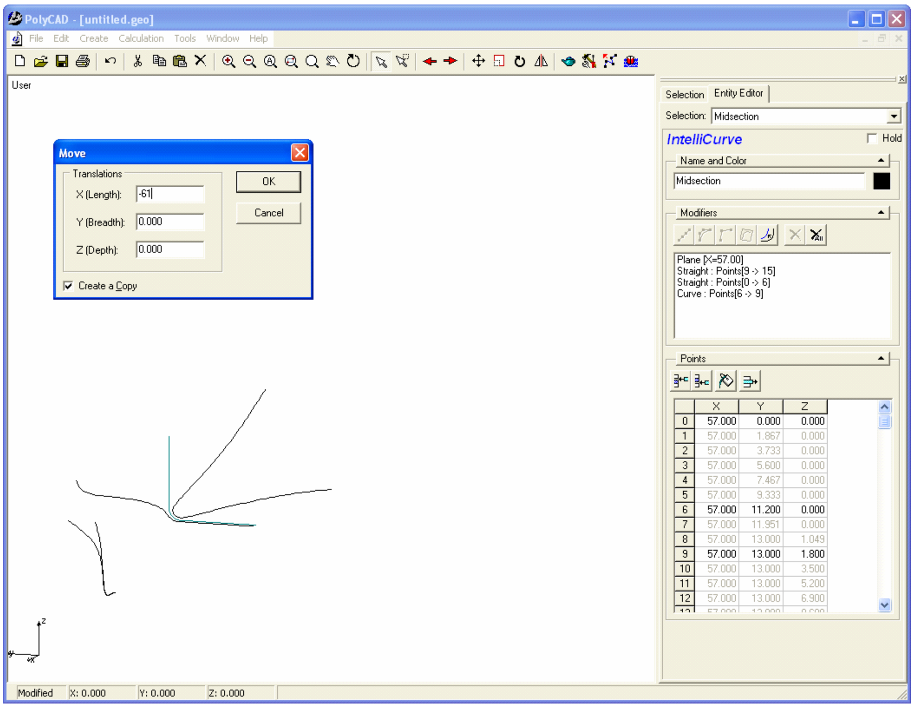


Figure 27.14, the midsection is copied aft to produce the transom curve.

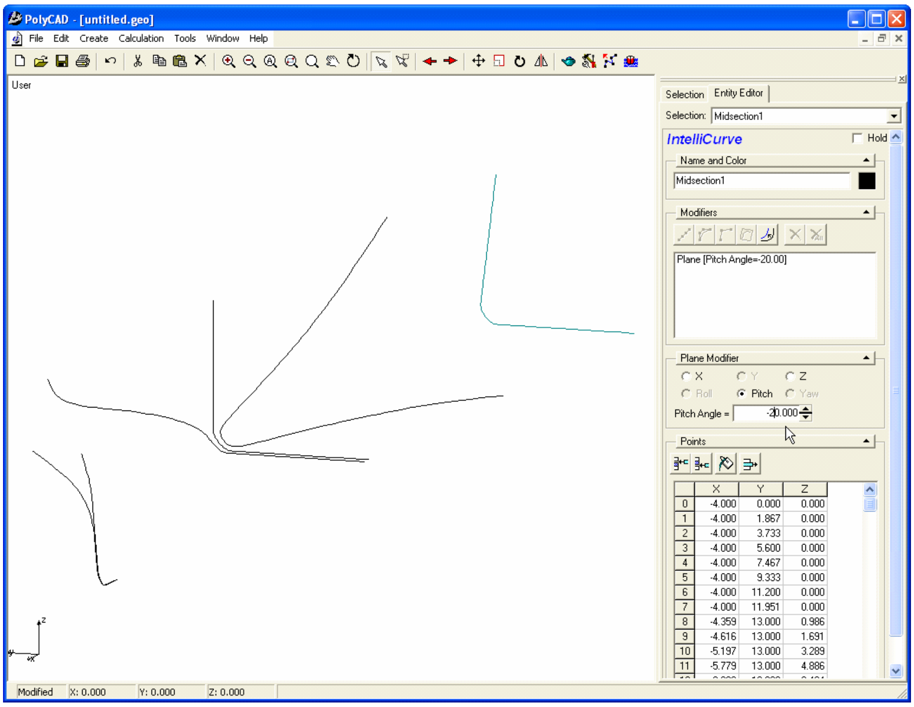


Figure 27.15, the transom is inclined by pitching the plane modifier by 20°.

