İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

COMPUTER AIDED FAIRING OF SHIP HULL FORMS

PhD Thesis by
Ebru NARLI, Msc.

(508950001012)

100706

Date of submission : 31 May 1999

Date of defence examination: 29 November 1999

Supervisor (Chairman): Assoc. Prof. Dr. Kadir SARIÖZ 15.12.1999

Members of the Examining Committee Prof. Dr. Reşat BAYKAL (I.T.U.) 17.12.1999

Prof. Dr. Eşref ADALI (I.T.U.) 20.12.1999

Prof.Dr. Nihat TEKİN (Y.T.U.) 24.12.1999

Prof.Dr. Abdülkerim KAR (M.U.) 20.12.1999

NOVEMBER 1999

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

TEKNE FORM YÜZEYLERİNİN BİLGİSAYAR DESTEKLİ DÜZGÜNLEŞTİRİLMESİ

DOKTORA TEZİ
Y. Müh. Ebru NARLI
(508950001012)

Tezin Enstitüye Verildiği Tarih : 31 Mayıs 1999
Tezin Savunulduğu Tarih : 29 Kasım 1999

Tez Danışmanı :	Doç. Dr. Kadir SARIÖZ
Diğer Jüri Üyeleri	Prof. Dr. Reşat BAYKAL (İ.T.Ü.)
	Prof. Dr. Eşref ADALI (İ.T.Ü.)
	Prof. Dr. Nihat TEKİN (Y.T.Ü.)
	Prof. Dr. Abdülkerim KAR (M.Ü.)

KASIM 1999

PREFACE

I would like to express my sincere gratitude to my thesis supervisor, Dr. Kadir SARIÖZ for his guidance and encouragement. I am grateful for the valuable advise, critical appraisal, and many suggestions given by the members of the Faculty of Naval Architecture and Ocean Engineering. I would also like to thank to my qualification jury, particularly Prof. Dr. Reşat BAYKAL, Prof. Dr. Nihat TEKİN, Prof. Dr. Eşref ADALI, and Assoc. Prof. Dr. Ahmet ALKAN for their suggestions and critics.

The author is deeply indebted to her parents for their support, patience and understanding.

May, 1999 Ebru NARLI

TABLE OF CONTENTS

ABBREVIATION

AP	: Aft Perpendicular
C^n	: n^{th} degree parametric continuity
CAD	: Computer Aided Design
CAGD	: Computer Aided Geometric Design
CASD	: Computer Aided Ship Design
CN	: Closeness Number
DWL	: Design Waterline
FP	: Fore Perpendicular
FN	: Fairness Number
G^n	: n^{th} degree geometric continuity
NPL	: National Physical Laboratory
NURBS	: Non-Uniform Rational B-Splines
SAC	: Sectional Area Curve
US	: United States
2D	: Two-Dimensional
3D	: Three-Dimensional

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$a(x)$	: Longitudinal distribution of the immersed sectional area
A	: Total area under the SAC indicating fullness
$A(x)$	: Distribution of sectional areas below the design waterline along the length of the ship
A_{WP}	: Design waterplane area
$b(u)$	: Curve's binormal vector
$b(x)$	: Longitudinal distribution of design waterline breadth
B	: Beam
B_i	: Coordinates of the defining polygon vertices, de Boor points
B/T	: Beam to draught ratio
C_B	: Block coefficient
C_M	: Midship area coefficient
C_P	: Prismatic coefficient
C_{PA}	: Afterbody prismatic coefficient
C_{PF}	: Forebody prismatic coefficient
C_{WP}	: Waterplane area coefficient
δC_P	:Added slice area representing the required change in prismatic coefficient
δC_{PA}	: Change in afterbody prismatic coefficient
δC_{PF}	: Change in forebody prismatic coefficient
dM	: Longitudinal transfer of moment of the strip
dS	: Differential area
dx	: Infinitesimal length
d_ω	: Differential area
D	: Depth
e_1, e_2	: Directions of principal curvatures
E	:Young's modulus of elasticity determined by the material properties of the beam
E	: Error of approximation for least squares method
E	: Elastic bending energy
$f(x)$	: Dependent variable corresponding to offset values
F	: Three-dimensional vector functions
F_{ik}	: Cubic spline blending functions
$F(x)$	: Objective function
$g(x)$	: Analytic function, approximating polynomial
$g_i(x)$	: Constraint functions
G	: Three-dimensional vector functions
h_a	: Centroid of the added area in the afterbody
h_f	: Centroid of the added area in the forebody
h_i	: Weights associated with defining polygon vertices
H	: Mean curvature
$H(t)$	: Hermite polynomials

$\mathbf{I}$	: Moment of inertia determined by the cross sectional shape of the beam
$\mathbf{I}$	: Parameter interval
$\mathbf{J_{n,i}}$	: Bezier or Bernstein basis functions
$\mathbf{K}$	: Gaussian curvature
$\mathbf{L}$	: Ship length
$\mathbf{L/B}$	: Length to beam ratio
$\mathbf{LCB}$	: Longitudinal centre of buoyancy
$\mathbf{LCF}$	: Longitudinal centre of flotation
$\mathbf{L_i(t)}$	: Lagrange polynomial basis functions
$\delta\mathbf{LCB}$	: Required shift of LCB
$\mathbf{M(x)}$	: Bending moment along the length of the beam
$\mathbf{M_{j,l}}$	: Normalised B-spline basis functions of degree l-1
$\mathbf{n}$	: Unit surface normal
$\mathbf{n_j(t)}$	: Newton polynomials
$\mathbf{n,m}$	: Number of points
$\mathbf{n(u)}$	: Curve's normal vector
$\mathbf{N_{i,k}}$	: Normalised B-spline basis functions of degree k-1
$\mathbf{p}$	: Parallel middle body length
$\delta\mathbf{p}$	: Required change in parallel middle body length
$\mathbf{P_i}$	: Given interpolation/approximation points
$\mathbf{P(t)}$	: Position vector of any point on the curve
$\mathbf{Q(u,v)}$	: Surface definition
$\mathbf{Q_u,Q_v,Q_{uv}}$	: Parametric partial derivatives of a surface
$\mathbf{r}$	: Multiplicity of parameter value t
$\mathbf{R_c}$	: Radius of curvature of any curved line
$\mathbf{R(x)}$	: Radius of curvature of the beam
$\mathbf{S}$	: Sum of squared principal curvatures
$\mathbf{t}$	: Arbitrary curve parameter
$\mathbf{t_{min}}$	: Minimum parameter value
$\mathbf{t_{max}}$	: Maximum parameter value
$\mathbf{t(u)}$	: Curve's tangent vector
$\mathbf{T}$	: Draught
$\mathbf{x}$	: Independent variable corresponding to sectional positions
$\mathbf{x_a}$	: Afterbody centroid of the original sectional area curve
$\mathbf{x_b}$	: Longitudinal centre of buoyancy
$\mathbf{x_f}$	: Forebody centroid of the original sectional area curve
$\mathbf{x_f}$	: Longitudinal centre of flotation
$\mathbf{x_i}$	: Elements of knot vector
$\mathbf{x_n,y_n,z_n}$	: Longitudinal, horizontal, vertical co-ordinates for the variant form geometry
$\mathbf{x,y,z}$	: Longitudinal, horizontal, vertical co-ordinates of the parent form geometry
$\mathbf{x_1,x_2,\ldots,x_n}$	: Design variables
$\mathbf{x(t),y(t),z(t)}$	: Real-valued functions of parameter t
$\delta\mathbf{x}$	: A vertical strip length
$\delta\mathbf{x}$	: Linear shift of dimensionless sectional area ordinates
$\mathbf{[X]}$	: Knot vector denoting parameterisation of the B-spline curves
$\mathbf{X(s)}$	: Arc length parameterisation of the curve
$\mathbf{X'(t)}$	: First derivative

$\mathbf{X''(t)}$	: Second derivative
$\mathbf{X(t),X(u)}$	: Regular parameterisation of the curve, vector-valued function
$\mathbf{w_i}$	: Weights factors
$\mathbf{y}$	: Offset values
$\mathbf{y(x)}$	: Mathematical spline curve
$\mathbf{y'(x)}$	: First derivative of the mathematical spline curve
$\mathbf{y''(x)}$	: Second derivative of the mathematical spline curve
$\bar{y}$	: Vertical centre of buoyancy
$\mathbf{u,v,w}$	: Parametric directions
Δ	: Displacement
∇	: Displaced volume
θ	: Required angle of shift for adjusting the LCB position
$\Re^3$	: Euclidean space
Δ_j	: Coefficients of monomials
δ_{ij}	: Kronecker delta
δ_i	: Deviation at the i^{th} offset
$\phi(t)$	: Basis functions for least squares method
$\kappa(u)$	: Curvature of a curve
$\tau(u)$	: Torsion of a curve
κ_1,κ_2	: Principal normal curvatures

COMPUTER AIDED FAIRING OF SHIP HULL FORMS

SUMMARY

This thesis describes the problem of fairing of ship hull forms and introduces novel procedures which can be used in the early stages of the design process to produce fair hull forms. These procedures range from B-spline fitting of two-dimensional ship lines to complex optimisation formulations where the designer may specify various objective functions and geometric constraints to obtain a three-dimensional fair hull form.

Development of a three-dimensional fair hull form is one of the main design requirements of the ship design process. This form must satisfy the design objectives and constraints of the problem in terms of performance, arrangement, safety, cost etc. The fairness of the hull form will be required to improve hydrodynamic performance, producibility characteristics and aesthetic properties.

The traditional solution of the fairing problem is the use of physical splines and weights which was introduced in the 18th century and has been successfully used for many ship types. This method is based on the successive fairing of ship lines on three different planes in an iterative manner. Provided that the designer has sufficient experience and time, the resultant form should have three-dimensional fairing characteristics. The process has no objective measures and the fairing characteristics of the resultant form greatly depends on the designer's experience and ability. Moreover, excessive time and experienced personnel will be required.

However, in the preliminary design stage there may be several alternatives to be investigated for further elaboration in a very limited time schedule. Furthermore, modern performance analysis procedures require more accurate hull form definitions.

This problem, i.e., the development of several accurately defined fair hull forms in the preliminary design stage necessitates the use of computer aided design methodologies.

Fairing is a part of most ship design packages commercially available today. In many cases the fairing procedures are based on interactive routines where the designer visually observes the form and interactively modifies it until satisfacatory fairness is achieved. Alternatively the designer is presented with curvature plots which will help him identify the regions of unfairness. These procedures can be seen as the computerised version of the manual fairing method and hence suffer the same drawbacks, i.e., the need for excessive time and experienced personnel.

The designer clearly needs automated procedures in which the fairness is defined in an objective manner and achieved within the boundaries of the design problem. Hence, this thesis attempts to develop automated fairing procedures to be used in the preliminary design stage. Three novel computational fairing procedures are

developed in the context of this thesis. The efficiency and the applicability of these novel procedures are shown in realistic ship hull form fairing problems. The basic principles and limitations of these methods are briefly described in the following paragraphs.

The first proposed approach is based on B-spline approximating (or fitting) of ship lines and hull surfaces, and referred to as *forward fairing of ship hull forms*. This process is carried out in an iterative manner until satisfactory fairness is achieved. This approach closely represents the manual fairing procedure in which the designer deals with each design curve separately and the global fairness is assumed to be achieved when all the curves on three different planes are maintained. The difference stems from the fact that the sum of the squares of curvature calculated at offset points are introduced as an indicator of fairness, and hence hull forms can be evaluated in terms of a numerical criterion. The main problem of this computational fairing approach is due to the deviation of offset points in each iteration. In many cases many iteration are needed and cumulative deviations may result in final forms too different from the original form to be acceptable. However, this problem may be avoided by using an alternative approach of B-spline fitting or by using affine transformation to the final form.

The second alternative fairing procedure is based on modification of curvature curves and surfaces, and named as *inverse fairing of ship hull forms*. Since the fairness is closely related to curvature, fair ship lines and hull surfaces can be obtained by modifying the curvature. Several techniques may be used to represent the curvature curves or surfaces. These techniques range from simple interpolating polynomials to complex NURBS surfaces. Applications indicate that representation of curvature curves or surfaces by mathematical spline curves or surfaces can be used to improve the fairness of ship hull forms with acceptable penalty in closeness to the original form.

Ultimately, the fairing of ship hull forms is formulated as an optimisation problem as the main goal of an automated fairing procedure is to hide all of the numerical details of the fairing process from the designer. This *optimisation approach for fairing of ship hull forms* is based on a non-linear direct search method in which the fairness of the hull surface is the objective function to be optimised subject to geometric constraints such as position and normals. One of the main concerns of the designer in a fairing problem is not to deviate too much from the original form in order not to degrade performance characteristics already obtained. Therefore, a closeness constraint may be used to ensure that the deviations between the parent and variant designs are not excessive. Additional constraints such as surface area or hull volume can also be defined. The results of the optimisation process indicate that provided that the designer can specify his objectives and constraints clearly, the procedure will generate fair hull forms which satisfies the constraints of the design problem.

The developed procedures have been applied to actual ship lines and hull forms to prove the capability and flexibility of the methodologies in practical problems.

TEKNE FORM YÜZEYLERİNİN BİLGİSAYAR DESTEKLİ DÜZGÜNLEŞTİRİLMESİ

ÖZET

Bu tezde gemi tekne formlarının düzgünleştirilmesi problemi tanıtılmakta ve ön dizayn aşamasında tekne formlarının düzgünleştirilmesi için kullanılabilecek yeni yöntemler sunulmaktadır. Bu yöntemler karmaşıklık açısından büyük farklılıklar gösterebilmektedir. Örneğin, problem basit olarak tekne formlarını uygun dereceden bir B-spline yüzeyi ile temsil etmek şeklinde tanımlanabileceği gibi çok daha karmaşık olarak bir nonlineer optimizasyon problemi olarak da tanımlanabilir.

Tekne formlarının üç boyutlu düzgünlüğü hidrodinamik performans ve üretim kolaylığı açısından mutlaka sağlanması gereken bir özelliktir. Bu amaçla kullanılan konvansiyonel yöntem olan fiziksel tirizler ve ağırlıklar 18. yüzyıldan beri başarı ile uygulanmaktadır. Bu yöntemde üç boyutlu tekne formu üç ayrı düzlemde iki boyutlu dizayn eğrileri ile temsil edilir ve iteratif bir tarzda uygulanan düzgünleştirme işlemi sonunda üç boyutlu düzgün bir form elde edilebilir. Yeterli zaman ve deneyimli uzman bulunması durumunda bu yöntem oldukça başarılıdır. Yöntemin temel dezavantajlarından biri düzgünlük kriterinin uzmana bağlı olarak değişmesi ve aynı probleme çok farklı çözümler üretilebileceği gerçeğidir.

Günümüzde yaygın rekabetin hüküm sürdüğü bir ortamda gemi ön dizaynı çok sayıda alternatifin kısa zaman dilimleri içinde geliştirilmesini ve güvenilir gelişmiş performans analiz yöntemleri kullanarak incelenmesini zorunlu kılmaktadır. Bu tür performans analiz yöntemleri hassas ve detaylı tanımlanmış tekne formları gerektirmektedir. Bu durumda dizayner ön dizayn aşamasında kısa zaman dilimlerinde hassas ve detaylı olarak tanımlanmış düzgün alternatif tekne formları geliştirmek durumundadır.

Tekne formu düzgünleştirme prosedürü halen çeşitli bilgisayar destekli gemi dizayn paket programlarında yaygın olarak kullanılmaktadır. Genel uygulamalar interaktif olarak gerçekleşmekte ve dizaynerin dizayn form eğrilerini interaktif olarak düzgünleştirmesi gerekmektedir. Genellikle, eğrilik eğrisi veya yüzeyi kullanılarak formun ne derece düzgün olduğu dizaynere sunulmaktadır. Bu yöntemler konvansiyonel yöntemin sahip olduğu tüm dezavantajlara sahiptir. Yani, objektif bir kriter yoktur, deneyimli personel gereklidir ve işlem zaman alıcıdır.

Ön dizayn aşamasında çok sayıda hassas ve detaylı tanımlanmış düzgün tekne formları geliştirmek durumunda olan dizayner otomatik olarak düzgünleştirme işlemi yapacak prosedürlere gereksinim duyacaktır. Bu tezde bu amaca yönelik olarak ön dizayn aşamasında kullanılabilecek üç yeni tekne form düzgünleştirme yöntemi geliştirilmiştir. Geliştirilen tüm yöntemlerin başarısı ve uygulanabilirliği gerçek tekne form ve eğrilerinin düzgünleştirme problemlerine uygulanarak gösterilmektedir.

Geliştirilen ilk yöntem *tekne form eğri ve yüzeylerinin doğrudan düzgünleştirilmesi* olarak adlandırılmakta ve yöntemin esası, konvansiyonel tiriz ve ağırlık metodundan esinlenerek düzgünleştirmenin insan müdahelesi olmadan bilgisayar ortamında gerçeklenmesine dayanmaktadır. Tekne formlarının üç boyutlu düzgünleştirme problemi iki boyuta indirgenmekte ve tekne form dizayn eğrileri uygun dereceden B-spline eğrileri ile temsil edilmekte ve yeterli düzgünlük sağlanana kadar bu işlem iteratif olarak farklı düzlemlerde devam etmektedir. İterasyonlar sonucu her üç düzlemde yeterli yaklaşıklık elde edildiğinde üç boyutlu düzgün bir form elde etmek mümkün olabilmektedir. Bu yöntemle üretilen formların düzgünlüğü nümerik bir değer olan düzgünlük sayısının değeri ile değerlendirilmektedir. Düzgünlük sayısının tarifi düzgünlüğün saptanmasında en güvenli kriter olan eğrilik eğrisi yardımıyla, eğrilik eğrisini iyi bir yaklaşımla temsil ettiği varsayılan ikinci türev değerlerinin karelerinin toplamı olarak yapılmaktadır. Sonuçta elde edilen form, iterasyon sayısına bağlı olarak orijinal formdan farklı olabilmektedir. Bu tür durumlarda uygulanacak bir afin dönüşüm ile orijinal formun sahip olduğu form ve performans karakteristiklerini korumak mümkün olabilecektir.

Geliştirilen ikinci yöntem *tekne formlarının geriye dönük düzgünleştirilmesi* olarak adlandırılmaktadır. Tekne formlarını düzgünleştirmek üzere eğrilik yüzeyleri uygun matematiksel yüzeyler ile temsil edilmekte ve geri dönüşüm ile tekne formu elde edilmektedir. Tekne formlarının düzgünlüğünü en iyi temsil eden kriterler eğrilik yüzeyine dayandığı için bu yüzeylerin düzgünleştirilmesi ile elde edilen yeni yüzeyler de orijinal yüzeye göre çok daha düzgün olmaktadır. Eğrilik yüzeylerini temsil etmek üzere çok farklı matematiksel ifadeler kullanılabilmektedir. Eğrilik yüzeylerini temsil etmek üzere avantajları ve başarısı kanıtlanmış B-spline teknikleri seçilmiştir. Kullanılan matematiksel ifadelerin eğrilik yüzeyini yakın temsili sağlanarak düzgünleştirilmiş yüzeyin orijinal yüzeye yakınlığı sağlanabilmektedir.

Geliştirilen son yöntem *tekne formlarının düzgünleştirilmesi için optimizasyon yaklaşımı* olarak adlandırılmıştır. Tekne formlarının düzgünleştirilmesine yönelik çalışmalarda temel amaç tüm ara işlemlerin dizaynerden gizlenmesi olduğundan bu amaca yönelik olarak tekne form eğrilerinin düzgünleştirilmesi işlemi bir non-lineer optimizasyon problemi olarak formüle edilmektedir. Bu formülasyonda dizayn değişkenleri olarak tekne formunu oluşturan ofset noktaları, amaç fonksiyonu olarak ise tekne yüzeyinin düzgünlüğünü belirleyen eğrilik yüzeyine bağlı fonksiyonlar kullanılmaktadır. Değişik geometrik veya performansa bağlı kısıtlar kullanılarak problemin çözüm alanı sınırlanabilmektedir. Bu formülasyon ile gerçekleştirilen çalışmalar amaç fonksiyonunun ve kısıtların doğru seçilmesi durumunda düzgün tekne formlarının kolay ve hızlı bir şekilde elde edileceğini göstermektedir.

Geliştirilen yöntemler gerçek tekne form eğri ve yüzeylerine uygulanarak pratik problemlerdeki uygulanabilirlikleri kanıtlanmaktadır.

1. INTRODUCTION

The first level of the ship design process is to determine the main dimensions and produce a fair hull form with the desired form characteristics. Hence, fairing of ship hull forms is one of the main tasks in a ship design process. Since the great majority of hull forms are free form-empirical shapes rather than mathematical forms, a fairing process is generally required in all stages of hull form design. The fairing problem may have different characteristics depending on the design stage and the fairness of a hull form should be improved as the design is developed.

During the early stages of the design process there is little data available to produce a fair hull form and the designer often initiates the process with a rough sketch, a successful previous design, or few form parameters. Prediction of performance characteristics of ships has tended in the past to be based on simplified analytic approximations to the hull form and hence three-dimensional fairness, accurate enough to carry out basic hydrostatic and hydrodynamic calculations was sufficient in the preliminary design stage. This was necessary in order to reduce the complexity of the analysis to a practicable computational level. However, today ship designers require more accurate predictions on the performance characteristics of their designs and modern computer processing power now available enables more extensive analyses to be carried out. Therefore, the designer needs to have a precisely defined fair hull surface even in the first stages of ship design process.

Today most modern shipyards use improved production technology based on pre-fabricated sections and blocks, which require high degree of accuracy. Rectification of errors at the production stage will be costly and may well build residual stresses into the hull. Hence it is of vital importance to have an accurately defined three-dimensional fair hull surface before the construction stage is reached.

Fairing of ship hull forms is, traditionally, achieved by battens and weights in which excessive time and experienced personnel are required. This process is based on the assumption that a three dimensional hull form can be represented by three two-dimensional orthogonal planes representing section lines, waterlines and buttock lines. Therefore, the complex problem of three-dimensional fairness is reduced to three sets of two-dimensional fairing problems. Each ship line on three planes can be faired independently, however in order to achieve three-dimensional fairness these

curves must be compatible, i.e., an iterative process, which would require time, and experience is required. Obviously, the manual graphical fairing method employs no objective criteria and uses a visual judgement that incorporates the experience of the person who is performing the fairing. Indeed, it is quite common for two-experienced loftsmen, starting with the same initial data to produce two different sets of subjectively fair hull forms.

The advent of computer in the shipbuilding, since the early fifties, changed the scope, style, and methodology of mathematical ship hull form definition in a dramatic way. Today modern shipbuilding around the world can no longer be imagined without intensive use of computers and computational methods of ship geometry definition. The production of ship parts is performed by numerically controlled machine tools, which require accurate computer based ship geometry information as input.

Today, modern computing technology has provided numerous software systems to develop ship geometry design and production information which are now a common feature in many shipyards and design offices. This technology is particularly useful in the preliminary design stage of ships where a large number of alternatives need to be developed and analysed. These software systems generally employ two distinct approaches, i.e., form creating and form fairing.

The form creating approach requires the designer to plot a defining polygon (or control net) which roughly approximates the desired curve (or surface). A B-spline curve (or surface) of suitable order is automatically plotted on screen and the designer can judge whether he is satisfied with the curve. It is not clear what type of measures will be used to judge the form apart from the designer's eyes. Obviously, there are many objectives and constraints to be satisfied in a hull form design process, hence it is doubtful that the designer can control all these aspects in an interactive process.

The form fairing applications, on the other hand, are based on interactive modification of design curves with the aid of curvature plots. The coupled sign variations on the curvature curve are indicators of the erroneous points on the original curve. The designer eliminates the undesirable features of the curve and relocates these points and checks the new corresponding curvature plot. This process is repeated until a curve of acceptable fairness is achieved. The process is based on visual observations and the effectiveness of the process depends mainly on the skills of the person controlling the process. This approach gives to designer complete

control of the process, however, it is also doubtful in many cases that all erroneous points can be identified visually.

Interactive techniques based on form creating or form fairing are often praised as the optimal tool to create ship hull forms. Yet this interactive control has several drawbacks, some of which are listed below:

- The procedure makes extensive use of human resources, which tend to be more and more expensive and minimal use of computer, which gets cheaper.
- It is assumed that the user can identify the data points to be modified as well as the amount of correction to be made where may be doubtful in many cases.
- Fairness is a global feature of the curve, whereas interactive changes are local. The sequence of local modifications may not always lead to a fair curve, which is not much deviated from the original curve.
- This approach may be viewed as the computerised version of conventional manual fairing method and hence experienced personnel and excessive time would be required.
- This approach is not compatible with an automated hull form design process because of its interactive nature.

Ideal solution of the problem of fairing of hull forms can be achieved by using computers extensively with no human interaction, which is clearly desirable in CASD environment in order to increase the automation in the design process. Although many researchers have studied on this subject the optimum solution of the surface-fairing problem can not be considered as solved yet. Various algorithms are proposed such as local fairing by knot removal, local and global fairing based on energy minimisation etc., but all requiring human interruption.

Three novel methods, for fairing ship hull forms in an automated manner, are developed and presented in this thesis. It is presumed that an initial hull form which satisfies the objectives and constraints of the design problem in terms of arrangement, space, powering, stability, seakeeping etc. is available and the designer needs to modify this form in order to improve fairness characteristics for detailed calculations or production purposes. The details of the developed procedures are given in Chapters 5, 6 and 7.

First, *Chapter 2* presents a brief review of hull form design procedure in the preliminary design stage. It is shown that the safest method to generate a fair hull form, which satisfies the design requirements, is to distort a parent hull with satisfactory performance characteristics. The details of various linear and non-linear variation methods are described and the advantages and drawbacks are discussed in this chapter. All techniques are applied to a typical fishing vessel form based on well-known BSRA trawler series (**Pattullo and Thomson (1965)**). The variation or distortion of a parent hull will inevitably require a fairing process even if the parent hull has satisfactory fairing characteristics. (**Narlı et al, 1999**)

Chapter 3 briefly describes the mathematical models used in representing ship hull forms. It is shown that a wide range of mathematical models from simple polynomials to complex non-uniform rational B-splines is available. All these models are shown to have drawbacks, which limit their application in practical hull form representation problems. Typical applications are also presented.

Since the fairness of a curve or surface is related to its geometric characteristics the basic geometrical properties of curves and surfaces are discussed in *Chapter 4*. It is shown that the curvature of a curve or surface is the most important geometric parameter in a fairing problem. Several fairing measures based on mean or Gaussian curvature are described and applied to typical ship lines.

Two forward fairing procedures are introduced in *Chapter 5*. These procedures are based on the approximation of a given form plan by mathematical functions such as polynomials, cubic splines, Bézier curves and B-splines. The experience has shown that B-spline curves provide the best alternative for approximating ship lines. As the order of the B-spline curves is increased smoother curves can be obtained but the deviation from the original data points will increase. This feature of B-spline curves has led to the development of practical forward fairing procedures, which are described in detail in this chapter.

The first forward fairing procedure is based on successive application of B-spline curves for ship waterlines and section lines in an iterative manner until a satisfactory degree of fairness is achieved. The offset points of each waterline in the lines plan is assumed to be the polygon vertices for the B-spline representation and the modified offsets are then obtained from the B-spline curve. These offsets then become the polygon vertices for the corresponding section lines. The process is successively applied until satisfactory levels of fairness and deviation from original offsets are obtained. This algorithm is applied for fairing the lines of a distorted mathematical

hull form and the parent hull of a well-known high-speed displacement hull form series, and a naval corvette form. (**Narlı, 1997**)

The second forward fairing procedure adopted in this study is similar in its iterative manner with the first approach. Here, the successive application of B-spline curves starts with section lines and continues with B-spline curve fit for waterlines. The B-spline curve fit process provides the polygon vertices, which yield a B-spline curve representation of the original offset points. By selecting suitable number of polygon vertices one can increase the degree of fairness with increasing penalty on closeness to original points. If the number of polygon vertices is the same as the offset points, the generated B-spline curve from this polygon fits through all offset points. This feature of B-spline curve fit satisfies the closeness requirement, as there is no deviation from original points. However, the fitted curve may develop unwanted oscillations. Therefore, a compromise between the degree of fairness and closeness should be sought by changing the number of polygon vertices. This process is also applied for fairing the lines of a distorted mathematical hull form and the parent hull of a well-known high-speed displacement hull form series. (**Narlı and Sarıöz, 1998**)

It is a well-known fact that the best indicator of fairness of a curve or surface is its corresponding curvature plot, and they provide a powerful means for curve and surface interrogation. Thus, by improving the curvature plot of a curve or surface it is possible to increase fairness characteristics. An inverse fairing approach based on this fact is presented in *Chapter 6*. Several mathematical techniques are used to represent curvature plots and the original offsets are obtained by double integration. The details of this inverse fairing procedure is presented and applied for fairing the lines of a distorted mathematical hull form.

In *Chapter 7*, the fairing problem is defined as an optimisation procedure in which the objective function is a fairness measure to be optimised subject to constraints related to geometric or performance characteristics of the hull form. It is shown that the problem is highly non-linear and a non-linear optimisation technique based on Hooke and Jeeves algorithm is used. Although excess computational work is required, the optimisation techniques are shown to be practical with the availability of fast computers and produces high quality surfaces of hull forms. The details and the efficiency of this algorithm is presented by applying for fairing the lines of a distorted mathematical hull form.

While the objective is to obtain three-dimensional fairness it is reasonable to assume that designer should preserve the hull form characteristics which may be selected for the best hydrodynamic performance. Therefore, there should be certain limits on

variation from the original lines plan. These limits are expressed in terms of conventional naval architectural parameters, i.e., the block coefficient, longitudinal centre of buoyancy etc.

In order to assess the efficiency and flexibility of the developed procedures, a test case is provided in *Chapter 8*. The selected form is based on a high-speed semi-displacement vessel. This form is the result of a preliminary design study and hence does not possess the necessary fairing qualities for further design calculations and production purposes. The aim of this chapter is to improve the fairing properties by using three novel procedures within the geometric limits of practical design and to show the relative merits of these fairing procedures.

Finally, *Chapter 9* presents the principal conclusions and recommendations.

All test curve and surface definitions used in the applications and numerical results are presented in Appendices.

2. A REVIEW OF SHIP HULL FORM DESIGN PROCESS

One of the fundamental tasks in the hull form design process is development of a three-dimensional hull form which satisfies the specified design requirements. Since the hull form has drastic implications on most of the techno-economic performance characteristics, it should be specified clearly in the early stages of ship design.

Hull form design stage has the primary objective of meeting the desired overall shape characteristics, often expressed in terms of numerical form parameters, related to the principal curves defining the hull form. The complete set of hull lines, i.e., body plan, waterlines etc., defining the form should be generated with sufficient accuracy for subsequent design calculations. The resulting lines plan should meet all explicit form specifications and should not violate any of the implicit, intuitive roles of hull form design with respect to fairness.

A rational ship hull form design methodology, as shown in **Figure 2.1**, will require the establishment of design objectives, development of feasible alternatives, and evaluation of each alternative against the objectives in order to choose the best alternative as the final design. The design objectives may include; minimum powering requirements, maximum internal volume, maximum stability, and maximum producibility which may have to be satisfied simultaneously in many cases. The second stage is the development of alternative designs which can potentially meet the design objectives. A large number of these evaluations are required, so the capability to automatically evaluate the performance of a candidate hull design without human intervention is crucial for the success of the design process. This process must be carried out in a systematic manner by using naval architecture form parameters which are varied with other parameters being constant.

The designer, in general, has three different choices to create a three dimensional hull form which satisfies the objectives and constraints of the design problem;

- Interpolation from series of parent designs (standard series approach)
- Lines creation from geometric hull form parameters (form parameter approach)
- Lines creation by distortion of a parent hull (variation and distortion of an existing design)

These techniques are briefly described in the following sections. (**Nowacki et al, 1995**)

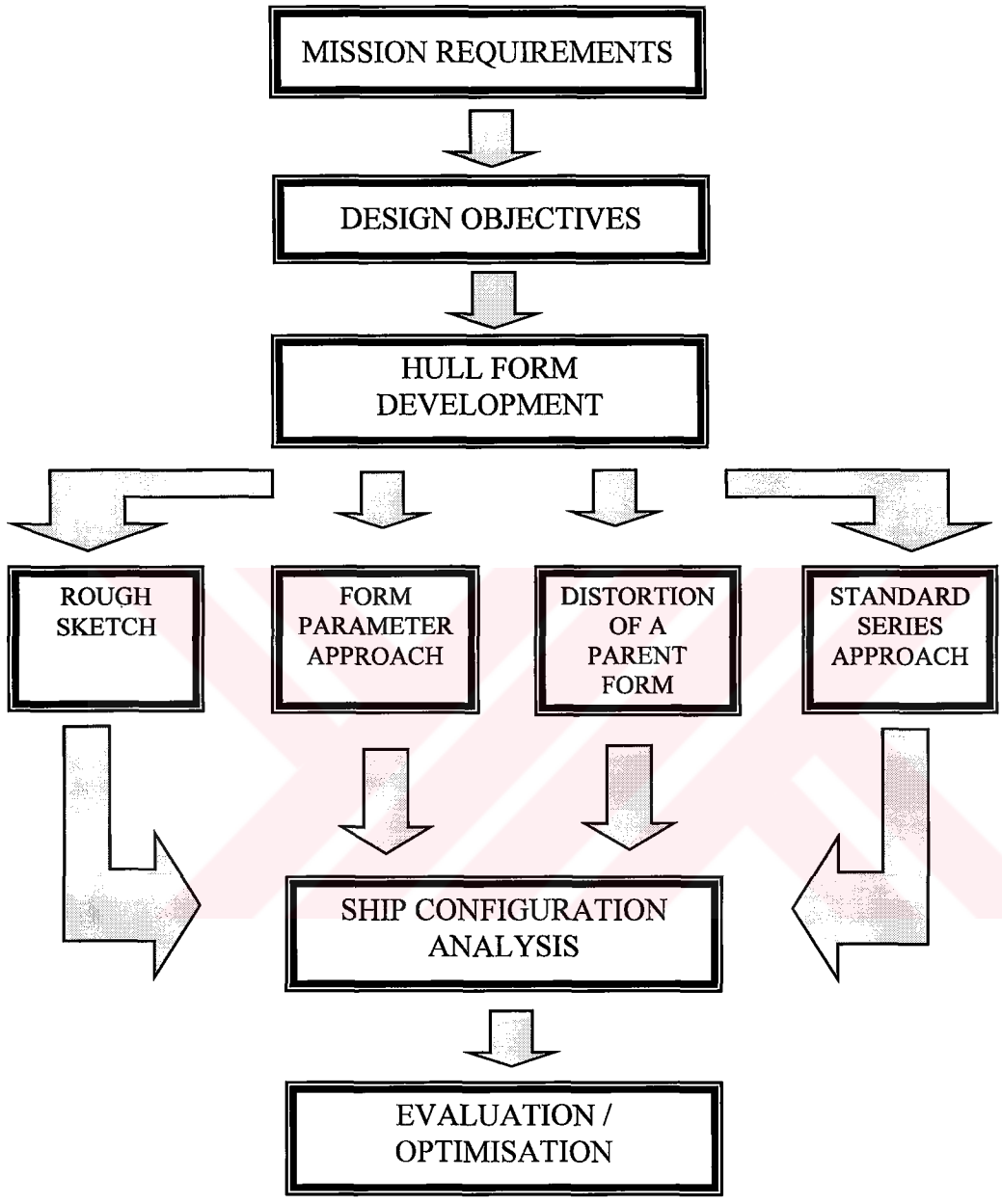


Figure 2.1. General structure of a hull form design methodology.

2.1. Standard Series Approach

The basic idea of the standard series approach is simply to interpolate a desired new hull form within the variety of designs available of systematic hull form series. Since most systematic series are developed for resistance and/or propulsion model tests, this may offer the advantage of predicting powering performance directly from the test results.

The standard series method is attractive due to its simplicity. Obviously, if one starts from one of the successful variants of the standard series, the outcome of the process is the development of a new design with favourable hydrodynamic characteristics. However, standard series are based on specific parent forms optimised for a specific mission. Therefore, it is rarely possible to apply a standard series based form directly for a new design.

Standard series are generally generated from a parent design or series of parent designs. In most series, variation and distortion techniques are applied to parent forms in order to produce series of hull forms. It is obvious that only some simple variations in hull form with respect to the proportions of the main dimensions, block coefficient, and in some cases, the centre of buoyancy are included in the standard series. The range of variation in the series is limited to interpolation, therefore, extrapolation should be avoided.

One of the drawbacks of this method is that, few form parameters can be varied independently. The designer has to accept the outcome for the associated dependent variables. Further, the parent forms can not be always sufficiently similar or the variations may not be closely spaced to rely on the fairness of the interpolated lines.

Development of standard series is expensive and time consuming. Therefore, it is not possible to generate standard series for the new ship types as they emerge and the standard series quickly become outdated.

Some of the well-known series are given in **Table 2.1.**

Table 2.1. Typical standard series.

	Ship type	Range of Hull Form Parameters						Documentation
		$L/\nabla^{1/3}$	L/B	B/T	C_B	C_P	LCB	
Taylor (1915)	Twin Screw Cruiser	4.85-11.26		2.25-3.75	0.444-0.796	0.48-0.86	0	SAC, Offset tables, Profiles
Series 60 (1950)	Single Screw Merchant	4.56-7.49	5.5-8.5	2.5-3.5	0.60-0.80	0.614-0.805	2.5A-3.5F	SAC, Offset tables, Profiles
BSRA (1961)	Single Screw Merchant	4.2-6.3	5.33-8.37	2.12-3.96	0.55-0.85	0.570-0.852	3.0A-3.5F	SAC, Offset tables, Profiles
Series 64 (1965)	High Speed Round Bilge	8.04-12.40	8.454-17.734	2.0-4.0	0.35-0.55	0.63	6.6A	SAC
BSRA Trawler (1965)	Single Screw Trawler	4.35-5.10	4.3-5.8	2.0-3.5	0.53-0.63		2.90A-1.09F	SAC, Offset tables, Profiles
WEBB Trawler (1966)	Single Screw Trawler	3.85-5.22	3.20-5.75	2.3	0.42-0.53	0.55-0.70		SAC, Offset tables, Profiles
MARAD (1987)	Single Screw Bulk Carrier	5.25-14.00	4.5-6.5	3.0-4.5	0.80-0.875	0.805-0.880	2.5F	Profiles, Offset tables

2.2. Form Parameter Approach

In this approach, hull form lines are generated from basic hull form parameters, hence the process does not require a basis design. These form parameters are normally related to the offsets at specific stations, slopes and curvatures at the ends of the curve, and integral properties like sectional area and the longitudinal centre of buoyancy. The basic ship curves are generated mathematically satisfying the specified values of these form parameters. Thus, the problem is basically a curve design problem.

The crucial first step of form parameter approach is the determination of initial set of form parameters for the principal curves of the hull form. By using the set of initial design requirements the designer will be required to specify the following hull form design curves

- Sectional area curve, (SAC)
- Design waterline, (DWL)
- Stem, stern and keel profile,
- Deck line

The starting point for the above design curves may be a rough manual sketch, an accurate lines plan of comparable parent design, or empirical guidelines.

Each basic design curve (SAC, DWL, ...) is designed from its own parameter inputs. These curves are assumed to be sufficient to develop the complete lines plan. In general, the design of a hull form starts with the layout of a sectional area curve which represents the distribution of sectional area A(x) below the design waterline for given ship dimensions. The area below this curve corresponds to the displacement of the underwater body. Distribution of the displacement and location of the centre of buoyancy are characteristic features of the design and influence decisively the hydrodynamic properties of the hull form. Therefore, in the design of sectional area curve, satisfying the desired curve form parameters is of maximum importance. Subsequently, from the sectional area curve, the underwater section areas A(x) at the stations are determined. The sections are laid out and waterlines and buttocks are examined to comply with section lines. The modifications should be achieved if necessary to obtain the best possible ship lines.

In **Figure 2.2** schematic curves of longitudinal distributions of the immersed sectional area, a(x), and design waterline, b(x), of a typical ship hull form are shown. This figure contains useful information concerning characteristics of the underwater hull geometry. The area under sectional area curve provides the displaced volume and its longitudinal centre, x_b, depicts the centre of buoyancy.

$$\nabla = \int_0^L a(x)dx \qquad x_b = \frac{\int_0^L a(x)xdx}{\int_0^L a(x)dx}$$

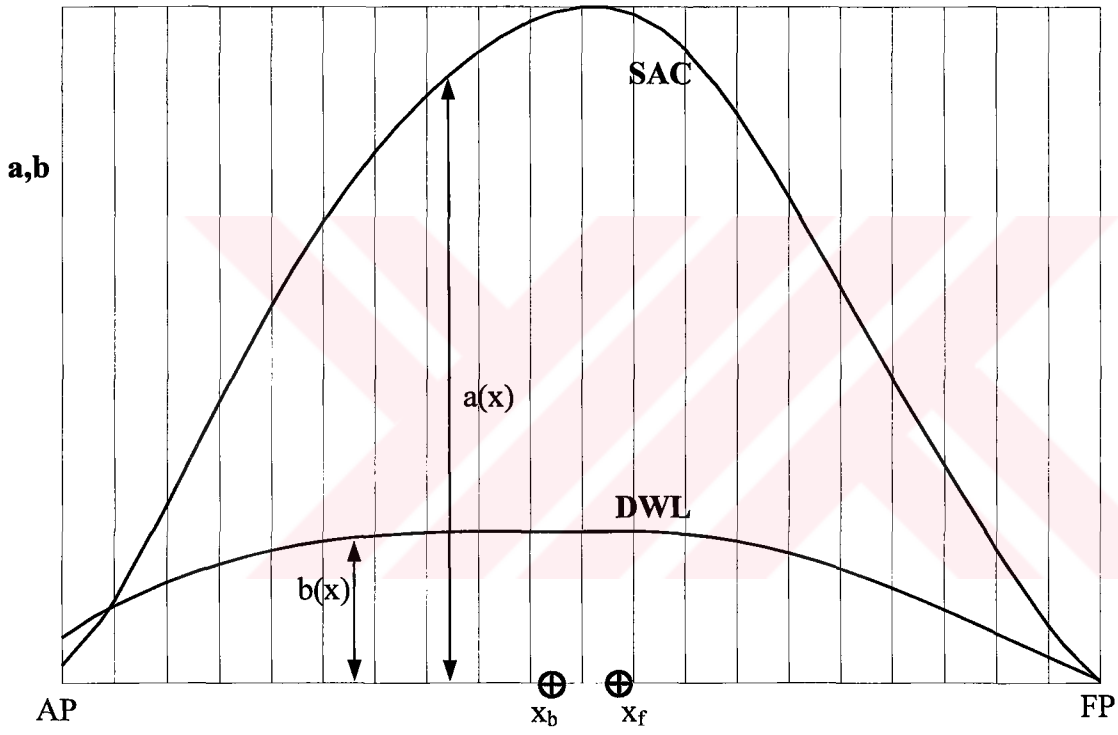


Figure 2.2. Sectional Area (SAC) and Design Waterline (DWL) curves for a typical ship hull form.