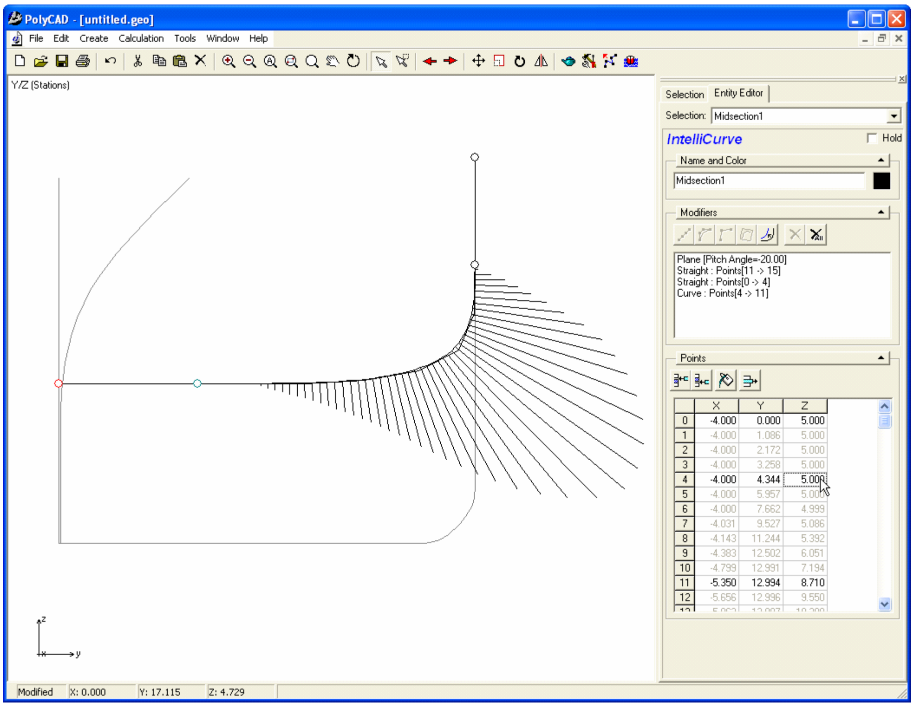


Figure 27.16, modifiers are used to shape the transom curve.

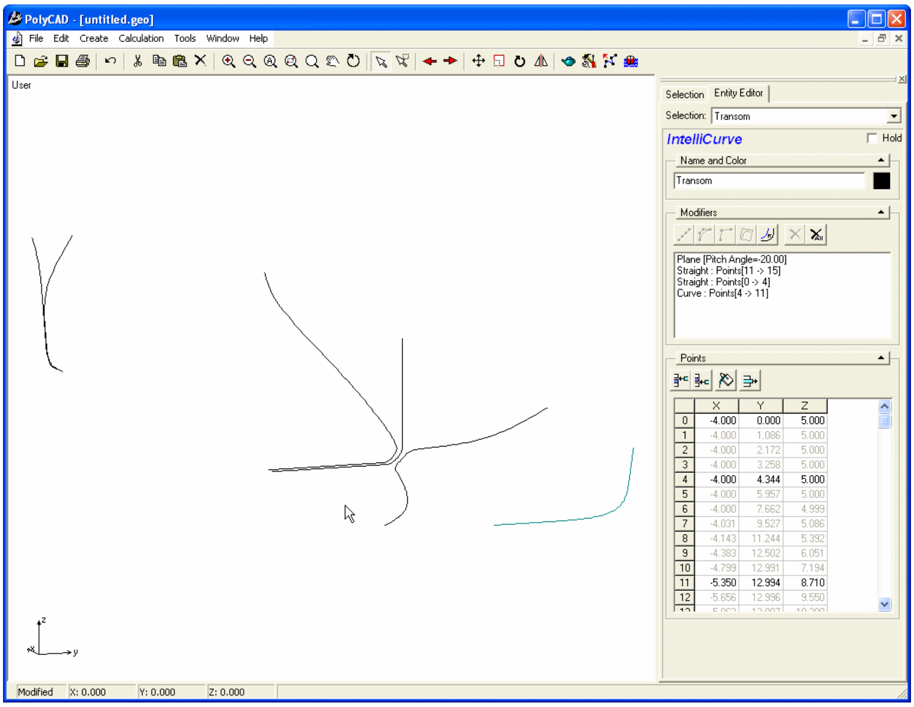


Figure 27.17, all the curves necessary to produce an initial hull form have been defined

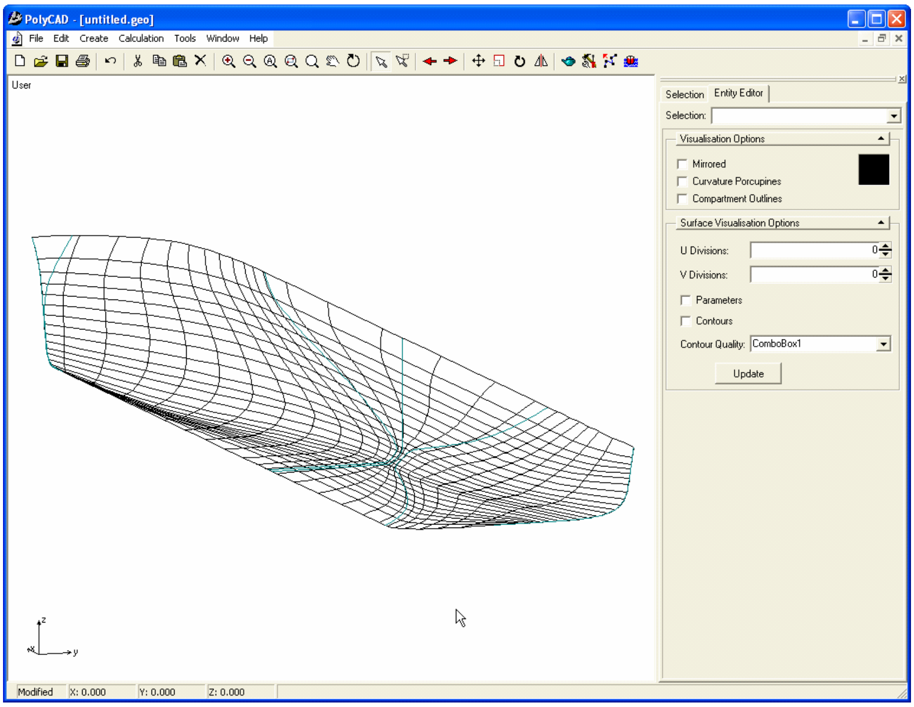


Figure 27.18, the parameter lines of the initial hull surface produced over the curves.

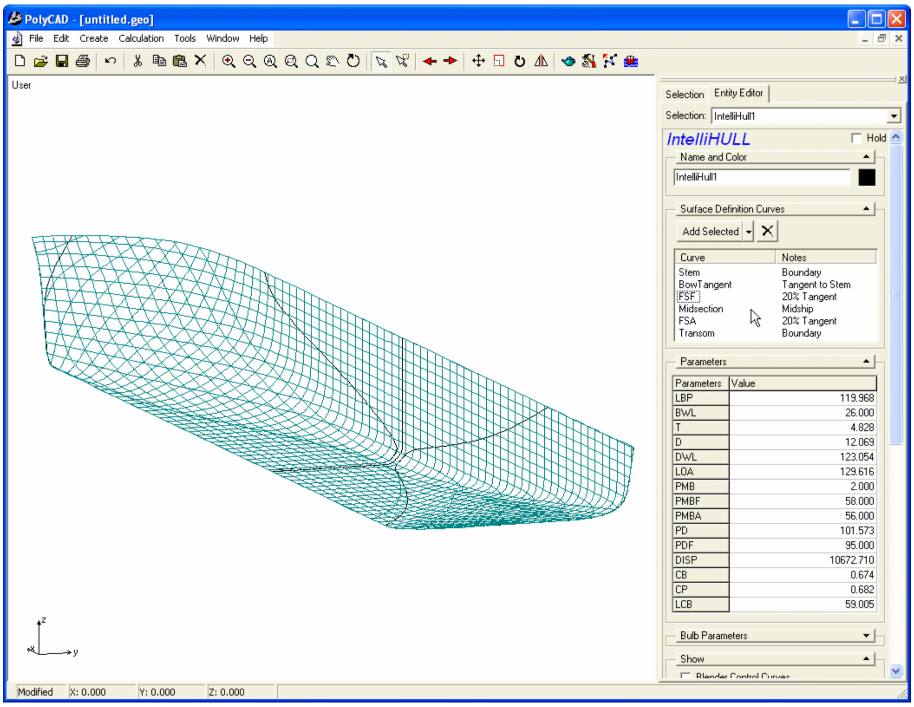


Figure 27.19, the contours of the hull surface. Note that the deck line is not horizontal so some curves need updating.

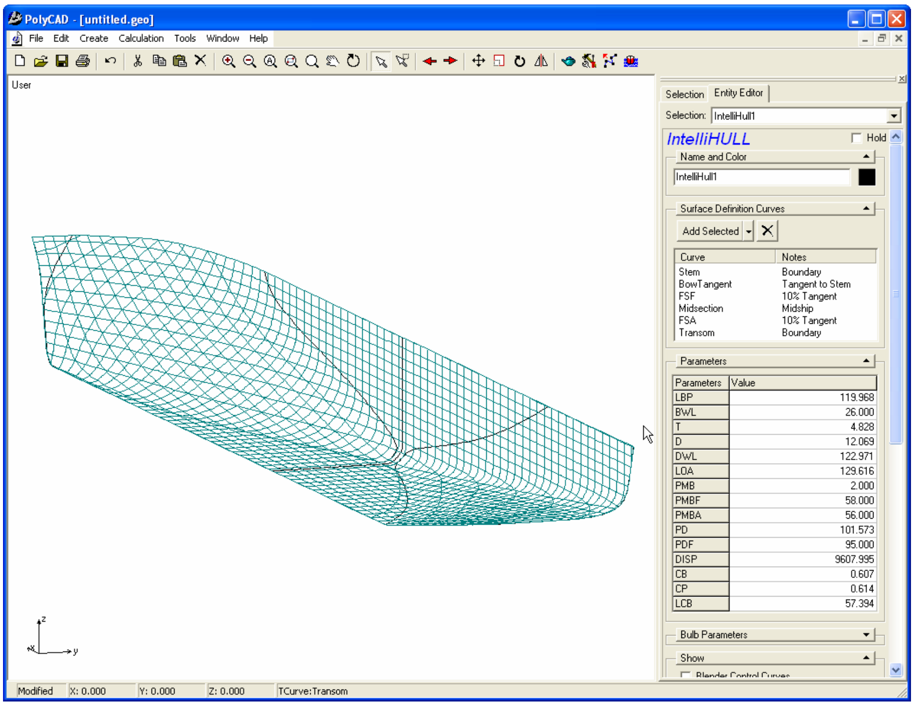


Figure 27.20, the flat of side tangents are adjusted from the default 20% to 10% percent.