Similarly the area under the beam distribution provides the design waterplane area, A_{WP}, and its longitudinal centre, x_f, is the centre of flotation.

$$A_{WP} = \int_0^L b(x)dx \qquad x_f = \frac{\int_0^L b(x)xdx}{\int_0^L b(x)dx}$$

The procedure of hull form design using form parameter approach is described in **Figure 2.3.** This procedure suggests that an early hull form may be defined by using a set of design curves such as the sectional area curve, design waterline, keel contour, deck outline and sheer. However, an initial design based merely on a few basic curves will rarely meet the designer's full intention right away and to examine other aspects of the form the designer must look at other views and sections just as in manual design. (**Nowacki et al, 1995**)

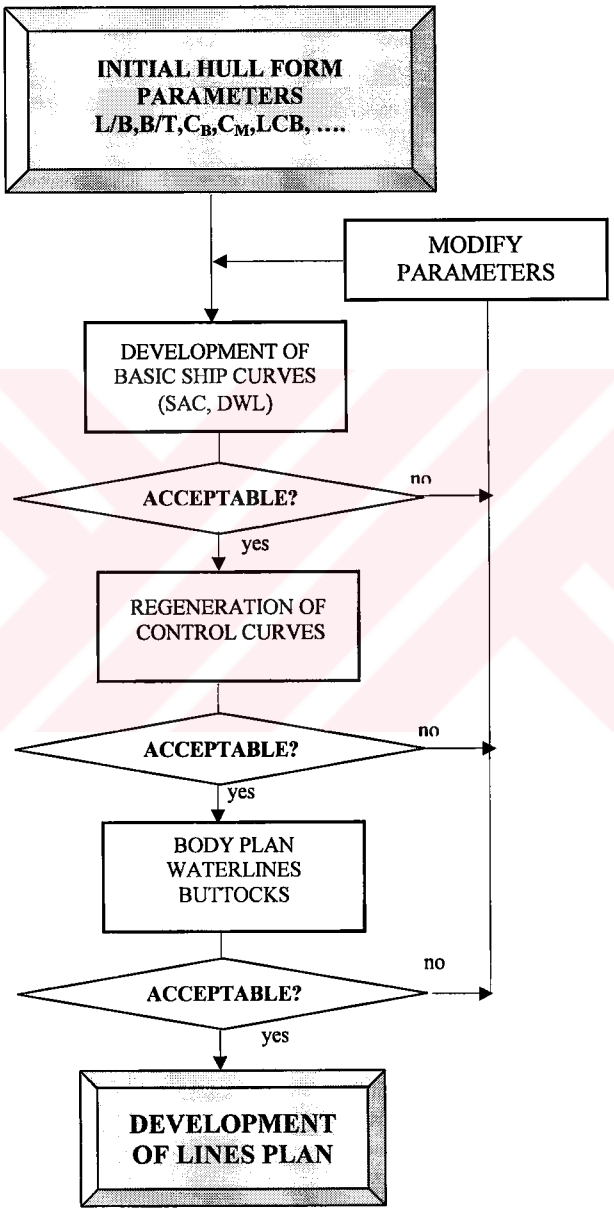


Figure 2.3. Form parameter approach.

2.3. Distortion of a Parent Hull

Because of the risk involved with starting the hull form design process from scratch, the designer most frequently initiates the design process with a parent form which has satisfactory performance characteristics. However, this parent form may not be suited completely to new design requirements and certain variation may be needed. With the aid of suitable mathematical operations, a slight modification of parent design produces the new hull form. The pursued goal of these procedures is basically to change any desired hull form parameters while keeping others constant. However, in practice, due to the complex interdependencies among the hull form parameters, the outcome of the process is the limitation of varying few parameters while keeping the others fixed or accepting the resulting changes without the possibility of intervention.

These changes usually achieved by distortion transformations, which can either be linear or non-linear. Some cases that require variations from the parent hull are

- Increased capacity of the new design may require in the size of the vessel,
- Main dimensions may need to be increased e.g., to improve stability,
- Fullness of the design may need to be changed,
- Longitudinal distribution of buoyancy may be altered.

The changes in main dimensions can be simply achieved by multiplying the offset values by corresponding constant expansion or contraction factors. This does not affect the hull form characteristics. Several techniques have been developed to modify the hull form characteristics of a given parent form. The earlier applications are based on the linear distortion of the sectional area curve in order to change the position of LCB or fullness. The modified form is obtained by shifting the ordinates of the parent form along the ship length. More recent methods such as the shape averaging method use three-dimensional hull forms instead of two-dimensional design curves.

The application of distortion techniques is limited by the type and geometric characteristics of parent hull form. For example, a hull with a bulbous bow cannot be produced from a non-bulbous form. Similarly, it would not be feasible to change the fullness of a bulk carrier form more than few percent. For many ship types the upper limit of variation of block coefficient is about 10%, however for full forms this may be reduced to only 2-3 %. The longitudinal centres of buoyancy and flotation may be

changed by ±2 %. Best results will naturally be obtained if the new form differs slightly from the parent design.

All linear and non-linear techniques are briefly described **Narlı et al (1999)** and the applicability and efficiency of the methods are presented with applications to a typical fishing vessel form based on well-known BSRA trawler series. (**Pattullo and Thomson, 1965**).

2.3.1. Linear Distortion Techniques

2.3.1.1. Swinging the Sectional Area Curve

This method enables to change LCB position by keeping the fullness constant. The sectional area curve of the parent ship represented by solid line ABC and the derived curve represented by the dotted line are illustrated in **Figure 2.4.**

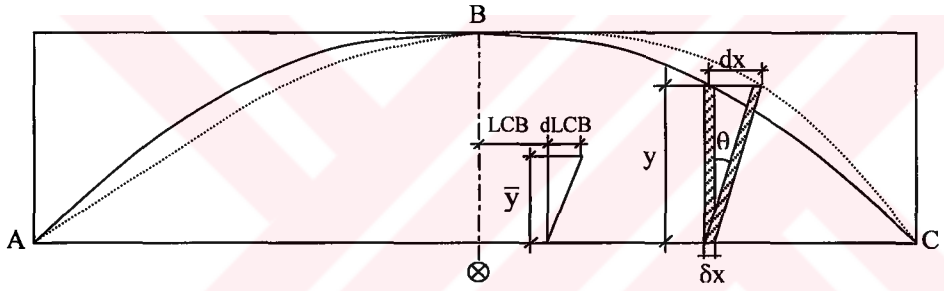


Figure 2.4. Swinging the sectional area curve to change LCB position.

Let us consider a thin vertical strip of δx. The area of the element is δxy. Longitudinal transfer of moment of the strip due to swinging is

$$dM = \delta x \, y \frac{1}{2} dx = \frac{1}{2} y^2 \tan\theta \, \delta x$$

$$M = \tan\theta \frac{1}{2} \Sigma(y^2 \, \delta x)$$

Vertical Moment : $\frac{1}{2} \Sigma(y^2 \, \delta x) = A \, \bar{y}$

Longitudinal Moment : $A \, \bar{y} \tan\theta = A \, dLCB$

$$\tan\theta = \frac{dLCB}{\bar{y}}$$

where

θ : required angle of shift for adjusting the LCB position

A : total area under the sectional area curve ABC, indicating fullness

$\bar{y}$: the vertical centre of buoyancy

dLCB : required change in LCB position

Once the new positions of transverse sections are determined the modified offsets can be obtained directly from the waterlines plan of the parent form. An example of this variation can be seen in **Figure 2.5,** where the longitudinal position of the centre of buoyancy is shifted aft by 2%. The thin and thick lines correspond to the parent and variant forms, respectively. The results presented in **Table 2.2** indicate that the method enables the designer to change the longitudinal position of the centre of buoyancy while the fullness is kept constant. However, it should be noted that the properties of the waterplane area (C_{WP} and LCF) will also change in an uncontrolled manner. Furthermore, there is no control over the parallel middle body (or maximum section) in the derived form as can be seen in **Figure 2.5.** The definitions of the parent and variant hull forms obtained by swinging the sectional area curve can be found in Appendix B.

Table 2.2. Main particulars of the parent and variant form obtained by swinging the sectional area curve.

	Parent Form	**Variant Form**
Length between perpendiculars (L_{BP})	30.480 m	30.480 m
Beam (B)	6.100 m	6.100 m
Draught (T)	2.515 m	2.515 m
Displacement (Δ)	244 m^3	244 m^3
Block coefficient (C_B)	0.522	0.522
Midship area coefficient (C_M)	0.887	0.887
Prismatic coefficient (C_P)	0.588	0.588
Waterplane area coefficient (C_{WP})	0.753	0.744
Longitudinal centre of buoyancy (LCB)	% 1.34 L (aft)	% 3.35 L (aft)
Longitudinal centre of flotation (LCF)	% 4.06 L (aft)	% 5.22 L (aft)

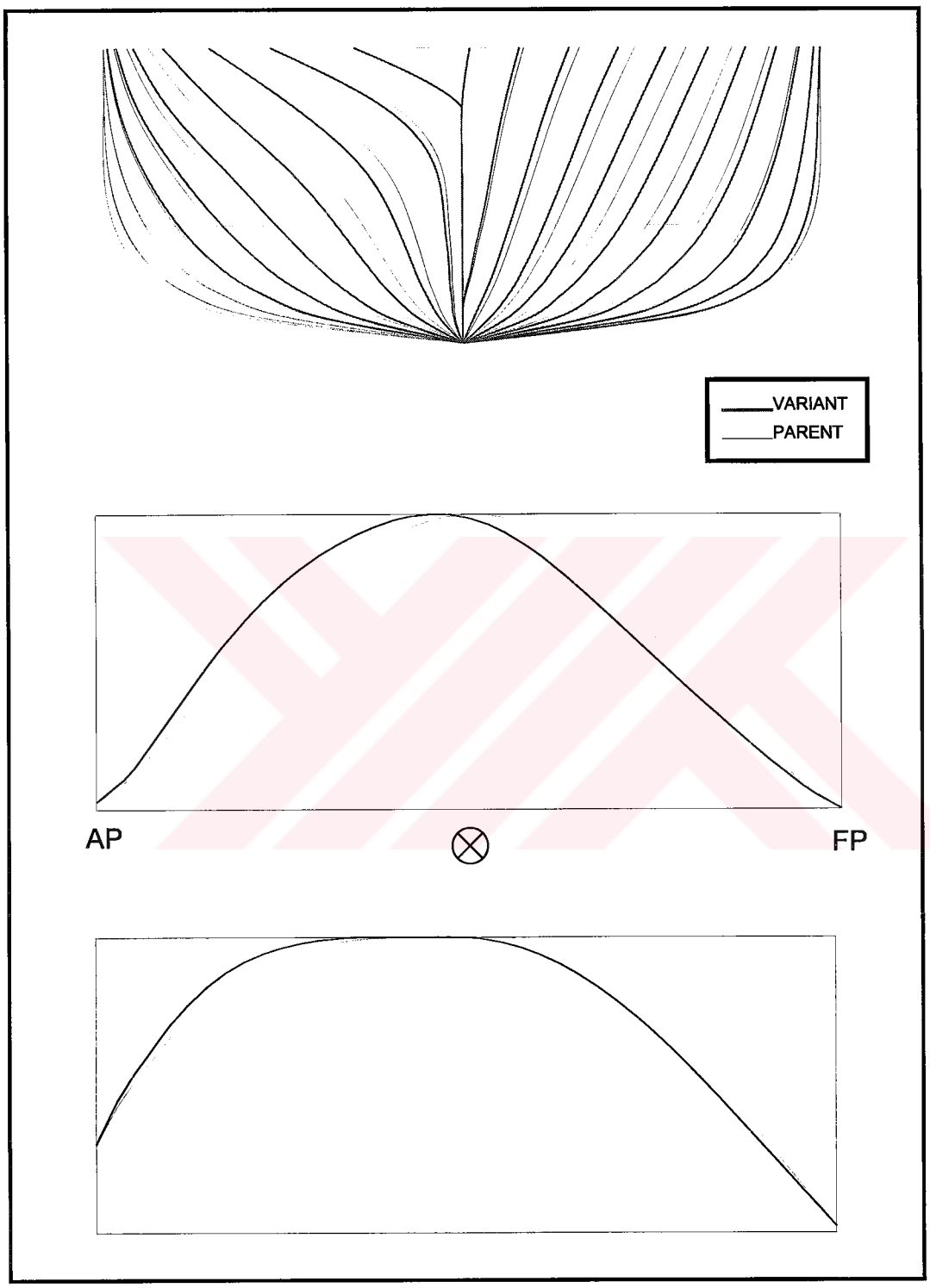


Figure 2.5. Body plan, sectional area and loaded waterline curves for the parent and variant form derived by swinging the sectional area curve.

2.3.1.2. One-Minus Prismatic Method

The basis of this method is to adjust the sectional area curve of the basic ship form by contracting or expanding the entrance and run and reducing or increasing the parallel middle body length as necessary. New offsets can be obtained directly from the parent design (see **Lackenby (1950)**). The new form is subjected either to expansion or contraction depending on the desired form characteristics. These adjustments are likely to influence some of the geometric particulars like LCB (longitudinal centre of buoyancy) position, C_P (prismatic coefficient) and the extent of the parallel middle body in both the fore and after bodies.

The curve in **Figure 2.6** represents the sectional area curve of the parent ship for one half of the body. For convenience, the terms given in this study are not separated for entrance and run, and valid for both halves of the ship. It is necessary to consider this half body and maximum sectional area ordinate as equal to unity. Therefore, the area under the curve becomes numerically equal to the prismatic coefficient of the half body and the added slice represents the change in C_P, indicated by δC_P.

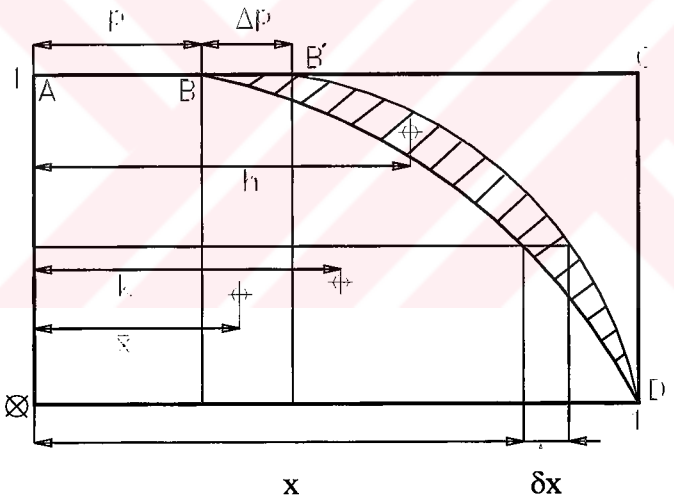


Figure 2.6. Geometrical derivation of shifting function.

According to **Figure 2.6** the linear shift of dimensionless sectional area ordinates (δx) is obtained by using the proportionality of the areas before and after the distortion procedure. Hence, δx is derived as follows:

$$\frac{BB'D}{BCD} = \frac{\delta x}{1-x} \qquad \Rightarrow \qquad \delta x = \frac{\delta C_P}{1-C_P}(1-x)$$

It can clearly be seen from **Figure 2.6** that the variation of C_P produces inevitable changes in the parallel middle body length (p), according to the modifications of the entrance and run, e.g., at x = p

$$\delta p = \frac{\delta C_P}{1 - C_P}(1 - p)$$

In order to change the total prismatic coefficient (C_P) and/or the longitudinal centre of buoyancy (LCB), the required changes in fore and afterbodies must be determined. This can be achieved by taking moments as follows

$$\delta C_{PF} = \frac{2[\delta C_P(h_a + LCB) + \delta LCB(C_P + \delta C_P)]}{(h_f + h_a)}$$

$$\delta C_{PA} = \frac{2[\delta C_P(h_f - LCB) - \delta LCB(C_P + \delta C_P)]}{(h_f + h_a)}$$

where

C_P : prismatic coefficient of the parent form

δC_P : the required change in prismatic coefficient

LCB : the distance of the LCB in the parent form (positive forward)

δLCB : the required shift of the LCB

δC_{PF} : the change in forebody prismatic coefficient

δC_{PA} : the change in afterbody prismatic coefficient

h_f : centroid of the added area in the forebody

h_a : centroid of the added area in the afterbody

The exact values of levers h_f and h_a can be calculated by the following relations

$$h_f = \frac{C_{PF}(1 - 2\bar{x}_f)}{1 - C_{PF}} + \frac{\delta C_{PF}}{2(1 - C_{PF})^2}[1 - 2C_{PF}(1 - \bar{x}_f)]$$

$$h_a = \frac{C_{PA}(1 - 2\bar{x}_a)}{1 - C_{PA}} + \frac{\delta C_{PA}}{2(1 - C_{PA})^2}[1 - 2C_{PA}(1 - \bar{x}_a)]$$

where C_{PF} and C_{PA} are the prismatic coefficients for fore and afterbodies, and $\overline{x}_f$ and $\overline{x}_a$ are the centroids of the original fore and afterbodies. However, since δC_{PF} and δC_{PA} are not known h_f and h_a cannot be determined exactly and the second term may be ignored, i.e.,

$$h_f = \frac{C_{PF}(1-2\overline{x}_f)}{1-C_{PF}}$$

$$h_a = \frac{C_{PA}(1-2\overline{x}_a)}{1-C_{PA}}$$

This technique of form distortion is useful and relatively simple to apply but there are some restrictions, which are:

- The parallel middle body length and the prismatic coefficient cannot be varied independently,
- The prismatic coefficient of the fore and aft halves can not be adjusted,
- The process cannot be applied to some types of forms, e.g., ships which has no parallel middle body,
- There is limitations in the range of longitudinal shift of sections,
- The maximum longitudinal shift of sections is restricted to the ends.

An application of this variation procedure is shown in **Figure 2.7** where the block coefficient is increased by 5% and the position of the centre of buoyancy is shifted aftwards by 2%. As indicated in **Table 2.3**, the increase in C_B was achieved exactly, however the actual difference in LCB is slightly different which is considered to be due to the approximation of levers h_a and h_f. The detailed definition of the variant hull form obtained by 1-C_P method is presented in Appendix B.

Table 2.3. Main particulars of parent and variant form derived by 1-C_P method.

	Parent Form	**Variant Form**
Length between perpendiculars (L_{BP})	30.480 m	30.480 m
Beam (B)	6.100 m	6.100 m
Draught (T)	2.515 m	2.515 m
Displacement (Δ)	244 m^3	256.5 m^3
Block coefficient (C_B)	0.522	0.548
Midship area coefficient (C_M)	0.887	0.887
Prismatic coefficient (C_P)	0.588	0.618
Waterplane area coefficient (C_{WP})	0.753	0.760
Longitudinal centre of buoyancy (LCB)	% 1.34 L (aft)	% 3.44 L (aft)
Longitudinal centre of flotation (LCF)	% 4.06 L (aft)	% 5.07 L (aft)

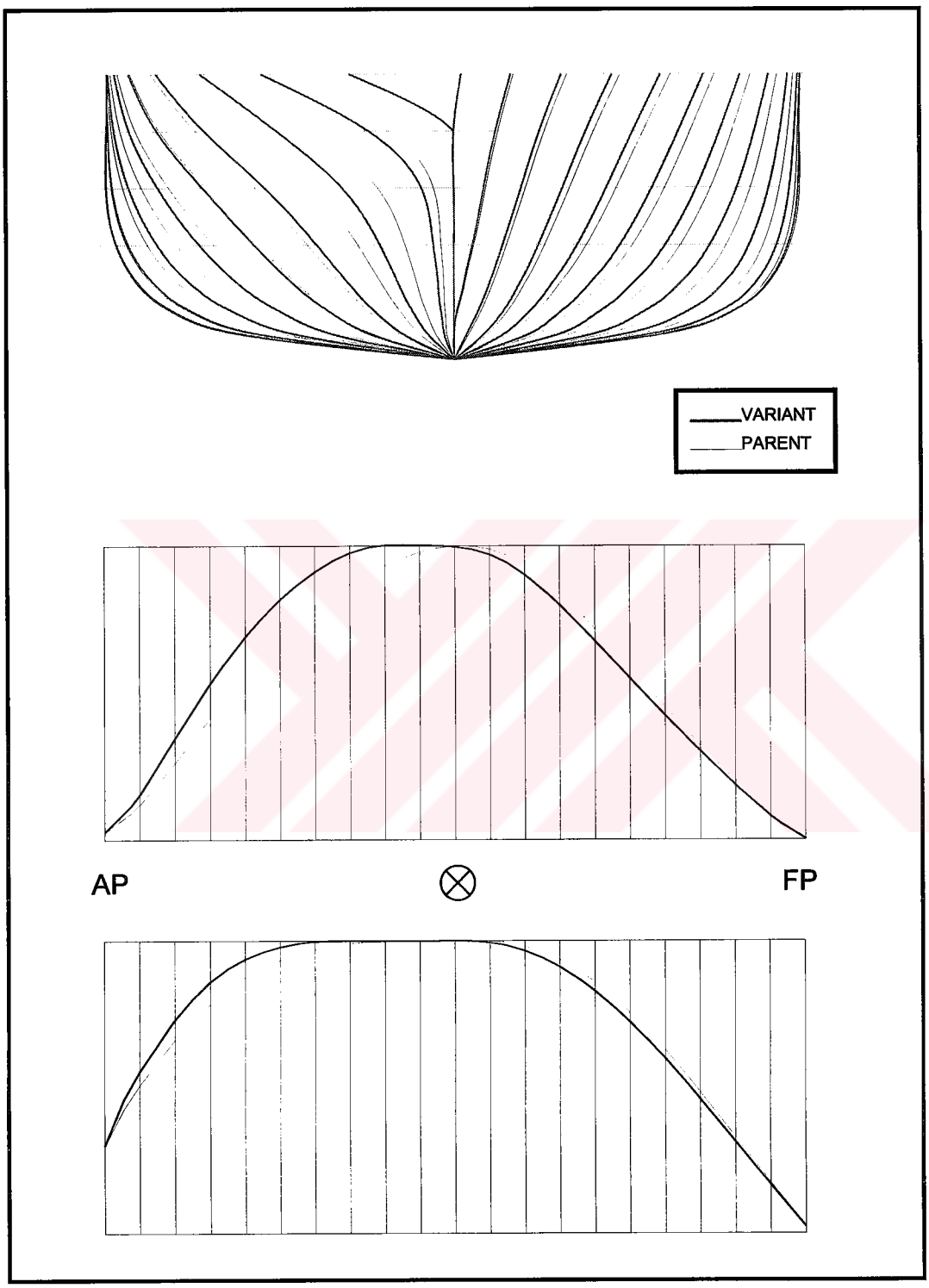


Figure 2.7. Body plan, sectional area and loaded waterline curves for the parent and variant forms derived by using the one-minus prismatic method.

2.3.1.3. Modified One-Minus Prismatic Method

This method, also known as the Lackenby's method was developed as a quadratic version of the one-minus prismatic method whereby the lengths of the parallel middle body can be controlled independently of LCB and the prismatic coefficient. This generalised method includes almost all of the cases related to any extent of parallel middle body. (**Lackenby, 1950**)

Lackenby assumes that δx is proportional to x(1-x) which is a maximum when x=1/2 therefore the added (or removed) area is concentrated in the centre. The required shifts of ordinates for forebody and afterbody is obtained by the following general formula

$$\delta x = (1-x)\left\{\frac{\delta p}{1-p} + \frac{x-p}{C_P(1-2\bar{x}) - p(1-C_P)}\left[\delta C_P - \delta p\frac{1-C_P}{1-p}\right]\right\}$$

where p is the parallel middle body and δp is the required change in parallel middle body. The first approximation to the lever h for each part of the sectional area is given by

$$h = C_P\left\{\frac{2\bar{x} - 3k^2 - p(1-2\bar{x})}{C_P(1-2\bar{x}) - p(1-C_p)}\left[1 - \frac{\delta p}{\delta C_P}\frac{1-C_P}{1-p}\right] + \frac{\delta p}{\delta C_P}\frac{1-2\bar{x}}{1-p}\right\}$$

The Lackenby's method was applied to the parent form to increase the block coefficient by 5% and to shift the position of the centre of buoyancy forward by 2%. The results are shown in **Figure 2.8** and **Table 2.4**. The offsets of the variant hull form derived by Lackenby's method are given in Appendix B.

The main difference between the one minus prismatic and the Lackenby's method can be seen in the sectional area curve comparisons, i.e., the midship section is same for the parent and variant forms because there was no parallel middle body change requirement.

Therefore, by using this technique the parallel middle body length and the prismatic coefficient can be varied independently however, the main restriction of the method is the dependent variation of the characteristics of the sectional area and the design waterline curves.

Table 2.4. Main particulars of parent and variant forms obtained by Lackenby's method.

	Parent Form	Variant Form
Length between perpendiculars (L_{BP})	30.480 m	30.480 m
Beam (B)	6.100 m	6.100 m
Draught (T)	2.515 m	2.515 m
Displacement (Δ)	244 m^3	256.5 m^3
Block coefficient (C_B)	0.522	0.548
Midship area coefficient (C_M)	0.887	0.887
Prismatic coefficient (C_P)	0.588	0.618
Waterplane area coefficient (C_{WP})	0.753	0.782
Longitudinal centre of buoyancy (LCB)	% 1.34 L (aft)	% 0.88 L (aft)
Longitudinal centre of flotation (LCF)	% 4.06 L (aft)	% 2.28 L (aft)

2.3.1.4. Moor's Method

The linear distortion methods described in the above sections, do not allow independent variation of the characteristics of the sectional area and the design waterline curves, i.e., C_B, LCB and C_{WP}, LCF. To remedy this, within the framework of linear distortion approach, **Moor (1970)** defined a section shape factor as *(section area curve ordinate)/(waterline curve ordinate)* using sectional area curve and waterline curve ordinates for each section and for the parent and variant hulls. Therefore, each section for a new form corresponding to the section shape factor curve can be derived from the parent by selecting the parent section with the required ordinate in the section shape factor curve and multiplying its transverse offsets by the ratio of the ordinate of the required waterline curve at the required section to the ordinate of the parent waterline curve at the parent section. The Moor's method is applied to the parent form in order to change C_B by 5%, C_{WP} by 5% and shift LCB and LCF forward by 1%. As can be seen in **Figure 2.9** and **Table 2.5** the method can handle all these variations independently. The offsets of the variant hull form derived by Moor's method are given in Appendix B.

Table 2.5. Main particulars of parent and variant forms derived by Moor's method.

	Parent Form	Variant Form
Length between perpendiculars (L_{BP})	30.480 m	30.480 m
Beam (B)	6.100 m	6.100 m
Draught (T)	2.515 m	2.515 m
Displacement (Δ)	244 m^3	256.4 m^3
Block coefficient (C_B)	0.522	0.548
Midship area coefficient (C_M)	0.887	0.887
Prismatic coefficient (C_P)	0.588	0.618
Waterplane area coefficient (C_{WP})	0.753	0.790
Longitudinal centre of buoyancy (LCB)	% 1.34 L (aft)	% 0.18 L (aft)
Longitudinal centre of flotation (LCF)	% 4.06 L (aft)	% 2.86 L (aft)

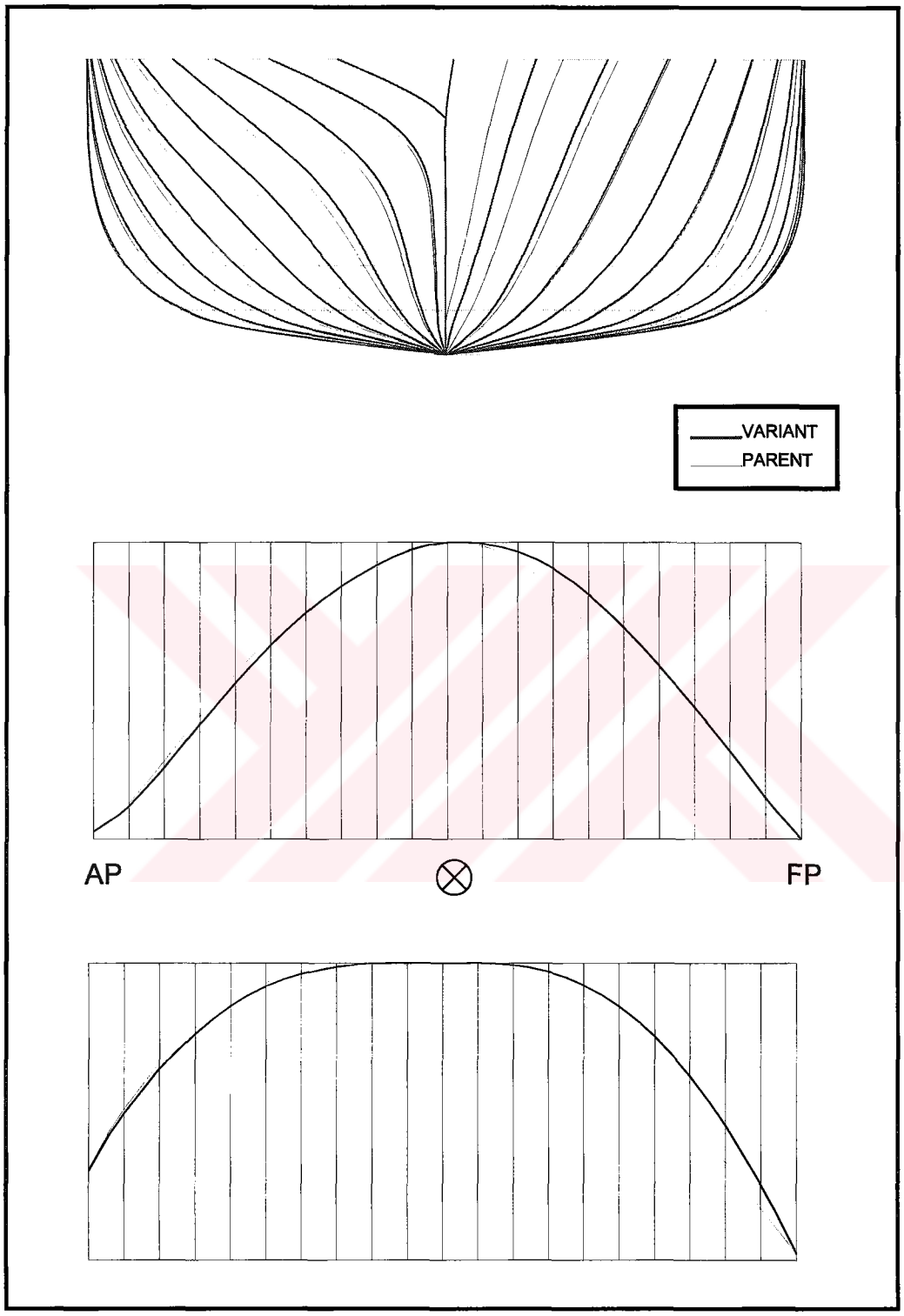


Figure 2.8. Body plan, sectional area and loaded waterline curves for the parent and variant forms derived by using the Lackenby's method.

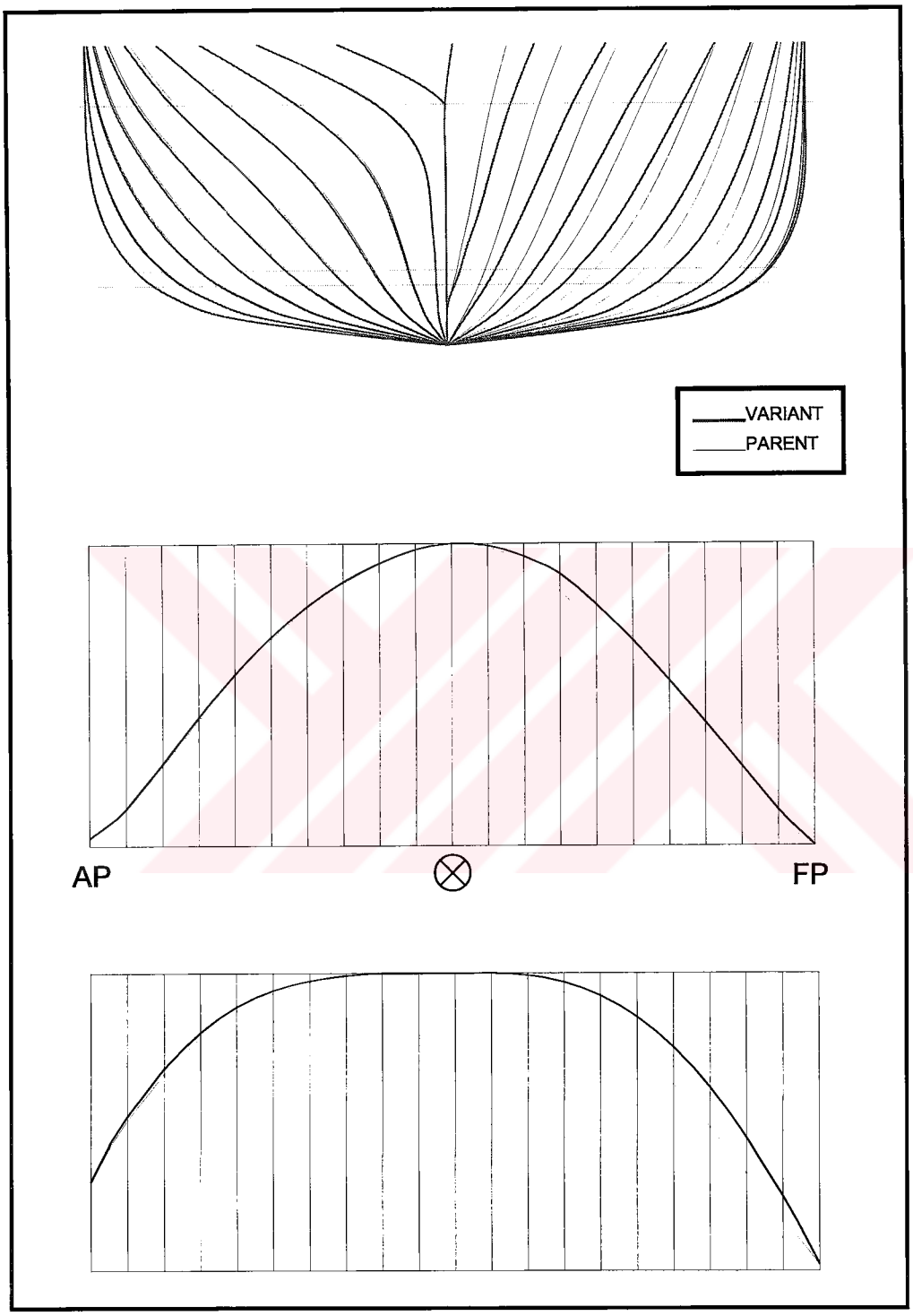


Figure 2.9. Body plan, sectional area and loaded waterline curves for the parent and variant forms derived by using the Moor's method.

2.3.2. Non-Linear Distortion Techniques

Söding (1967) proposed a method of distorting the surface of an existing hull by using simple standard functions in the following general form:

$$xn = x + F_1(x) \cdot F_2(y) \cdot F_3(z) \cdot G(F_1'(x), F_2(y), F_3(z))$$
$$yn = y + F_4(x) \cdot F_5(y) \cdot F_6(z) \cdot G(F_4(x), F_5'(y), F_6(z))$$
$$zn = z + F_7(x) \cdot F_8(y) \cdot F_9(z) \cdot G(F_7(x), F_8(x), F_6'(z))$$

where x,y,z represent the parent form geometry and xn,yn,zn are the corresponding points on the surface of the variant form. F are three dimensional vector functions and G functions are introduced to prevent gross distortions at curve endings, but not generally affecting the overall transformation.

The selection F functions are arbitrary and they are normally in the form of polynomials. These include, in addition to scale and area curve transformations, standard functions to change the bilge radius, rise of floor, U- or V-ness of sections, stem and stern profiles, flare or tumblehome and local waterline, section or buttock shapes.

Affine distortions of a parent hull (modification of main dimensions by simple ratio) are simply accomplished within the generalised method. For instance, to increase or decrease the length, $F_1(x)$ is set equal to the ratio of new to parent length minus 1, $F_2(y)$ and $F_3(z)$ set equal to 1, and $F_4(x)$ through $F_9(z)$ may be ignored since y and z coordinates are not altered. To modify the beam, $F_4(x)$ and $F_6(z)$ are set to 1, and $F_5(y)$ is set equal to the ratio of new to parent beam minus 1.

Modification of sectional area curve only requires distortions on the x axis, so only function $F_1(x)$ needs to be addressed. Specific functions may be derived to achieve a number of distortions including to change bilge of radius, length of parallel middle body, rise of floor, U- or V-ness of sections etc. Details of this functions are given by **Rabien (1996)**. A simple application of the method is presented in **Figure 2.10** and **Table 2.6**. The offsets of the variant trawler form obtained by using this non-linear method can be found in Appendix B. In this simple case all F functions were kept unit with the exception of $f_9(z)$ which was selected as a linear polynomial.

The method is versatile and, in theory, any kind of variation is possible by selecting appropriate functional. However, in most cases it is not possible to establish a direct connection between the F functions and the characteristics of the resulting variant

form. This may be a great disadvantage in the systematically development of a large number of alternatives.

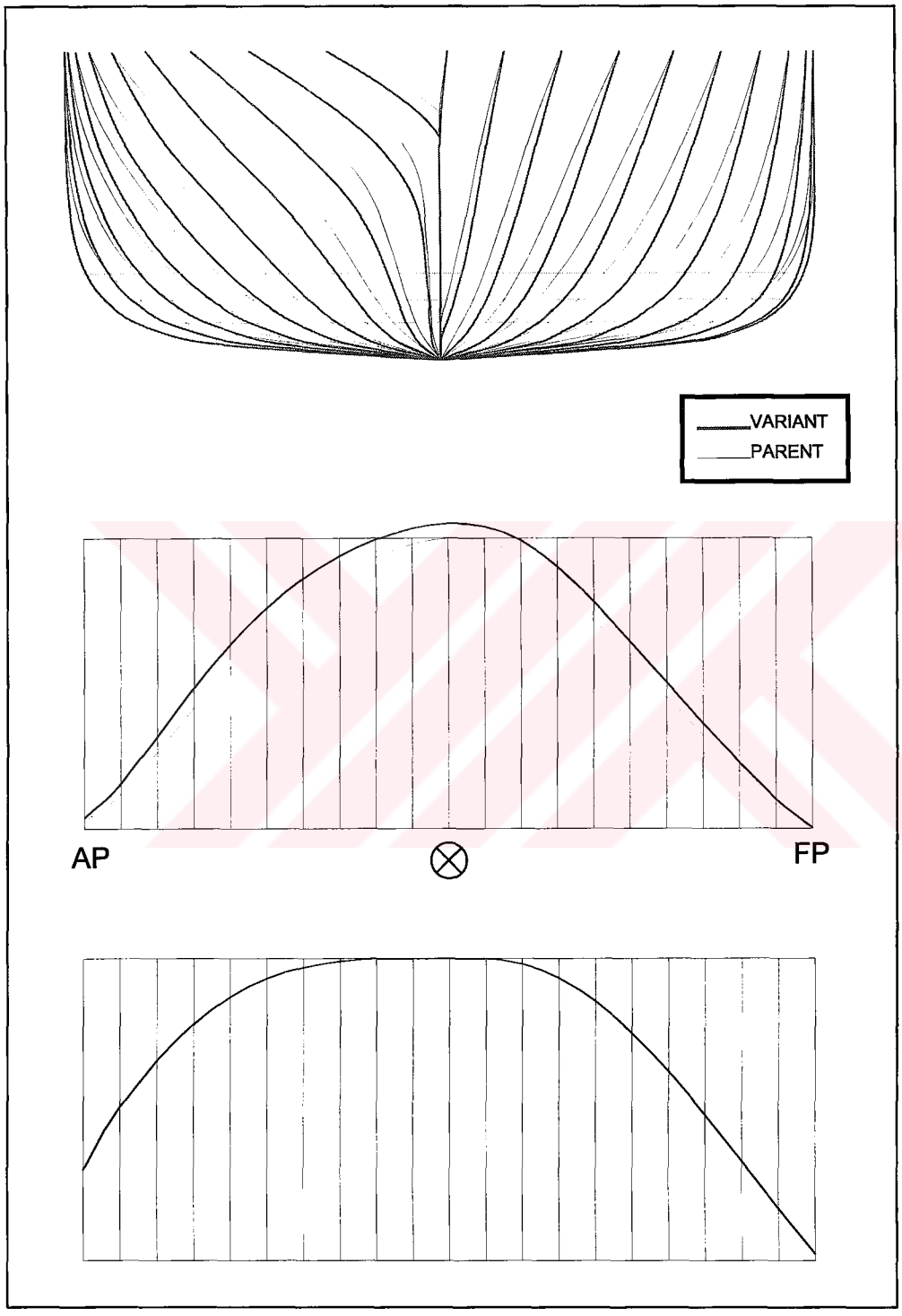


Figure 2.10. Body plan, sectional area and loaded waterline curves for the parent and variant forms derived by using a non-linear distortion method.

Table 2.6. Main particulars of parent and variant form derived by non-linear distortion.

	Parent Form	Variant Form
Length between perpendiculars (L_{BP})	30.480 m	30.480 m
Beam (B)	6.100 m	6.100 m
Draught (T)	2.515 m	2.515 m
Displacement (Δ)	244 m^3	266.1 m^3
Block coefficient (C_B)	0.522	0.569
Midship area coefficient (C_M)	0.887	0.932
Prismatic coefficient (C_P)	0.588	0.610
Waterplane area coefficient (C_{WP})	0.753	0.753
Longitudinal centre of buoyancy (LCB)	% 1.34 L (aft)	% 1.61 L (aft)
Longitudinal centre of flotation (LCF)	% 4.06 L (aft)	% 4.06 L (aft)

2.3.3. Shape Averaging Method

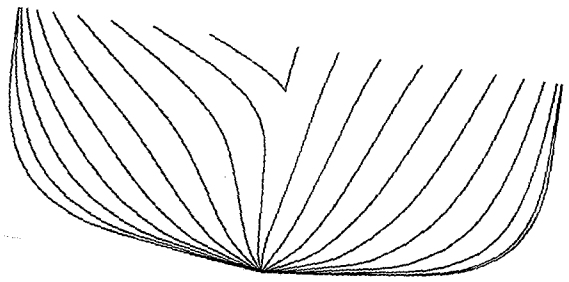
This method is relatively new compared to the previously mentioned methods, and it may find potential applications varying widely from simple designs to complicated industrial products. (i.e., car, ship, etc.) A typical application of the method is presented by **Chen and Parent (1989)**. Shape averaging is the process of extracting a typical representation from the designer's input shapes. This technique can be used to create new forms by blending global features of related or unrelated shapes, thus stimulating the generation of new ideas as well as creating the average product. The method would allow a designer to combine primitive shapes to gradually to generate more complicated objects (e.g., a teardrop and a car shape can be averaged to explore the aerodynamic aspects of car body styling). Hence, this approach provides the designer a useful means to study, create and evaluate forms quantitatively.

The promising averaging process, which may be used as a variation tool in preliminary ship design, is syntactic averaging. The shapes are treated as geometric entities and the averaging is concerned only with geometric properties of elements such as size and distance. In general, the method consists of two parts: The correspondence determination, which is a pre-processing step for averaging, and averaging according to these correspondences. In the correspondence process, a base hull form that has the greatest number of vertices is chosen to be paired with another hull form. The vertices are matched on the basis of having the minimum distance between them and to have a unique match, therefore both hulls will contain the same number of vertices. After this process, the hulls will be ready for averaging.

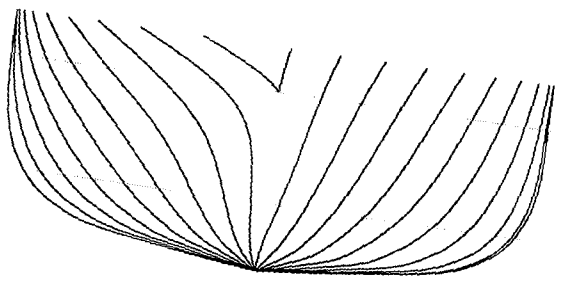
Averaging involves processing each set of correspondences at one time and extracting the average point from that set under various weights. Each shape can be assigned a various weight. Depending on the weights assigned, a series of results that either interpolate or extrapolate the input shapes can be generated. The results of interpolation indicate a smooth transformation from one shape to another depending on the weights assigned, whereas the extrapolation scheme indicates the trendy shapes.

Shape averaging, as a form generation tool can be used for the creation of original design concepts as well as generation of conventional medium shapes. Herein, this shape averaging process is applied to a well-known BSRA trawler series, in order to obtain a variant form with 20% larger block coefficient referred to as *final form*. The *parent form*, intermediate solutions labelled as *variant forms* and the *final form* are shown in **Figure 2.11.** The variant forms are obtained by using linearly varying weight functions. These weight functions are chosen as to obtain forms of fullness varying between 0% to 20%. The offset tables of parent, variant and final forms can be found in Appendix B. Applications indicate that the method proves to be a successful form variation tool and may find wide application areas. However, as shown in the results, many other form parameters are also changed. This may not be acceptable in some studies where form parameters are systematically changed.