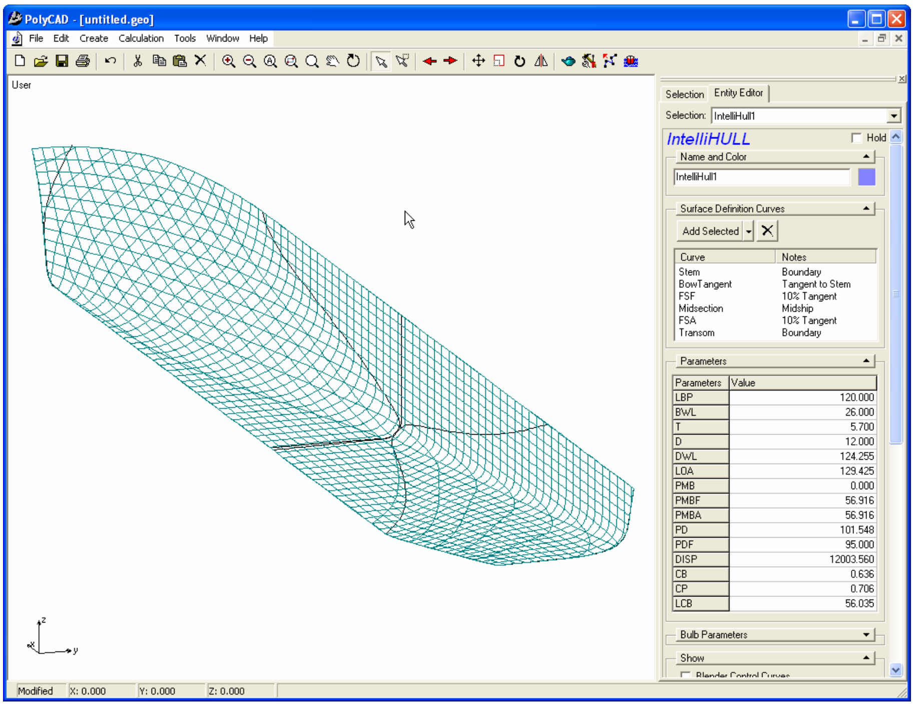


Figure 27.21, the definition curves are corrected so that the depth is consistent. Parameters are used to set the main particulars, LBP = 120m, BWL = 26m, D = 12m, T = 5.7m, PMB ˜ 0 and PDF = 95m.

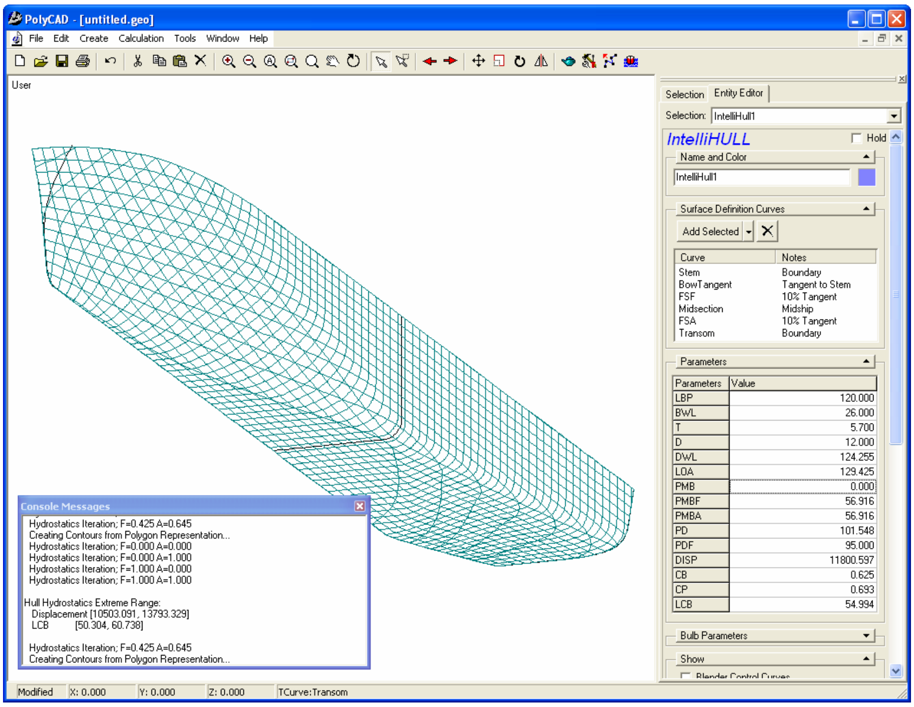


Figure 27.22, the hydrostatics parameters are adjusted to come close to the desired values. However, while the individual values of the hydrostatics are within the extreme range, the run is too full to allow the hydrostatics to be achieve together.

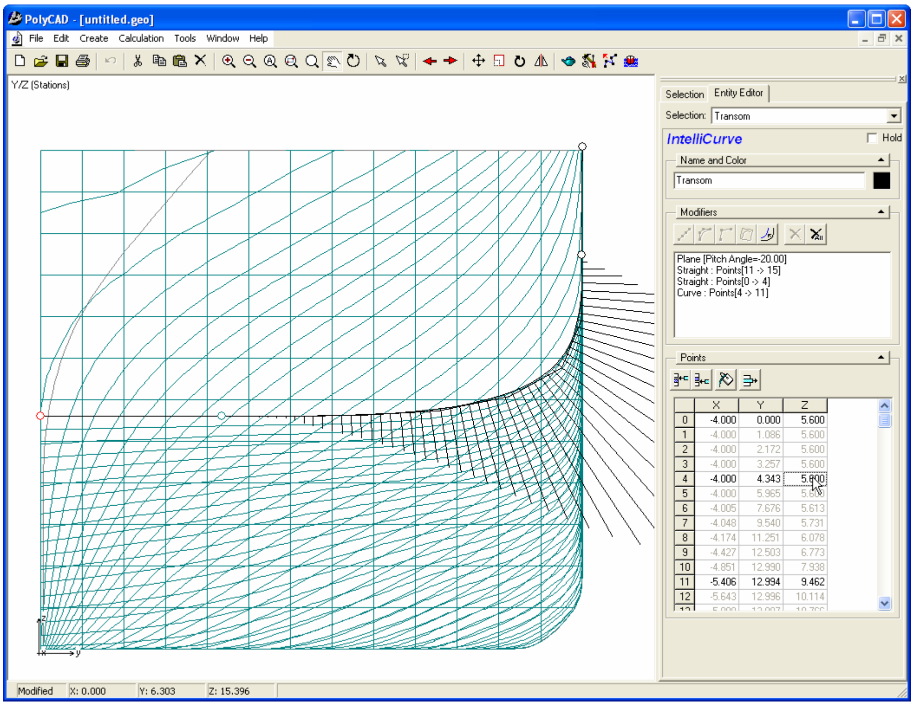


Figure 27.23, the transom curve is adjusted to reduce the immersion from 0.7m (z=5.0m) to 0.1m (z=5.6m) and the FSA curve is updated to further smooth the surface.