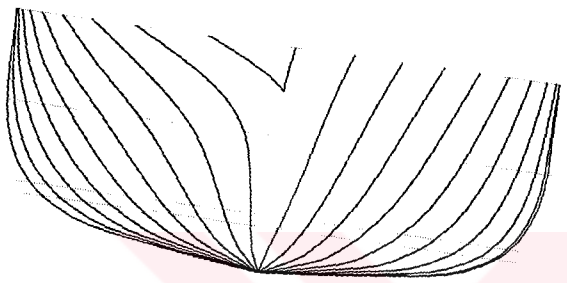
Ultimately, the use of linear and non-linear distortion techniques enables the ship hull form designer to generate feasible hull form alternatives by using a parent hull form with satisfactory hydrodynamic performance properties. In this chapter, alternative hull form distortion techniques have been described and advantages and drawbacks are presented. It has been observed that the selection of the variation tool should be based on the design objective which could be as simple as to change the block coefficient by x% or as complex as to modify the hull form to provide better streamlines for a bilge keel. It is worth noting that Moor's method emerges as the most powerful linear distortion technique as it can handle independent variations of hull form characteristics. However, there may be cases where non-linear techniques are clearly desirable and hence should be applied. (**Narlı et al, 1999**)



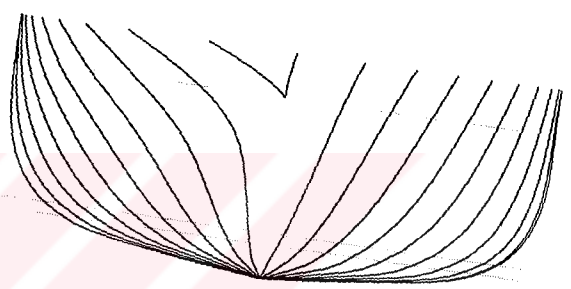
PARENT



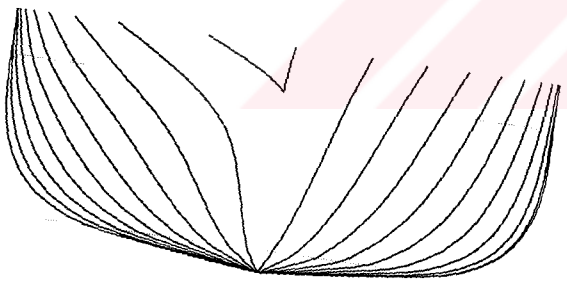
VARIANT1



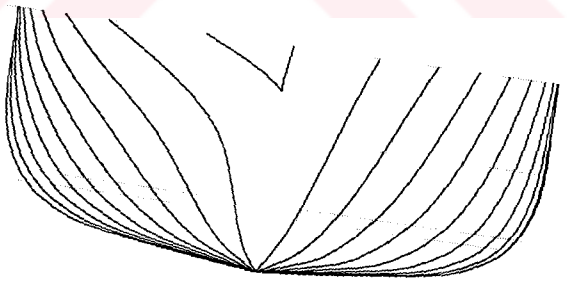
VARIANT2



VARIANT3



VARIANT4



FINAL FORM

Figure 2.11. Body plans for the parent, intermediate and final variant forms derived by using the shape averaging method.

3. MATHEMATICAL REPRESENTATION OF SHIP LINES AND SURFACES

The design and representation of ship hull forms by means of mathematical methods has a long tradition. Simple geometric shapes such as circles, ellipses etc. have received considerable attention and can be found in ancient ships due their relatively easy production capabilities. However, practical applications have not been possible until modern numerical techniques and powerful digital computers became available.

The benefits of representing the hull form by a precise mathematical description are:

- The need for full scale lofting procedure is eliminated since accuracy could be obtained initially to whatever degree desired.
- The mathematical representation can be powerful as well as useful when coupled with high-level computer graphics environment.

The Swedish researcher **Chapman (1760)** first initiated the research in the field of mathematical techniques for ship curves. He had mentioned the use of family of parabolas for waterlines and other ship curves on the hull surface. **Nystrom (1863)** expanded Chapman's work by using parabolas of varied order with fractional as well as integral components, to make up both waterlines and sections.

A pioneering work on mathematical ship lines was achieved by **D.W. Taylor (1915)** who represented his parent ship by mathematical expressions and then derived a series of ship forms by systematic variation of geometric form parameters. His objective was to produce a series of lines whose shapes were similar and whose form parameters varied systematically. Taylor published the concept of generating the sectional area curve, waterlines and transverse sections from form parameters. He approximated the sectional area curve, and the waterlines of the ship by fifth-order and fourth-order explicit polynomials, respectively, in the following general form.

$$y(x) = \sum_{i=0}^{n} a_i x^i$$

Using this approach Taylor developed the parent hull shapes investigated in his classical systematic hull form series known as Taylor series. He was the first naval architect to use mathematical representation for the systematically variation of hull shape by a family of form parameters. A number of US warships were delineated according to Taylor's method.

Although the use of ordinary polynomials is a simple and attractive approach to represent ship hull lines, a flat bottom based on a constant rise of floor cannot be reproduced, nor can a fixed bilge radius or flat side of large extent. It may therefore be presumed that these simple polynomials cannot be used for all ship types.

Research efforts on ship resistance, propulsion etc., have produced major contributions to the art of mathematical representation of ship lines. **Weinblum (1934)**'s contributions in these developments has had the most crucial effect. He has the objective of developing hull forms with desired physical properties. In his work the principles of parametric lines creation were further extended in connection with the systematic variation and hydrodynamic optimisation of hull forms.

Benson (1940) used fifth degree polynomials to represent waterlines. The coefficients of these polynomials are determined on basis of given form parameters. Benson developed a fairing process in vertical direction which involves an expression of the form parameters as a function of the vertical co-ordinate.

Watanabe (1946) used waterline polynomials of the same type as **Benson (1940)** but with terms only even orders up to the 10^{th}. The coefficients of the polynomials are determined on the basis of form parameters such as area, statical moment, and angles of entrance and run. The vertical fairing is performed by expressing each waterline parameter as a function of the vertical variable.

Theilheimer (1957) used cubic spline curves to represent ship lines. The waterlines are expressed as polynomials which must pass through the given half breadths. The fairness of the waterlines are tested by studying the second differences and corrected in an iterative manner. The continuity in vertical direction is achieved by expressing the polynomial coefficients as functions of vertical distance.

Rösingh and Berghaus (1959) use ordinary polynomials to represent ship hull surface. The smoothing of the lines is achieved by double integration of the second derivatives. Thus, from the coordinates of the original waterlines a generating function is obtained after double differentiation. The generating functions are represented by polynomials with two coefficients, which are determined by the least squares method.

The smoothing in vertical direction is achieved by expressing polynomial coefficients as functions of vertical distance.

In order to reduce the degree of ordinary polynomials **Pien (1960)** modifies the hull surface before the application of polynomials. Bow and stern stations are assumed to be the vertical boundary lines of the hull surface and the modified hull surface is represented by a relatively low degree polynomial expression. Hence, the surface is determined as a polynomial,

$$y(x,z) = \sum_{i=1}^{N} \sum_{j=1}^{M} a_{ij} x^i z^j$$

which is solved by the least squares method for the unknown a_{ij} coefficients.

Kerwin (1960), in order to represent ship lines mathematically, used an approximation function of the form

$$g(x,z) = \sum_{n=1}^{N} \sum_{m=1}^{M} a_{mn} \psi_m \phi_n$$

where ψ_m and ϕ_n are m^{th} and n^{th} polynomials representing waterlines and sections, respectively. The coefficients a_{mn} are determined from the condition that g(x,z,) is to approximate the given offsets f_{ij} in the sense of least square. Therefore,

$$E = \sum_{j=1}^{J} \sum_{i=1}^{I} \left\{ f_{ij} \sum_{n=1}^{N} \sum_{m=1}^{M} a_{mn} \psi_{mj} \phi_{ni} \right\}^2$$

should be minimised. I and J are the number of sections and waterlines, respectively.

Taylor (1963) used least squares method and Chebyshev polynomials to represent sectional area curve and other design curves. It is shown that Chebyshev polynomials approximate a given function so that the maximum deviation is less than compared to any other polynomial of the same degree.

In determination of the coefficients of these high-order polynomials, increase in degree led to numerical trouble due to ill-conditioned systems of equations.

The representation of ship form or of its elements in terms of a set of shape parameters has first been undertaken with the aim of systematic variation and description of hull form families, e.g., **D.W. Taylor (1915).** Later the purpose has been extended into the design of new ship forms from given set of parameters. **Williams (1964)** has described a design procedure based on this technique. The waterline parameters are read off or computed from the drawing and then represented by polynomials which are cross faired in the draught direction. Thus, similar sets of parameters are used for all waterlines.

Miller and Kuo (1963) developed a fairing procedure in which the ship is subdivided into several regions such as the ends, parallel middle body, chine, bottom and bilge are treated separately. In the main parts of fore and after-body, the waterlines are approximated by polynomials complying with some six to ten integral and differential parameters. The coefficients of the polynomials are obtained and faired vertically in the sense of least squares.

Von Kerczek and Tuck (1969) have introduced conformal mapping functions for representing the underwater sections of the hull. They demonstrated the feasibility of representing the underwater sections of practical hull forms by Theodorsen conformal mapping functions. They had obtained rather exact close fits for Series 60 hull forms defined by a given set of offset points

The advantages of explicit polynomials for lines creation and mathematical representation led these researchers to develop fairing procedures mainly based on explicit polynomial representations. Also, polynomials were the only class of functions that can be used for representation of ship lines at that time. However, besides their simplicity, flexibility, and practical computational advantages they have certain limitations. The difficulties arise specifically because of their oscillating nature with increasing degrees. Moreover, it is difficult to represent shapes with vertical tangents. Hence, these drawbacks have led the researchers to develop better alternative mathematical techniques.

In the early fifties, the availability of computers in shipbuilding changed the scope, and methodology of mathematical ship form definition in a dramatic way. The production of ship parts is performed by numerically controlled machine tools, which require accurate, and computer based ship geometry information as input. Hence, intensive use of computers and computational methods for ship hull form definition is required, and novel mathematical techniques for curve and surface representation were introduced. These methods were generally based on parametric as opposed to the earlier explicit, $y = f(x)$ or implicit $F(x,y) = 0$ representations. Hence, the problems,

such as infinite tangency, multi-varied curves associated with the explicit representation are overcome. The major breakthrough in Computer Aided Geometric Design (CAGD) field was the development of the theory of spline curves and surfaces. (i.e., Bezier curves, and their generalisation to B-spline curves) They have been used effectively in the field of hull form design. The obvious advantages of physical splines have led the researcher's to simulate it mathematically. A great deal of work has been devoted to the mathematical modelling of the draftman's spline. The earliest work related to physical splines dates to the 18^{th} century, to a study of *elastica* by James and Daniel Bernoulli, and Leonhard Euler. Elastica is idealised thin beams. The Bernoulli brothers postulated that the work required to bend a thin beam was proportional to the square of the curvature, and Euler has derived the differential equation of the curve and classified its forms. However, the notion of the *mathematical spline* was first introduced by **Schoenberg (1946),** a piecewise polynomial function of degree n whose segments meet with C^{n-1} continuity.

The spline curves in general have solved the problems of connection (continuity) encountered in the previous methods and obtained in two ways; as a curve that interpolates through given data points (curve interpolating techniques) or as the result of interactive manipulation of the defining control polygon (curve approximating techniques).

One of these interpolatory approaches is the most popular cubic splines which are in general piecewise analog of the physical elastic spline that has long been used in shipbuilding industry to produce fair ship curves. **Holladay (1957)** introduced the cubic spline for function interpolation and integration. The curve he developed minimised the squared curvature of the curve. He also noted that for curves with modest slopes, $f'(x)<<1$ the cubic spline has provided a good approximation to the bending of a thin rod.

A curve produced by a spline is described by cubic polynomials between adjacent weights and the curve is continuous up to the curvature. The mathematical spline basis was derived from its physical counterpart; (the elastic spline made of wood, plastic or steel) by observing that, for small deflections, the shape assumed by the physical spline was a piecewise cubic polynomial and thus referred to as *cubic spline function.* The mathematical equivalent is modelled by using cubic polynomials over each span of the curve. Therefore, the equation for a single parametric cubic spline segment can be given by

$$P(t) = \sum_{i=1}^{4} B_i t^{i-1} \qquad t_1 \le t \le t_2$$

where t_1 and t_2 are the parameter values at the beginning and end of the curve segment. P(t) is the position vector of any point on the cubic spline segment. The superiority of the interpolating cubic spline representation stems from the fact that, among all interpolating function methods, it has the minimum value for the integral of the square of the curvature. Thus, the total bending energy stored in the spline is minimum. However, this interpolating curve fitting technique results in a subjectively "fair" curve as it is constrained to pass through all specified points. They lack some of the desirable properties possessed by most approximation techniques. Therefore, one may usefully sacrifice this interpolating behaviour in favour of higher degree of smoothness by adopting an approximating function built up as a linear combination of basis B-spline function over each span of the curve.

The work following Holladay's introduction of cubic splines concerned methods for achieving an improved approximation to the minimum energy curves and eliminating the unwanted wiggles or oscillations sometimes observed in cubic splines. First **Asker (1962)** introduced several approaches. He introduced a method allowing the stiffness of the spline to be varied along its length. He presented two versions of variable stiffness, eliminating the problem of extraneous wiggles. In the first scheme stiffness varies in a piecewise constant fashion resulting in a C^1 cubic spline, and in the second stiffness varies in a piecewise linear fashion resulting in a C^2 quartic spline. He applied this interpolating spline function in the numerical design of ship lines.

Ferguson (1964) introduced the parametric cubic spline curve by applying cubic spline function interpolation to vector-valued data. He described curve segments as vectors, using parameters. Hence, a *Ferguson curve segment* is a cubic vector function with respect to a parameter obtained by specifying the position and tangent vectors of the end points, and can be written in terms of its end points and end tangents as

$$p(t) = P(0)(1-3t^2+2t^3) + P(1)(3t^2-2t^3) + P'(0)(t-2t^2+t^3) + P'(1)(-t^2+t^3)$$
$$0 \le t \le 1$$

Another attempt to reduce the occurrence of oscillations yield a mathematical analog to the physical phenomenon of adding tension along the direction of the spline curve. The earliest attempts by **Schweikert (1966)** resulted in exponentially based alternatives to the natural spline. Thus, he proposed the use of *splines under tension.* This was an attempt to improve the controllability of the spline. By suitably adjusting

the tension parameter, the production of loops in the spline can be prevented. Thus, tension parameter can be used as a smoothing parameter to dampen the undesired oscillations. However, these exponential splines were expensive to calculate. The basics of the theory of the splines under tension are similar to that of energy minimising splines, namely cubic splines. It is known that cubic spline function minimises the bending energy proportional to the integral of the square of its curvature. For the case of splines in tension one more integral term is included into this minimisation problem. That is the arc length of the spline where the tension is obtained by pulling on the ends of the spline. Thus, this leads to decreasing the arc length. The solution curve is obtained by minimising the following expression:

$$\int_0^{\ell} |X''(t)|^2 dt + p^2 \int_0^{\ell} |X'(t)|^2$$

where the first term represents the curvature, and second arc length. The free parameter $p \geq 0$ denotes global tension parameter.

Another attempt to solve most of the problems encountered in cubic spline representations is the introduction of *parabolically blended curves* by **Overhauser (1968)**, however at the expense of being only C^1 continuous. (i.e., only first derivative continuity is maintained at the internal joints) Although, a parabolically blended curve is computationally inexpensive and easy to implement, due to its lack of higher-order continuity, it has found limited application areas.

Cline (1974), and **Pilcher (1974)** further developed splines under tension. Alternatively, due to the expense of computing these exponential curves, **Nielson (1974)** introduced the ν-spline as a C^1 piecewise cubic polynomial. ν-splines extend splines in tension by allowing tension to vary along the curve providing greater control over the curve shape.

Hagen (1985) introduced the τ-spline as a generalisation of the ν-spline. τ-splines are piecewise quintics exhibiting G^2 continuity. The developments continued with **Foley (1987)**. He introduced the cubic weighted ν-spline, a generalisation of C^2 cubic splines, weighted splines, and ν-splines. **Meier and Nowacki (1987)** described a new type of splines that minimises a fairness functional resulting in a higher degree interpolant. Similarly, **Pottmann (1990)** extends τ-splines defining a new minimisation functional. Hence, Pottmann's spline results in a G^3 continuous interpolant.

Previously discussed interpolating spline techniques are characterised by the fact that the derived mathematical curve passes through each and every point. However, this interpolating feature cannot be accepted as a general requirement in many applications, especially in hull form design. Due to various reasons (e.g., misreading, transcribing, etc.), the data points may possess unintentional erroneous points, and some modifications should be necessary. Specifically, these have led the researchers to develop novel approximation techniques mainly based on control polygon techniques where the shape of the curve approximates the control polygon. **De Casteljau (1959)** at Citroen and **Bezier (1966)** at Renault have initiated the research in this area. They have independently developed *Bezier curves and surfaces*. They have recombined the terms of the Ferguson cubic segment in a way that makes the physical meaning of the vector coefficients more apparent. De Casteljau's development, slightly earlier than Bezier's was never published, so the whole theory of polynomial curves and surfaces in Bernstein form now bears the Bezier's name. Their techniques had the important property, missing in any interpolation method of guaranteeing that smooth shapes would be generated from smooth data. The variation diminishing property is crucial to the approximation and design of shapes.

Bezier curves are defined by vertices which form an open polygon. The resulting curve is tangent to the first and last span of the polygon and passes through the first and last vertex where the order of the curve is strictly related to the number of the polygon vertices. The curve does not pass through the interior vertices but has the characteristic form defined by the polygon. A parametric Bezier curve is defined by

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t), \quad 0 \le t \le 1; \qquad J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

where B_i represents the coordinates of the defining polygon vertices and $J_{n,i}$ the Bezier or Bernstein basis function.

From Bezier curves to B-splines was a natural progression. Bezier curves are used to derive the theory of rational and non-rational B-spline curves. In Schoenberg's original work, the B-spline basis had been cast in terms of truncated power series, which was convenient for analysis but at the same time numerically troublesome. The simultaneous appearance, of papers by **de Boor (1972)**, **Cox (1972)** and **Riesenfeld (1973)**, **Gordon and Riesenfeld (1974)**'s research giving a numerically stable method for computation.

B-splines are the piecewise counterpart of the Bernstein polynomials. Depending on the basis function selected, two different types of representation can be obtained; Bezier or B-spline curves. They have found wide applications in hull form design. *B-spline curves* are piecewise polynomials, which are broken into segments each being defined by a polynomial. The B-spline curve can be defined as

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \qquad t_{min} \le t \le t_{max} \quad , \quad 2 \le k \le n+1$$

$$N_{i,1}(t) = \begin{cases} 1 & \text{if} \quad x_i \le t \le x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

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

$$\begin{aligned} x_i &= 0 && 1 \le i \le k \\ x_i &= i - k && k+1 \le i \le n+1 \\ x_i &= n - k + 2 && n+2 \le i \le n+k+1 \end{aligned}$$

where k is the order of the curve, n+1 the number of defining polygon vertices (B_i), $N_{i,k}(t)$ the basis functions associated to each vertex of the defining polygon, t the curve parameter varying from t_{min} to $t_{max}(n+2-k)$ and x_i the elements of a knot vector. The shape of the curve is controlled by the vertices of the polygon like in Bezier representation but the basic advantage of B-splines is the added flexibility gained by the introduction of knot vectors, and specification of the degree of B-spline curves independent of the number of the defining polygon vertices. There are also various control handles in B-spline curves which makes it a useful tool for representation and fairing of ship curves. Control can be performed by changing the order of the curves, repeating the control vertices or by changing the number and position of the vertices.

Versprille (1975) introduced the non-uniform rational B-splines (NURBS) to computer aided geometric design applications. The main advantage of the rational form over the integral form of the B-spline is the ability to represent conic curves exactly. The rational form of the B-spline is described using the control points of the integral B-spline (B_i), augmented with weights at each control point. The weight acts as a shape control handle, thus increasing the curve's degree of freedom. Increasing the value of the weight associated with a control polygon vertex draws the curve closer

to the associated vertex. Rational B-spline curves are generalisation of non-rational B-spline curves, and can be expressed in the form

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

where, h_i's represent weights associated with the defining polygon vertices. It is obvious that when all $h_i = 1$, the rational form reduces to non-rational integral B-spline curves. Rational B-splines provide a single precise mathematical form capable of representing the common analytical shapes. (e.g., lines, circles, all types of sculptured surfaces) used in computer graphics and computer aided design environment. Therefore, have found wide application areas more recently in hull form design.

The natural extension of curves are surfaces, and the major breakthrough in the field of mathematical surface description methods is the introduction of **Coons (1964)**'s surfaces, a generalised form of Ferguson surfaces. The method uses normalised cubic splines for all four boundary curves. Cubic blending functions are used to define the interior of the patch. A bi-cubic surface patch is defined by the four position vectors at the corners, eight tangent vectors, two at each corner, the four twist vectors at the corners, and the four cubic blending functions. The Coons surfaces provide a flexible and powerful surface design tool, however possess a number of problems mainly, connection, and control problems. Details of Coons surfaces can be found in **Coons (1974)**. Most of these problems are overcome by the extension of Bezier curves to surfaces. A cartesian or tensor product Bezier surface is defined by

$$Q(u,w) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,j} J_{n,i}(u) K_{m,j}(w)$$

where $B_{i,j}$ represents the control net vertices, and $J_{n,i}$, $K_{m,j}$ are the Bezier or Bernstein basis functions in the u and w parametric directions. Most of the problems (higher degree due to high number of vertices, lack of local control etc.) encountered in Bezier surfaces are overcome B-spline surfaces. B-spline surfaces are the natural extension of the Bezier surfaces, and can be given in the form

$$Q(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M_{j,\ell}(w) \qquad u_{min} \le u \le u_{max}, \qquad 2 \le k \le n+1,$$

$$w_{min} \le w \le w_{max}, \qquad 2 \le \ell \le m+1$$

where $N_{i,k}(u)$ and $M_{j,\ell}(w)$ are the B-spline basis functions of degree k-1, and ℓ-1 in the bi-parametric u and w directions, respectively.

Most of the curve and surface representation methods discussed in the foregoing have been used for ship hull lines definition and surface design. (e.g., cubic splines and splines in tension are used in the U.S. Navy CASDOS System, and Bezier curves are used in the Unisurf system for hull form design) However, compared with other mathematical tools, the integral/rational B-splines can be seen as the most powerful and flexible technique for developing efficient fairing procedures, and widely used in CASD (Computer Aided Ship Design) applications.

Research and development of more than three decades in the area of CAGD has produced numerous methods for constructing curves and surfaces. The sequence of development of mathematical curve and surface description methods summarised in the foregoing is illustrated in **Figure 3.1.**

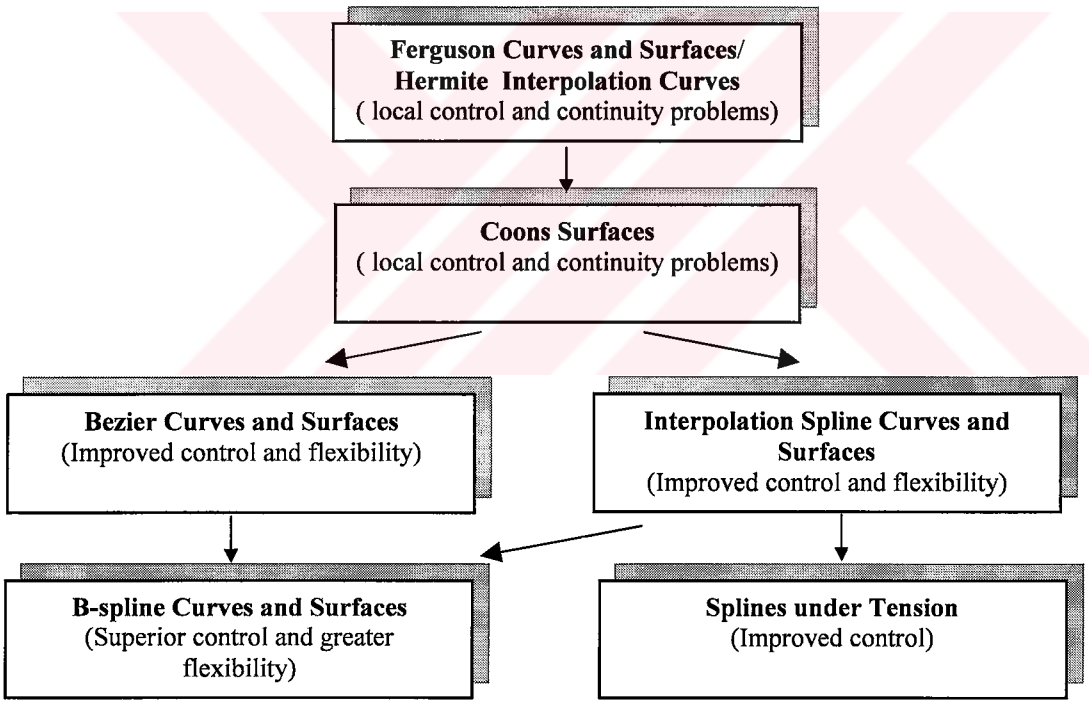


Figure 3.1. The sequence of development of mathematical curve and surface description methods. (**Yamaguchi , 1988**)

3.1. Polynomial Representation of Ship Lines

In general, free form ship curves, i.e., frame, waterline, buttock lines, are given in tabular forms as follows:

x_o	x_1	x_2	...	x_n
y_o	y_1	y_2	...	y_n

Mathematical representation of ship lines will require to generate an approximating function that will allow an estimation of the offset value y for $x \neq x_i$, i = 1, 2, ... , n. It may be assumed that these offset points are defined by a non-analytic function y = f(x). We wish to approximate f(x) by a suitable analytic function g(x) which has the general characteristic shape.

The best form of an analytical function g(x) which will represent the unknown original function represented by discrete offset points depends on many factors, such as the characteristic form of f(x), the source and accuracy of offset points, and accuracy requirements for the approximation. Hence, it is obvious that the more we know about the function f(x), the greater is the likelihood of finding the best approximating function. For instance, if the offset points suggests that f(x) should be in linear form in x, then we would probably begin by attempting to fit the offset points with a line.

The most common, and possibly the simplest method of representing a set of points by mathematical functions g(x) are those involving linear combinations of simple functions drawn from a class of functions $[g_i(x)]$ of the form:

$$g(x) = a_o g_o(x) + a_1 g_1(x) + \ldots + a_n g_n(x)$$

The classes of functions frequently encountered can be found in **Table 3.1.**

Table 3.1. Some classes of functions.

Monomials (Taylor Series)	x^i	; i = 0,1,...,n
Trigonometric func. (Fourier Series)	sinkx, coskx	; k = 0,1,...,n
Exponentials	$e^{b_i x}$	; i = 0,1,...,n

Linear combination of monomials leads to polynomial of degree n

$$f(x) \approx g(x) = a_o + a_1x + a_2x^2 + \ldots + a_nx^n = \sum_{i=0}^{n} a_i x^i$$

Linear combination of the Fourier functions leads to

$$f(x) \approx g(x) \quad = a_o + a_1\cos x + a_2\cos 2x + \ldots + a_n\cos nx + b_1\sin x + b_2\sin 2x + \ldots + b_n\sin nx$$

$$= a_o + \sum_{k=1}^{n} a_k \cos kx + \sum_{k=1}^{n} b_k \sin kx$$

Linear combination of exponentials leads to

$$f(x) \approx g(x) = a_o e^{b_o x} + a_1 e^{b_1 x} + \ldots + a_n e^{b_n x} = \sum_{i=0}^{n} a_i e^{b_i x}$$

where f(x) is a dependent variable corresponding to e.g., offsets and x is an independent variable corresponding to e.g., sectional positions.

The algebraic polynomials are the most important and popular approximating functions. Polynomials are easy to evaluate and their sums, products and differences are also polynomials. Polynomials can be differentiated and integrated easily, yielding other polynomials in both cases.

All these advantages of the polynomials would be of little value if there were no analytical justification for believing that polynomials can yield good approximations for a given function f(x). It is implied by good approximation that the discrepancy between an approximating polynomial g(x) and f(x), i.e., the error in the approximation, can be made arbitrarily small. This theoretical justification exists in the form of Weierstrass approximation theorem which states that

If f(x) is continuous in the closed interval [a,b], (that is $a \le x \le b$) then, given any $\varepsilon > 0$, there is some polynomial $g_n(x)$ of degree n, such that

$$\left| f(x) - g_n(x) \right| < \varepsilon \qquad a \le x \le b$$

The polynomials can be simply divided into two categories; interpolating and approximating polynomials. Examples of interpolating and approximating polynomials are illustrated in **Figures 3.2.a** and **3.2.b**, respectively.

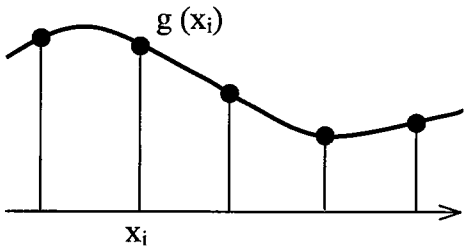


Figure 3.2.a. An interpolating polynomial.

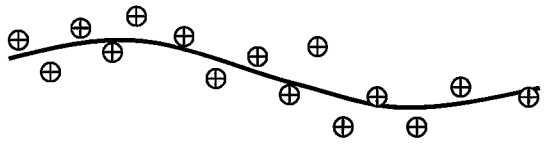


Figure 3.2.b. An approximating polynomial.

3.1.1. Interpolating Polynomials

Polynomial interpolation is the most fundamental of all interpolation concepts, however, today, interpolation polynomials are mostly of theoretical value rather than practical (i.e., the construction of elementary concepts of mathematical representation). Faster and more accurate methods based on polynomial methods have been developed, (e.g., piecewise polynomials or splines) and are widely used in hull form representation and fairing studies.

3.1.1.1. Interpolation of Curves Using Monomials

We wish to evaluate the interpolating function for n+1 distinct points P_i associated with appropriately selected parameter values t_i.

$$p_n(t) = \sum_{j=0}^{n} A_j t^j \qquad A_j \in \Re^3 \qquad t \in [a,b]$$

The most obvious criterion for determining the A_i coefficients of $p_n(t)$ is to require that

$$P_i = p_n(t_i) = \sum_{j=0}^{n} A_j (t_i)^j \qquad i = 0,1,...,n$$

We refer to the points P_i as interpolation points, and t as parameter values. The pairs (t_i, P_i) describes the nodes of the interpolation problem. The nth degree polynomial $p_n(t)$ must reproduce P_i exactly for the n+1 arguments $x = x(t_i)$ but in no way guarantees accurate approximation of P_i for $t = t_i$.

The formulation of the problem is in vector-valued form. Thus, the interpolation of functions in $\Re^2$ corresponds to choosing the interpolation points $f(x_i)$, and the associated parameter values to be the x- coordinates x_i.

The solution of simultaneous n+1 equations produces the coefficients A_i of the approximating function. The n+1 equations can be given in matrix form as:

$$\begin{bmatrix} 1 & t_o & \cdots & t_o^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & \cdots & t_n^n \end{bmatrix} \begin{bmatrix} A_o \\ \vdots \\ A_n \end{bmatrix} = \begin{bmatrix} P_o \\ \vdots \\ P_n \end{bmatrix}$$

There is a unique solution for the A_i, and hence there is only one polynomial $p_n(t)$ which exactly produces P_i values.

This type of polynomial may be ideal for the cases where fairing process is not necessary, assuming that the given offset points do not possess erroneous points. However, in a fairing problem, the offset points need some corrections, and this technique can not be accepted as suitable at all. Moreover, for high values of n (degree of the polynomial) the linear system of equations can produce ill-conditioning. Also, it is computationally expensive to change one of the knots, since the entire system has to be resolved.

3.1.1.2. Lagrange Interpolation Polynomials

We can eliminate some of the drawbacks of using monomials as basis functions by replacing with Lagrange polynomials. The general form of Lagrange interpolating polynomial is given by the following equation:

$$p_n(t) = \sum_{j=0}^{n} L_j(t) P_j$$

where P_j is the given interpolation points and L_j is the Lagrange polynomial basis functions. L_j is defined by the property that

$$L_i(t_j) = \delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

where δ_{ij} is the Kronecker delta. In explicit form L_i of nth degree is given as

$$L_i(t) = \frac{(t-t_o)(t-t_1)...(t-t_{i-1})(t-t_{i+1})...(t-t_n)}{(t_i-t_o)(t_i-t_1)...(t_i-t_{i-1})(t_i-t_{i+1})...(t_i-t_n)} = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{(t-t_j)}{(t_i-t_j)}$$

Figure 3.3 shows Lagrange polynomials defined by the above formulation.

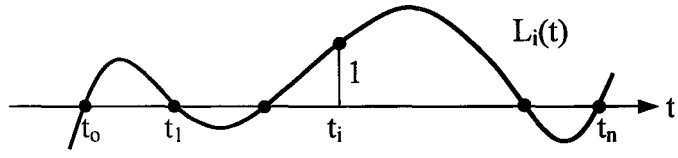


Figure 3.3. Lagrange polynomials.

For clarity, a second-degree Lagrange polynomial can be derived using the above formula:

$$p_2(t) = \sum_{j=0}^{2} L_j(t)P_j = \frac{(t-t_1)(t-t_2)}{(t_o-t_1)(t_o-t_2)}P_o + \frac{(t-t_o)(t-t_2)}{(t_1-t_o)(t_1-t_2)}P_1 + \frac{(t-t_o)(t-t_1)}{(t_2-t_o)(t_2-t_1)}P_2$$

The basic advantage of Lagrange interpolation technique is the computational simplicity though computational work increases with additional knots. All basis functions have to be recomputed. This disadvantage can be prevented using Newton basis functions described in the following section.

Another serious problem arises with increasing number of points ($n \geq 5$). The interpolating Lagrange polynomial shows oscillations around the defining points. For quite reasonable data points and parameter values, the polynomial interpolant exhibits wild wiggles that are not inherent in the data. Consequently, this method of interpolation is not shape preserving and can not be regarded as suitable to implement in a fairing procedure.

3.1.1.3. Newton Interpolation Polynomials

The general form of Newton interpolating polynomial is defined by :

$$p_n(t) = \sum_{j=0}^{n} n_j(t) A_j$$

The Newton polynomials are denoted by n_j and defined by the following expression:

$$n_i = (t - t_o)(t - t_1)...(t - t_{i-1}), \qquad n_o(t) = 1$$

The Newton polynomials are illustrated in **Figure 3.4.**

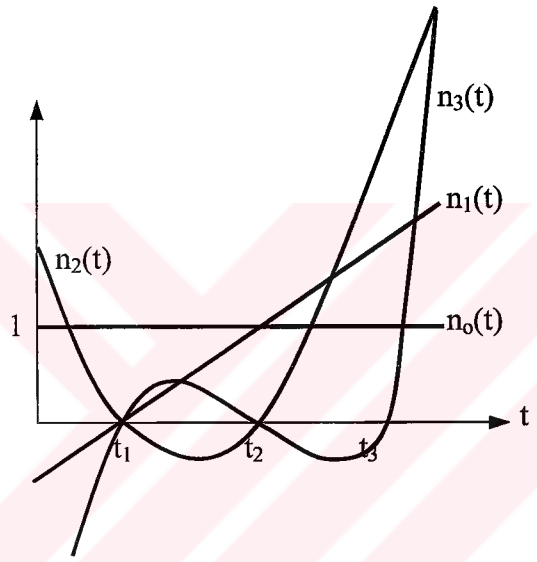


Figure 3.4. Newton polynomials.

where A_j are the coefficients for the interpolation problem and can be found directly using the following system of equations.

$$\begin{aligned}
p_n(t_o) &= P_o = A_o \\
p_n(t_1) &= P_1 = A_o + A_1(t_1 - t_o) \\
p_n(t_2) &= P_2 = A_o + A_1(t_2 - t_o) + A_2(t_2 - t_o)(t_2 - t_1) \\
&\vdots \\
p_n(t_n) &= P_n = A_o + A_1(t_n - t_o) + A_2(t_n - t_o)(t_n - t_1) + \cdots + A_n(t_n - t_o)\cdots(t_n - t_{n-1})
\end{aligned}$$

The coefficients are obtained as:

$$A_o = P_o$$
$$A_1 = \frac{P_1 - P_o}{t_1 - t_o}$$
$$A_2 = \frac{(P_2 - P_1)/(t_2 - t_1) - (P_1 - P_o)/(t_1 - t_o)}{t_2 - t_o}$$
$$\vdots$$

etc.

To make systematic calculations, the idea of *divided differences* is introduced:

$$[t_i\, t_k] = \frac{P_i - P_k}{t_i - t_k}$$

Then the coefficients of the Newton polynomial can be written as:

$$A_o = P_o$$
$$A_1 = [t_1\, t_o] = \frac{P_1 - P_o}{t_1 - t_o}$$
$$A_2 = [t_2\, t_1\, t_o] = \frac{[t_2\, t_1] - [t_1\, t_o]}{t_2 - t_o}$$
$$A_3 = [t_3\, t_2\, t_1\, t_o] = \frac{[t_3\, t_2\, t_1] - [t_2\, t_1\, t_o]}{t_3 - t_o}$$
$$\vdots$$
$$A_n = [t_n\, t_{n-1} \cdots t_1\, t_o] = \frac{[t_n \cdots t_1] - [t_{n-1} \cdots t_0]}{t_n - t_o}$$

The Newton polynomials are easy to implement in computational fairing procedures. However, when modification of the initial offset points is necessary, the method can not be considered as suitable because of its interpolating nature.

3.1.1.4. Hermite Interpolation Polynomials

Hermite interpolation is a generalised form of Lagrange interpolation. Moreover, higher-order derivatives are included to the interpolation problem. We have n+1 parameter values corresponding to offset points and higher-order derivatives, $P_k, P_k', \cdots, P_k^{m_k}$ respectively.

The Hermite interpolation curve should therefore satisfy,

$$\begin{aligned} H(t_k) &= P_k \\ H'(t_k) &= P'_k \\ &\vdots \\ H^{m_k}(t_k) &= P^{m_k} \end{aligned}$$

The solution of this interpolation problem should satisfy the above conditions with lowest possible degree polynomial. The degree of the curve should be at most:

$$N = \sum_{k=0}^{n} (m_k + 1) - 1$$

For simplicity, the maximum order derivative can be taken as $m_o = m_1 = \ldots = m_n = 1$ and the degree of polynomial becomes $N = 2n+1$. Thus, we can write the interpolation polynomial as:

$$H(t) = \sum_{j=0}^{n} P_j f_{nj}(t) + \sum_{j=0}^{n} P'_j g_{nj}(t)$$

where $f_{nj}(t)$ and $g_{nj}(t)$ are polynomials of degree $2n+1$. Then with $m_k=1$ we need

$$\begin{aligned} f_{nj}(t_k) &= \delta_{jk} & g_{nj}(t_k) &= 0 \\ f'_{nj}(t_k) &= 0 & g'_{nj}(t_k) &= \delta_{jk} \end{aligned} \qquad j,k = 0,1,\ldots,n$$

Polynomials satisfying the above conditions can be constructed in terms of the Lagrange polynomials

$$L_{nk}(t) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{(t - t_j)}{(t_k - t_j)} \qquad j = 0,1,\ldots,n$$

Thus, the polynomials $f_{nj}(t)$ and $g_{nj}(t)$ are defined as:

$$\begin{aligned} f_{nk}(t) &= \left[1 - 2L'_{nk}(t_k)(t - t_k)\right] L^2_{nk}(t) \\ g_{nk}(t) &= (t - t_k) L^2_{nk}(t) \end{aligned} \qquad k = 0,1,\ldots,n$$

An advantage of this interpolation technique is that not only position vectors but also tangent vectors are under control. With a good choice of derivatives at the boundary

points, the Hermite interpolation curve can produce satisfactory results. However, the main disadvantages are; for n+1 position vectors it has the degree N=2n+1 (much higher than Lagrange's), thus it tends to oscillate around the defining interpolatory points P_k. Also, the position vectors for the curve or surface measured from a model or drawing are relatively more reliable than tangent vectors. Consequently, poor representations can likely to be obtained since the derivative values have great influence on the shape of the curve. To avoid these drawbacks, the idea had arisen to abandon the infinite differentiability property (a feature of ordinary polynomials) and to replace full Hermite polynomial by piecewise Hermite polynomials, which is known as Ferguson interpolation. It can give first-derivative continuity without severe oscillation problems.

3.1.2. Approximating Polynomials

In a fairing problem, where some of the offset points need to be altered to improve fairness characteristics, the interpolating polynomials would not be useful as they pass through the offset points. Moreover, defining offset points for sections or waterlines are usually large and it may be impossible to find an interpolating polynomial of suitable order. Even if such a polynomial exists it may have a tendency to oscillate. Therefore, alternative techniques are clearly needed to represent ship lines, which may approximate the curve rather than interpolating. One such approach, known as the least squares method, is based on the minimising the square of the differences between the polynomial and the offset points. The details of this approach are given in the following section.

3.2.2.1. Least Squares Polynomial

The most popular of the approximating functions is the least squares method. The aim is the minimise the sum of the squares of the discrepancies between the unknown function defined by n+1 points P_i and the approximating function. If the deviation δ_i at the ith offset P_i from the approximating function $X(t_i)$ is given by

$$\delta_i = X(t_i) - P_i$$

Then, the least squares criterion is to minimise the overall error of approximation

$$E = \sum_{i=0}^{n} |\delta|^2 = \sum_{i=0}^{n} (X(t_i) - P_i)^2$$

It may be desirable to introduce weights w_i so that different weight factors can be assigned for each initial offset point.

$$E = \sum_{i=0}^{n} w_i (X(t_i) - P_i)^2$$

We can introduce $X(t_i)$ may have the form

$$X(t_i) = \sum_{j=0}^{m} A_j \phi(t) \qquad m < n$$

where A_j are the coefficients and $\phi(t)$ are basis functions. If m is selected to be equal to n, the error E is exactly zero and the least squares polynomial is identical with interpolating polynomial. So m should be less than n.

We try to determine the coefficients of the approximating polynomial so that error E is minimised. The necessary condition for the overall error to be minimised:

$$\frac{\partial E}{\partial A_k} = -2\sum_{i=0}^{n} w_i (P_i - X(t_i)) \frac{\partial X(t_i)}{\partial A_k} = -2\sum_{i=0}^{n} w_i (P_i - X(t_i))\phi_k(t_i) = 0$$

Substituting the expression of $X(t_i)$ in this equation leads to the system of linear equations:

$$\underbrace{\sum_{i=0}^{n} w_i P_i \phi_k(t_i)}_{[P,\phi_k]} = \sum_{j=0}^{m} A_j \underbrace{\sum_{i=0}^{n} w_i \phi_j(t_i)\phi_k(t_i)}_{[\phi_j,\phi_k]} \qquad k = 0,1,\ldots,m$$

Then, this system of linear equations can be written in matrix form:

$$\begin{bmatrix} [\phi_o,\phi_o] & \cdots & [\phi_o,\phi_m] \\ [\phi_1,\phi_o] & \cdots & [\phi_1,\phi_m] \\ \vdots & \ddots & \vdots \\ [\phi_m,\phi_o] & \cdots & [\phi_m,\phi_m] \end{bmatrix} \begin{bmatrix} A_O \\ A_1 \\ \vdots \\ A_m \end{bmatrix} = \begin{bmatrix} [P,\phi_o] \\ [P,\phi_1] \\ \vdots \\ [P,\phi_m] \end{bmatrix}$$

This system of linear equations does determine the A_i uniquely, and the solution coefficients do actually produce the minimum possible value of E. However, standard methods for solving linear systems may either produce no solution at all, or magnify data errors due to ill conditioning for large number of offsets.

Low degree polynomials both reduce the computational requirements and also reduce numerical instabilities that arise with higher degree curves. Clearly, these instabilities cause undesirable oscillations. Though low degree polynomials seem advantageous under these circumstances, reasonable representations can not be achieved due to large deviations from the original points. Increasing the degree of the polynomial to obtain a better fit would be likely to lead to oscillatory problems.

An example is given in **Figure 3.5**, where least squares polynomial is applied to a hard chine section for varying degrees. The drawbacks of this method such as oscillatory nature with increasing degree, and inability to represent knuckles etc. can clearly be seen from the figure.

We can conclude that polynomials both interpolating and approximating are unsuitable to implement in a representation or fairing procedure. It is possible to generate a curve of degree n-1 provided that the offset points of n is given. However, the solution curve may have a tendency to oscillate due to ill conditioning. Also, another serious problem arises when a ship line with straight parts is considered. Polynomials can not contain straight parts unless the entire curve is straight. In this case, the curve can be subdivided to overcome this problem, with the expense of more computational effort.

Further descriptions of polynomials and applications to typical ship sections and waterlines, e.g., bulbous, chine, etc. can be found in **Narlı (1995)**.

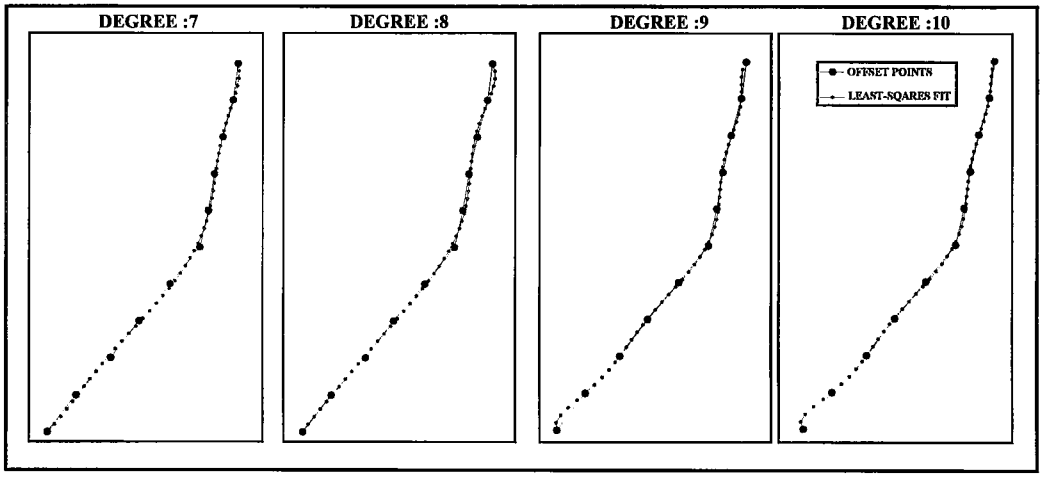


Figure 3.5. Application of least squares polynomial to a hard chine section.

3.2. Spline Techniques in Hull Form Design

There is a substantial class of curves and surfaces based either on the interpolation or on the approximation of points that are supplied by the user. However, the interpolation and approximation methods discussed in previous chapters are not adequate for the representation of ship lines. It is a general requirement that the curvature of the curves and surfaces should not vary too much, i.e., it should appear to be smooth. Clearly, most of the classical interpolating functions, e.g., polynomials have a tendency to oscillate illustrating large deviations from the desired curve, thus it may not be a good idea to use polynomials to fit free form ship curves. An entire curve is not easily defined by a single analytic function. One way to improve this situation is to divide the given interval into smaller sub-intervals, and to construct an approximating curve consisting of pieces of curves. Each interval is described by a separate analytic function to form a piecewise representation. Implementation of this approach produces smoother curves where the curvature of the curve shows variation diminishing property. This piecewise approach is employed in *spline curves and surfaces* in general for reasons of flexibility. Hence, parametric spline curves and parametric tensor product surfaces have become standard tools in CAGD due to their distinguished properties. Different processes for defining splines have been developed. All spline functions can be expressed by a linear combination of basis functions. These basis or blending functions differ from simple cubic splines to Bernstein and Basis spline functions, each having its own advantages as well as limitations.

The spline curves may be characterised especially in two categories: *curve fitting* and *curve approximating (or fairing)* techniques. The former is characterised by the fact that the derived mathematical curve passes through each and every data point. A mathematical curve is generally obtained from given or digitised points. (cubic splines, parabolically blended curves etc.) Alternatively, for the latter, the basics of the approach stems from the fact that the mathematical description of a space curve is generated "ab initio", i.e., without any prior knowledge of the curve shape or form. Bezier curves and their powerful generalisation to B-spline curves are the distinguished specimens of this category. These two techniques are characterised by the fact that few if any points on the curve pass through the control points used to define the curve. These methods are frequently referred to as *curve fairing or control polygon techniques* and have become standard tools in geometric modelling of free form curves and surfaces.

3.2.1. Cubic Splines

The word spline, an East Anglican dialect word as stated in **Bartels et al (1987)** is derived from the name of a tool used by loftsman in ship design offices for ship lofting process. Hence, the spline function is the mathematical model of this drafting tool. The physical spline is a thin elastic beam, usually made of wood, metal or plastic and used to draw smooth curves through given points. The material spline is held fixed in certain places by heavy weights called ducks, which constrain the curve to pass through these points. The type of support may be regarded as hinged, thus the system of spline and ducks may be accepted as a simply supported, thin, continuous elastic beam. Assuming small deflections and using linearised beam theory one can show:

- The deflection in each span can be represented by a cubic polynomial.
- The bending moments, which are proportional to the second derivatives of the deflection are continuous through the entire length of the spline, that is, the curve is C^2 continuous.
- The third derivatives are discontinuous at the supports unless the supporting force happens to be zero.

Furthermore, it will be useful to note that this property of cubic splines is referred to as minimal curvature property and, among all interpolating curves y(x), the cubic spline is the curve which minimises $\int_0^{\ell} (y'')^2 dx$.

A physical spline can be considered as a thin elastic beam, therefore it follows the Euler's equation and the bending moment along the length of the beam is obtained as follows;

$$M(x) = \frac{EI}{R(x)}$$

where

M(x) : bending moment along the length of the beam

E : Young's modulus of elasticity determined by the material properties of the beam

I : moment of inertia determined by the cross sectional shape of the beam

R(x) : radius of curvature of the beam

Radius of curvature for the beam is expressed as

$$\frac{1}{R(x)} = \frac{y''}{(1+(y')^2)^{3/2}}$$

The deformations of the spline is assumed to be sufficiently small; ($y' << 1$), hence R(x) simply reduces to

$$R(x) = \frac{1}{y''(x)}$$

Where y'' is the second derivative of the mathematical spline curve y(x) with respect to x. Therefore,

$$y''(x) = \frac{M(x)}{EI}$$

If the weights are assumed to be simple supports then the variation of the bending moment M(x) between the weights becomes a linear function with A and B the constants as follows:

$$M(x) = Ax + B$$

Substituting this into the above equation yields

$$y''(x) = \frac{Ax + B}{EI}$$

Integrating this twice shows the physical spline to be cubic polynomial functions between each pair of adjacent supports as follows:

$$y(x) = Ax^3 + Bx^2 + Cx + D$$

Thus, mathematical counterpart of physical spline can be modelled by using cubic polynomials between weights. In **Figure 3.6** the elastic spline and ducks are illustrated.

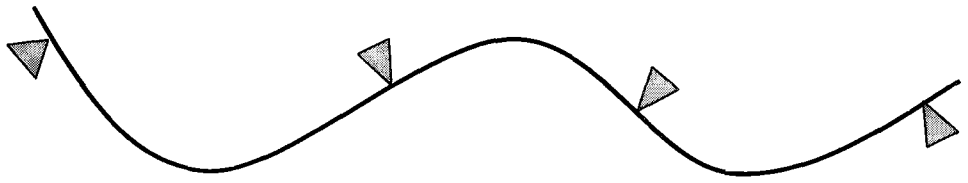


Figure 3.6. Physical spline and weights.

It is shown above that the equation for a single parametric cubic spline segment is given by

$$P(t) = \sum_{i=1}^{4} B_i t^{i-1} = B_1 + B_2 t + B_3 t^2 + B_4 t^3 \qquad t_1 \le t \le t_2$$

where t_1 and t_2 are parameter values at the boundaries of the segment. P(t) is the position vector of any point on the cubic spline segment. Thus, P(t) is a vector-valued function having the components [x(t), y(t), z(t)]. The coefficients B_i of the spline segment is derived by specifying four boundary conditions.

The terms defined at the boundaries of the each spline segments are denoted by the following notations given in **Table 3.2.**

Table 3.2. The boundary terms of spline segments.

Position Vectors	Tangent Vectors	Curvature Vectors
$P_1, P_2, \ldots, P_n$	$P_1', P_2', \ldots, P_n'$	$P_1'', P_2'', \ldots, P_n''$

where n is the number of points. Differentiating the spline segment equation yields:

$$P'(t) = \sum_{i=1}^{4} B_i (i-1) t^{(i-2)} = B_2 + 2B_3 t + 3B_4 t^2 \quad t_1 \le t \le t_2$$

If we assume $t_1 = 0$, the four boundary conditions for the segment will become

$$P(0) = P_1 \qquad P(t_2) = P_2 \qquad P'(0) = P_1' \qquad P'(t_2) = P_2'$$

Solving for unknown B_i yields

$$P(0) = P_1 = B_1$$
$$P'(0) = P_1' = B_2$$
$$P(t_2) = P_2 = B_1 + B_2 t_2 + B_3 t_2^2 + B_4 t_2^3$$
$$P'(t_2) = P_2' = B_2 + 2B_3 t_2 + 3B_4 t_2^2$$

B_3 and B_4 are obtained using the above equations,

$$B_3 = \frac{3(P_2 - P_1)}{t_2^2} - \frac{2P_1' + P_2'}{t_2} \qquad B_4 = \frac{2(P_1 - P_2)}{t_2^3} + \frac{P_1' + P_2'}{t_2^2}$$

Thus, we can obtain the equation for a single cubic spline segment in terms of position and tangent vectors as follows:

$$P(t) = P_1 + P_1' t + \left[\frac{3(P_2 - P_1)}{t_2^2} - \frac{2P_1' + P_2'}{t_2}\right] t^2 + \left[\frac{2(P_1 - P_2)}{t_2^3} + \frac{P_1' + P_2'}{t_2^2}\right] t^3$$

Since this equation defined for only one segment, in order to obtain the overall curve equation, these adjacent piecewise segments should be combined. Two adjacent segments can be seen in **Figure 3.7**.

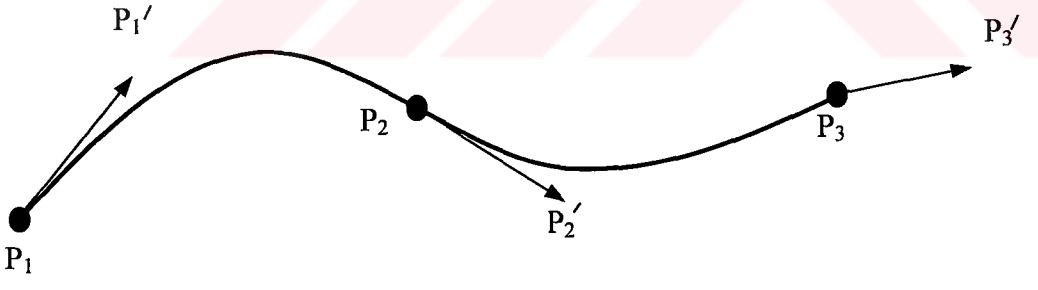


Figure 3.7. Two adjacent segments.

The same equation holds for both segments, provided that the position vectors P_1, P_2, P_3, the tangent vectors P_1', P_2', P_3' and the parameter values t_2, t_3 are known. However, it is unlikely that the tangent vector P_2' at the intermediate point is known, but it can be determined by imposing the second order continuity condition for the cubic spline curves. Thus, differentiating the spline segment equation yields

$$P''(t) = \sum_{i=1}^{4} B_i (i-1)(i-2) t^{(i-3)} = 2B_3 + 6B_4 t \quad t_1 \le t \le t_2$$

The parameter range of the first spline segment $P_1(t)$ is $t_1 = 0 \le t \le t_2$ and the curvature at the end of the first segment is:

$$P_1''(t_2) = 2B_3 + 6B_4 t_2$$

The parameter range of the second spline segment $P_2(t)$ is $t_2 = 0 \le t \le t_3$ and the curvature at the beginning of the second segment is:

$$P_2''(0) = 2B_3$$

For any two adjacent spline segments, equating the second derivatives at the common internal joint (end of the first segment and beginning of the second segment):

$$P_1''(t_2) = P_2''(0) \quad \Rightarrow \quad [2B_3 + 6B_4 t_2]_1 = [2B_3]_2$$

$$2\left[\frac{3(P_2 - P_1)}{t_2^2} - \frac{2P_1' + P_2'}{t_2}\right] + 6t_2\left[\frac{2(P_1 - P_2)}{t_2^3} + \frac{P_1' + P_2'}{t_2^2}\right] = 2\left[\frac{3(P_3 - P_2)}{t_3^2} - \frac{2P_2' + P_3'}{t_3}\right]$$

By collecting tangent terms at one side, the unknown tangent vectors can easily be obtained:

$$t_3 P_1' + 2(t_2 + t_3)P_2' + t_2 P_3' = \frac{3}{t_2 t_3}\left[t_2^2(P_3 - P_2) + t_3^2(P_2 - P_1)\right]$$

For simplification, the above equation can be written in matrix formulation and the unknown tangent vector P_2' can easily be computed.

$$\begin{bmatrix} t_3 & 2(t_3 + t_2) & t_2 \end{bmatrix} \begin{bmatrix} P_1' \\ P_2' \\ P_3' \end{bmatrix} = \left[\frac{3}{t_2 t_3}\left[t_2^2(P_3 - P_2) + t_3^2(P_2 - P_1)\right]\right]$$

These results can be generalised for n data points to give n-1 piecewise cubic spline segments. **Figure 3.8** illustrates n piecewise cubic segments.

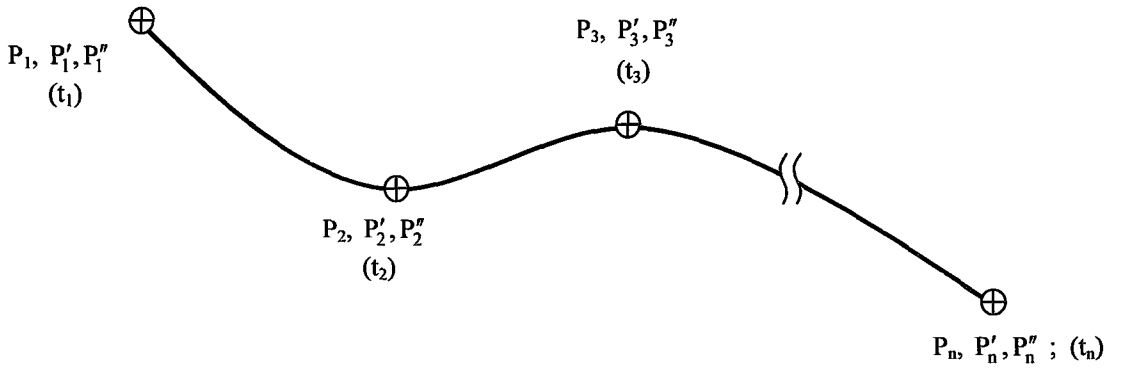


Figure 3.8. Illustration of n piecewise cubic segments.

Similarly, the linear system of equations can be constructed for n-1 piecewise cubic segments. Although the problem is indeterminate, with the assumption that end tangent vectors P_1', P_n' are known, the problem becomes determinate and can be written in matrix notation as:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ t_3 & 2(t_2+t_3) & t_2 & 0 & \cdots & 0 \\ 0 & t_4 & 2(t_3+t_3) & t_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & t_n & 2(t_n+t_{n-1}) & t_{n-1} \\ 0 & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_1' \\ P_2' \\ P_3' \\ \vdots \\ \vdots \\ P_n' \end{bmatrix} = \begin{bmatrix} P_1' \\ \frac{3}{t_2 t_3}\left[t_2^2(P_3 - P_2) + t_3^2(P_2 - P_1)\right] \\ \vdots \\ \vdots \\ \frac{3}{t_{n-1}t_n}\left[t_{n-1}^2(P_n - P_{n-1}) + t_n^2(P_{n-1} - P_{n-2})\right] \\ P_n' \end{bmatrix}$$

Once the tangent vectors are P_k' are computed, the B_i coefficients for each spline can be determined as follows:

$$B_{1k} = P_k,$$
$$B_{2k} = P_k',$$
$$B_{3k} = \frac{3(P_{k+1} - P_k)}{t_{k+1}^2} - \frac{2P_k' + P_{k+1}'}{t_{k+1}},$$
$$B_{4k} = \frac{2(P_k - P_{k+1})}{t_{k+1}^3} + \frac{P_k' + P_{k+1}'}{t_{k+1}^2}$$

So the piecewise cubic spline equation can be written for each spline segment in matrix form as:

$$P_k(t) = \sum_{i=1}^{4} B_{ik} t^{i-1} = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} B_{1k} \\ B_{2k} \\ B_{3k} \\ B_{4k} \end{bmatrix} \quad 0 \le t \le t_{k+1}, \quad 1 \le k \le n-1$$

To simplify the calculations, the coefficients B_{ik} can be substituted in the above equation and rearranged to give the overall cubic spline equation as follows:

$$P_k(u) = \begin{bmatrix} F_1(u) & F_2(u) & F_3(u) & F_4(u) \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ P'_k \\ P'_{k+1} \end{bmatrix} \quad 0 \le u \le 1, \quad 1 \le k \le n-1$$

where $F_{ik}(u)$ are called blending functions, and can be written as:

$$F_{1k}(u) = 2u^3 - 3u^2 + 1$$
$$F_{2k}(u) = -2u^3 + 3u^2$$
$$F_{3k}(u) = u(u-1)^2 t_{k+1}$$
$$F_{4k}(u) = u(u^2 - u) t_{k+1}$$

Figure 3.9 shows the blending functions for $t_{k+1} = 1$.

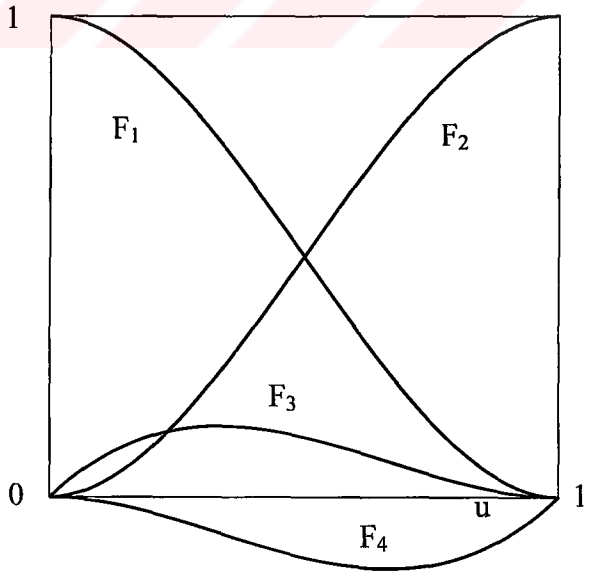


Figure 3.9. Cubic spline blending functions.

Consequently, it has been shown that a piecewise cubic spline curve is determined by the position vectors, tangent vectors, and the parameter values.

The parametric piecewise cubic spline curve is advantageous since it is the lowest degree curve which allows a point of inflection and which has the ability to twist through space. Although the use of cubic spline has proven useful in shipbuilding industry, a number of disadvantages still exist. Specifically, parametric cubics never reduce exactly to a conic section, they poorly approximate asymptotic curves, and they can exhibit oscillations. (e.g., the continuation of a straight line by a circular arc) The reason of the oscillations is due to its global nature. The cubic spline is influenced by all defining data points, a change in any one segment affects all segments, hence a local modification involves the recomputation of the entire spline. Consequently, there is lack of local control over the curve shape. Moreover, the third derivative is piecewise continuous. Discontinuities in third derivative can thus induce unwanted inflection points at certain locations along the curve. Although such an oscillating curve is C^2 continuous, it is unacceptable.

Four typical ship sections are chosen for the application of cubic spline. These sections and their first and second derivatives are shown in **Figure 3.10.** Cubic spline representation of these sections are shown in **Figure 3.11** with their corresponding derivatives. The drawbacks of cubic spline fitting at corners and flat parts are clearly visible.

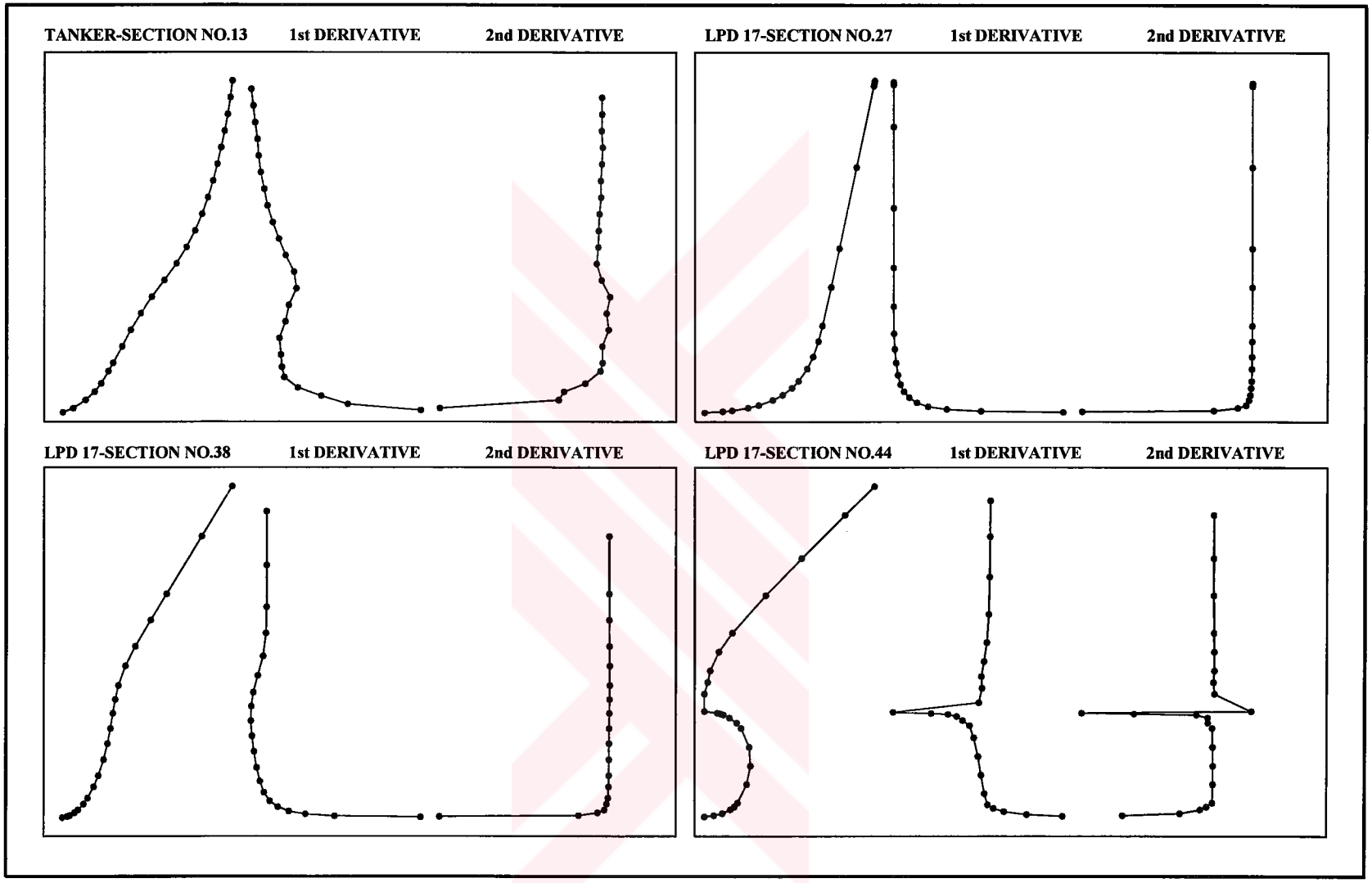


Figure 3.10. Typical ship sections with first and second derivatives.

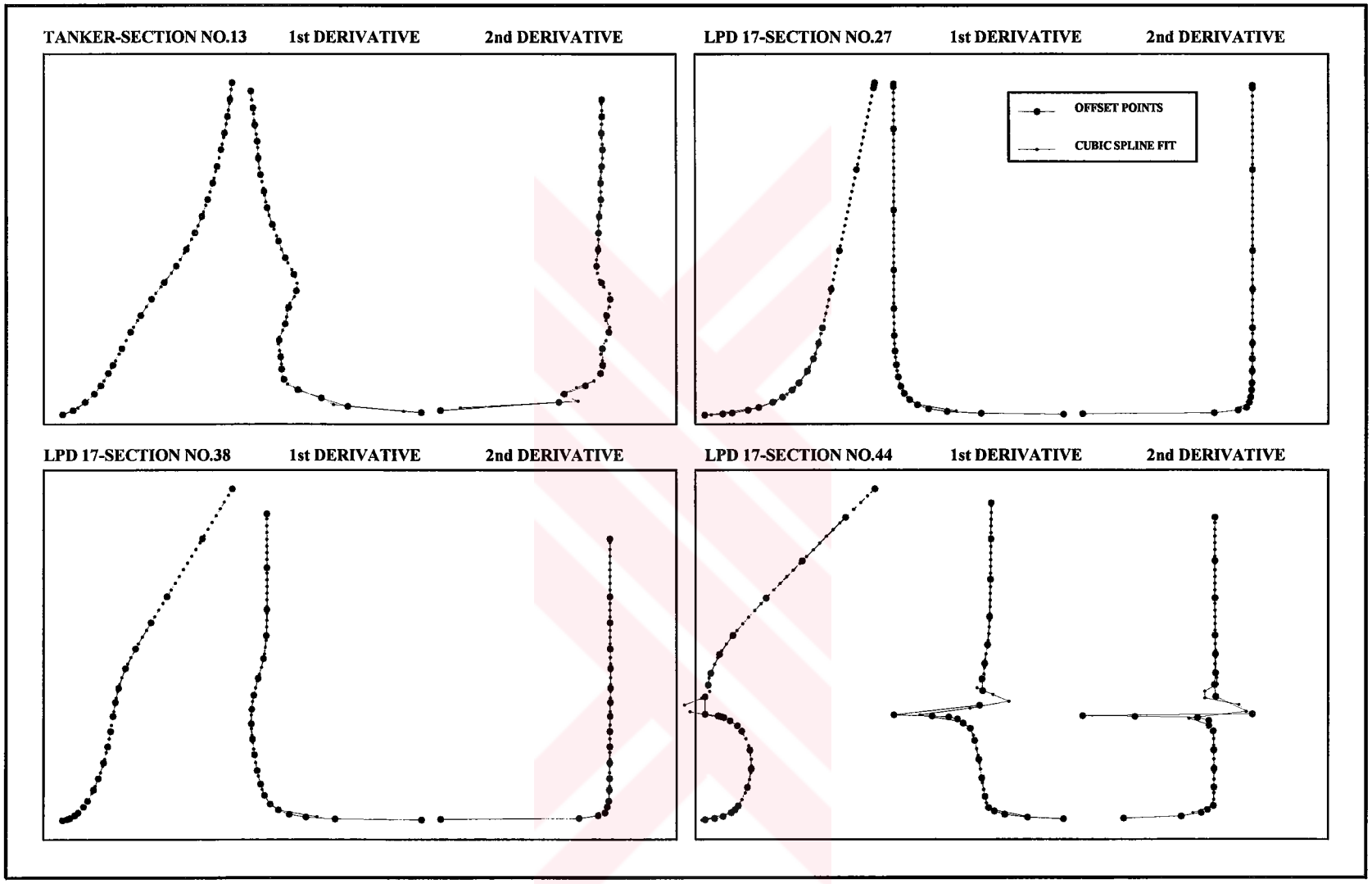


Figure 3.11. Cubic spline representation of typical ship sections.

3.2.2. Bezier Curves

Bezier curves and surfaces, originally developed by **Bezier (1966)**, constitute one of the earliest attempts to develop a flexible and intuitive interface for computer aided design. Bezier curves and surfaces have been used for many years by Renault for designing outer panels of automobiles. Also, this powerful control polygon technique has found wide applications in shipbuilding industry for defining ship curves and surfaces. The difference in this technique from aforementioned curve fitting techniques is that the curve is obtained from its defining polygon and not constrained to pass through given data points. Thus, the limitations of the interpolating methods can be overcome by using Bezier curves. The shape and the order of the curve is controlled by the use of easily controlled input parameters. A Bezier curve is associated with the vertices of a control polygon namely, the Bezier polygon which uniquely defines the shape of the curve. The curve shape tends to follow the shape of the polygon.

Bezier curves are defined by a recursive algorithm, known as *de Casteljau* algorithm. This algorithm is probably the most fundamental one in the field of curve and surface design, although it is surprisingly simple. The essence of the algorithm is repeated linear interpolation. This simple and very intuitive geometric construction leads to a powerful theory.

De Casteljau Algorithm

The *de Casteljau* algorithm can be derived geometrically. Lets consider three control polygon points; B_o, B_1, B_2 and obtain the quadratic expression illustrated in **Figure 3.12**.

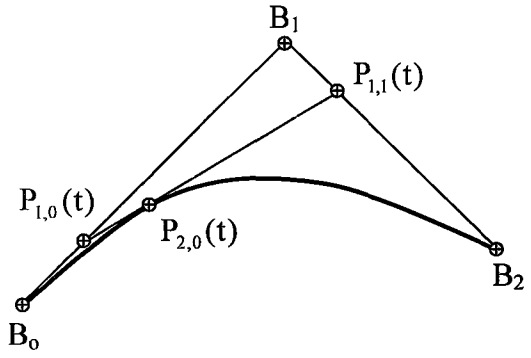


Figure 3.12. The de Casteljau algorithm.

We can construct the polynomial equations $P_i(t)$ using these polygon vertices B_i by linear interpolation as follows:

$$P_{1,0}(t) = (1-t)B_o + tB_1$$
$$P_{1,1}(t) = (1-t)B_1 + tB_2$$
$$P_{2,0}(t) = (1-t)P_{1,0}(t) + tP_{1,1}(t)$$

Substituting the first two equations into the third yields the quadratic equation of the curve formed by three defining polygon vertices, B_o, B_1, B_2

$$P_{2,0}(t) = (1-t)^2 B_o + 2t(1-t)B_1 + t^2 B_2$$

where the first subscripts denote the degree of the curves. It can clearly be seen from **Figure 3.12** that $P_{2,0}(t)$ lies within the convex hull of the defining polygon and the polygon formed by B_0, B_1, B_2 is called the Bezier or control polygon of the Bezier curve $P_{2,0}(t)$.

A sequence of linear interpolation yields a polynomial curve of arbitrary degree n:

$$P_{r,i}(t) = (1-t)P_{r-1,i}(t) + t\,P_{r-1,i+1}(t) \qquad r = 1,2,...,n \qquad i = 0,1,...,n-r$$

However, to facilitate further developments, it is generally required to have an explicit representation of Bezier curves. So, this recursive algorithm should be extended to yield mathematical basis of Bezier curves defined by non-recursive formula of Bernstein polynomials.

Thus, Bezier curves are expressed in terms of *Bernstein polynomials,* $J_{n,i}(t)$ defined explicitly by,

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

where the binomial coefficients are given by

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

$J_{n,i}(t)$ is the i^{th} n^{th} order Bernstein basis function. They are constructed to yield the following recursion of *de Casteljau* algorithm:

$$J_{n,i}(t) = (1-t)J_{n-1,i}(t) + tJ_{n-1,i-1}(t)$$

A property of Bernstein polynomials is that for any given value of the parameter t, the summation of the basis function is precisely one.

$$\sum_{i=0}^{n} J_{n,i}(t) = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} = [t + (1-t)]^n = 1$$

The relation among Bernstein polynomials is illustrated in **Figure 3.13.**

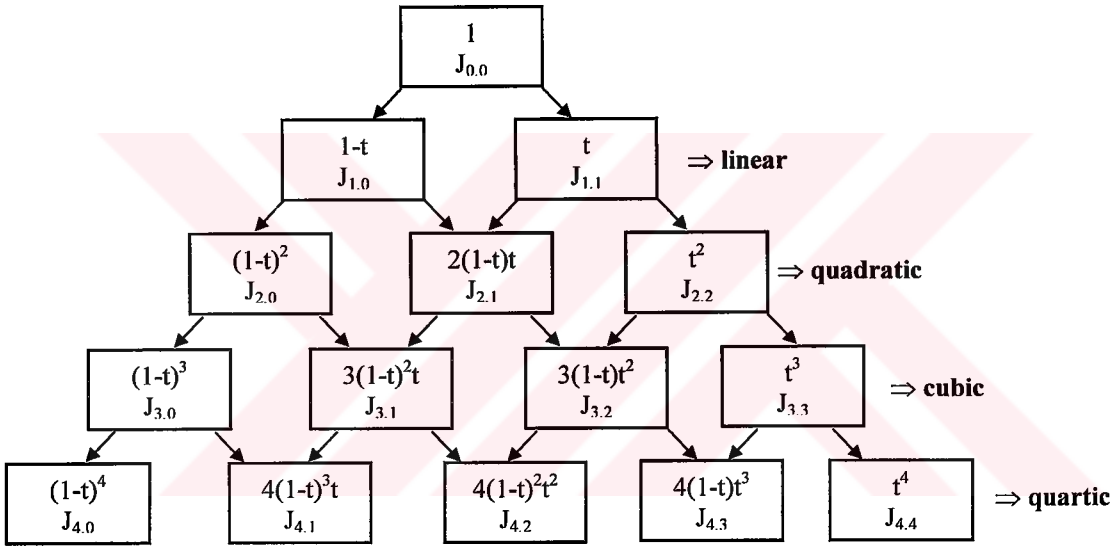


Figure 3.13. The relation among varying degrees of Bernstein polynomials.

Bernstein polynomials for quadratic and cubic cases can be seen in **Figure 3.14.**

A parametric Bezier curve is defined by

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t) \qquad 0 \le t \le 1$$

where B_i represent the position vectors of the n+1 vertices of the characteristic polygon of the Bezier curve and $J_{n,i}(t)$ Bernstein basis functions. This curve passes only through first and last points of the defining polygon; B_0 and B_n, and the tangents are in the direction of the vectors B_0B_1 and $B_{n-1}B_n$, as shown in **Figure 3.15.**

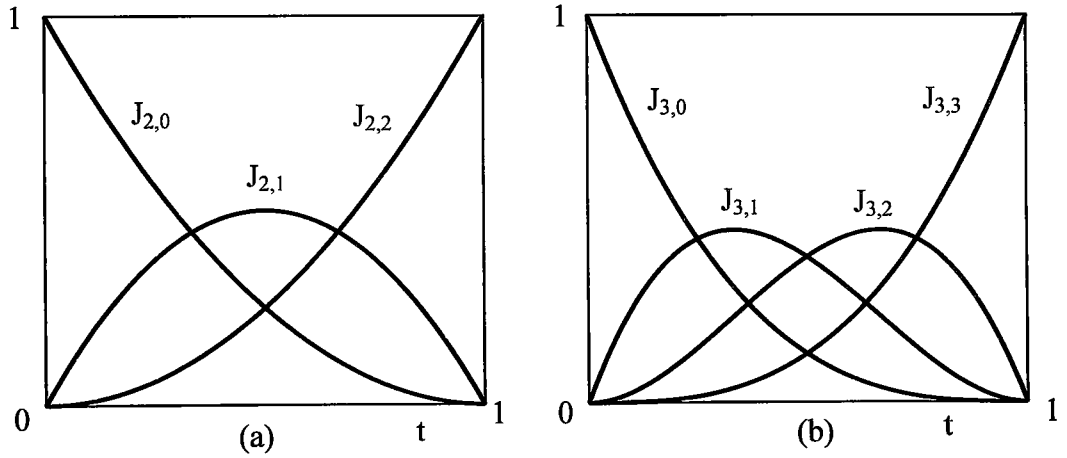


Figure 3.14. Bernstein blending functions. (a) quadratic case (n=2), three polygon points, (b) cubic case (n=3), four polygon points.

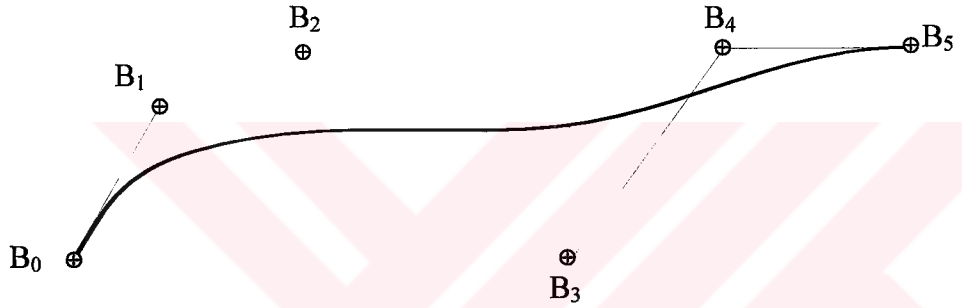


Figure 3.15. An example of a fifth degree Bezier curve and its defining polygon.

The equation of the Bezier curve can also be expressed in a matrix form as follows:

$$P(t) = [J][G]$$

where $[J] = [J_{n,0}\ J_{n,1}\ \ldots\ J_{n,n}]$ and $[G] = [B_0\ B_1\ \ldots\ B_n]^T$. As an example cubic case can be considered (n=3), and defined in matrix notation:

$$P(t) = \begin{bmatrix}(1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3\end{bmatrix}\begin{bmatrix}B_0\\B_1\\B_2\\B_3\end{bmatrix}$$

By collecting the parameter coefficients, the above equation for cubic Bezier curves can be decomposed into a more convenient form:

$$P(t) = [T][N][G] = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

This matrix representation can be generalised as:

$$[T] = \begin{bmatrix} t^n & t^{n-1} & \cdots & t & 1 \end{bmatrix}$$

The terms of the matrix N(n,n) are derived from the following equation:

$$N_{i+1,j+1} = \begin{cases} \binom{n}{j}\binom{n-j}{n-i-j}(-1)^{n-i-j} & 0 \le i+j \le n \\ 0 & \text{else} \end{cases}$$

where i, j denote rows and columns, respectively.

$$[N] = \begin{bmatrix} \binom{n}{0}\binom{n}{n}(-1)^n & \binom{n}{1}\binom{n-1}{n-1}(-1)^{n-1} & \cdots & \binom{n}{n}\binom{n-n}{n-n}(-1)^0 \\ \binom{n}{0}\binom{n}{n-1}(-1)^{n-1} & \binom{n}{1}\binom{n-1}{n-2}(-1)^{n-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \binom{n}{0}\binom{n}{1}(-1)^1 & \binom{n}{1}\binom{n-1}{0}(-1)^0 & \cdots & 0 \\ \binom{n}{0}\binom{n}{0}(-1)^0 & 0 & \cdots & 0 \end{bmatrix}$$

$$[B] = \begin{bmatrix} B_0 & B_1 & \cdots & B_{n-1} & B_n \end{bmatrix}^T$$

Basic properties of Bezier curves can be summarised as follows:

- The degree of the Bernstein basis polynomial is one less than the number of the vertices of the control polygon.
- The Bezier curve segments are convex combinations of the control polygon vertices, hence lies within the convex hull of the defining polygon (*convex hull property*).

- Only the first and last vertices of the control polygon lie on the curve (*end point interpolation*), the other vertices define the shape, order and derivatives of the curve.

- The tangent vectors at the ends of the curve have the same direction as the first and last polygon spans, respectively.

- The curve exhibits *variation-diminishing* property. Hence, no plane has more intersections with the Bezier curve than with its characteristic polygon, in other words, Bezier curve wiggles no more than its control polygon.

- The curve is invariant under affine transformation (*affine invariance*). The shape of a Bezier curve is determined only by its vertices, independent of any coordinate system.

- A multiple vertex will act as a weighting factor, and the curve will follow the polygon closer in that region at the expense of obtaining higher degree curves.

- Raising the degree of the defining polynomial curve by adding an additional vertex to the defining polygon increases the flexibility of a Bezier curve.

- A change in a vertex affects the whole Bezier curve and re-computation of all points on the curve is required.

Two characteristics of the Bernstein Basis Bezier curves limit the flexibility of the resulting curves. First, the number of the specified polygon vertices fixes the degree of the curve. The second limitation is due to the global nature of Bernstein polynomials. All of the defining polygon vertices influence the shape of the curve hence the curve is a blend of their values. This lack of local control eliminates the ability to perform a local change within the curve.

Using the above formulation, Bezier splines are applied to typical ship sections as can be seen in **Figure 3.16**. The applications indicate that Bezier curves can provide far representations with some deviation from the original offset points. This limits the application of Bezier curves and surfaces in mathematical representation of ship lines. However, this method has found wide application areas in the field of *ab initio* design where an experienced designer creates a new hull form from limited data available.

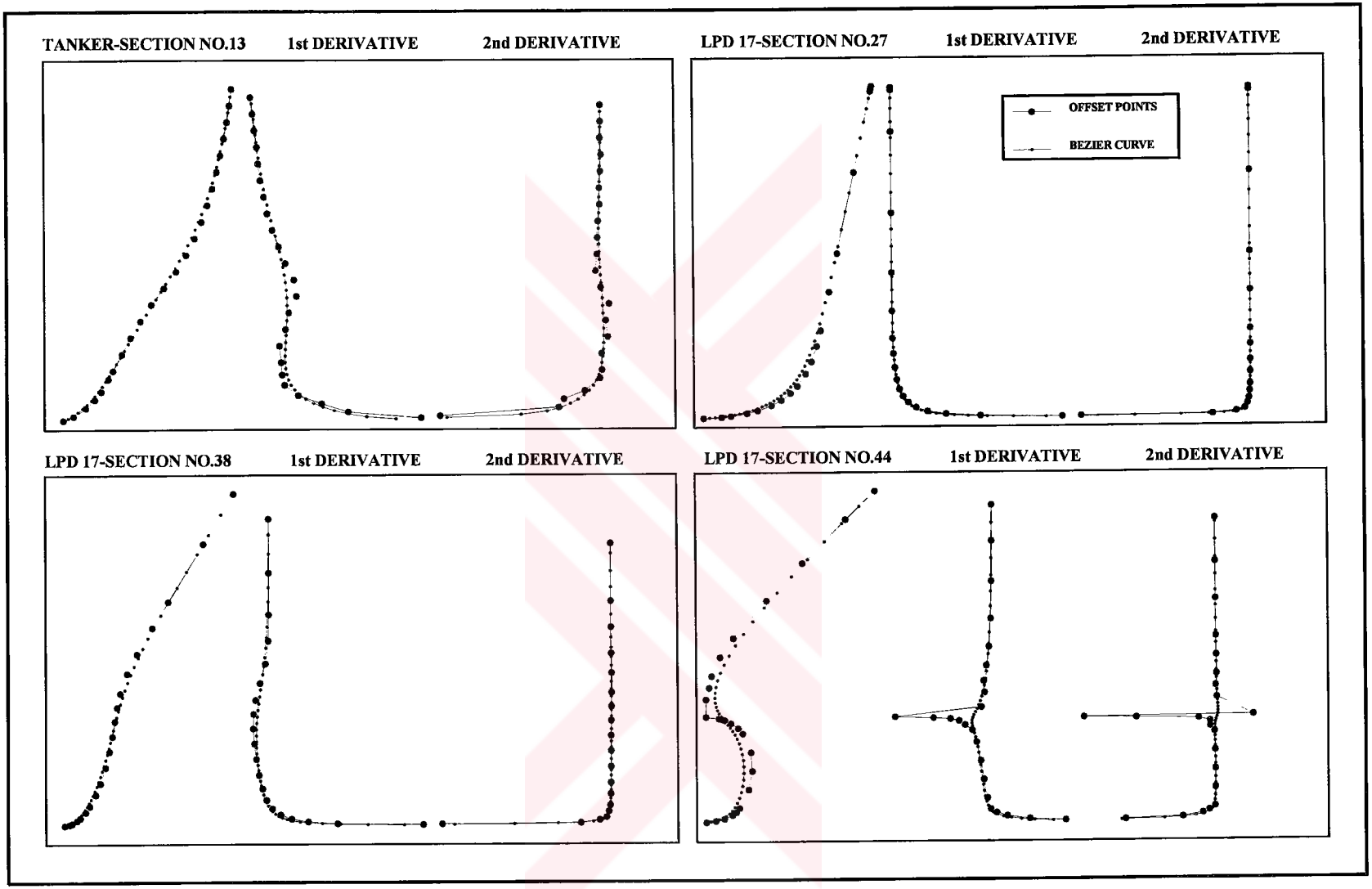


Figure 3.16. Bezier approximation for typical ship sections.

3.2.3. B-Spline Techniques

Many 3D computer graphics modelling and animation systems use a B-spline representation for curves and surfaces because of their unique geometric properties such as smoothness and controllable C^n parametric continuity between patches. These advantages have made B-spline curves and surfaces popular in CASD applications and today many ship hull form design software use B-spline techniques.

3.2.3.1. B-Spline Curves

B-spline curves, within the context of approximation techniques, are the result of a further intensive development of the Bezier curves described in the previous subsection. They are a special class of polynomials with unique properties that result in several practical advantages in comparison to other spline functions and polynomials. The Bernstein basis blending functions used in the Bezier approximation are replaced with a polynomial spline basis that has superior characteristics. **Schoenberg (1946)**, introduced the polynomial spline called the B-spline and used for statistical data smoothing. His paper started the modern theory of spline approximation but it was applied to Computer Aided Geometric Design (CAGD) by **Riesenfeld (1973)** and **Gordon and Riesenfeld (1974)**. The application of this theory in CAGD has made it possible to avoid the drawbacks of Bezier curves. Specifically, fixed degree of curves due to the number of defining polygon vertices, and global nature of Bernstein basis. Higher flexibility and superior geometric properties of B-spline curves have caused it to become a standard tool of computational geometry and geometric modelling.