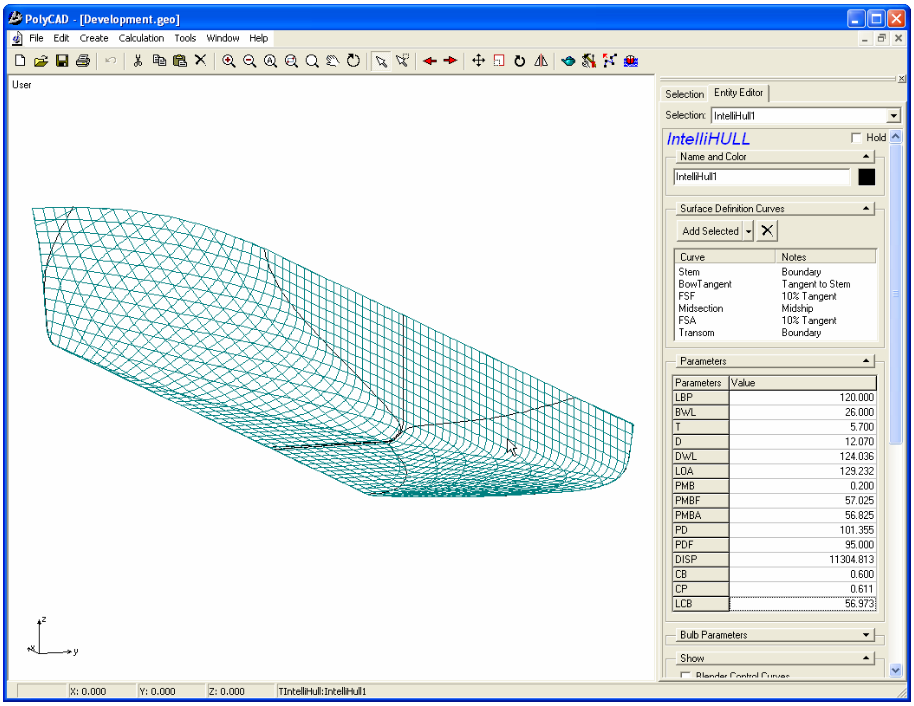


Figure 27.24, hydrostatics can now be achieved. C_B = 0.6 (? = 11305 ± 1 Tonne) and LCB = 57 ± 0.1m

When compared to existing techniques, the ease at which this technique can create the hull form surface is quite tremendous. The screen shot images show the development of the hull form over a period of around fifteen minutes. By comparison, it would take the best part of a day to develop this hull form in the best current manually interactive hull form design tool. The concept of taking some basic curves representing the shape and forming a surface around them is possible and the results are very pleasing. Considering the limited resources that have been used in this project, it would be interesting to see how good a system a commercial developer could create in the existing tools.

The actual surface generation technique is geometrically quite simple. The surface is constructed using blending curves to position the control vertices and the blending curves are developed using simple geometric techniques such as basic vector calculations. This, alone, is not the only factor in the surface development. The interface tools are very important, allowing the user to build up the definition by applying the geometric constraints. However, the user is not obliged to use them and is free to develop the geometry manually if so wished. The implementation shows that providing the correct interface to the surface generation technique is just as important as the technology itself. Particularly in hull surface design, the user has a particular need to define and maintain specific shapes in the geometry. Consequently, the approach that is provided by this technique is very well suited to the hull surface design discipline.

While this technique makes the development of the hull form much quicker and easier, the rate at which a hull surface can be developed does create a few problems when there is the need to rely on experience. The time consuming manual definition process involves a lot of iteration and manipulation allowing the designer to identify the arrangement of definition curves for the particular hull shape. If there is a need to reformat the definition structure of the surface, it is seen as an acceptable part of the development process. The interesting thing that happens with a more rapid approach to hull form design is that as the surface is formed in a similar way to the manual process, the same amount of skill and experience is required to develop the surface, just without the need for all the manipulation. However, as the system forms the surface so quickly, the user does not get the time to become familiar with appropriate definition arrangement to form the surface. The development of the bulk carrier hull form illustrated this problem, (18.5.2). Many hours were spent toiling with the wrong surface definition, even using direct manipulation to improve the shape. However, as a good solution was not being reached, an attempt was made with different arrangement on the stern curve. This formed exactly the right surface shape straight away without the need for direct manual manipulation. In hindsight, it was obvious that the original stern curve deformed the surface in a way that would not result in the correct hull form shape. However, these type of conclusions cannot be drawn until experience in the shape of hull form surfaces, with NURBS in this case, has been obtained. New techniques that speed up complex operations are usually expected to require less skill to use. However, in this case, the same skill is required. The technique allows the user to perform the same design operations but removes the need for considerable manipulations. As a consequence, the designer must provide skill and experience in much more concentrated amounts to keep up with the tool. Overall, the results of the tool are very pleasing and it is good to see that a significantly better approach can be found to initial hull form design.

28. APPENDIX 7 – HULL FORM TRANSFORMATION IMPLEMENTED BY PARAMETRIC MODIFICATION

Once the designer has constructed an initial hull surface, it can be analysed with respect to the other areas of the ship that it affects. The analysis of detailed surface characteristic is not usually performed until a little later in the design process. In the initial stages, modification of the main particulars is performed until vessel fits the specification provided by the owner. In present hull surface design tools this will be very laborious because the surface modification is implemented manually. The design tool may provide separate functions to transform the surface. However these rarely provide exactly the right transformation required by the designer.