A B-spline curve or surface is specified by a set of points called the control vertices. Although these vertices do not generally lie on the generated surface curve or surface, like in Bezier curves, their positions completely determine its shape. These defining vertices form the control polygon of the curve. The generated curve or surface tends to mimic the overall shape of the control polygon, and manipulation of a control vertex causes a modification in the resulting shape over a limited range.

B-spline basis also contains the Bernstein basis as a special case but B-spline functions are defined locally. This non-global behaviour is due to the fact that each vertex B_i is associated with a unique basis function. Thus, each vertex affects the shape of a curve only over a range of parameter values where its associated basis function is non-zero. The B-spline basis also allows the order of the basis function and hence the degree of the resulting curve to be changed without changing the

number of defining polygon vertices. Moreover, it is possible to introduce new control points without increasing the polynomial degree, and even to change the order of continuity between neighbouring spline segments

B-spline Basis:

B-spline basis functions can be derived and expressed in different ways. The most convenient definition was independently given by **Cox (1971)** and **de Boor (1972)**. The recursive formulation of the i^{th} normalised B-spline basis functions of order k are expressed as:

$$N_{i,1}(t) = \begin{cases} 1 & \text{if } x_i \le t \le x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,k}(t) = \frac{(t - x_i)}{x_{i+k-1} - x_i} N_{i,k-1}(t) + \frac{(x_{i+k} - t)}{x_{i+k} - x_{i+1}} N_{i+1,k-1}(t) \quad 1 \le i \le n+1$$

The convention $\frac{0}{0} \equiv 0$ is adopted here for numerical purposes.

It can clearly be seen from recursive Cox-de Boor relation that in order to calculate a specified basis function of order k, all lower order basis functions up to 1 has to be computed. Thus, this dependence forms a triangular pattern and illustrated in **Figure 3.17.**

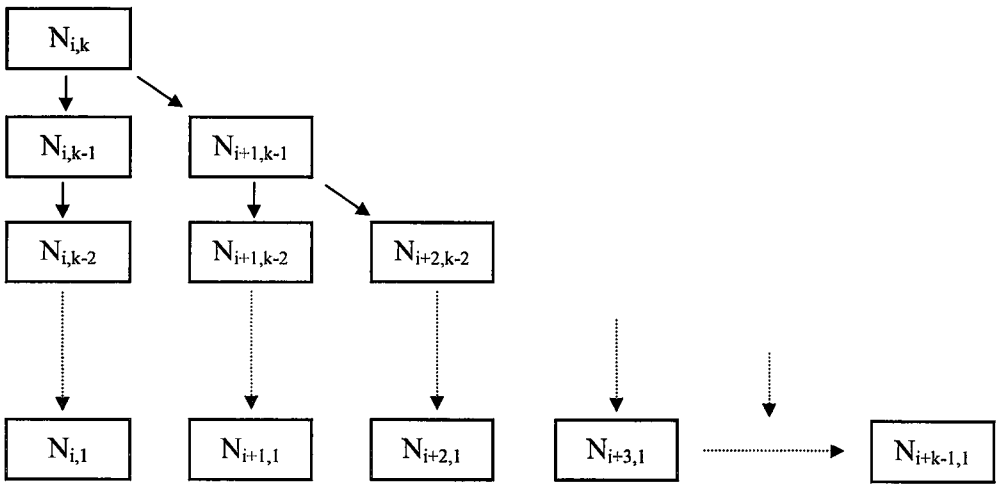


Figure 3.17. Dependence of B-spline basis functions.

The values x_i in normalised B-spline functions are elements of a knot vector satisfying the relation $x_i \leq x_{i+1}$, i.e., monotonically increasing series of real numbers. This knot vector denoted with X shows the parameterisation of the basis function, which affects its properties. The elements of the knot vector X are parameter values t and they vary from t_{min} to t_{max} denoting the parameter range along the curve.

The normalised B-spline basis functions are positive, and have local support. Some of their properties can be expressed in mathematical terms:

- $N_{i,k}(t) > 0 \quad \text{for } x_i < t < x_{i+k} \quad \text{(positivity)}$
- $N_{i,k}(t) = 0 \quad \text{for } x_1 < t < x_i \,, \quad x_{i+k} < t < x_{n+k+1} \quad \text{(local support)}$
- $\sum_{i=0}^{n+1} N_{i,k}(t) = 1$
- $N_{i,k}(t)$ has continuity C^{k-2} at each of the knots

The choice of the knot vector has a significant influence on the B-spline basis functions $N_{i,k}$ and hence on the resulting curve. Thus, B-spline bases are classified through knot vector types:

a) Uniform and non-uniform B-spline bases

b) Periodic and non-periodic B-spline bases

The first classification is the type of parameterisation, and the latter determines the resulting curve to be open or closed curve.

In general, *uniform knot vectors* are evenly spaced, and begin at zero and incremented to some maximum value or normalised in the range between 0 and 1, e.g.,

$$X = [0 \;\; 0.25 \;\; 0.50 \;\; 0.75 \;\; 1.0]$$

An *open uniform knot vector* has multiplicity of knot values at the ends of the knot vector equal to the order k of the B-spline basis function, e.g.,

$$X = [0 \;\; 0 \;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 3 \;\; 3], \quad \text{for } k = 3$$

An open uniform knot vector can be computed by the following expressions:

$$\begin{aligned} x_i &= 0 && 1 \le i \le k \\ x_i &= i-k && k+1 \le i \le n+1 \\ x_i &= n-k+2 && n+2 \le i \le n+k+1 \end{aligned}$$

Open uniform B-spline basis functions are computed using recursive Basis functions and the above open uniform knot vector expression. An example is illustrated in **Figure 3.18** for order three (k=3) and for four defining polygon vertices (n+1=4). The knot vector is obtained as

$$X = [0\ 0\ 0\ 1\ 2\ 2\ 2].$$

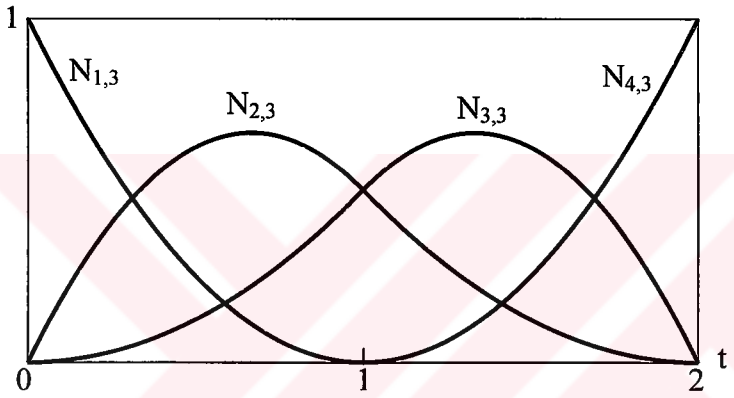


Figure 3.18. Open (non-periodic) uniform B-spline basis functions, k = 3, n + 1 = 4.

It is important to note that when the number of defining polygon vertices is equal to the order of the B-spline basis, and an open uniform knot vector is used, the B-spline basis reduces to Bernstein basis. If we consider four polygon vertices, and order is chosen as four (k =4), a cubic Bezier/B-spline curve is generated, and the knot vector becomes;

$$X = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1].$$

Non-uniform knot vectors are formed of unequally spaced parameter values. They can also possess multiple internal knot values. An example for non-uniform basis functions is illustrated in **Figure 3.19.** The order is taken as three (k=3) for five defining polygon vertices (n+1=5). The knot vector which contains an interior repeated knot value is taken as

$$X = [0\ 0\ 0\ 1\ 1\ 3\ 3\ 3]$$

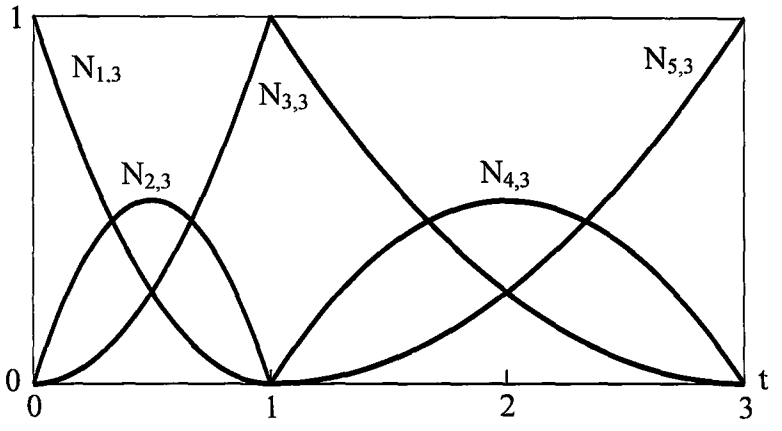


Figure 3.19. Open non-uniform B-spline basis functions, k = 3, n + 1 = 5.

An *open non-uniform knot vector* can be computed using the following formulae. The interior knot values are proportional to the chord distances between polygon vertices:

$$x_i = 0 \qquad 1 \le i \le k$$

$$x_{i+k} = \frac{[i/(n-k+2)]c_{i+1} + \sum_{j=1}^{i} c_j}{\sum_{i=1}^{n} c_i}(n-k+2) \qquad 1 \le i \le n-k+1$$

$$x_i = n-k+2 \qquad n+1 \le i \le n+k$$

where $c_i = |B_{i+1} - B_i|$. For equally spaced polygon vertices the result reduces to an open uniform knot vector.

Both uniform and non-uniform knot vectors can be *periodic (closed)*. In this case, the knot values range from generally 0 to t_{max} = n+k, and the number of knot values is n+k+1, e.g., for k = 3 and for four polygon vertices n+1 = 4, the knot vector for the periodic basis function is; X = [0 1 2 3 4 5 6].

For a given order k, uniform knot vectors yield periodic uniform basis functions that satisfy

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

This property of periodic basis functions is illustrated in **Figure 3.20.**

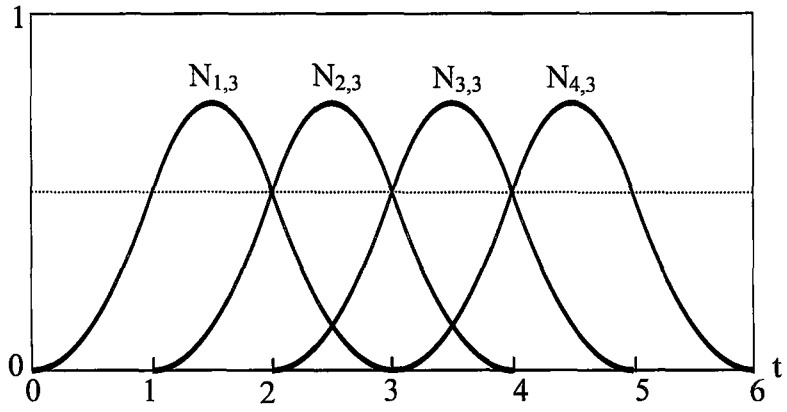


Figure 3.20. Periodic (closed) uniform B-spline basis functions, k = 4, n +1 = 4.

De Boor Algorithm:

De Boor algorithm allows the computation of points on a B-spline curve without explicit background knowledge of the B-spline basis functions described in the foregoing. The algorithm is generalisation of de Casteljau algorithm which is used for obtaining Bezier curves. The basis of the algorithm is constructing B-spline curve points from defining control polygon points which are also called de Boor points. The algorithm, which yields B-spline curve points for the corresponding parameter value t, is defined by the following recursive expression:

$$B_i^j(t) = \frac{x_{i+n-j} - t}{x_{i+n-j} - x_{i-1}} B_{i-1}^{j-1}(t) + \frac{t - x_{i-1}}{x_{i+n-j} - x_{i-1}} B_i^{j-1}(t)$$

$$j = 1,\ldots, n-r$$
$$i = I - n + k + 1,\ldots, I + 1$$

where x represents knot vector values, n degree of the curve, and r multiplicity of parameter value t, usually taken as zero (r = 0). Parameter value t should be in the interval $t \in [x_I, x_{I+1}]$. Control polygon points are denoted with; $B_i^0(t) = B_i$ in the above recursive formulation. The value of the B-spline curve at parameter value t is:

$$P(t) = B_{I+1}^{n-r}(t)$$

An example is given for the case of four defining polygon vertices, namely de Boor points, and evaluation of a B-spline curve P(t) of degree n = 3 at parameter value t is

considered. Parameter value t is assumed to be in the interval $[x_2, x_3]$, and the B-spline curve value at t is computed from de Boor algorithm as follows:

$$B_3^3(t) = \frac{x_3 - t}{x_3 - x_2} B_2^2(t) + \frac{t - x_2}{x_3 - x_2} B_3^2(t)$$

Figure 3.21 shows the geometric interpretation of the Boor algorithm as repeated linear interpolation in this case.

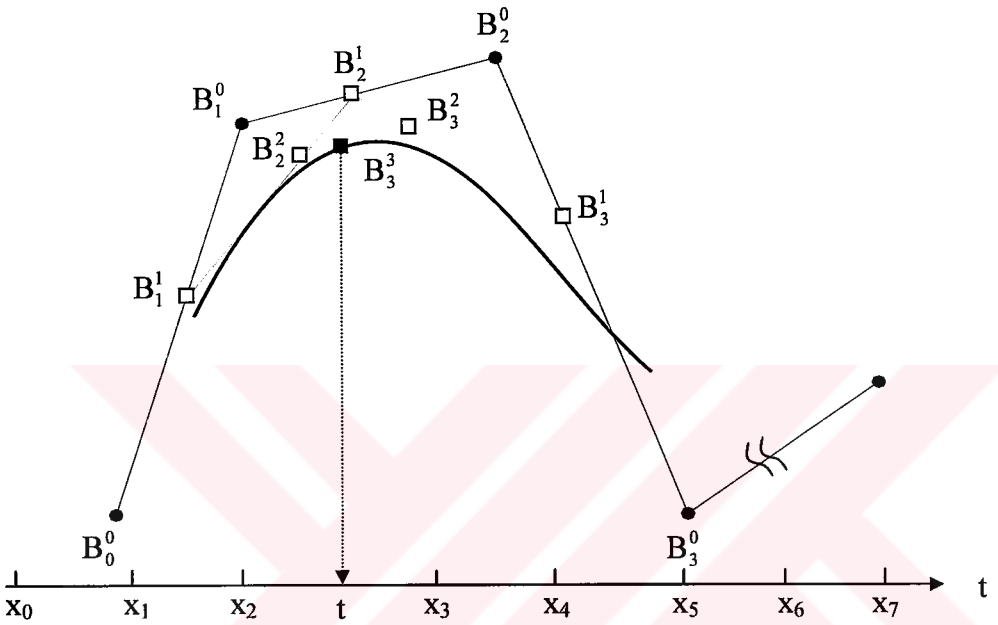


Figure 3.21. The de Boor algorithm for the k = 4 case.

Mathematical Definition of B-spline Curves:

The general form of a B-spline curve is given by

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \qquad t_{min} \le t \le t_{max}, \qquad 2 \le k \le n+1$$

where P(t) represent position vectors along the B-spline curve, and B_i control polygon vertices, also called de Boor points. $N_{i,k}(t)$ are the normalised B-spline basis functions. A B-spline curve is the linear combination of defining polygon vertices and B-spline basis functions.

The uniform B-spline basis is considered to be standard spline basis in CAGD and hence used in the context of this thesis. An open uniform B-spline curve and its defining polygon is illustrated in **Figure 3.22**.

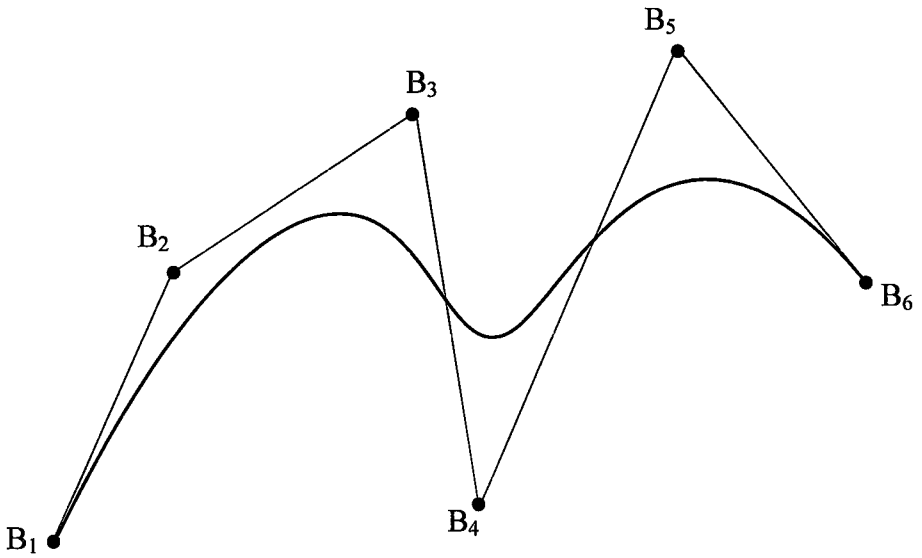


Figure 3.22. An open uniform B-spline curve of order k = 4, and its defining polygon.

Due to the flexibility of B-spline curves, various natural shape control handles can be used to modify the shape of the curve. These can be listed as:

- The type of the knot vector, and basis function,
- The order of the basis function,
- The number and position of control polygon vertices,
- Multiple polygon vertices,
- Multiple knot values.

The effect of using different types of knot vectors on B-spline curves is illustrated for comparison in **Figure 3.23**. The non-uniform B-spline curve is chord length proportional parameterised and for equally spaced polygon vertices the resulting curve reduces to evenly spaced integer interior knot values, i.e., an open uniform knot vector.

It can clearly be seen from **Figure 3.23** that non-uniform B-spline curves do not greatly differ from uniform B-spline curves unless the relative distances between polygon vertices differ radically.

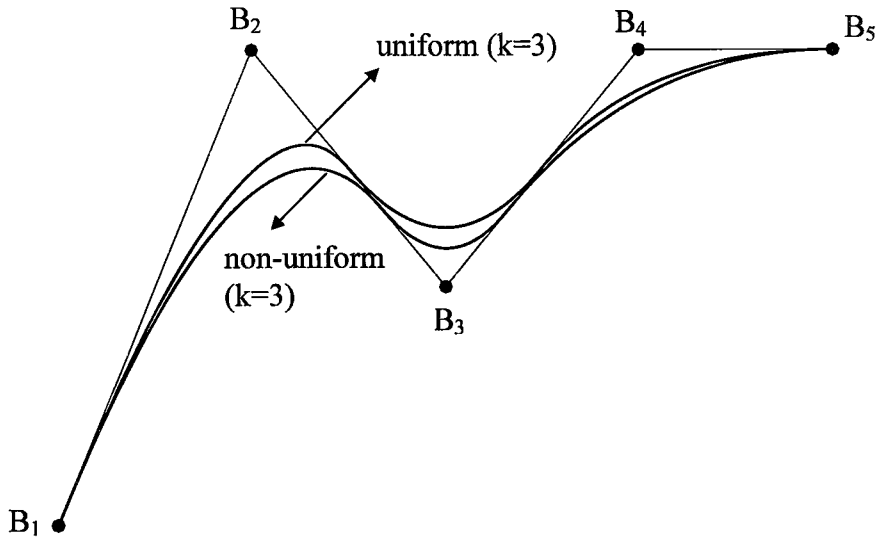


Figure 3.23. Comparison of quadratic open uniform and non-uniform B-spline curves.

The effect of varying order and multiple polygon vertices on B-spline curves is illustrated in **Figure 3.24** and **3.25**, respectively.

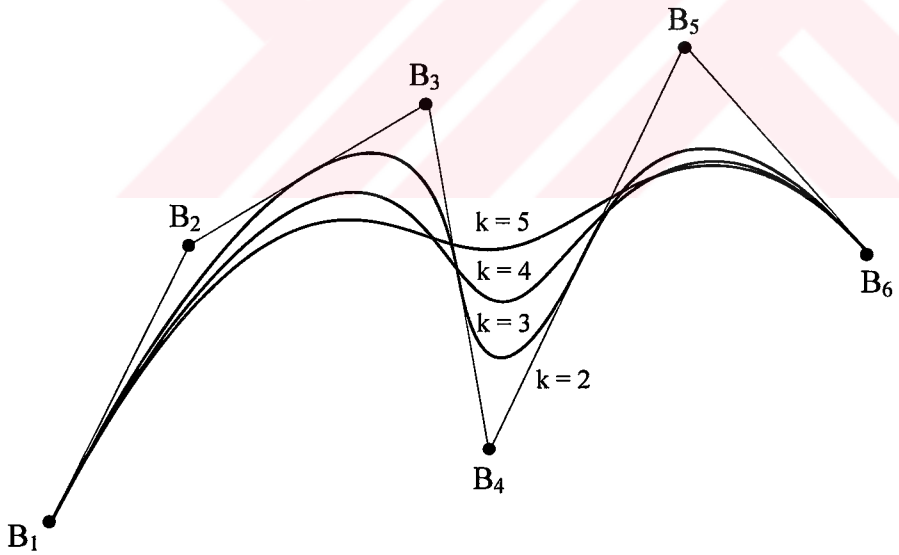


Figure 3.24. Effect of varying order on a B-spline curve.

Basically, the order the B-spline curve determines how close the curve will follow the defining polygon. As the order decreases, the generated curve lies closer to the defining polygon. For order two (k=2), the curve is the identical to the defining polygon.

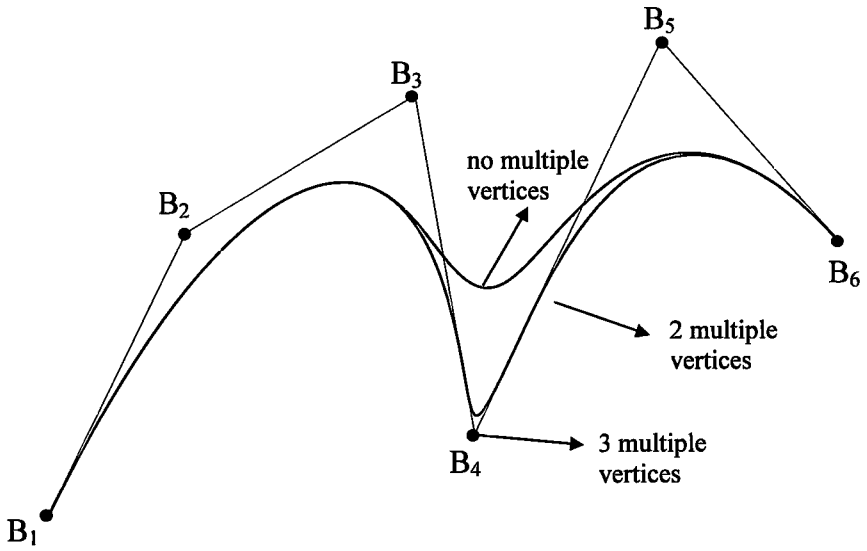


Figure 3.25. Effect of multiple vertices at B_4 on a B-spline curve of order four, k=4.

Using double or triple vertex produces knuckles depending on the order of the spline. This ability is a common requirement in ship hull form design.

Basic properties of B-spline curves can be summarised as follows:

- They allow a high degree of local variation without propagation along the entire curve, since each control vertex is associated with a unique basis function. Each curve segment of a B-spline curve of order k is controlled by adjacent k vertices of the defining polygon, and completely unaffected by other control vertices. Conversely, the effect of moving a vertex is confined to k segments. This local control effect is illustrated in **Figure 3.26**. The three curves of order four (k=4) are obtained by changing the position of the fifth vertex of the control polygon successively from B_5 to B_5' and B_5''. The modified polygons are denoted with dashed lines. Clearly, the curve is affected over the neighbouring $\pm k/2$ spans around the displaced vertex.

- B-spline curve segments of degree k-1 join with continuity of the parametric k-2 derivative vectors. Thus, an entire B-spline curve of degree k-1 is everywhere continuous along with its first k-2 derivatives, C^{k-2}.

- The associated knot vector of a B-spline curve denotes its parameterisation, and controls the degree of the continuity between polynomial segments. If multiple interior knot values are used to yield sharp corners or cusps, (e.g., multiplicity r, $x_i = x_{i+1} = \ldots = x_{i+r-1}$, $r \leq k-1$) then the differentiability of the basis function is reduced to C^{k-r-1} at that knot value due to introducing a span of zero length. This property is also useful when interpolating first and last polygon vertices by making the multiplicity of the first and last knot equal to the order of the spline (*end point interpolation*).

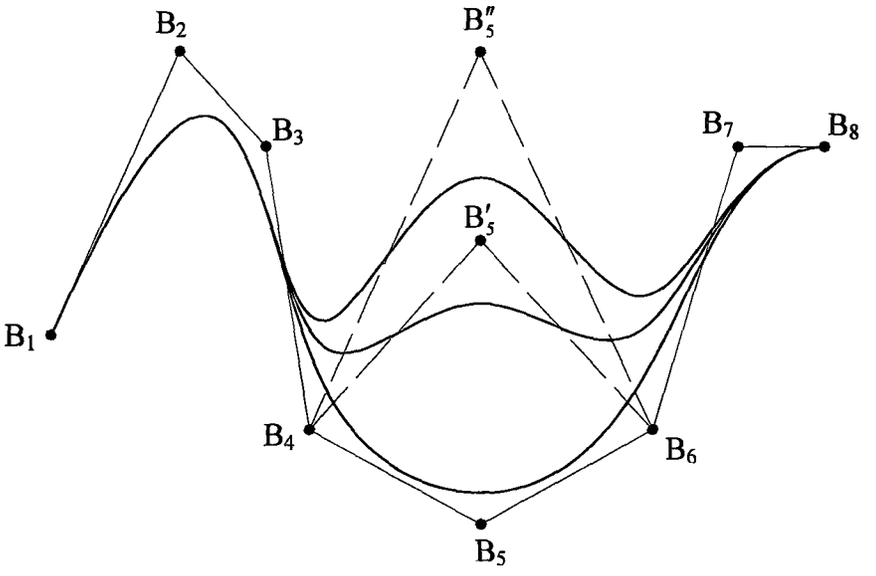


Figure 3.26. Effect of vertex modifications over a B-spline curve.

- Though the order of the B-spline representation is independent of the number of control vertices, the maximum order is confined with the number of defining polygon points. It is a general requirement that the order must be sufficiently high to offer enough freedom to satisfy various constraints, however it may be desirable to maintain the order as low as possible.

- Although the generated curve reflects the shape of the control polygon, it tends to produce a smoother curve due to B-splines *variation diminishing property*.

- A point on a B-spline curve of degree k-1 is a convex combination of k control vertices. The set of all possible convex combinations of these vertices is their convex hull. Thus, each order k B-spline segment lies within the convex hull of its k defining control vertices. Since the entire B-spline curve is composed of a sequence of these segments, it will pass through the union of the convex hulls of each segment (*convex hull property*). **Figure 3.27** shows the convex hulls of the control polygons for several B-spline curves of varying orders.

- They can be controlled interactively by defining vertex polygons and can be made to closely resemble the defining polygon.

- The relationship between a B-spline curve and its control polygon is invariant under affine transformation. Any affine transformation can be applied to the curve by applying it to its defining control polygon (*affine invariance*).

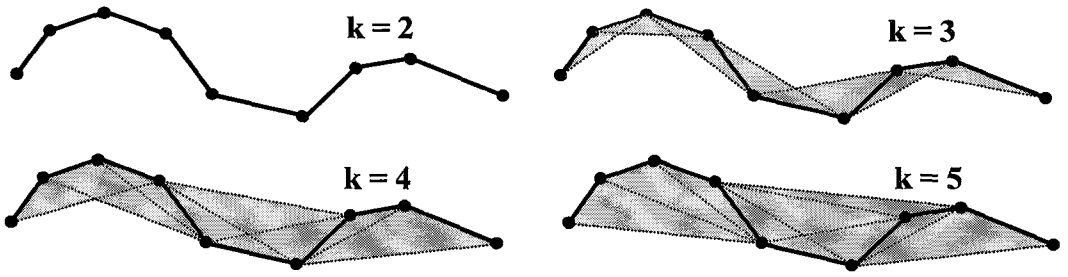


Figure 3.27. Convex hulls of B-spline segments of various orders.

- They allow slope and curvature (first and second order derivatives) to possess discontinuity along the curve enabling flat part or knuckles to be represented by a single spline definition. A B-spline curve, which contains a line segment is illustrated in **Figure 3.28**.

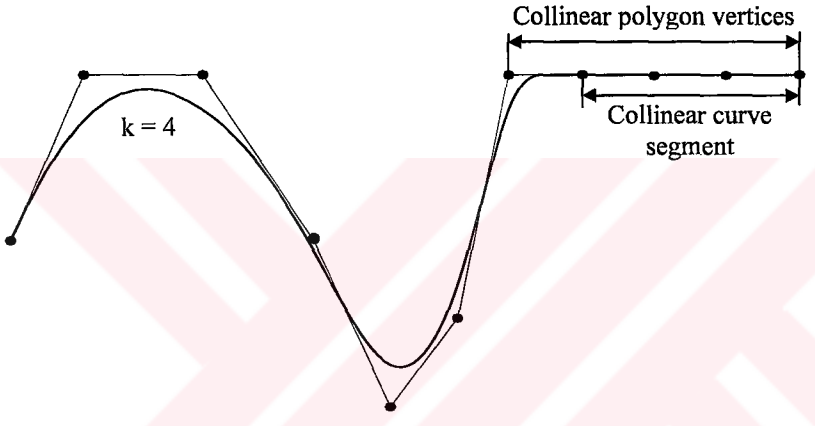


Figure 3.28. Effect of collinear control points over a B-spline curve.

- The flexibility of a B-spline curve can be increased by raising the order of the defining B-spline basis and hence of the defining polynomial segments. Also, the flexibility can be increased by inserting additional knot values into the defining knot vector.

B-splines have sufficient degrees of freedom to capture any possible curve shape likely to be met in ship curve design. Therefore, in the context of the thesis, B-splines have been used both for representation of ship curves and surfaces, and in developing effective fairing procedures. B-splines of low order ($4 \leq k \leq 6$) generally give satisfactory results in designing and fairing procedures. In **Figure 3.29**, cubic uniform B-spline representation of typical ship sections is presented. It can be seen from the applications that B-splines curves are likely to represent any possible shape, including knuckles.

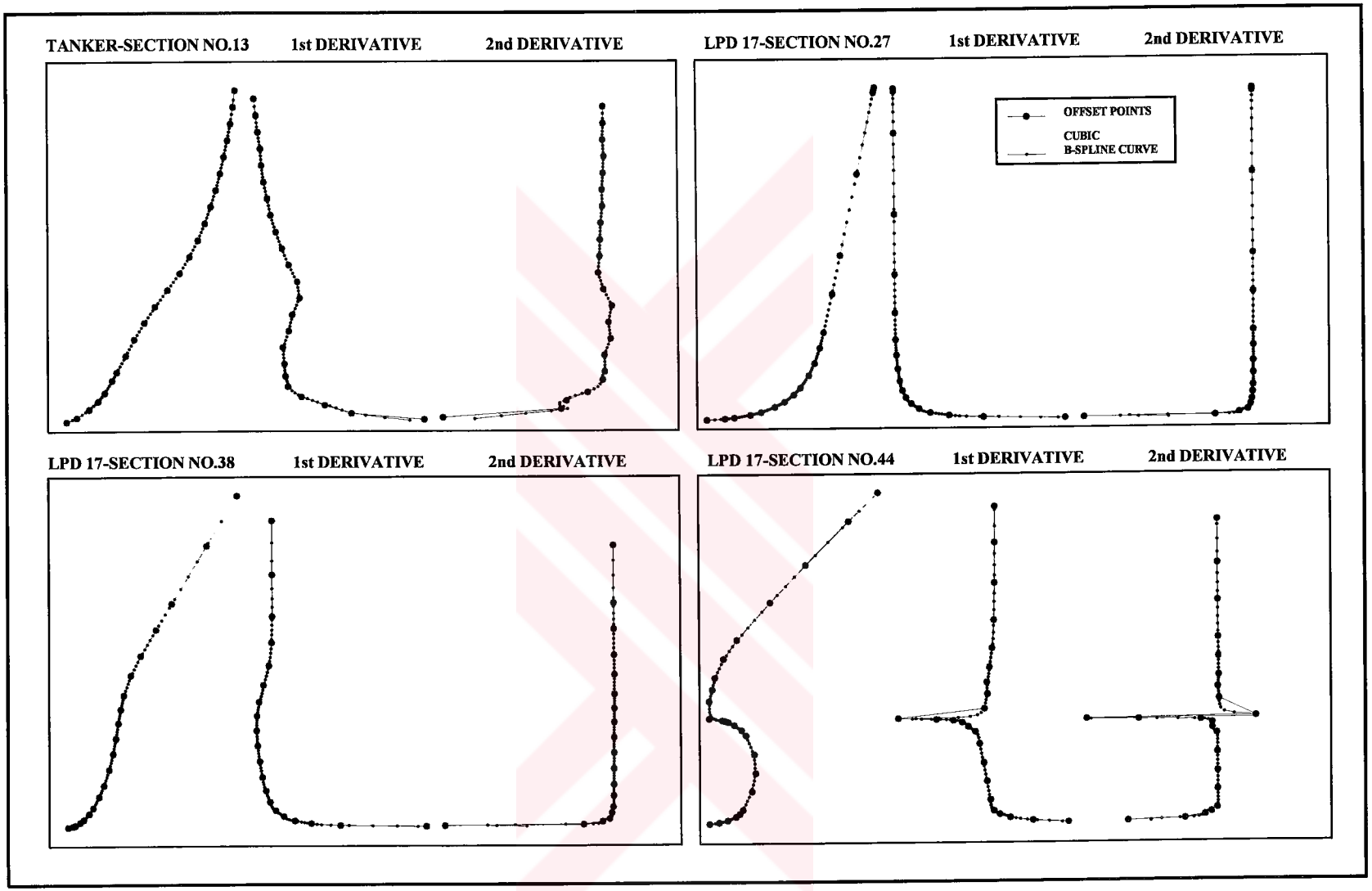


Figure 3.29. Cubic uniform B-spline representation of typical ship sections.

3.2.3.2. B-Spline Surfaces

Tensor product B-spline surfaces are obtained as a straightforward generalisation of the B-spline curves, therefore share most of the properties of B-spline curves demonstrated in the previous section. They play an important role in current surface design methods. Tensor product B-spline surfaces can be computed using the following expression

$$Q(u,w)=\sum_{i=1}^{n+1}\sum_{j=1}^{m+1}B_{i,j}\,N_{i,k}(u)M_{j,\ell}(w) \qquad u_{min}\le u\le u_{max}, \qquad 2\le k\le n+1,$$
$$w_{min}\le w\le w_{max}, \qquad 2\le \ell\le m+1$$

where $N_{i,k}(u)$ and $M_{j,\ell}(w)$ are the B-spline basis functions of degree k-1, and ℓ-1 in the bi-parametric u and w directions, respectively. The set of control points are usually referred to as *control net*. The expressions of the B-spline basis functions are given similarly to that of B-spline curves,

$$N_{i,1}(u)=\begin{cases}1 & \text{if } x_i\le u\le x_{i+1}\\ 0 & \text{otherwise}\end{cases}$$

$$N_{i,k}(u)=\frac{(u-x_i)}{x_{i+k-1}-x_i}N_{i,k-1}(u)+\frac{(x_{i+k}-u)}{x_{i+k}-x_{i+1}}N_{i+1,k-1}(u) \qquad 1\le i\le n+1$$

$$M_{j,1}(w)=\begin{cases}1 & \text{if } y_i\le w\le y_{i+1}\\ 0 & \text{otherwise}\end{cases}$$

$$M_{j,\ell}(w)=\frac{(w-y_i)}{y_{i+\ell-1}-y_i}M_{j,\ell-1}(w)+\frac{(y_{i+\ell}-w)}{y_{i+\ell}-y_{i+1}}M_{j+1,\ell-1}(w) \qquad 1\le i\le m+1$$

where x_i and y_i are elements of knot vectors, and defined as follows:

$$[X]=[x_1\ \ldots\ x_{n-k+2}], \qquad [Y]=[y_1\ \ldots\ y_{m-\ell+2}]$$

The number of defining vertices $B_{i,j}$ in the u and w parametric directions is n+1 and m+1, respectively. A degree (k,ℓ) tensor product B-spline surface is a piecewise polynomial surface defined over a rectangular domain. Clearly, the shape of a B-spline surface is greatly influenced by the type of knot vectors in both directions. Open, uniform, non-uniform knot vectors can be used. Although different types of knot vectors can be used for u and w parametric directions, it is not common, and used only for specific occasions. (e.g., a cylindrical surface of varying cross-sectional

area, an open knot vector is used for one parametric direction and a periodic knot vector is used for the other.)

Figure 3.30 shows an example of a bi-cubic B-spline surface and its defining control net. The defining control net is formed of a 4×4 polygon net, and the order in both parametric direction is four.

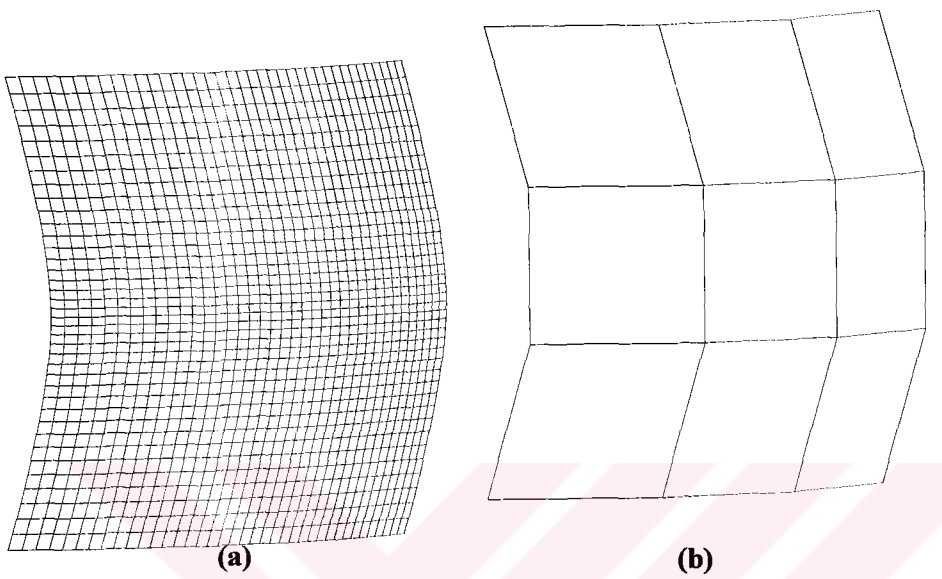


Figure 3.30. (a) Bi-cubic B-spline surface **(b)** Its control net.

A tensor product B-spline surface has the following properties, most of them inherited from B-spline curves.

- The maximum order of the surface in each parametric direction is equal to the number of defining polygon vertices in that direction.

- The continuity of the surface in each parametric direction is two less than the order in each direction provided that knot vectors do not contain multiple knot values. Thus, a B-spline surface of degree k-1, and ℓ-1 in u and w parametric directions is everywhere continuous along with its first k-2 (C^{k-2}), and ℓ-2 ($C^{\ell-2}$) derivatives. For multiple knots the differentiability decreases by one for each increase in multiplicity. Multiple knots can thus be used to generate local knuckles or discontinuity in surfaces.

- The surface is invariant with respect to an affine transformation. Hence, the surface is transformed by its defining polygon net. This is a consequence of the fact that the B-spline basis functions are positive and sum to one.

- The influence of a single polygon vertex is limited to ± k/2, ± ℓ/2 spans in each parametric direction.

- The B-spline surface reduces to a Bezier surface if the number of defining polygon vertices is equal to the order in each parametric direction.

- The surface lies within the convex hull of the defining polygon net formed by taking the union of all convex hulls of k, ℓ neighbouring polygon net vertices. This strong convex hull property for B-spline surfaces follows directly from B-spline curves.

- Tensor product B-spline surfaces can only be used if the surface to be modelled can be associated with a rectangular domain.

As a consequence of strong convex hull properties originated basically from B-spline basis, a B-spline surface can contain flat regions and lines of sharp discontinuity. This particular characteristic is especially required in hull form design environment as some ship forms are likely to be formed of these parts. (e.g., a hard chine form contains lines of sharp continuity, and a full ship is likely to contain a flat part)

Applications of B-spline surfaces for a mathematical hull form which was first suggested by **Wigley (1934)**, and a high-speed displacement type hull form used as the parent hull for NPL series described by **Bailey (1976)**, are shown in **Figures 3.31** and **3.32**, respectively. Their defining polygon net is formed of hull form offset points. Since a typical hull form is conventionally defined by offset points over a rectangular domain, it is common to represent the hull surface by tensor product B-spline surfaces.

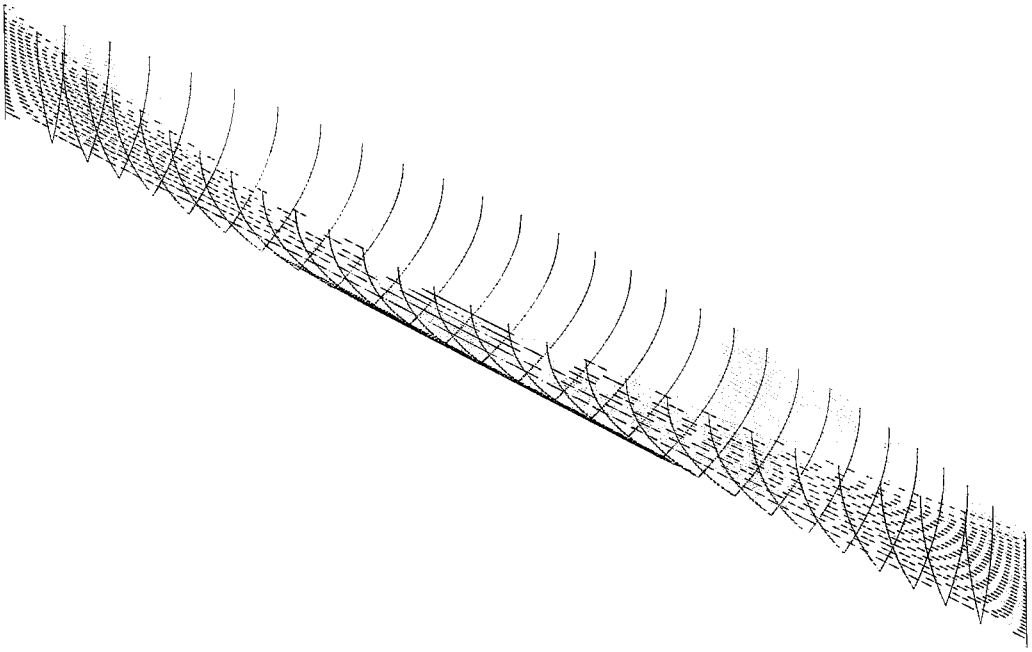


Figure 3.31.a. Bi-cubic B-spline surface wireframe model of Wigley hull form.

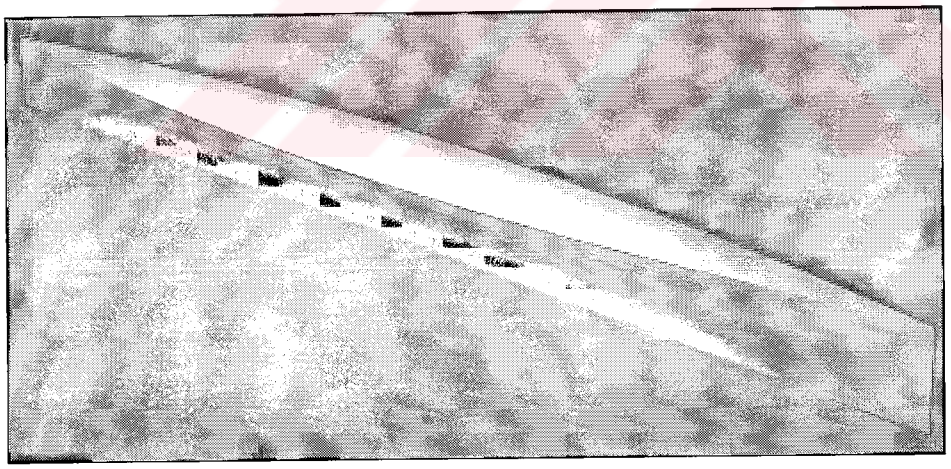


Figure 3.31.b. Bi-cubic B-Spline surface representation of Wigley hull form.