The practical transformation of the hull form definition was one of the key aspects that this project found important to cover. But unlike present tools, the best approach was found to be by providing transformations through the use of form parameters and to provide the interface to the transformations within the design environment so that they are available at the same time as the geometrical definition. The implementation provides all the important form parameters and these invoke custom transformations which use form topology to modify the surface definition. There are four groups of transformations:

- Transformation of Transverse Sectional Shape
- Transformation of the Parallel Middle Body
- Transformation of Length
- Transformation of Hydrostatic Particulars

The transformation of sectional shape is usually handled by affine transformation procedures. These apply a global transformation to the whole of the definition. However, as detailed in Chapter 6 these global transformations do not have the ability to take account of the changes that will occur to particular shapes and hence unwanted deformation in the definition takes place. Instead of using affine transformations, the form topology definition breaks the hull form into areas which can be transformed separately which, through the form topology structure links, will affect the shape of the whole hull surface. Consequently, in the transformation of sectional shape, it is only necessary to transform the shape of the midship section. However, in the implementation, each definition curve is analysed for cases when the geometrical constraint tools have not been used. Figure 28.1 and Figure 28.2 illustrate the transformation of breadth and depth respectively. The approach to design the transformations is detailed in Chapter 14. The diagrams illustrate that transformations can be developed which minimise unwanted deformation to the hull form when global changes are applied. In the case of the transformation of breadth, (Figure 28.1), the result shows minimal deformation around the bilge radius area. A clearer example of the reduced deformation is shown when the depth is changed, (Figure 28.2). All the sections between the midship section and the transom have remained practically the same with the only changes being in the vertical extent of the flat of side. This diagram illustrates exactly what these types of transformation functions are trying to achieve. Practical modification with minimised change in the shape of the surface.

The approach taken to the implementation of longitudinal changes in shape, such as modification to the parallel middle body, is achieved with processes very similar to standard practice. However, advantages are gained by using geometric constraint tools on the surface result in it being only necessary to modify the curves representing the extents of the parallel middle body. The geometric constraints take care of applying the transformation to the surface definition itself. Figure 28.3 shows an example of a transformation to the parallel middle body.

Changes in the length of the vessel can be achieved using a similar approach to the modification of the parallel middle body, (Figure 28.4). However, unlike the parallel middle body transformations, which rely on the scaling of the two curves defining the extent of the parallel middle body, changes in length is achieved by moving whole definition curves. Only when it is impossible to move curves does it become necessary to scale the extent of the hull form by scaling factors. This process ensures that the shapes of particular areas of the hull form are accurately maintained for as long as possible. A detailed review of the diagram shows that for the area of the hull surface, the shape is exactly the same as the original hull form surface.

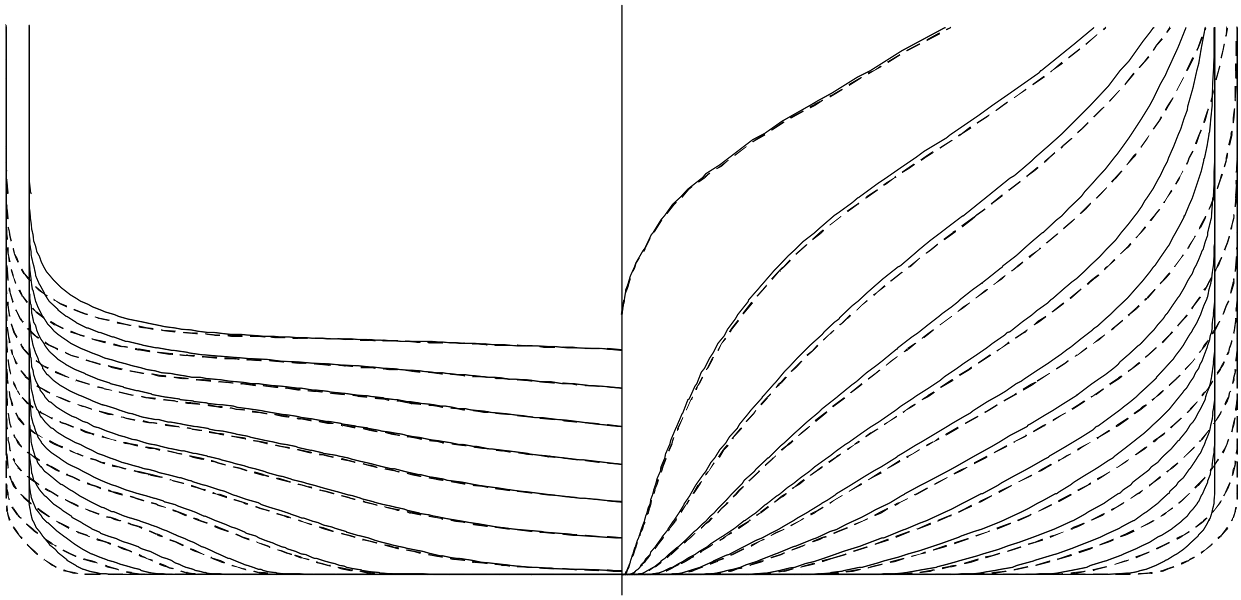


Figure 28.1, the transformation of Beam from 26.0m to 27.0m. Bilge radius shape is not deformed by the operation.