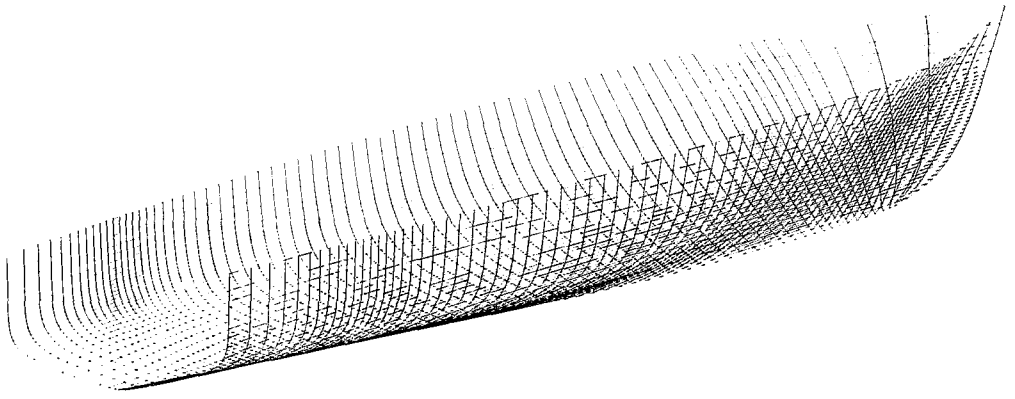


Figure 3.32.a. Bi-cubic B-spline surface wireframe model of NPL hull form.

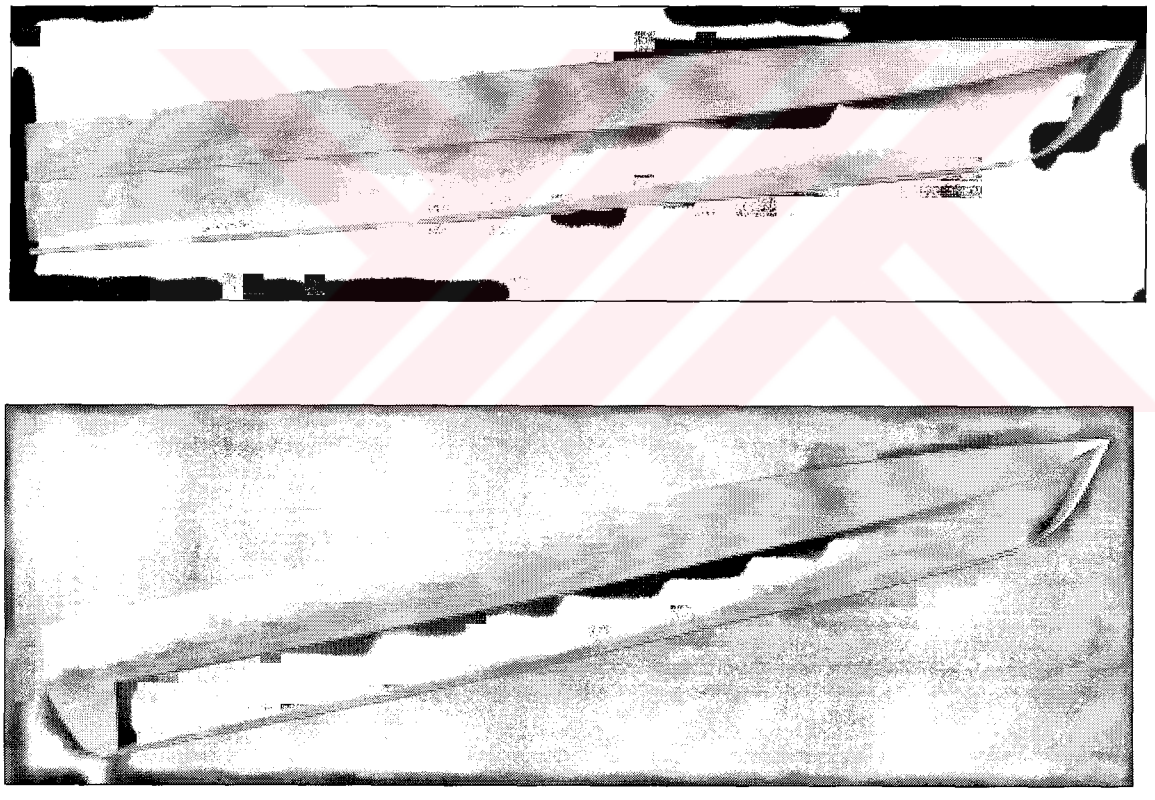


Figure 3.32.b. Bi-cubic B-Spline surface representation of NPL hull form.

3.2.4. Comparison of Spline Techniques

Detailed discussion of spline techniques (i.e., cubic splines, Bezier and B-spline curves) can be found in **Barnhill and Riesenfeld (1974), Faux and Pratt (1981), Yamaguchi (1988), Rogers and Adams (1990), Lyche and Schumaker (1992), Hoschek and Lasser (1993)**. Some of the theoretical characteristics of the spline methods are listed in **Table 3.3**. Compared characteristics are computer storage, computing difficulty, continuity, local control capabilities, and available shape control parameters or handles, and the ability to represent a knuckle.

Table 3.3. General characteristics of spline curves.

	Cubic Spline	Bezier Curves	B-spline Curves
Computer Storage (Input)	• Offset points • End tangent vectors	Offset points	• Offset points • Order
Computing difficulty	Requires matrix inversion (high)	Bernstein-basis functions are computed (medium-low)	B-spline basis functions + knot vector computed (medium-high)
Degree	Cubic	Depends on the number of offset points. (n-1)	User defined. (maximum degree n-1)
Continuity	Up to second derivative. (C^2)	Two less than number of offset points. (C^{n-2})	Two less than order of the basis functions. (C^{k-2})
Local control	No local control	No local control	High local control
Shape control handles	• Location of offset points • End tangent vectors	• Location of offset points • Number of offset points	• Location of offset points • Number of offset points • Order of the basis function • Type of the knot vector • Multiple knot/polygon vertices
Ability to represent a corner	Requirement of curve subdivision	Requirement of curve subdivision	Defining multiple(k-1) polygon vertices or knot values at that point

4. GEOMETRIC PROPERTIES OF SHIP HULL FORMS

In this chapter some elementary techniques and properties of curves and surfaces are discussed and identified that will be useful for the description of hull form geometry and fairing methodologies. Curves and surfaces for both design, approximation and fairing are characterised by those properties.

4.1. Parametric Curve Representation

In Computer Aided Geometric Design (CAGD) environment it is more common to define curves parametrically in terms of a single scalar parameter. Thus, as stated by **Farin (1993),** *the exploration of the use of parametric curves and surfaces can be viewed as the origin of CAGD.* The reason of the popularity of this representation is due to its greater flexibility and easy-to-manipulate property. The conventional scalar-valued, explicit and implicit forms in the Cartesian coordinate system are only capable of describing a small class of curves and surfaces. For example, such a function can not represent a non-planar curve twisted in space. Moreover, it can not be used to describe a multiple-valued curve or surface. Also, the choice of the coordinate system should have no effect on the shape of the curve. It is obvious that all these crucial requirements can only be satisfied with parametric representation.

In CAGD environment, the curves are generally specified as polynomial functions of parameter t, where t is an arbitrary curve parameter and usually lies in the range of $0 \leq t \leq 1$. The curve can be expressed with a differentiable vector valued function on an interval *I*, the function X(t) is called the *parameterisation* of the curve.

$$X(t) = [x(t), y(t), z(t)] \quad t \in [a, b]$$

Parametric representation can be conceptualised as a mapping from parameter space to Euclidean space. For a given parametric value, the coordinates of a point on the curve or surface is obtained. This mapping is illustrated for a space curve in **Figure 4.1.**

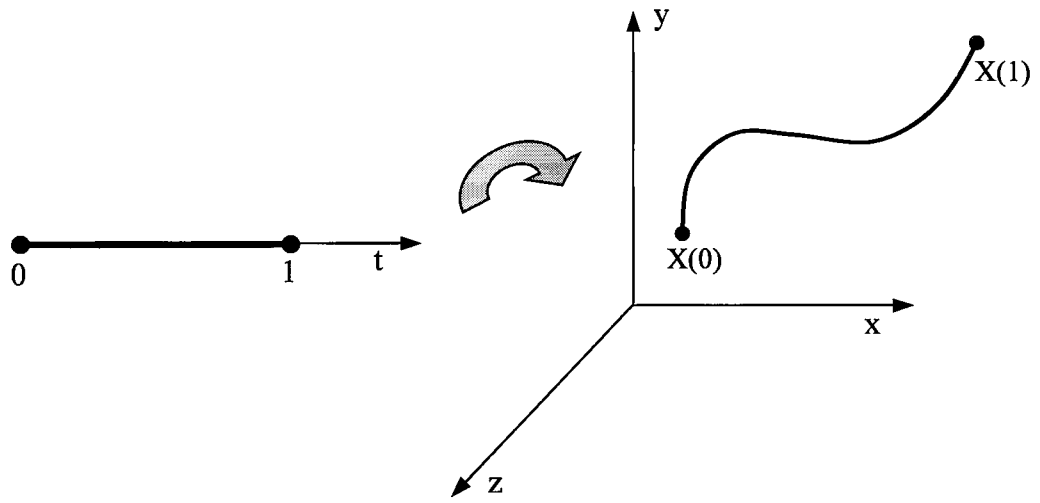


Figure 4.1. Mapping from t parameter space into $\Re^3$ space for a spatial curve.

A parameterisation is said to be *regular* provided that it is at least continuously differentiable, and satisfies the following condition

$$|X'(t)| \neq 0 \text{ for all } t \in I$$

Another important way to represent a curve parametrically, which has obvious geometric significance, is to parameterise the curve in terms of arc length where the arc length of curve can be given as,

$$s(t) = \int_{t_0}^{t} |X'(t)|\,dt \qquad \text{where} \qquad |X'(t)| = \sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2}$$

Under *arc length parameterisation* $|X'(s)| = 1$ everywhere. Although arc length is an important concept, it is primarily used for theoretical considerations and for the development of curve algorithms.

4.2. Geometric Characterisation of Curves

Geometric character of a curve is usually described in terms of tangent, curvature and torsion quantities. A trihedral frame, which is called the Frenet Frame is introduced in differential geometry that will facilitate the description of these properties. This frame is composed of the curve's tangent (t), normal (n), and binormal (b) vectors. Each vector can be defined as follows:

Tangent of the Curve:

At any point on a curve in the plane, the line best approximating the curve that passes through this point is tangent to the curve. The tangent t(u) at a point X(u) is the direction of the curve at X(u) where u is the curve parameter. Thus, the tangent is in the direction of the first derivative of the curve.

$$t(u) = \frac{X'(u)}{|X'(u)|}$$

Binormal of the Curve:

The binormal of the curve at a point on X(u) is perpendicular to the plane, the curve lies at X(u). Thus, the binormal is perpendicular to the first and second derivatives.

$$b(u) = \frac{X'(u) \times X''(u)}{|X'(u) \times X''(u)|}$$

Normal of the Curve:

The normal of the curve at a point on X(u) is perpendicular to the tangent and binormal vectors.

$$n(u) = b(u) \times t(u)$$

These three vectors define a set of three planes; the *osculating plane*, the *normal plane*, and the *rectifying plane*.

The osculating plane is spanned by the tangent and normal vectors and is named for the best approximating tangent circle (osculating circle) that passes through the point on the X(u). The normal plane is spanned by normal and binormal vectors. The rectifying plane is spanned by tangent and binormal vectors, and is named for the fact that as it moves along a curve, it sweeps out a rectifying developable surface. The Frenet Frame of a space curve can be seen in **Figure 4.2.**

This special local coordinate system varies its orientation as parameter (t) traces out the curve. The fundamental properties of curves are curvature and torsion, which uniquely define the curve shape. They are intrinsic differential characteristics of curves and surfaces.

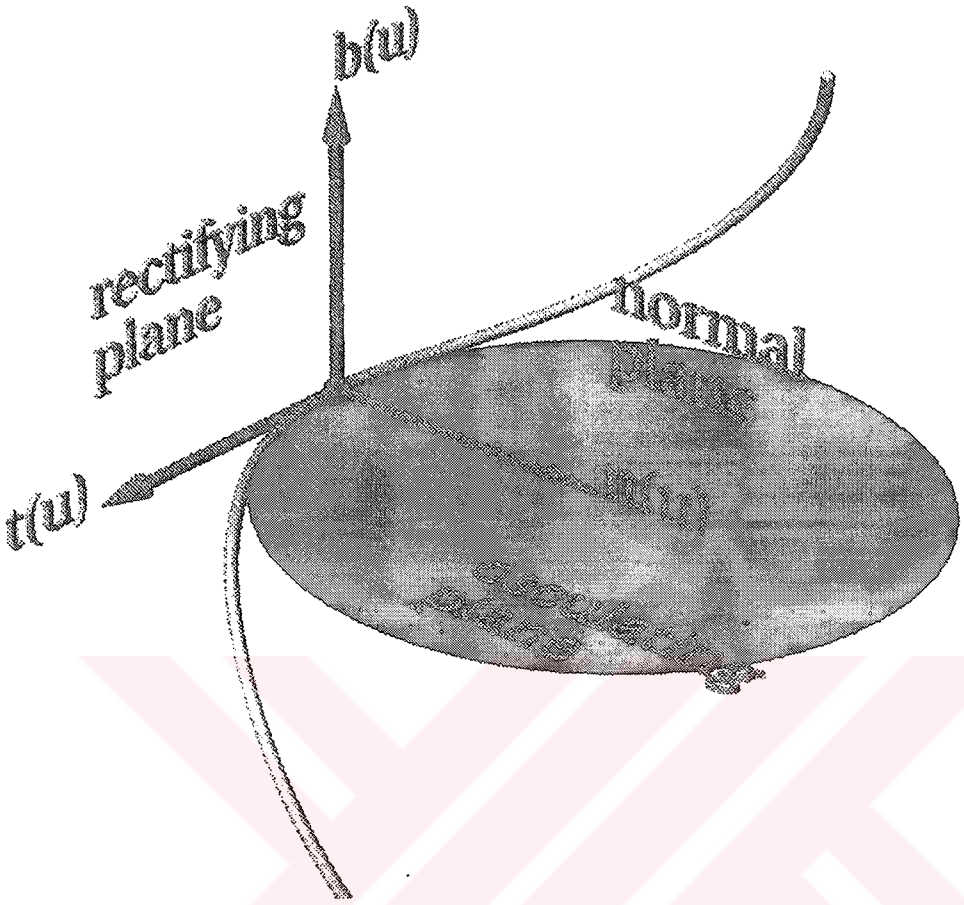


Figure 4.2. The Frenet Frame. **Moreton (1992)**

4.2.1. Curvature

Global and local shape characteristics of curves are completely described by curvature, and fairness criteria for curves are generally associated with the curvature properties, thus it has emerged as a necessity to introduce curvature defined in differential geometry. **Moreton (1992)** defines curvature as an instantaneous measure of how much the curve is bending in the osculating plane away from the tangent direction. The reciprocal of the radius of the best approximating tangent circle that passes through this point called *osculating circle* is the curvature of the curve at this point. The circle lying to the left of the curve denote positive, and conversely to the right of the curve denote negative curvature.

For an arbitrary curve, the curvature of a circle of radius R is given by $\kappa = 1/R$, and $R = 1/\kappa$ is called the radius of curvature at the point $u = u_0$. The radius of curvature can be interpreted geometrically as the radius of the circle whose first and second derivatives agree with those of the curve X(u) at the point $u = u_0$.

When a curve is arc length parameterised, the second derivative is equal to the curvature, thus $X''(u)$ is generally used as an approximation to the curvature. Most features of a curve can be conveniently expressed in terms of curvature κ, as listed in **Table 4.1. (Wesselink, 1996)**

Table 4.1. The relationship of curve features with curvature.

Curve Feature	Mathematical Definition
Inflection point	κ changes its sign
Convex curve	$\kappa > 0$
Concave curve	$\kappa < 0$
Flat point	$\kappa = 0$
Corner, chine, knuckle, break	Max $\lvert\kappa\rvert$
Accelerating/Decelerating curve	Monotonous increasing/decreasing $\lvert\kappa\rvert$

Frenet-Serret equations are direct results of these definitions:

$$t'(s) = \kappa(s)\, n(s)$$
$$b'(s) = \tau(s)\, n(s)$$
$$n'(s) = -\kappa(s)\, t(s) - \tau(s)\, b(s)$$

where $\tau(s)$ represents torsion of the curve, and s denotes arc length parameterisation.

The expression of the curvature can be derived directly from Frenet-Serret equations:

$$\kappa(u) = \frac{\left|X'(u) \times X''(u)\right|}{\left|X'(u)\right|^3}$$

where $X = X(u)$ is a vector-valued function of a parameter u defined on an interval I and the components of X will be real-valued functions of parameter u. Highly curved splines yield large curvature values. If $\kappa(u) = 0$ for all $u \in I$ than the curve reduces to

a straight line. If $\kappa(u) = 0$ holds locally, then the curve has a point of inflection or a flat point.

$$X''(s) = \kappa(s)\, n(s)$$

The Frenet Frame vectors, the curvature and torsion can be expressed in terms of arc length instead of regular parameterisation. Although simplified expressions can be obtained, no obvious advantage regarding computer implementation is achieved.

It is a well-known fact that curvature, along with torsion, completely characterises the shape, thus fairness of curves. The requirement of eliminating unnecessary variations applies directly to the curvature and torsion of a spatial curve. It is also worth noting that C^3 continuity is required for the torsion, hence curvature of a spatial curve to be continuous. Consequently, curvature plots of curves are frequently used as fairness indicators in most curve design schemes.

4.2.2. Torsion

Torsion is specifically used for describing a curve's shape, like the curvature of a curve. It is an instantaneous measure of how much the curve is bending away from or out of the osculating plane, hence a curve with zero torsion denotes a planar curve. $\tau(u)$ is the rate of change of the angle between $b(u)$ and $b(u=0)$, in other words torsion measures how far the curve is from being planar. The expression of torsion can be given as:

$$\tau(u) = \frac{\det\left| X'(u), X''(u), X'''(u) \right|}{\left| X'(u) \times X''(u) \right|^2}$$

The curvature and torsion are invariants of a curve, i.e., their values are independent of the parameterisation. Thus, they uniquely determine a curve's shape.

4.3. Geometric Characterisation of Surfaces

The idea for curves can further be developed for surfaces, such as a surface which bulges out in all directions (the surface of a sphere) is positively curved, i.e., if the surface lies all on one side of the tangent plane of a point taken on the surface. Conversely, a saddle shaped surface has negative curvature; every plane through a point on the saddle actually cuts the saddle surface in two or more pieces. In other words, a positive curvature value means the surface is locally either peak or a valley, and negative value means the surface has locally saddle points. The zero value indicates the surface is flat at least in one direction. Thus, the geometric character of a surface is described by position, surface normal and the principal curvatures.

If the surface is assumed to be represented by a vector-valued function of two variables, its equation becomes;

$$Q(u,v) = [x(u,v), y(u,v), z(u,v)], \quad [u,v] \in [a,b]$$

where u and v are arbitrary surface parameters, and the surface is parameterised regularly.

The unit surface normal is the most elementary differential characteristic of a surface and defines the tangent plane to the surface at a given point. It is computed from the partial derivatives of the surface such as;

$$n = \frac{Q_u(u,v) \times Q_v(u,v)}{\left| Q_u(u,v) \times Q_v(u,v) \right|}$$

Currently, the best mathematical technique for analysing surface characteristics is by means of Eulerian (orthogonal) nets of minimum and maximum curvature, and specifically Gaussian. At any point P on a surface, the curve of intersection of a plane contains the normal to the surface at a point on the surface, and the surface has curvature κ at that point. As the plane is rotated about the normal, the curvature changes. Euler, has found out that minimum and maximum curvatures have unique directions. The curvatures in these directions are called the principal curvatures κ_1 and κ_2. Moreover, the principal curve directions are orthogonal. Two combinations of the principal curvatures are of special interest, the mean and Gaussian curvatures. The mean curvature is defined as

$$H = \frac{\kappa_1 + \kappa_2}{2},$$

and the Gaussian curvature is

$$K = \kappa_1 \kappa_2$$

Dill (1981) has shown that for bi-parametric surfaces the mean and Gaussian curvatures can simply be expressed as

$$H = \frac{A|Q_w|^2 - 2BQ_uQ_w + C|Q_u|^2}{2|Q_u \times Q_w|^3}$$

$$K = \frac{AC - B^2}{|Q_u \times Q_w|^3}$$

where

$$A = |Q_u \times Q_w| Q_{uu}$$
$$B = |Q_u \times Q_w| Q_{uw}$$
$$C = |Q_u \times Q_w| Q_{ww}$$

As noted before, the sign of the Gaussian curvature serves to characterise the local shape of the surface, i.e., elliptic, hyperbolic, or conical. In **Table 4.2** surface types along with their sign conventions are given. Thus, the distribution of κ over the surface indicates the shape of the regions.

Table 4.2. Surface types.

$\kappa_1\kappa_2$		**Shape**
Same sign	> 0	Elliptic (bump or hollow)
Opposite sign	< 0	Hyperbolic (saddle point)
One/both zero	= 0	Cylindrical (flat point)

A convenient method for displaying the variation of a scalar quantity is by means of colour encoded maps, an even gradation of colour corresponding to a range of values for that quantity. A contour map is a sequence of planar curves lying on a surface, each parallel to a fixed reference plane. These level curves are typically evenly spaced and change shape in a smooth and continuous manner between successive levels. **Figure 4.3** shows a contour map for a test surface used in Figure 3.30, and **Figure 4.4** shows the contour map of a mathematical hull form named as Wigley form that has been generated by a set uniformly spaced sectioning planes. The mathematical expression of Wigley form can be found in Chapter 5.

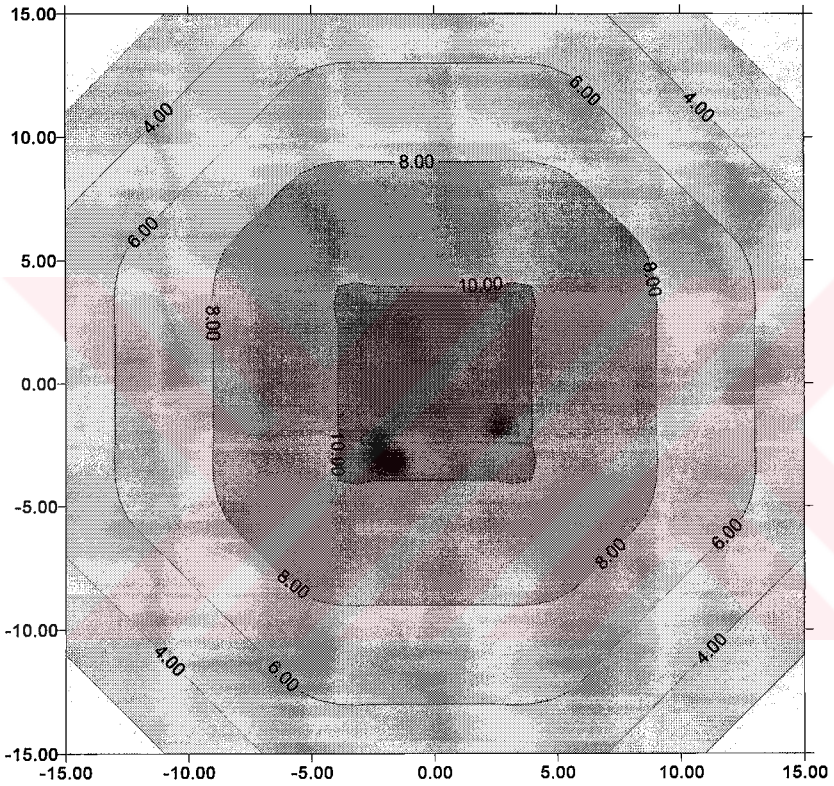


Figure 4.3. Colour encoded contour lines (y = const.) for the surface of Figure 3.30.

Surface interrogation aims at the extraction and visualisation of geometric properties of surfaces, and current surface analysis tools also use contour maps of curvature for detecting surface characteristics. The most effective technique is accepted as the colour encoded Gaussian curvature plots over the surface. Contour maps provide direct and natural interpretation of computer-generated surfaces. They are specifically used to derive detailed local information.

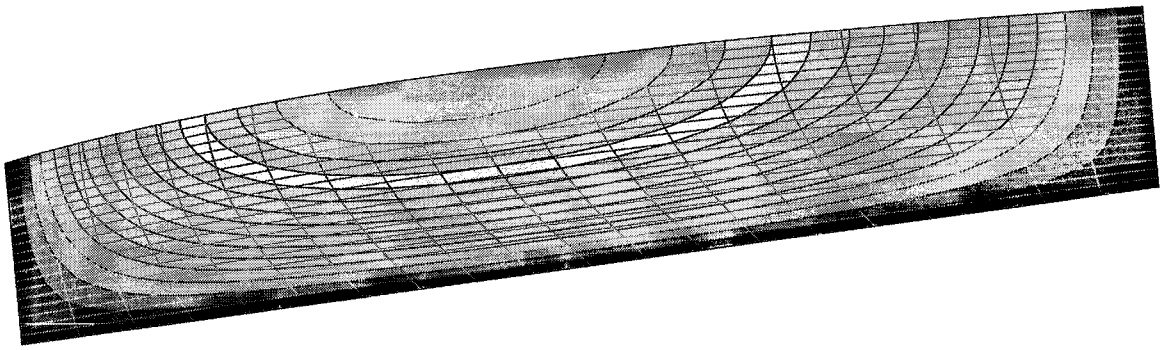


Figure 4.4. Colour encoded contour lines (y = const.) for Wigley form.

Figure 4.5 shows colour encoded Gaussian curvature image of a mathematical hull form (Wigley form). In this case, the expected smooth transitions between contour lines of Gaussian curvature can be seen from the figure as the hull form is intrinsically fair due to its mathematical base.

The mean curvature, on the other hand indicates whether the region is full or hollow, hence measures the deviation of a surface from a minimal surface. Positive values of mean curvature, (H>0) indicate local bumps, and similarly, negative values (H<0) indicate hollow surfaces.

Figure 4.6 shows encoded mean curvature image of Wigley form. The gradual transitions of contour lines of mean curvature can clearly be seen from the figure as expected. It is also worth noting that the contour lines of mean curvature appear to be rather smooth.

In practise, detailed curvature analyses on surfaces should be performed by using both Gaussian and mean curvatures.

In addition to the predefined mean and Gaussian curvatures, another fairness indicator can be defined using the principal curvatures of surfaces. This measure is also used as a standard fairness criterion for surfaces in engineering. It is defined by the following equation as the sum of squared principal curvatures of surfaces:

$$S = \kappa_1^2 + \kappa_2^2$$

This criterion approximates the strain (bending) energy in a thin rectangular elastic plate with small deflection. **Figure 4.7** gives the illustration of the colour-encoded sum of squared principal curvatures of Wigley form.

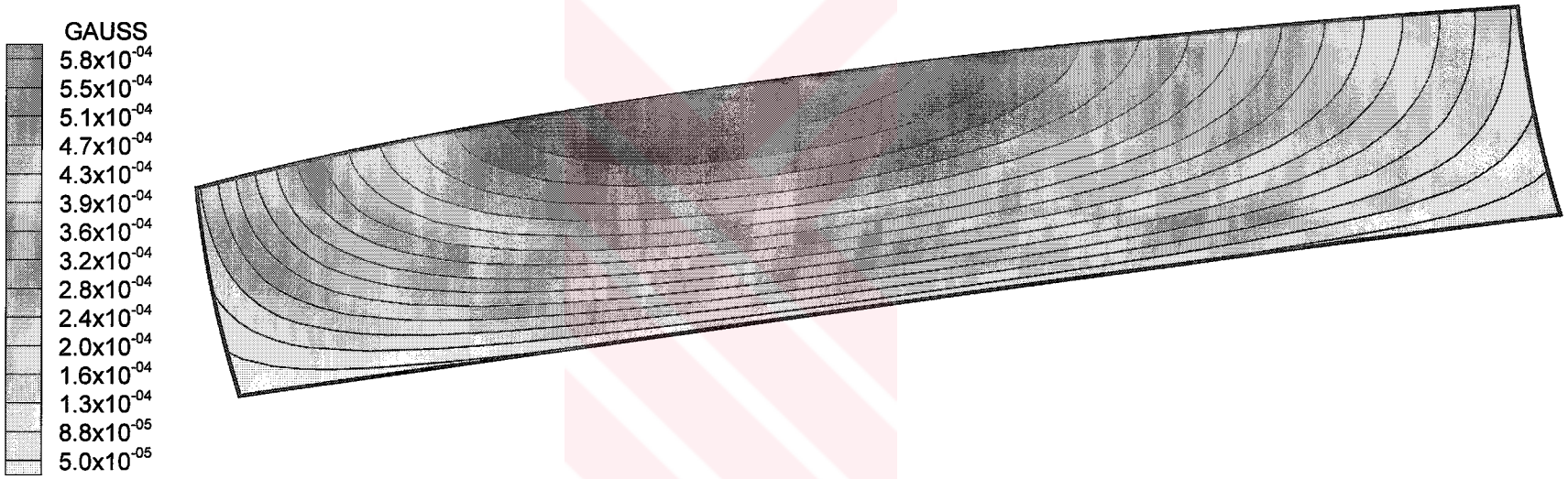


Figure 4.5. Color encoded Gaussian curvature image of Wigley form surface.

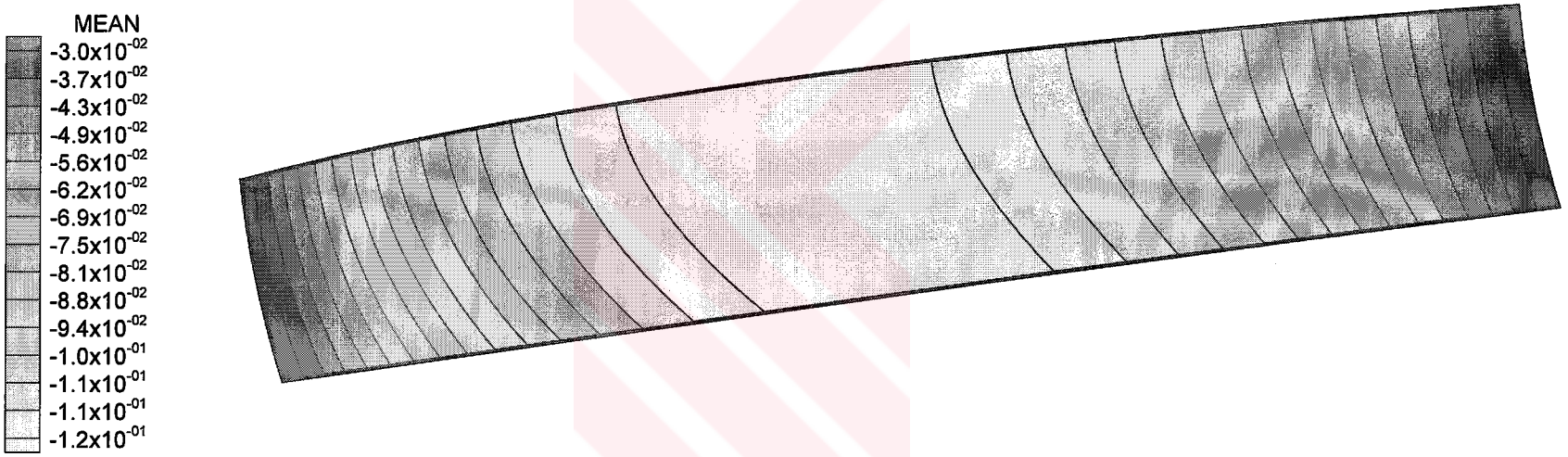


Figure 4.6. Color encoded mean curvature image of Wigley form surface.

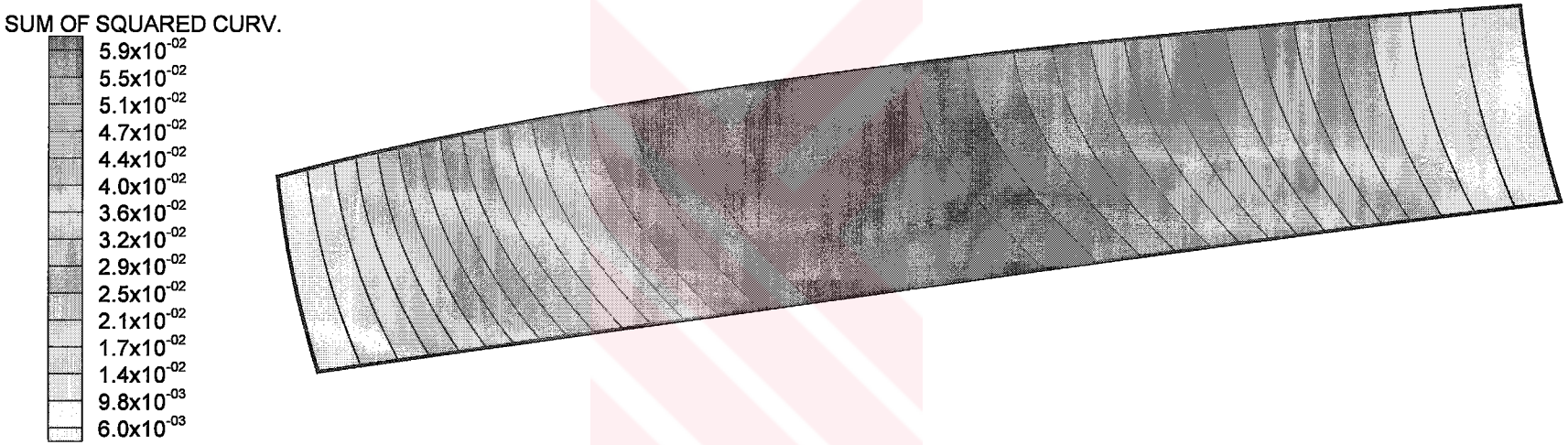


Figure 4.7. Color encoded sum of squared principal curvatures of Wigley form surface.

4.4. The Concept of Fairness

The inherent subjectivity of assessing the appearance of a curve makes the definition of fairness difficult. The difficulty arising is due to the fact that the definition depends on the application. Consequently, a variety of definitions can be found in literature, mostly associated with curvature properties of curves and surfaces for the judgement of fairness.

- The maximum rate of change of curvature must be as small as possible, **(Birkhoff, 1933).**
- The curve should be convexity preserving, **(Theilheimer and Starkweather, 1961).**
- **Atkins et al (1966)** give six requirements for a fair ship curve:

    1. Continuity of a curve
    2. Continuity of first derivative
    3. Continuity of second derivative
    4. Absence of extraneous inflection points
    5. Minimal deviation from the scaled offsets (by approximation)
    6. Good outlook to the eye.

- A frequency analysis of the radius of curvature plotted against arc length might give some measure of fairness, the lower the dominant frequency, the fairer the curve, **(Forrest, 1968).**
- A fair curve has minimum strain energy, **(Reese, 1985).**
- A curve's curvature plot must be almost piecewise linear, with only a small number of segments. Continuity of curvature is an obvious additional requirement, **(Farin et al, 1987).**
- In many design applications a gentle, gradual development of curvature along a curve is much desired and is often used as a subjective measure of curve fairness, **(Meier and Nowacki, 1987).**

- A curve is fair if its curvature plot consists of few monotone pieces, **(Farin and Sapidis, 1989).**

- **Bu-Qing and Ding-Yuan (1989)** suggests a plane curve is called fair if the following three conditions are satisfied:

    1. The curve should be G^2 (Geometric Continuity)

    2. There should be no unwanted inflection points on the curve

    3. The curvature of the curve should vary in an even manner.

        (i) The number of extreme points on the curvature should be as small as possible

        (ii) The curvature of the curve between two adjacent extreme points should vary almost linearly.

- **Calkins et al (1989)** defines the fairing process as follows:

  First and second differences are plotted and shown adjacent to the enlarged curve. Bumps or unfairness in the first and second differences reflect an unfairness in the original curve. The user then changes the point values of the original curve until the first and second differences are considered fair enough.

- Fairness is measured as the integral of the square of the second derivative of the curve, **(Nowacki et al, 1990).**

- A curve is characterised as fair, if its curvature plot is continuous, has the appropriate sign (if the convexity of the curve is prescribed), and it is as close as possible to a piecewise monotone function with as few monotone pieces as possible, **(Sapidis and Farin, 1990).**

- The properties desired of a fair curve are smoothness, shape preservation, absence of extraneous inflection points and the like, but a curve which satisfies all of these criteria as well as the original constraints may still fail to be fair. Fairness measures must depend only on the geometric invariants of the curve and be independent of the curve's parameterisation. A curve's shape should minimise either variation of the radius of curvature or the variation of curvature, **(Roulier et al, 1991).**

- A curve is fair if its curvature plot is continuous and consists of only a few monotone pieces, **(Farin, 1993).**

- A fair curvature plot should be free of any unnecessary variation, i.e., the distribution of curvature on a fair curve must be as uniform as possible, **(Pigounakis et al, 1996)**.

- A C^2 curve is considered fair if it minimises the integral of the squared curvature with respect to the arc length, **(Roulier and Rando, 1994)**.

- **Snaith (1998)** gives the definition of geometric fairness as devoid of unintended, localised;

    1. Erratically distributed, high frequency, high amplitude undulations, i.e., statistical variability / statistical error / noise

    2. Low frequency, low amplitude undulation involving inflections

    3. Flattening tendencies

    4. Bulging tendencies.

Based on these definitions the requirements for a fair curve or surface can be listed as follows:

→ The curve must be continuous along with its first and second derivatives,

→ The curve must be continuous free of extraneous inflection points causing the curve to have bumps, while possessing those specified by the naval architect,

→ Deviation must be as small as possible,

→ The curvature should vary gradually along the curve, i.e., maximum rate of change of curvature should be as small as possible.

All curves satisfying the four conditions stated above will look 'fair' or 'pleasing to the eye of an experienced loftsman and loftsman's batten has been used for many years widely and satisfactorily for interpolating and fairing of previously defined points.

4.5. Smoothing Process

Most hull representation techniques are sensitive to erroneous input data points. Thus, smoothing process should be applied to given points representing the ship lines as a pre-processing step of fairing in which the aim is to eliminate gross errors due to misreading a scale, transcribing the data points incorrectly. Therefore, reasonable set of points is obtained before initiating the fairing process.

The fundamental difficulty lies in distinguishing input human errors (e.g., misreading, etc.) from points that are inconsistent due to lack of fairness in the initial raw data. The former errors are regarded as random and hence should be eliminated. However, the latter may represent the designer's intentions (e.g., knuckles, chines, etc), and their character should be preserved.

The data points which serve as input to the lofting system, are traditionally obtained from offsets of a small scale preliminary design drawing prepared by the naval architect. This drawing is generally, one-fiftieth the size of the actual ship. The small size of the drawing, although beneficial to the architect in many ways, invariably causes small reading errors in the scaled offsets.

The first problem and step in smoothing process is to separate the erroneous points which do not at all reflect the naval architect's intention from the rest. Traditionally, in the mould loft, this operation is achieved by visual observations. Before proceeding with fairing process, the naval architect checks all the points to see if they seem to fit well with one another. Subsequently, the suspected points are rechecked with the lines plan and re-scaled if necessary. However, to automate this system in a computer environment seems difficult due to the possibility that it may cause the designer's intended bad points (e.g., knuckles) to be rejected. Therefore, in these conditions, automated system of lofting cannot be regarded as effective. However, the smoothing process can produce excellent results if applied with the basic assumptions of hull form design. These assumptions are:

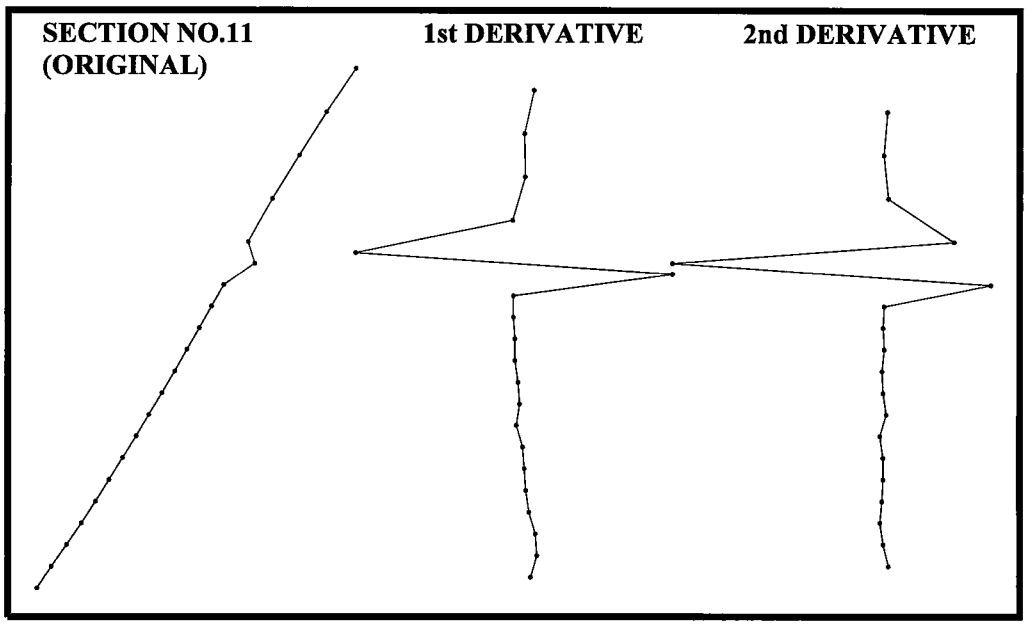
- The initial set of offset points for the hull surface to be analysed is assumed to be reasonable points and sufficiently describes the ship which was intended to be fair.

- The erroneous points are assumed to be due to human error being a small fraction of the total number of points.

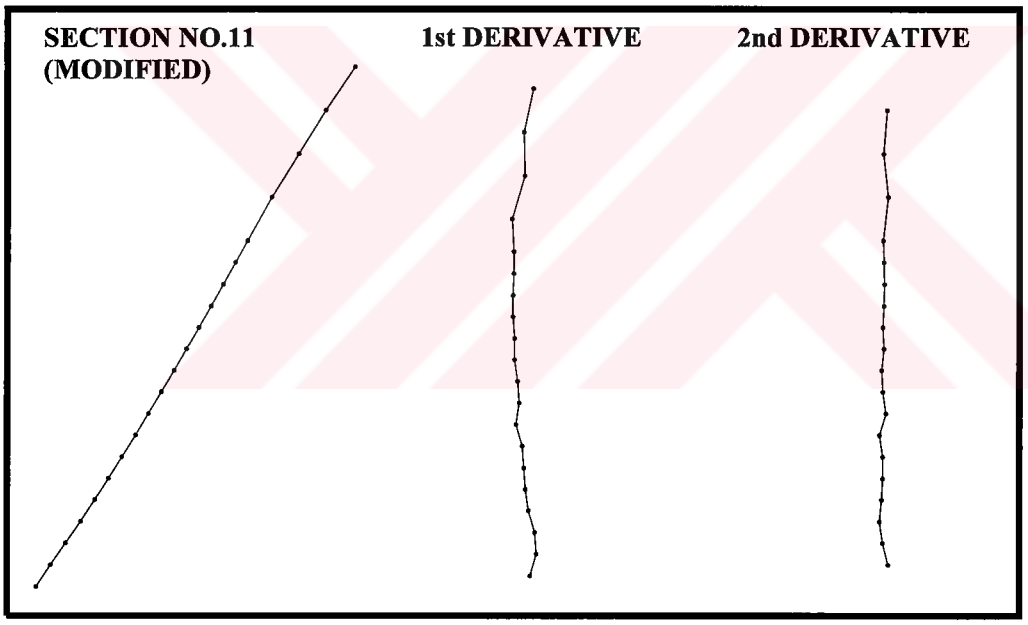
- The curves should not have too many or too closely spaced inflection points to prevent the occurrence of bumps.

In connection with the last assumption, one can easily verify that two closely spaced inflection points in a curve causes it to have bump in that region. Such bumps are not considered smooth and thus, inflection points need to be relatively isolated from one another to prevent the occurrence of bumps.

It is stated above that the fairness problem in hull form design can be investigated separately in preliminary and later stages of ship design. In the first problem, the hull form is roughly defined and designer is relatively free to suggest gross modifications in order to improve fairness. Under the assumption that number of data points that cause unfairness is small these points can be detected in the curvature curve by two consecutive sign changes. Modified original points can be obtained by a spline fit through data points excluding the erroneous one. This process which is referred to as "smoothing" should not be confused with a "fairing" problem where the number of data points to be modified is large and the relative errors in offsets are not large enough to be detected by a change of sign in curvature plot. A smoothing process may be extremely useful when applied before a fairing process in order to detect and eliminate gross errors which may occur during data input or transfer. A typical example is illustrated in **Figure 4.8.a** where the original section line, taken from **Bailey (1976)** obviously includes a misprinted offset value. The first and second differences magnify the error and a change of sign in curvature plot enables us to locate the erroneous point. This point is then eliminated and a cubic spline fit is applied to the remaining offsets. The new section line with smooth first and second differences are shown in **Figure 4.8.b.**



(a)



(b)

Figure 4.8. Smoothing process applied to a section line. (**Narlı and Sarıöz, 1998**)

5. FORWARD FAIRING OF SHIP HULL FORMS

The design of complex curved shapes such as ships and aircraft has always posed a problem and the technology employed has changed radically over the century. It was common practice in the past to lay out the design of a ship in the lofting room where elastic splines are used as a fairing tool to generate ship lines plans called manual lofting process. The results of this process were impressive, and precise hull geometry was produced that could be reliably used for production. The physical splines assumed shapes, which was both aesthetically pleasing and mechanically sound being the curve of minimum strain energy.

The effectiveness of manual lofting and drafting methods with the use of elastic battens and weights at that time had caused late initiation of the research for mathematical representation techniques. However, the obvious advantages in design calculations and production stages have motivated many researchers. Different mathematical methods varying from simple polynomials to complex rational/non-rational B-spline methods have been used in hull form design applications, and B-spline methods have overcome many problems encountered in other techniques and therefore preferably implemented in many CASD (Computer Aided Ship Design) software packages. However, even in state of the art NURBS based system one often faces undesirable shape feature in curves that must be eliminated, as a flawless network of curves is a necessary condition for creating an acceptable hull surface. The full manual lofting process is transferred into computer environment in these packages. This interactive fairing process is similar to the manual fairing process achieved by batten and weights. The only difference is that the corresponding curvature plots of curves, and the effect of modifications can be seen simultaneously. Curvature plots are extensively used by CAD researchers, and developers for inspecting curves and surfaces. Although this interactive process seems advantageous, judgement of fairness is again on subjective bases, depending on the person who is performing the process. Hence, this type of fairing can be doubtful in many cases as it is prone to human errors. Also, the fairing is a global feature of curves and surfaces, local modifications of curvatures are likely to produce unfair curves. Therefore, there is an extensive need of an efficient global fairing process based on an objective fairness criterion.

The early ship lines fairing computer software used polynomials to represent basic ship curves such as waterlines and section lines. The least square or area/moment methods employed by these early systems to compute the coefficients of the polynomials had difficulties of controllability requiring skilled personnel. The early ship lines fairing procedures were based on graphical means in which the actual shape of the hull was represented by lines drawn with views along the principal axes of the hull. Many weeks and sometimes months had to be spent obtaining fair lines, since fairness control was accomplished by visual inspections.

More recent applications, although achieved graphically, are based on more powerful and flexible mathematical representation techniques. They are based on various spline techniques which are found to be more suitable than the polynomial methods. It is widely accepted that B-spline methods emerged as providing more satisfactory results in most cases but it is still difficult to automate the process. Fairing procedures employed in most Computer Aided Ship Design (CASD) software are computerised versions of the manual method and hence have similar drawbacks.

Mathematical fairing of ship lines has been proposed as an alternative method by various authors. Mathematical ship fairing brings some principal advantages compared to graphical method.

In this chapter a flexible numerical procedure for fairing hull form design curves which form the three-dimensional ship body is presented. The traditional solution to the problem of fairing ship hull forms is to reduce the problem to simultaneous fairing of two-dimensional curves on three orthogonal planes, called the section lines, waterlines and buttock lines.

The approach adopted in this chapter is based on approximation of ship lines by B-splines of suitable order. It is shown that the degree of fairness can be improved by increasing the order of the B-spline, however this may result in excessive deviations from the original offset points. A balance between closeness and fairness can be identified by successive applications of B-spline to original offsets in an iterative manner.

An alternative approach based on B-spline fitting is presented in the next section. This approach is shown to minimise the deviation from original offset points.

In order to obtain three-dimensional fairness, two-dimensional ship lines on three orthogonal planes need to be faired simultaneously. The fairing process is first

applied to waterlines and the modified offsets are transferred to other planes. The iterative process is repeated until specified degree of fairness is achieved.

This procedure is applied for fairing the lines of a distorted mathematical hull form, the parent hull of a well-known high-speed displacement hull form series, and a high-speed naval corvette form.

While the objective is to obtain three-dimensional fairness it is reasonable to assume that designer should preserve the hull form characteristics which may be selected for the best hydrodynamic performance. Therefore, there should be certain limits on variation from the original lines plan. These limits are expressed in terms of conventional naval architectural parameters, i.e., the block coefficient, longitudinal centre of buoyancy etc. The results indicate that these procedures are particularly useful for high-speed displacement hull forms and can be used as practical design tools in the early stages of ship design.

5.1. Fairness and Closeness Metrics

Fairness is not a local property of a curve, and for a fair curve the maximum rate of change of curvature must be as small as possible along the curve. The radius of curvature R_c of any curved line is expressed as

$$R_c = \frac{\left[1+\left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}}{\frac{d^2y}{dx^2}}$$

At any selected point a curve has an effective radius of curvature equal to that of a circle. The absolute curvature, $1/R_c$, approaches zero with second derivative d^2y/dx^2 . Curvature plot of a curve can be used to detect geometric properties such as fairness. Two properties of a curve readily available from a plot of its curvature are the presence of inflection points and the variation of curvature. Therefore, the fairness problem in design of ship lines may be defined as:

- minimum rate of change of curvature,
- limited number of inflection points, and

- minimum deviation from given offset points within which the fitted lines must lie.

In many Computer Aided Hull Form Design Software the fairing process is based on interactive modification of curvature plots. However, this process is not suitable in an automated computational procedure and numerical criteria, which reflect visual observations are needed. For any ship line the sum of squares of second differences calculated at offset points may be used as an indicator of fairness. This sum which will be referred to as the *fairness number*, is proposed as a numerical criterion, to indicate a reduced rate of change of curvature as it gets smaller. It should be noted here that the second derivatives may be used as a good approximation for curvature, and as the ship lines are defined by discrete points, divided differences can be used instead of derivatives. Therefore, second differences can be used to compute the fairness numbers of the hull forms.

However, another criterion is needed to represent the similarity between the original and modified ship lines. This criterion may either be defined as the maximum absolute difference between the offset values or the sum of squares of differences. In this study, the first option is preferred and the absolute maximum difference is called the *closeness number*.

The ship line to be faired is defined as $[x_i, y_i : i = 1, \ldots, n]$ and the modified offsets are defined as $[x_i', y_i' : i = 1, \ldots, n]$. The fairness and closeness criteria will be given respectively as

$$FN = \sum_{i=1}^{n} \left(\frac{d^2 y}{dx^2} \right)^2 \qquad FN' = \sum_{i=1}^{n} \left(\frac{d^2 y'}{dx'^2} \right)^2$$

$$CN = \max \left| y_i' - y_i \right| \quad \text{where} \quad i = 1, \ldots, n$$

where FN and CN represent the fairness and closeness numbers respectively. The objective of the designer is to minimise both numbers simultaneously, however this is not possible in many cases and a compromise is needed. (**Narlı and Sarıöz, 1998**)

5.2. Fairing of Hull Forms by Iterative B-Spline Approximation

The B-spline approximation process adopted in this section is based on the approximation of a given form plan by mathematical functions. Various mathematical tools such as polynomials, cubic splines, Bézier curves and B-splines are available and the experience has shown that B-spline curves provide the best alternative for approximating ship lines. As the order of the B-spline curves is increased smoother curves can be obtained but the deviation from the original data points will increase. This feature of B-spline curves has led to the development of practical fairing procedures.

This procedure is based on successive application of B-spline curves for ship waterlines and section lines in an iterative manner until a satisfactory degree of fairness is achieved. The offset points of each waterline in the lines plan is assumed to be the polygon vertices for the B-spline representation and the modified offsets are then obtained from the B-spline curve. These offsets then become the polygon vertices for the corresponding section lines. The process is successively applied until satisfactory levels of fairness and deviation from original offsets are obtained.

In conventional CASD applications B-spline curves are used to represent ship lines which are defined by a small number of polygon vertices. These points are generally defined interactively to give a close approximation to the original offset points or alternatively the position of polygon vertices can be obtained from the position of offset points. When the number of offset points that define the design curve is relatively large (21 for a typical waterline) these offset points can be used as the defining polygon and the resulting B-spline curve will preserve the characteristic shape with not excessive deviation from the original offsets. As the degree of the B-spline increases fairness and deviation from the original points will also increase. This is illustrated in **Figure 5.1** where original offset points and B-spline curves which take these offsets as polygon vertices are shown on the left hand side of the figure. The degree of the B-spline curve increases downwards so does the fairness. This can be verified by visual observation of the first and second derivatives, which are shown in the middle and on the right hand side of the figure, respectively. It is also clear that as the degree of the B-spline curve is increased we obtain a poorer representation of the offset points.

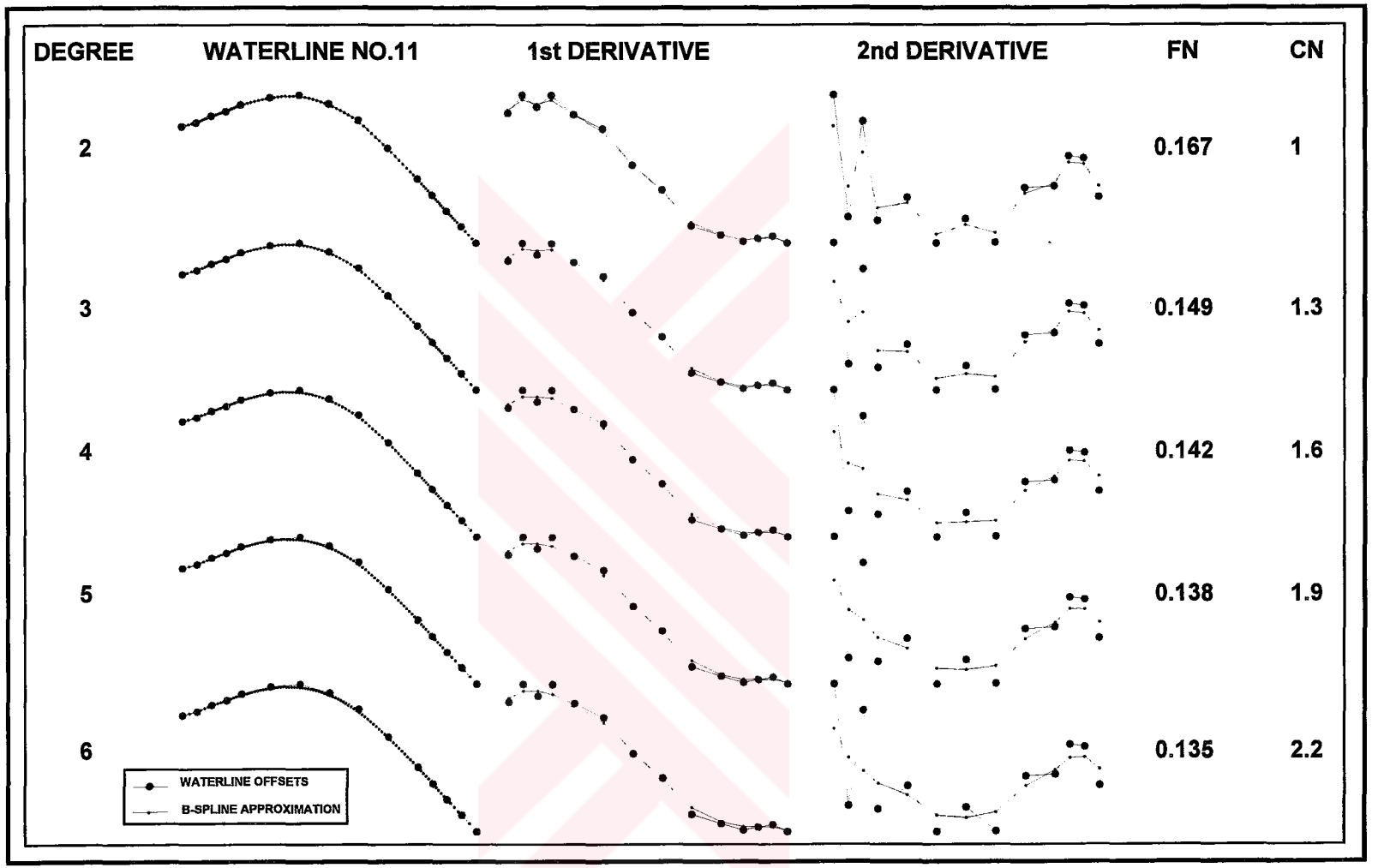


Figure 5.1. Effect of order in B-spline approximation of a waterline. **(Narlı and Sarıöz, 1998)**

This feature of B-spline representation can be applied in fairing of ship lines. The problem obviously is three-dimensional, however it can be reduced to fairing of two-dimensional lines on three two-dimensional planes, i.e., the waterlines, buttock lines and section lines. The fairing process, as illustrated in **Figure 5.2** starts with B-spline representation of waterlines. In many cases the number of offset points which define any waterline is more than 20, hence a low order (4th or 5th) B-spline curve will provide close representation of the original waterline. Then the modified offsets are transferred into body plan and they form the polygon vertices for new section lines. The order of B-spline curves will depend on the number of offset points and for cases with more than 10 offsets to define each section line, low order B-spline curves provide excellent representations.

To demonstrate the performance of the fairing procedure a mathematical hull form which was first suggested by **Wigley (1934)** selected. The hull surface, which has a fore-aft and port-starboard symmetry, is defined by the following equation:

$$y(x,z) = \frac{B}{2}\left\{1-(2x/L)^2\right\}\left\{1-(z/D)^2\right\} \qquad \text{where}$$

L : length

B : breadth

D : depth

x : distance from amidships, positive forward

y(x,z) : offset value at (x,z)

z : distance from deck line, positive downwards

This mathematical hull form, labelled as *Parent Hull* in **Figure 5.3**, has a fair surface and hence fair waterlines, buttocks and section lines. This form is then randomly distorted to obtain a starting point for the fairing process, which is based on successive approximation of ship lines by B-splines of suitable order. The order of the B-spline depends on the number of offset points and low orders should be preferred in order to minimise deviation from the original offsets. After five iterations a final form as shown in **Figure 5.3** is obtained.

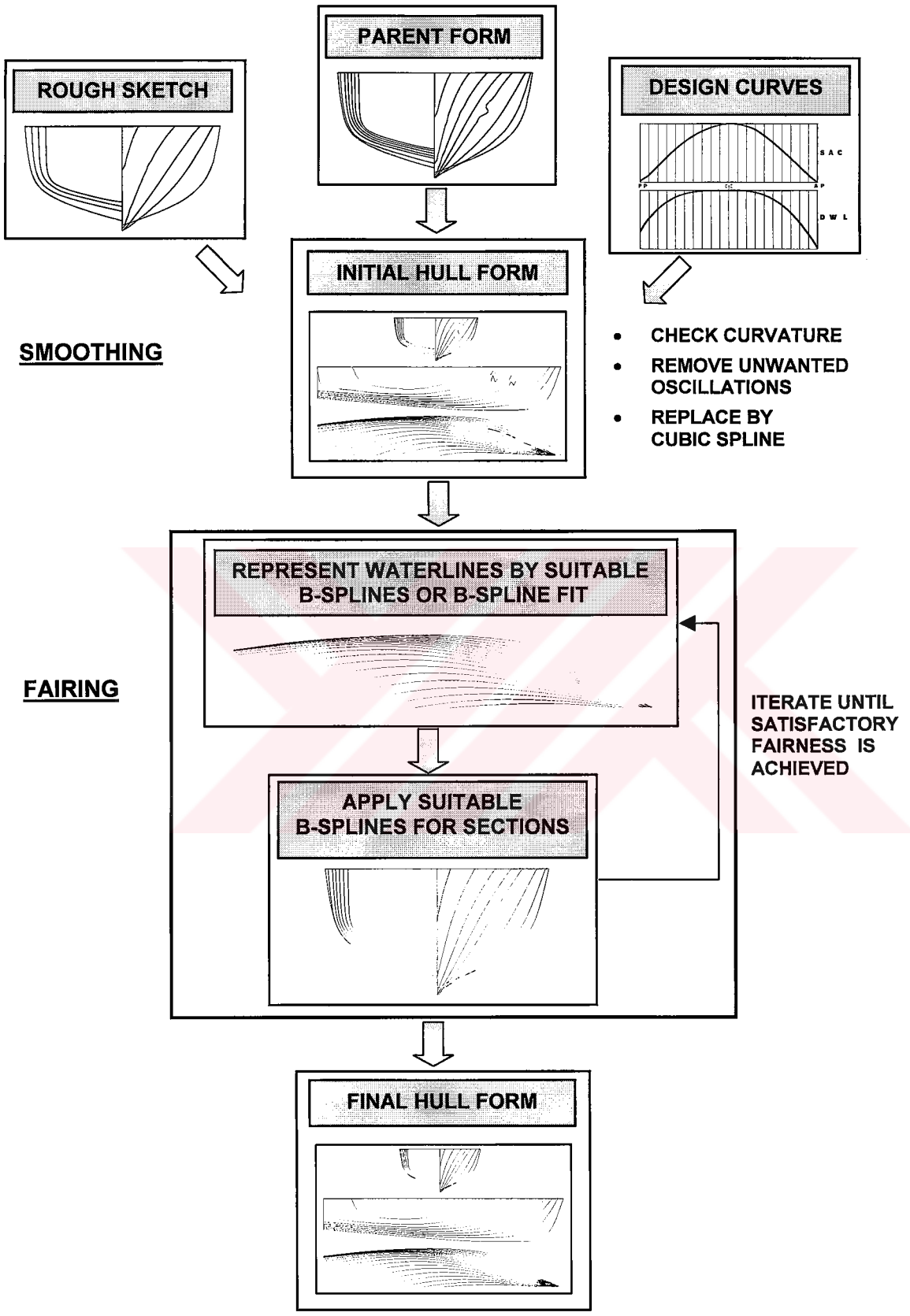


Figure 5.2. Hull form fairing process. **(Narlı and Sarıöz, 1998)**

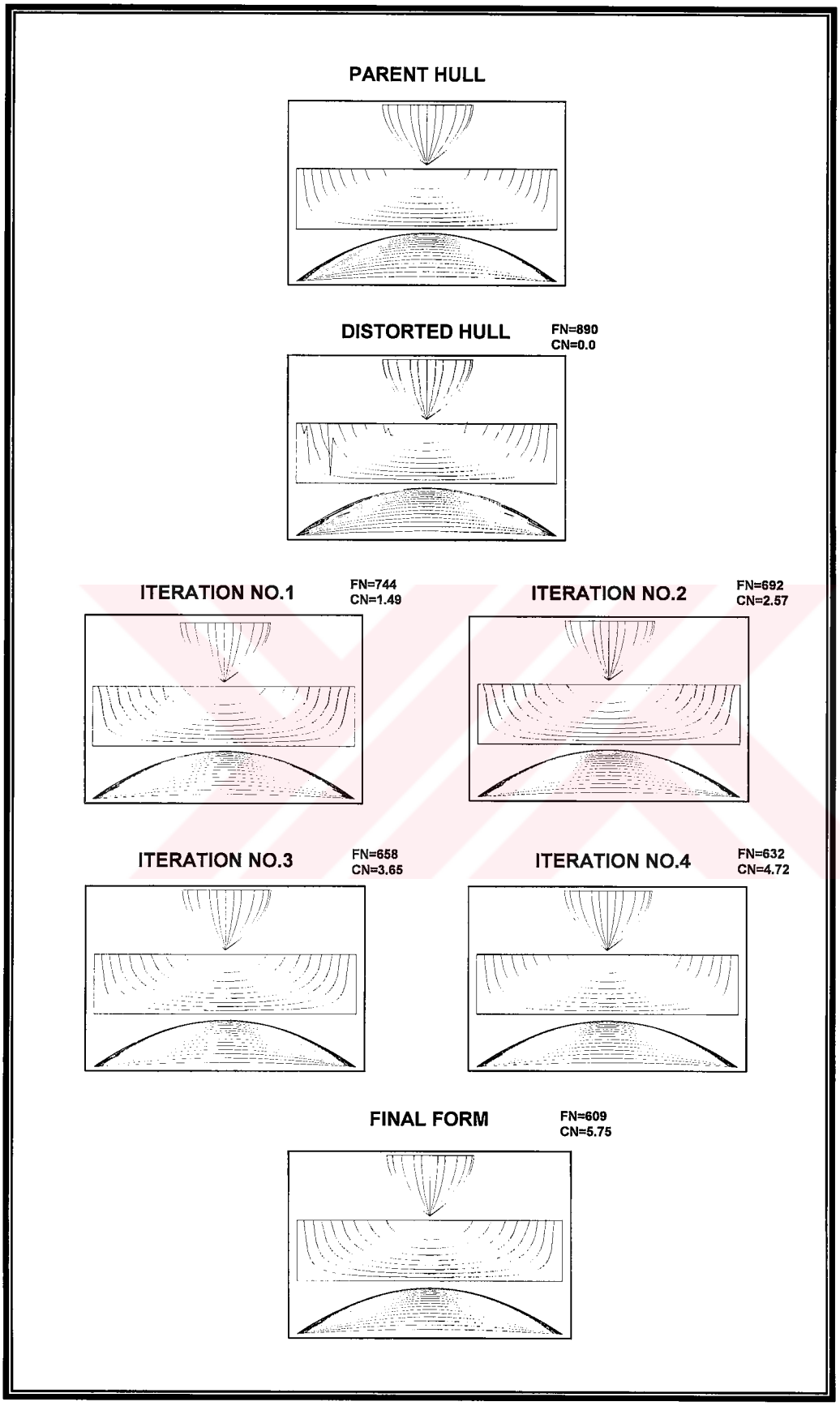


Figure 5.3. B-spline approximation of Wigley form. (**Narlı and Sarıöz, 1998**)

As can be seen from **Figure 5.3**, the final form has a smaller fairness number, which indicates fairer lines as can be observed visually. However, there is a penalty involved in the process, which is characterised by the closeness number, i.e., the maximum deviation between the original and faired offset points. As can be seen, this number increases with increasing fairness. In this case the process is terminated after 5 iterations in order to stay within specified deviation limits. Some geometric properties of each hull form is shown in **Table 5.1**.

These results indicate that the variation in geometric properties and displacement may be up to 6%, which may not be acceptable in many cases. These differences may be reduced by sacrificing in fairness, however a better way would be to apply affine transformation techniques that are described in Chapter 2. This would not detoriate the fairness characteristics already obtained but will provide a closer approximation to the original hull form.

Table 5.1. B-spline approximation of Wigley form.

	PARENT FORM	DISTORTED FORM	ITER. NO. 1	ITER. NO. 2	ITER. NO. 3	ITER. NO. 4	FINAL FORM
L (m)	16.000	16.000	16.000	16.000	16.000	16.000	16.000
B (m)	1.600	1.618	1.598	1.580	1.563	1.546	1.530
D (m)	1.000	1.000	1.000	1.000	1.000	1.000	1.000
C_{WP}	0.667	0.662	0.662	0.662	0.661	0.660	0.659
C_B	0.444	0.440	0.440	0.439	0.438	0.437	0.436
C_M	0.667	0.661	0.662	0.662	0.661	0.661	0.660
LCB (%)	0.000	0.018	0.012	0.012	0.010	0.009	0.008
$\nabla(m^3)$	11.375	11.393	11.240	11.092	10.946	10.803	10.662
FN	731.15	889.36	744.75	691.98	658.32	632.24	609.34
CN	0.000	0.000	1.490	2.570	3.650	4.720	5.750

As a second example the parent hull of NPL high-speed displacement hull series **Bailey (1976)** is selected. The original offsets include a misprinted value, which is detected and eliminated by the smoothing routine prior to the fairing process. The lines plan of the original hull form and the final form of the iterative fairing process are shown in **Figures 5.4.a** and **5.4.b**. Three-dimensional shaded images of the initial and final NPL forms are also shown in **Figure 5.5** and **5.6**, which clearly indicate the success of the process.

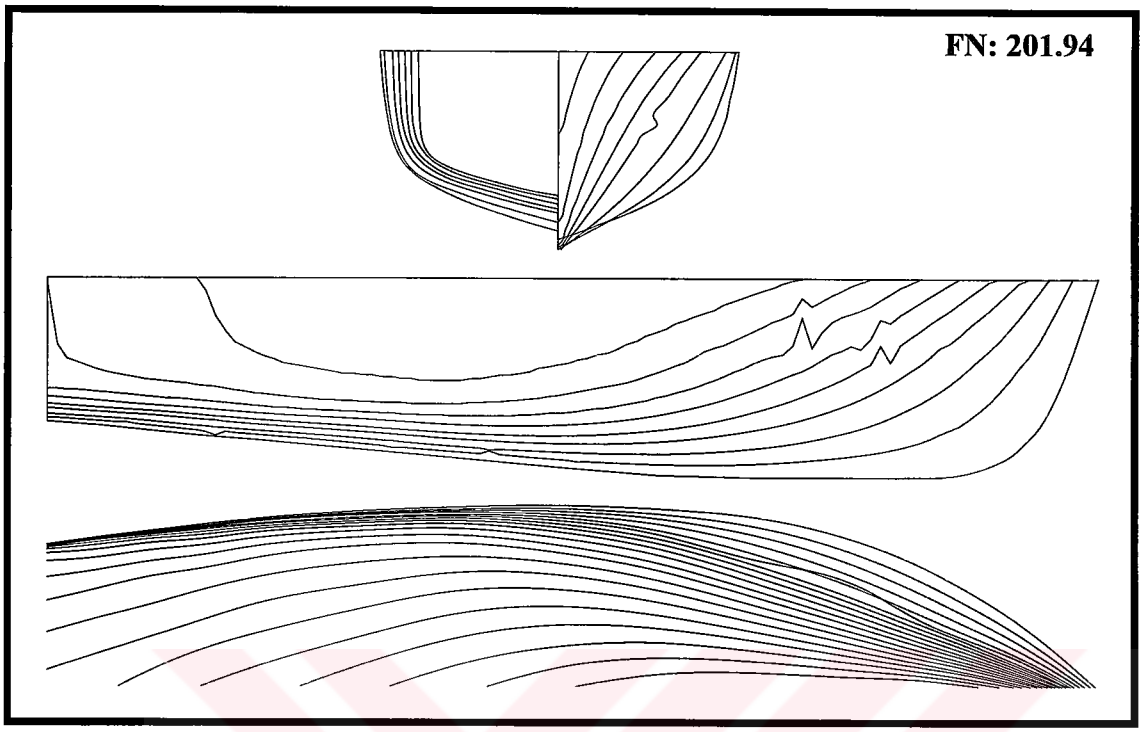


Figure 5.4.a. Initial NPL form. **(Narlı and Sarıöz, 1998)**

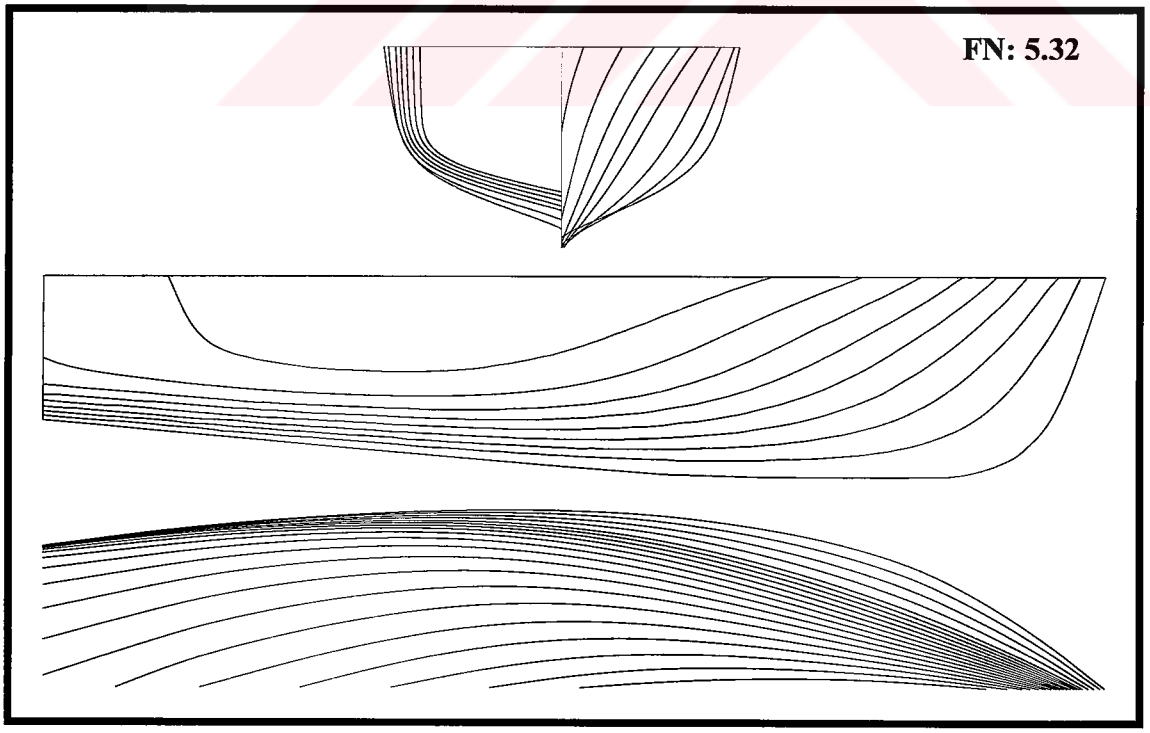


Figure 5.4.b. B-spline approximation of NPL form. **(Narlı and Sarıöz, 1998)**

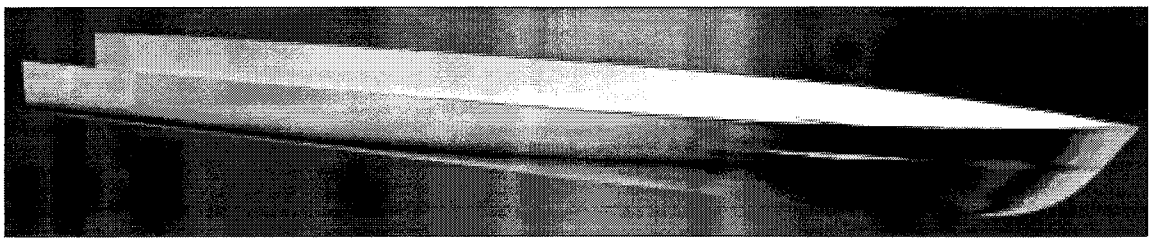


Figure 5.5.a. NPL initial hull form.

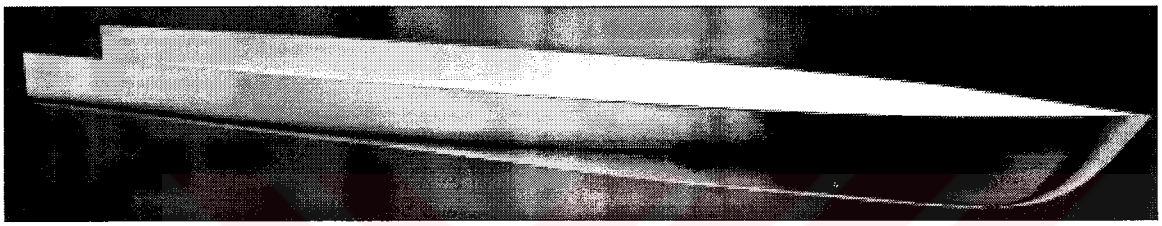


Figure 5.5.b. NPL final hull form.

Figure 5.6.a. NPL initial hull form.

Figure 5.6.b. NPL final hull form.

Some geometric properties of the parent form and the alternatives developed during the fairing process as well as those of the final form are given in **Table 5.2**.

Table 5.2. B-spline approximation of NPL parent hull.

	PARENT FORM	ITER. NO. 1	ITER. NO. 2	ITER. NO. 3	ITER. NO. 4	ITER. NO. 5	ITER. NO. 6	FINAL FORM
L (m)	2.540	2.540	2.540	2.540	2.540	2.540	2.540	2.540
B (m)	0.438	0.436	0.434	0.433	0.431	0.430	0.428	0.426
T (m)	0.240	0.240	0.240	0.240	0.240	0.240	0.240	0.240
C_{WP}	0.848	0.847	0.846	0.845	0.845	0.844	0.844	0.844
C_B	0.540	0.537	0.535	0.533	0.532	0.530	0.529	0.527
C_M	0.710	0.705	0.700	0.696	0.691	0.687	0.683	0.680
LCB (%)	-6.126	-6.186	-6.235	-6.283	-6.329	-6.373	-6.417	-6.549
$\nabla(m^3)$	0.144	0.143	0.142	0.141	0.140	0.139	0.138	0.137
FN	201.94	38.830	14.068	8.333	6.588	5.883	5.530	5.320
CN	0.0	4.890	5.175	5.443	5.704	6.048	7.045	8.001

Similar trends can be observed in both examples (Wigley and NPL form), i.e., a fair set of ship lines can be produced after a few iterations. However, there may be excessive deviations from the original form, which may not be acceptable by the designer. This is a direct consequence of B-spline approximation method, and the use of B-spline fitting, which is illustrated in the next section, could be used to obtain closer representations. Alternatively, the offsets of the faired hull may be multiplied by a constant obtained by comparing the beam or volume values of the original and final forms, or linear distortion techniques may be used to preserve the original hull form characteristics.

A typical corvette form has also been selected as a third application. The form labelled as the *parent form* can be seen in **Figure 5.7.a.** The process is applied with varying order of B-splines where low orders provide close representation of the original curves with the sacrifice in fairness so it has been decided to adopt high orders and number of iterations in order to produce a fairer form. After 15 iterations the *final hull form* as shown in **Figure 5.7.b** is obtained.

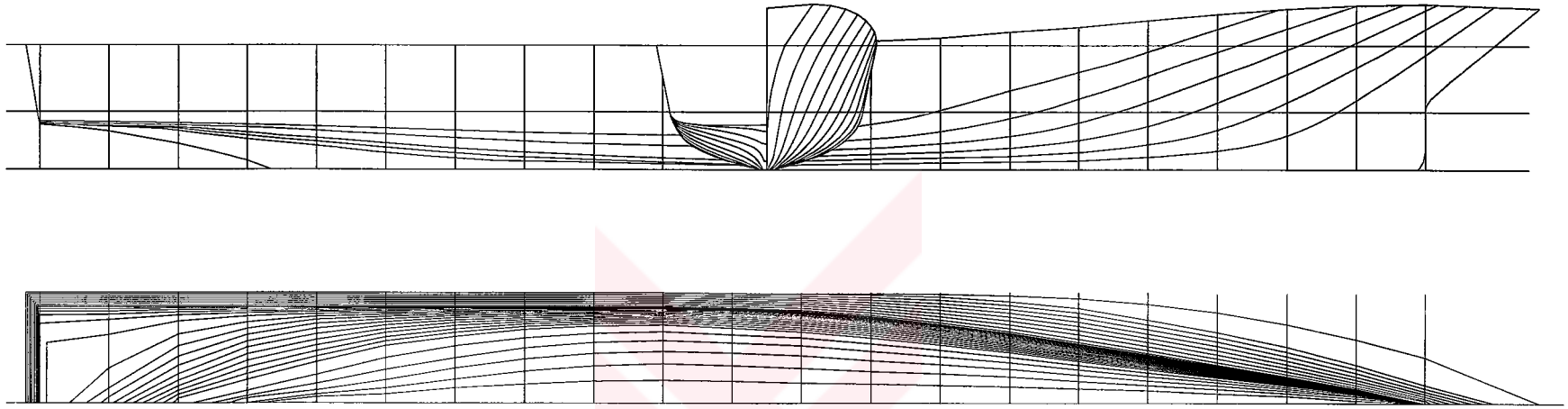


Figure 5.7.a. Original corvette (Parent form). **(Narlı, 1997)**

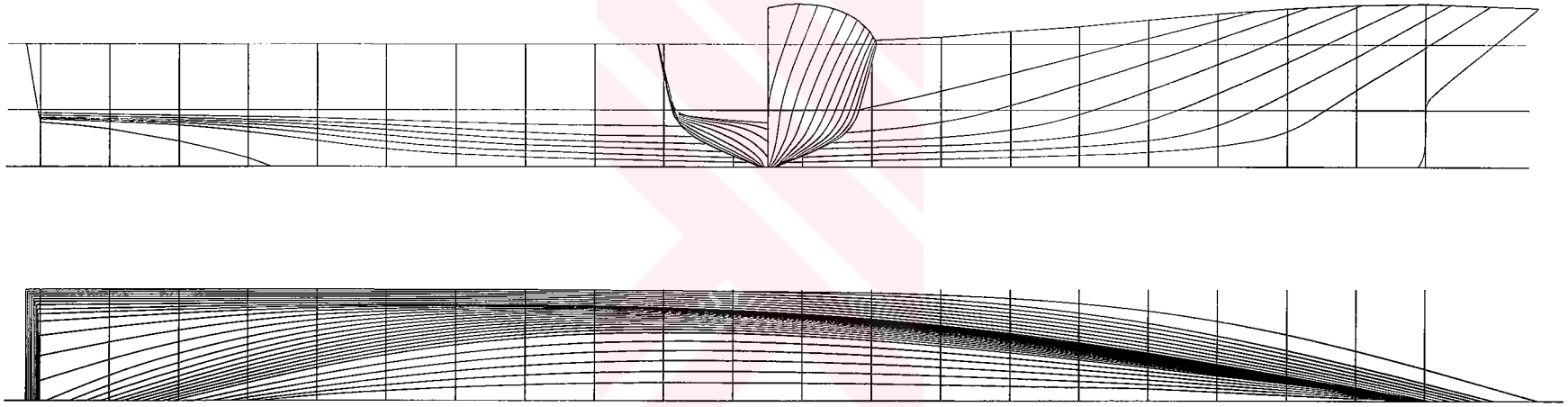


Figure 5.7.b. B-spline approximation of corvette (Final Form) **(Narlı, 1997)**.

The results of this application indicate that the variation in geometric properties and displacement may be up to %10, which may be unacceptable in many cases. In order to overcome this excessive deviation problem due to high number of iterations, and selection of high-order B-splines, linear distortion techniques, described in Chapter 2 are applied to the final form. Linear distortion techniques have found practical applications in deriving the lines for a new design from a similar basis ship which permits the longitudinal spacing of the transverse sections to be adjusted to suit the new curve of areas. Hence, a new acceptable form with same fairness characteristics, which is not much deviated from the initial form is obtained, and labelled as the *distorted form*. Thus, fairness characteristics already obtained will remain, providing a closer approximation to the original form.

The *final form* obtained as the product of the fairness process and *distorted form* obtained by applying one-minus prismatic method to the final form to preserve original hull form characteristics are shown in **Figure 5.8,** where dotted lines indicate the distorted form. Some geometric properties and fairness numerals of each hull form developed during the fairing process as well as those of the distorted form can be seen in **Table 5.3.**

Table 5.3. B-spline approximation of corvette form.

	PARENT FORM	ITER. NO. 1	...	ITER. NO. 15	FINAL FORM	DISTORTED FORM
L (m)	79.100	79.100	...	79.100	79.100	79.100
B (m)	12.600	12.600	...	12.600	12.600	12.600
T (m)	3.150	3.150	...	3.150	3.150	3.150
C_{WP}	0.796	0.794	...	0.773	0.771	0.803
C_B	0.464	0.459	...	0.418	0.415	0.469
C_M	0.708	0.704	...	0.645	0.642	0.659
LCB (%) (-Aft)	-3.890	-3.772	...	-2.790	-2.736	-3.750
$\nabla(m^3)$	1275.78	1262.38	...	1135.82	1128.10	1273.85
FN	0.150	0.120	...	0.053	0.051	0.078
CN	0.000	3.575	...	8.924	9.110	0.281

Figure 5.9.a and **5.9.b** depicts shaded images of parent and final forms of the corvette, respectively.

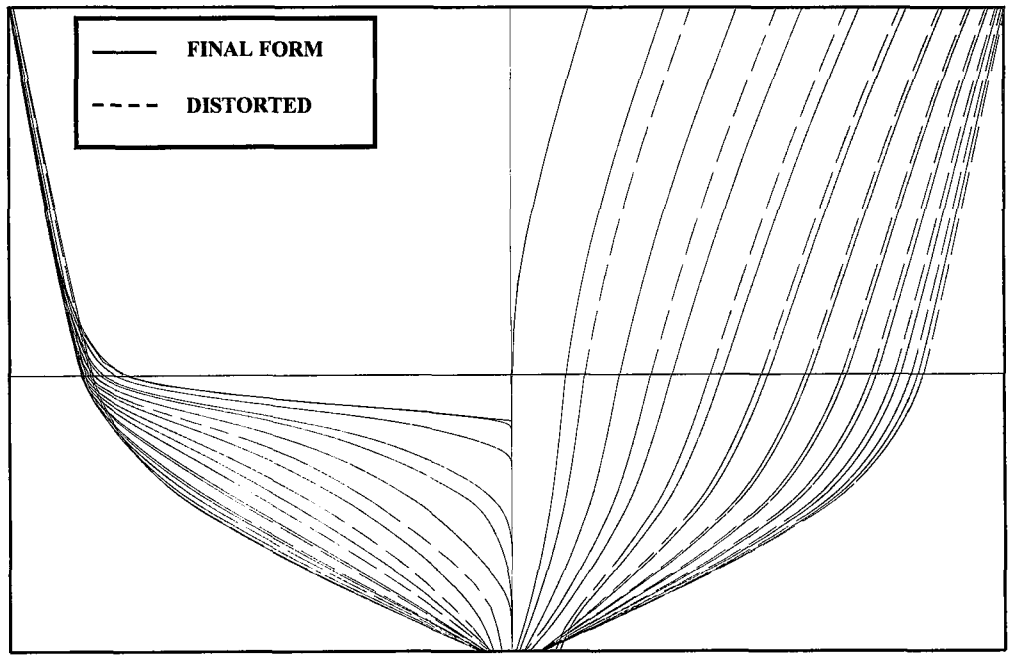


Figure 5.8. Final and distorted hulls of the corvette. (**Narh, 1997**)

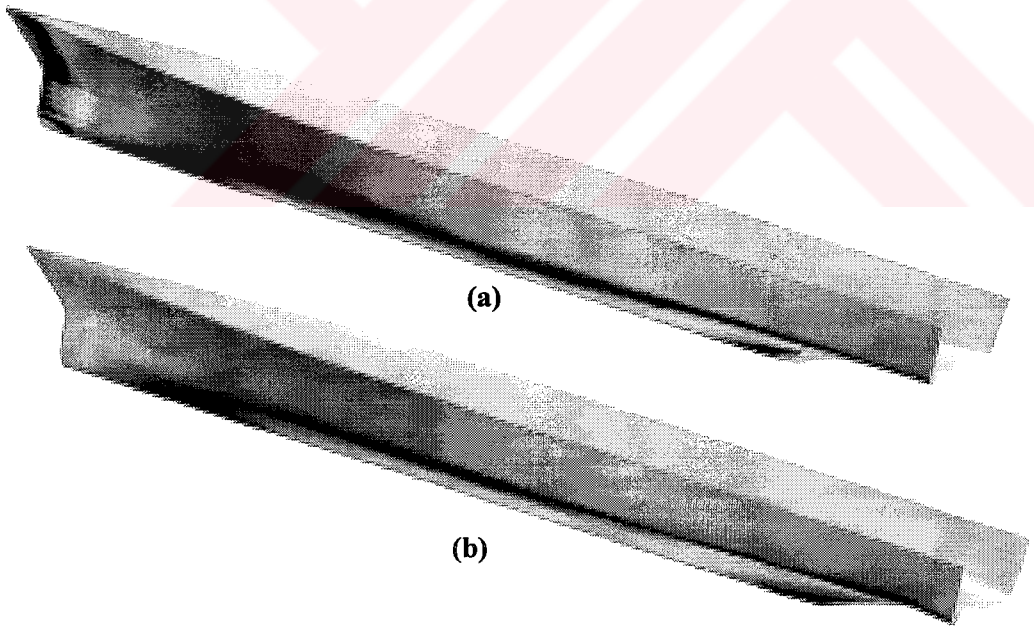


Figure 5.9. (a) Original corvette, **(b)** Final form.