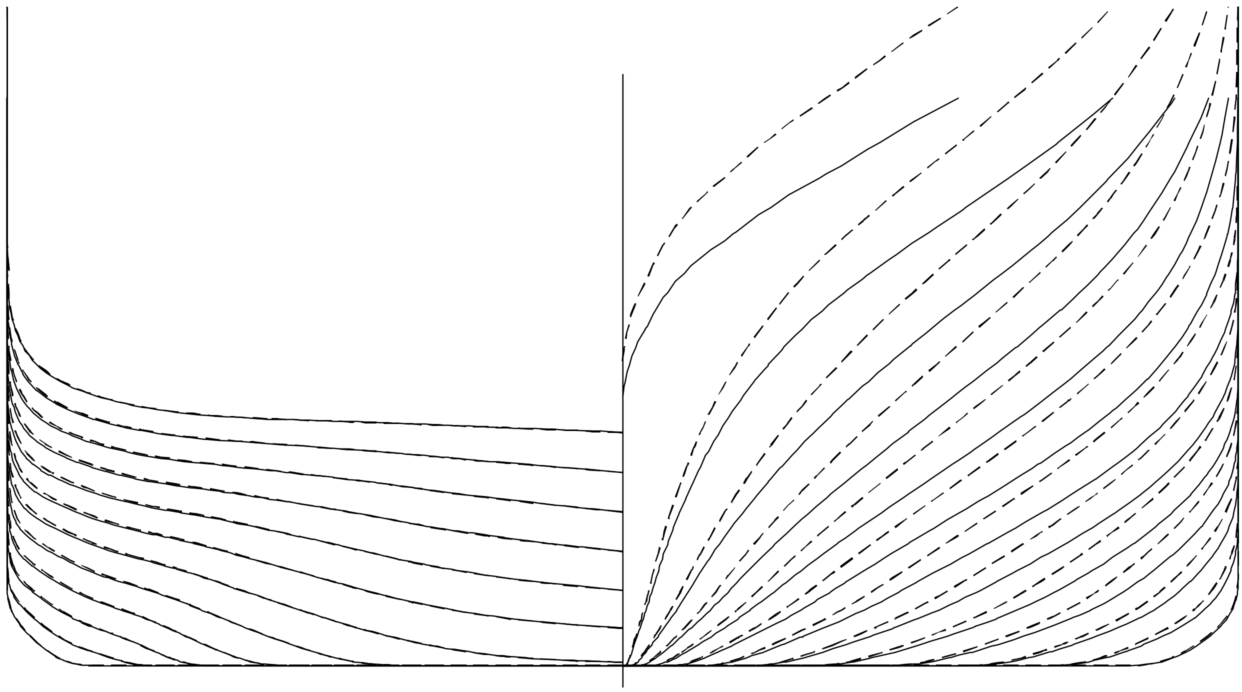


Figure 28.2, the transformation of Depth from 12.0m to 14.0m. Sections in the stern have hardly changed due to the ship section shape from the midship section to the transom.

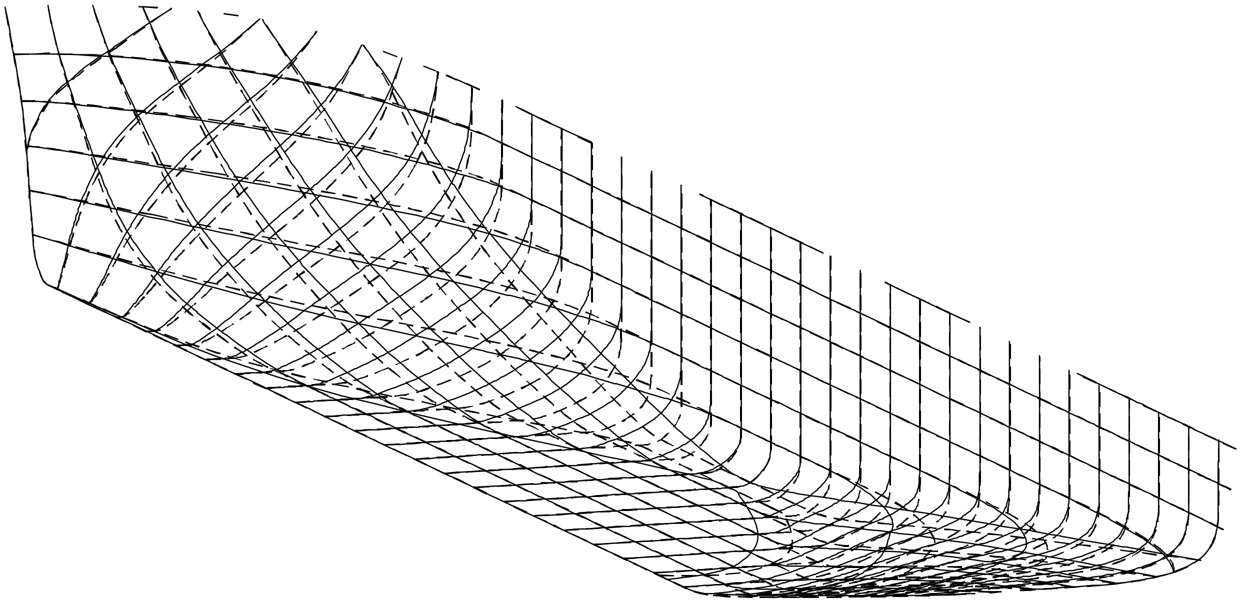


Figure 28.3, the transformation of parallel middle body from 0.0m to 20.0m.

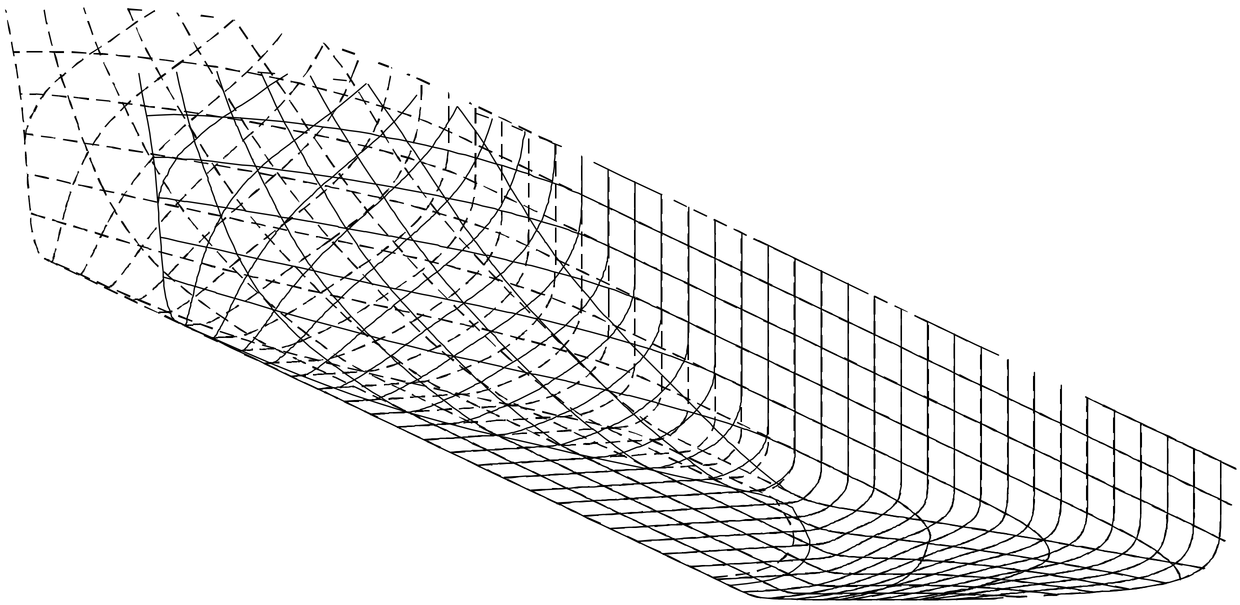


Figure 28.4, the transformation of LBP from 120.0m to 140.0m by increasing PMB. Note that the surface shape of the entrance remains the same.

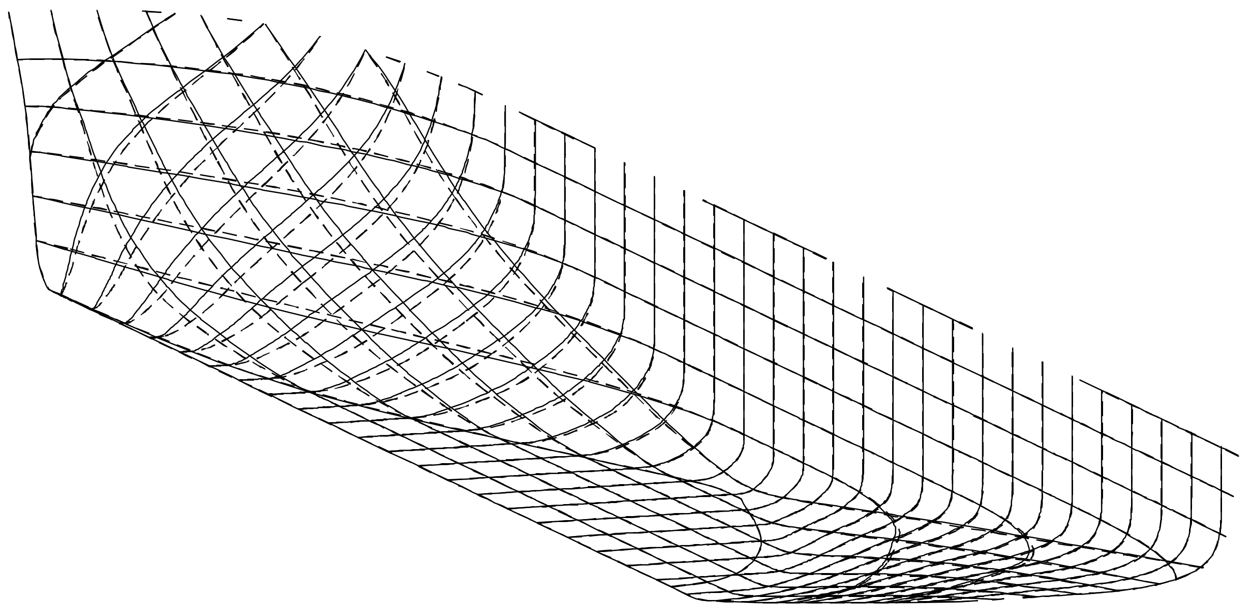


Figure 28.5, transformation of hydrostatics, C_B from 0.60 to 0.62 and LCB from 57.0m to 58.0m.

Modifications to the hydrostatic characteristics of the vessel are relatively simple if geometric form constraints are in place. The geometric constraints controlling the ends of the hull surface have been designed to work with an additional parameter which is used to control the volumetric characteristic. As there are two ends, there are two parameters resulting in the ability to modify the displacement and the longitudinal centre of buoyancy (LCB). The full modification scheme is given in more detail in Chapter 14. Figure 28.5 shows the example hull form modified to achieve particular hydrostatic properties. The transformation process achieves changes in shape quite subtly. Apart from the internal iteration processes, the transformation is a lot simpler than other techniques by Lackenby [26] or Hollister [44]. However, this transformation process does rely on a surface defined using form topology and geometric constraints.

The results of the parametric transformation procedures are very satisfactory. However, unlike existing tools which use separate transformation functions, the procedures implemented through form topology take full advantage of the structure and, as a consequence, they have a much simple implementation.