5.3. Fairing of Hull Forms by Iterative B-Spline Fitting

The second procedure adopted is similar in its iterative manner with the first approach. This useful fairing procedure is also obtained from B-spline curves, which is called the *B-spline Curve Fit*. Here, the successive application of B-spline curves starts with section lines and continues with B-spline curve fit for waterlines. The B-spline curve fit process provides the polygon vertices, which yield a B-spline curve representation of the original offset points.

By selecting suitable number of polygon vertices one can increase the degree of fairness with increasing penalty on closeness to original points. If the number of polygon vertices is the same as the offset points, the generated B-spline curve from this polygon fits through all offset points. This feature of B-spline curve fit satisfies the closeness requirement, as there is no deviation from original points. However, the fitted curve may develop unacceptable oscillations due to increasing number of polygon vertices requiring the inversion of a large ill-conditioned matrix. Therefore, a compromise between the degree of fairness and closeness should be sought by changing the number of polygon vertices.

In this approach, the offset data points are used to obtain the defining polygon to generate the B-spline curve passing through all data points. Hence, the B-spline curve in matrix form can be written as

$$[D]=[N][B] \qquad 2 \le k \le n+1 \le j$$

$$[D]=\begin{bmatrix} D_1(t_1) \\ D_2(t_2) \\ \vdots \\ D_j(t_j) \end{bmatrix} \qquad [N]=\begin{bmatrix} N_{1,k}(t_1) & \cdots & N_{i,k}(t_1) \\ \vdots & \ddots & \vdots \\ N_{1,k}(t_j) & \cdots & N_{i,k}(t_j) \end{bmatrix} \qquad [B]=\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_j \end{bmatrix}$$

Here, [D] represents the given set of offset data, [N] the basis functions, and [B] the defining polygon. In this case, number of defining polygon points is equal to the number of data points. So the matrix [N] is square and its inverse can be taken to obtain the defining polygon likewise

$$[B]=[N]^{-1}\,[D]$$

If the number of defining polygon points is less than the data points, the fitted B-spline curve does not pass through all data points but represent the original points very closely. Consequently, a fairer curve showing minimum deviation from original offsets is obtained. This feature of B-spline curve fit can be used in developing fairing processes. When fewer polygon points than data points are defined, the matrix [N] be no longer square. To make it square for taking the inverse, it should be multiplied by its transpose. Following this procedure the defining polygon is thus obtained.

$$[D]=[N][B]$$

$$[N]^T[D]=[N]^T[N][B]$$

$$[B]=\left[[N]^T[N]\right]^{-1}[N]^T[D]$$

Therefore, by using this defining polygon the B-spline curve that fairs the given data is obtained. More details about B-spline approximation and fitting can be found in **(Rogers and Adams, 1990)**.

The fairness and closeness characteristics of the new curve will depend on the number of polygon vertices selected. By increasing the number of vertices one can obtain closer representation with degrading fairness properties. An iterative process best achieves a compromise solution between fairness and closeness requirements where the number of original offset points determines the number of polygon vertices.

This procedure is also applied for fairing the lines of a distorted Wigley form and the parent hull of NPL series. The results generally indicate that one can obtain closer representations of the original hull form compared with the B-spline approximation procedure. Fairing of the distorted Wigley form by B-spline fitting process is illustrated in **Figure 5.10**. Compared with B-spline approximation this process produces smaller closeness numbers with a slight penalty in fairness number. Similar trends are observed in NPL case which is shown in **Figure 5.11**.

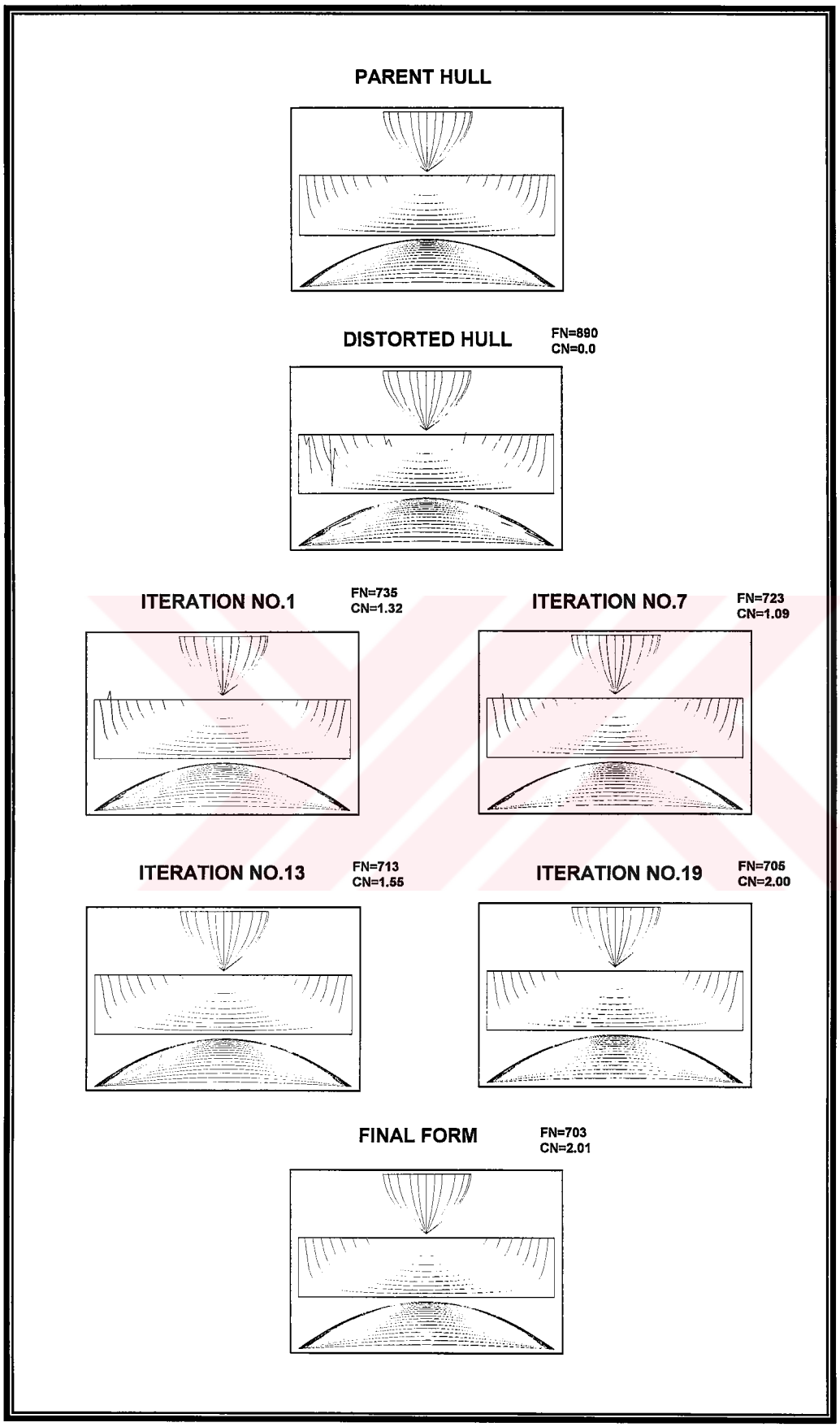


Figure 5.10. B-spline fitting of Wigley form. **(Narlı and Sarıöz, 1998)**

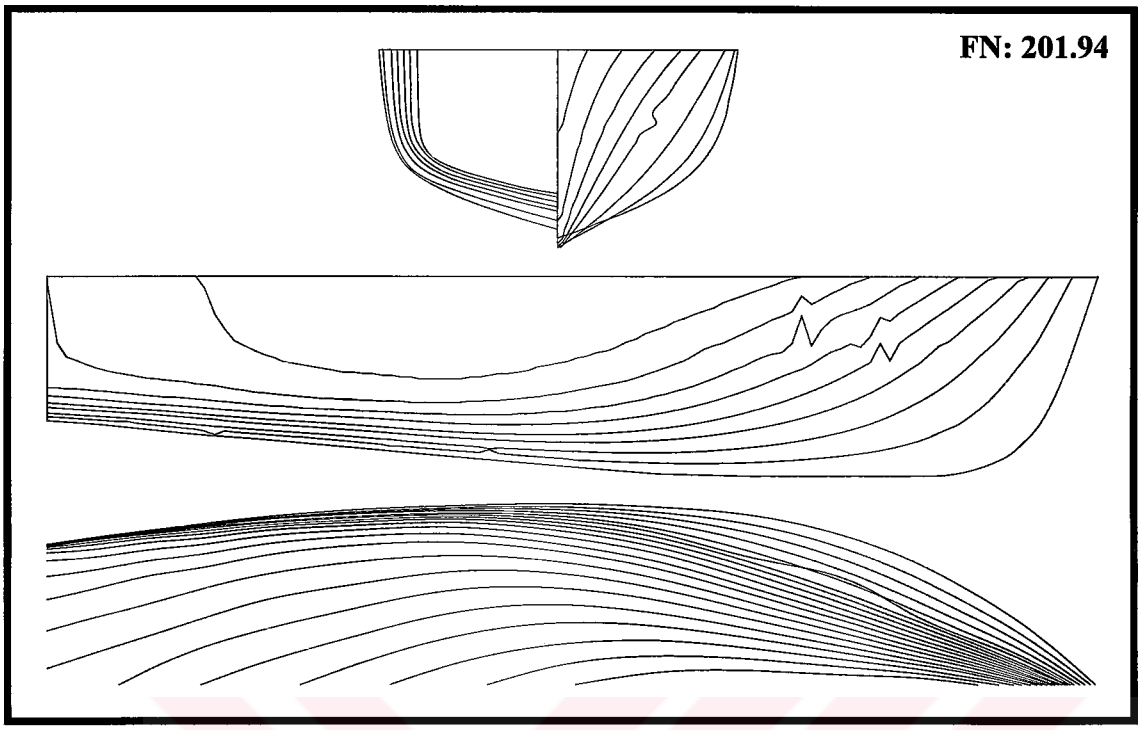


Figure 5.11.a. Initial NPL form. (**Narlı and Sarıöz, 1998**)

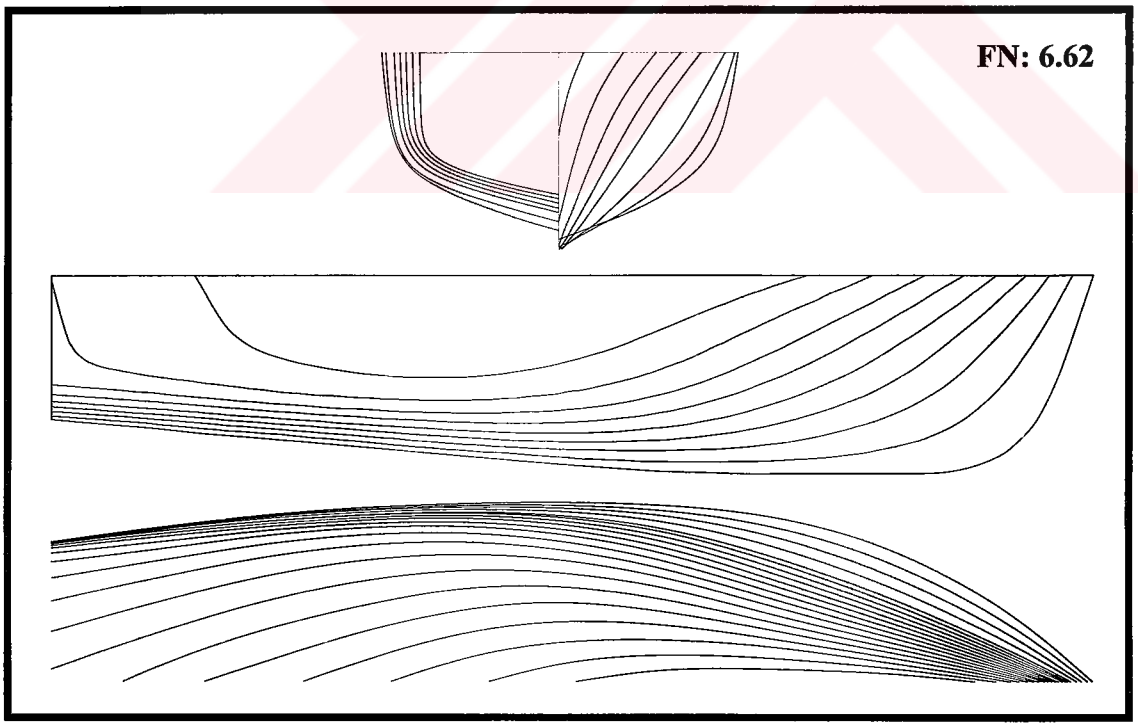


Figure 5.11.b. B-spline fitting of NPL form. (**Narlı and Sarıöz, 1998**)

Geometric properties of the original and final forms with those generated during the iterative process of fairing are presented in **Tables 5.4** and **5.5** for Wigley and NPL hull forms, respectively. These tables also indicate that B-spline fitting could produce closer representations to the original hull forms as required with a slight penalty in fairing characteristics.

Table 5.4. B-spline fitting of Wigley form.

	PARENT FORM	DISTORTED FORM	ITER. NO. 1	ITER. NO. 2	...	ITER. NO. 18	ITER. NO. 19	FINAL FORM
L (m)	16.000	16.000	16.000	16.000	...	16.000	16.000	16.000
B (m)	1.600	1.618	1.609	1.609	...	1.609	1.609	1.609
D (m)	1.000	1.000	1.000	1.000	...	1.000	1.000	1.000
C_{WP}	0.667	0.662	0.666	0.666	...	0.666	0.666	0.666
C_B	0.444	0.440	0.442	0.442	...	0.435	0.435	0.435
C_M	0.667	0.661	0.663	0.662	...	0.654	0.653	0.652
LCB (%)	0.005	0.018	0.011	0.013	...	0.008	0.008	0.008
$\nabla(m^3)$	11.375	11.393	11.380	11.367	...	11.205	11.196	11.186
FN	731.15	889.36	735.06	730.70	...	705.79	704.46	703.15
CN	0.000	0.000	1.117	1.220	...	1.931	2.005	2.008

Table 5.5. B-spline fitting of NPL parent hull.

	PARENT FORM	ITER. NO. 1	ITER. NO. 2	ITER. NO. 3	FINAL FORM
L (m)	2.540	2.540	2.540	2.540	2.540
B (m)	0.438	0.437	0.437	0.437	0.437
T (m)	0.240	0.240	0.240	0.240	0.240
C_{WP}	0.848	0.849	0.849	0.849	0.849
C_B	0.540	0.539	0.538	0.538	0.537
C_M	0.710	0.711	0.710	0.709	0.709
LCB (%)	-6.126	-6.170	-6.150	-6.130	-6.110
$\nabla(m^3)$	0.144	0.144	0.143	0.143	0.143
FN	201.940	6.658	6.628	6.623	6.618
CN	0.000	4.694	4.699	4.703	4.704

6. INVERSE FAIRING OF SHIP HULL FORMS

It is a well-known fact that fairness of curves and surfaces are closely related to their curvature characteristics, and they provide a powerful means for curve and surface interrogation. Thus, both the interactive and mathematical fairing procedures are largely concerned with improvement and control of curvature. There are specific requirements for a given ship line to be considered as fair, as mentioned in Chapter 4 (e.g., distribution of curvature on fair curve must be as uniform as possible, devoid of unintentional knuckles indicated by multiple sign changes in the curvature plot etc.). Whenever, all these requirements are fulfilled the generated curve or surface can be considered as fair. More recently, several algorithms have been developed based on modifications of curvature. Some worth noting can be listed as knot removal, knot insertion, and energy minimisation procedures. The knot removal and insertion algorithms are mainly based on local modifications of the curvature, and used in the development of local fairing algorithms. The essence of the method is removing offending knots by means of a curvature related fairness criterion and reinserting new ones. Details of these techniques can be found in **Farin et al (1987)**. However, the fairness is a global feature of curves, and eliminating erroneous points by local fairing methods does not necessarily produce acceptable results as local modifications of curvature do not affect curve's fairness characteristics. It is evident that erroneous points inherent in the given data indicated by multiple sign changes in the curvature curve should be eliminated, but a global fairing method is extensively required to eliminate randomly distributed relatively small erroneous points which do not indicated by sign changes in the curvature plot. Energy minimisation or variational methods are generally accepted as the most effective fairing methods. Some of the researchers in this field can be listed as **Moreton (1992)**, **Welch and Witkin (1992)**, **Welch (1995)** and **Wesselink (1996)**. Applications of these techniques for hull surface design will be addressed in Chapter 7.

Consequently, it is widely accepted that an efficient fairing methodology should improve curvature characteristics of curves (i.e., the maximum rate of change of curvature must be as small as possible). With these requirements in mind, a new global fairing algorithm is presented in this chapter. This approach has a simple base, but different in its manner from the previous forward fairing algorithm described in Chapter 5. The aim in this process is directed towards smoothing the curvature plot

of curves and surfaces. If a smooth curvature plot can be obtained, obviously its corresponding curve or surface will be smooth. The process starts with eliminating gross errors using smoothing process as described in Chapter 4, and continues with representing curvature plots with suitable mathematical curves and surfaces. Devoid of gross errors the curvature plot represented with approximating mathematical curve and surface methods is likely to produce fair curves and surfaces. The new improved curves/surfaces are obtained by an inverse procedure.

As the ship lines and surfaces are described by a set of discrete offset, divided differences can be used to represent derivatives, and thus curvature. The improved ship lines or the overall surface is obtained by integrating twice the improved curvature plot. However, the new curve/surface to be obtained should not be much deviated form the original. To accomplish both the fairness and closeness goals, various mathematical techniques have been tried and implemented in this process. It is found out that this closeness requirement can only be achieved by using B-spline curve and surface techniques of low order, and acceptable results are obtained. The detailed applications for typical ship sections and waterlines can be found in **Narlı (1995)**. A detailed description of B-spline techniques is given in Chapter 3.

Low order B-splines represent curvature curves with sufficient accuracy, and as the B-spline curvature plot is not constrained to pass through original values but closely mimic the overall curvature curve, a fair curvature plot can thus be obtained. This fairing process is first applied to typical ship curves and taken into account as a two-dimensional problem. The application of the process for three-dimensional problems is also considered. Distorted mathematical Wigley form is used to present the effectiveness of the process.

6.1. Two-Dimensional Problem – Inverse Fairing of Ship Curves

The first step in the application of inverse fairing procedure is the fairing of two-dimensional curves. A two-dimensional hull form design curve is assumed to be defined by a set of offset points. Section lines and waterlines are typical hull form design curves.

In order to apply the inverse fairing procedure the curvature plot of the curve is required. As the hull form design curves are defined by a set of discrete offset points, divided differences can be used to represent derivatives. Thus, the curvature of the design curve may be represented by the second divided differences.

The first crucial step in the inverse fairing procedure is the representation of the curvature plot by a suitable mathematical expression. A wide range of mathematical tools including polynomials and various types of splines are available, and experience has shown that B-splines provide the best results. The order of the B-splines determines the degree of the fairness. As the order of the spline representation increases the degree of fairness of the curvature plot will also increase.

The second crucial step is the evaluation of the offset points from the modified curvature plot by double integration. Since the derivation is carried out in a numerical manner, the integration is also achieved by using numerical procedures.

The fairness of the curvature plot greatly depends on the order of the B-spline used to represent the curvature curve. Fairness of the curvature curve and hence of the original curve can be improved by increasing the order of B-spline. However, the deviation from the original offset points will also increase with increasing order. Therefore, the designer must seek a compromise solution.

A typical example of application of inverse fairing procedure for a tanker section is shown in **Figure 6.1**. In each figure the original section, its first and second derivatives, the B-spline representation of curvature curve, and the modified curve are shown. In each case a different order of B-spline is used, i.e., 3, 4, 5, and 6 corresponding to quadratic, cubic, quartic, and quintic B-splines. The results clearly indicate that fairness of the curve can be improved by increasing the order of the B-spline, however, the deviation from the original offsets will also increase. Experience has shown that cubic and quartic B-splines provide satisfactory results in many cases.

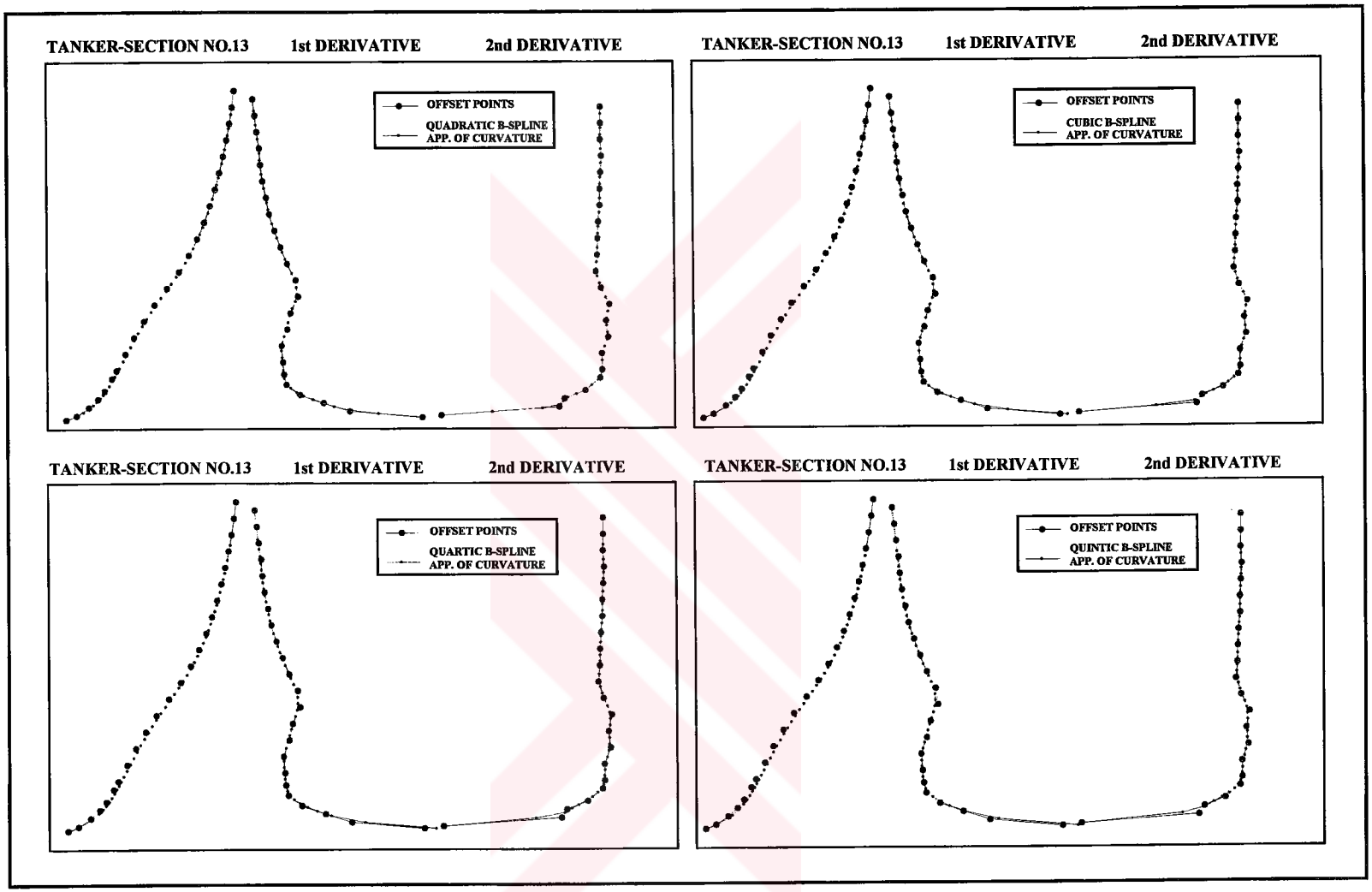


Figure 6.1. B-spline approximations of curvature for a typical tanker section for varying degrees.

6.2. Three-Dimensional Problem – Inverse Fairing of Ship Surfaces

The success of the inverse fairing procedure for two-dimensional curves have motivated the generalisation of the process for three-dimensional ship hull forms. In three-dimensional case the hull form is assumed to be defined by a number of offset points along the ship length, and in the vertical direction. It is also assumed that the curvature of the surface can be represented by second derivatives along the length and depth directions. The second derivatives in both directions were predicted by using suitable finite difference schemes.

The second derivatives, which are assumed to represent the curvature surface greatly, exaggerate the wrinkles and deformations on the hull surface. A typical example is shown in **Figure 6.2** where the second derivatives in length (u), and depth (w) directions are illustrated for a Wigley parabolic form, which was randomly distorted in order to be used as a test case in this thesis. As can be seen in the figure, the second derivatives in the (w) direction is far more obvious compared with the hull surface.

In a fairing problem second derivatives in u and w directions may not be sufficient on their own. More suitable measures to indicate the fairness of a surface are the mean and Gaussian curvatures, which are described as

$$\kappa_{mean} = \frac{1}{2}(\kappa_1 + \kappa_2)$$

$$\kappa_{gaussian} = \kappa_1 \kappa_2$$

where κ_1, κ_2 corresponds to principal curvatures.

The application of these measures to the distorted Wigley form is illustrated in **Figure 6.3.** These colour-coded maps clearly indicate the poor fairness characteristics of this particular form.

The principal of inverse fairing procedure is to represent the curvature plot by a suitable B-spline curve (or surface) and obtain the original offsets by double integration from the modified curvature coordinates. For a surface this process must be carried out in u and w parametric directions in a simultaneous manner.

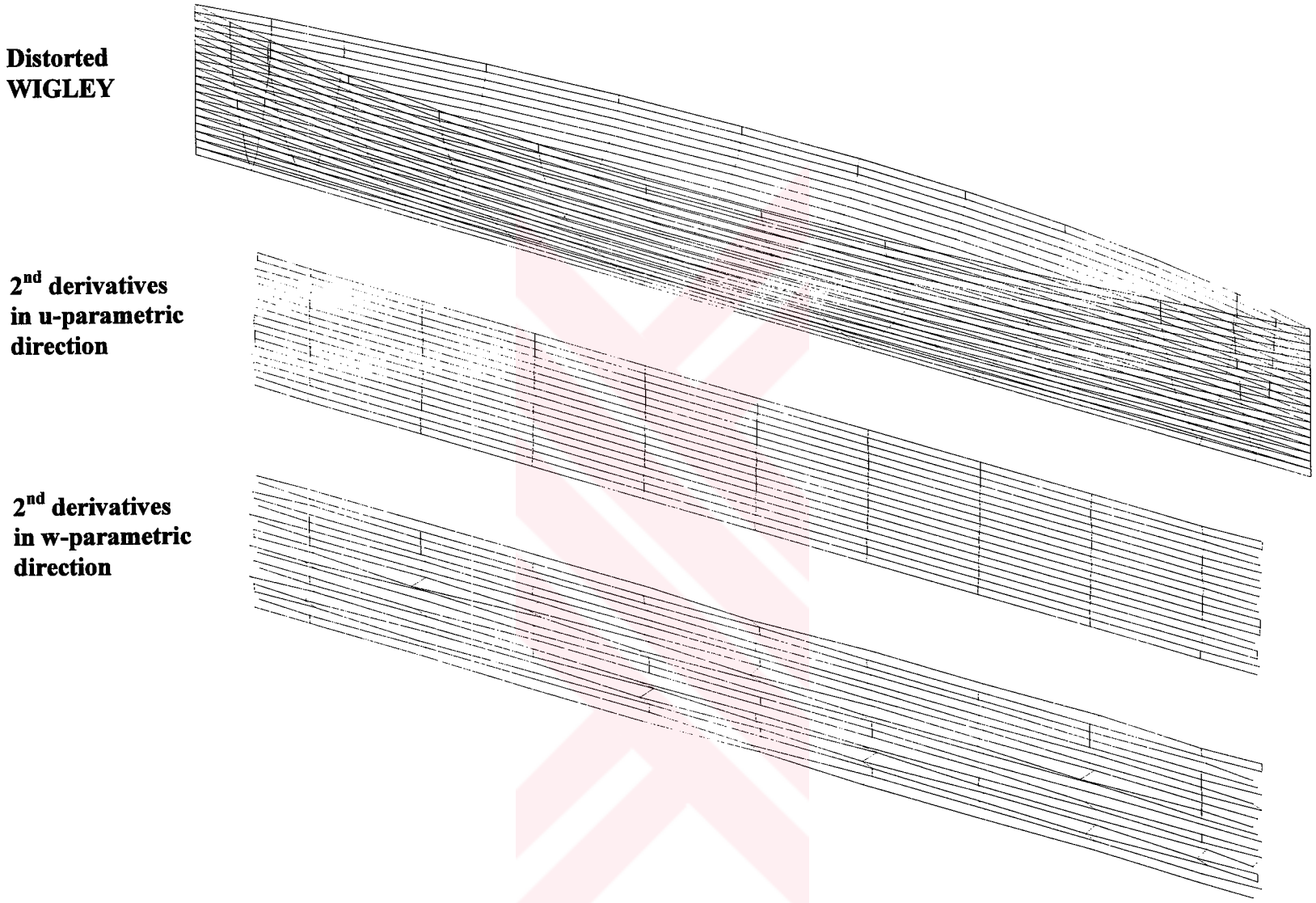


Figure 6.2. Distorted Wigley hull form and parametric second derivatives.

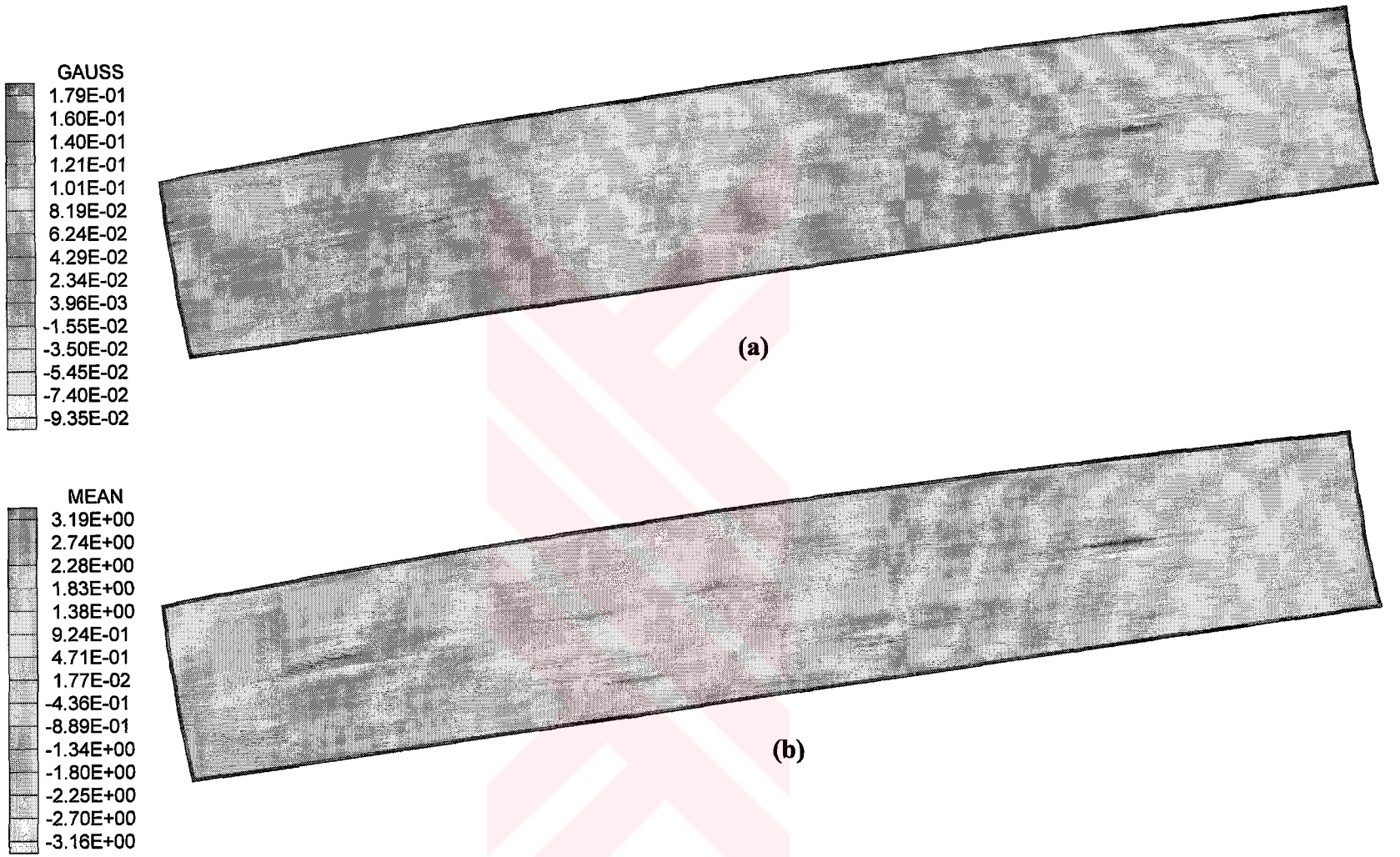


Figure 6.3. (a) Gaussian curvature map and **(b)** Mean curvature map of distorted Wigley hull form.

The selection of the suitable order of B-spline surface is again crucial and a compromise between the fairness and closeness characteristics must be achieved. Applications to several ship forms indicates that the best results can be obtained by using cubic and quartic B-spline surfaces to represent the curvature plot of the hull surface. It is also possible to select a B-spline surface with different orders in u and w parametric directions.

The results of the application of the inverse fairing procedure to the distorted Wigley form are shown in **Figures 6.4**, and **6.5. Figure 6.4** illustrates the second derivatives for the modified Wigley form which was obtained by representing the curvature plot of the distorted form by a cubic B-spline surface. **Figure 6.5** illustrates the mean and Gaussian curvature plots for the modified form which has much more improved fairness characteristics compared with the distorted form.

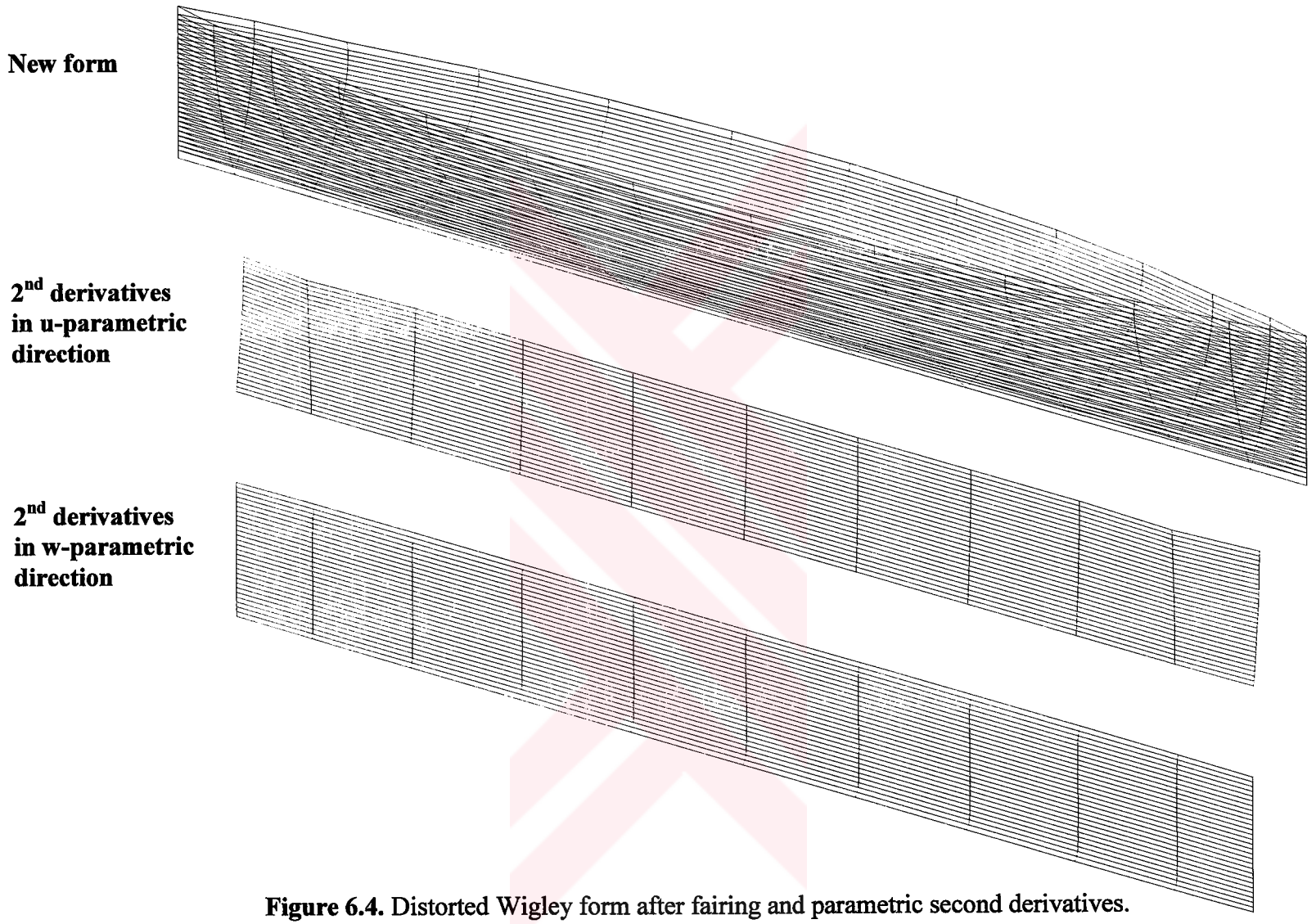


Figure 6.4. Distorted Wigley form after fairing and parametric second derivatives.

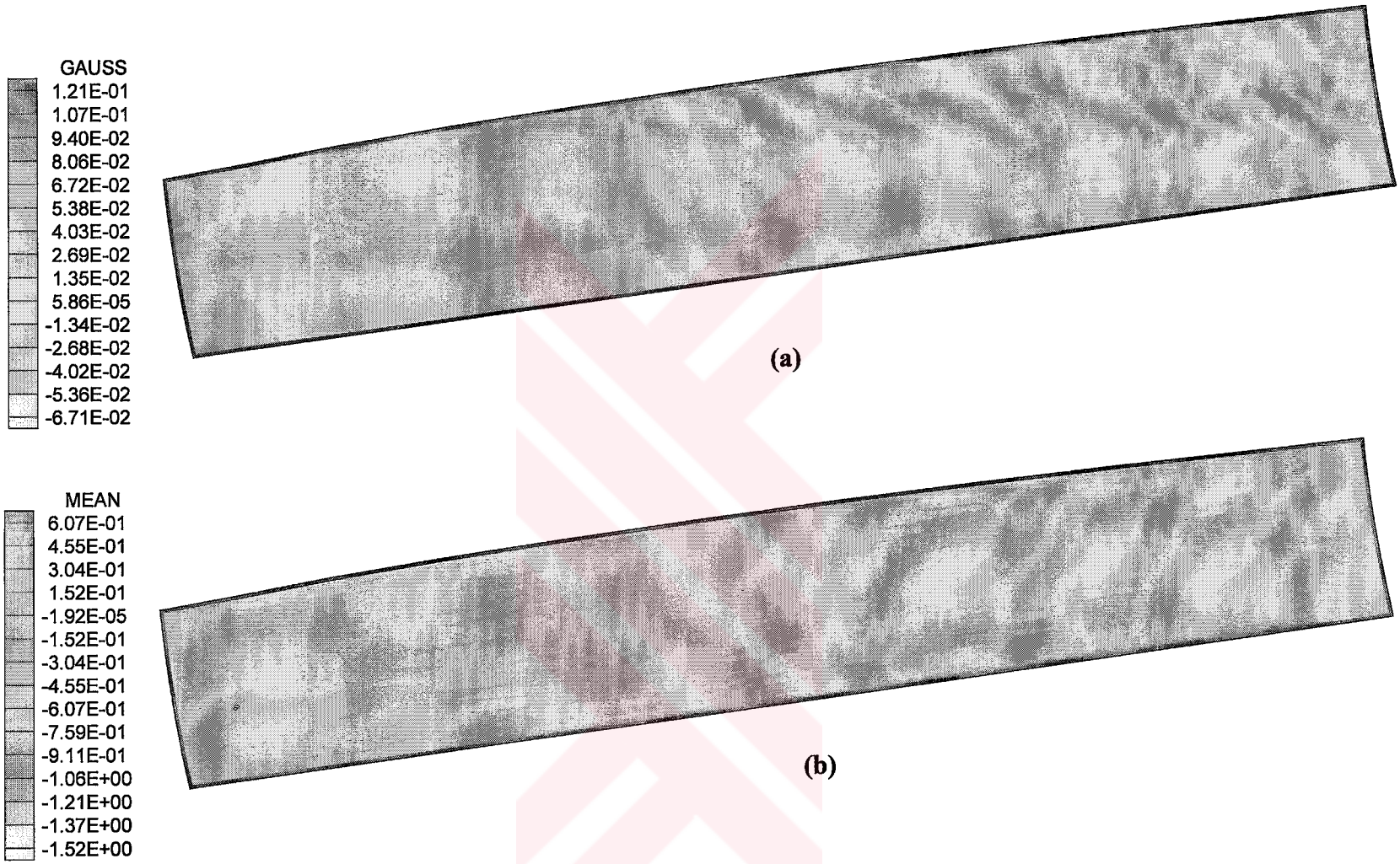


Figure 6.5. (a) Gaussian curvature map and **(b)** Mean curvature map of distorted Wigley form after inverse fairing.

7. AN OPTIMISATION APPROACH FOR FAIRING OF HULL FORMS

Hull form design process is a structured decision process from a set of requirements to a solution meeting technical and economic goals, therefore the problem can be regarded as a multi-criteria optimisation problem. Hull form fairing as a part of this design process can also be formulated in terms of an optimisation problem. A typical engineering optimisation problem is characterised by the following elements:

- Design variables
- Measure of merit
- Constraints

In the fairing problem of ship hulls, the free form parameters of the hull shape can be accepted as design variables, and the fairness measure can be chosen as measure of merit and geometric conditions, hydrodynamic properties may act as constraints. The selection of objective function and constraints is the vital part of an optimisation problem. For the problem of fairing ship hull forms many different types of objective functions and constraints can be used. In the following sections an optimisation problem is formulated and solved for fairing of ship hull forms.

7.1. Formulation of the Optimisation Problem

Formulation of an optimisation problem involves transcribing a verbal statement of the problem into a well-defined mathematical statement. This standard mathematical statement should be in the following general form

Minimise $f(x) = f(x_1, x_2, \dots, x_n)$

where $\mathbf{x}(x_1,x_2,\dots,x_n)$ represent the design variables.

Subject to $g_j(x) > 0 \quad j = 1, \dots, n$

Note that an equality constraint, such as A.x =B, can be converted into two inequality constraints as A.x>B and A.x<B.

Constraints define the boundaries of the feasible design space. In general two types of constraints are used in engineering problems.

- Constraints on design variables
  - Direct limitations on design variables (side constraints)

$$x_i^L \le x_i \le x_i^U \qquad \text{where} \qquad i = 1,2,...,n$$

  - Relations between design variables (variable linking)
- Constraints on system behaviour
  - Limits on system output
  - Physical laws governing the system

In a fairing problem, f(x) is a fairness measure which may have different forms as discussed in the following section. Both geometrical constraints such as position, tangent, curvature, etc. and practical design constraints such as area and volume are included in $g_i(x)$. For the purpose of the numerical treatment of these constraints, an internal penalty function technique can be applied which transforms the problem into a unconstrained optimisation problem where the objective function is

$$F(x, r_k) = f(x) + r_k \sum_{i=1}^{N} \frac{1}{g_i(x)}, \qquad r_k > 0$$

The optimisation problem as outlined is amenable to solution by non-linear programming techniques. Basically, the above function is taken as an object function with related geometric and functional constraints in the following sections. The **Hooke and Jeeves (1961)** direct search method, which is described in detail in the following section has been found to work well for the problem under discussion.

The structure of the optimisation based fairing procedure is illustrated in **Figure 7.1.**

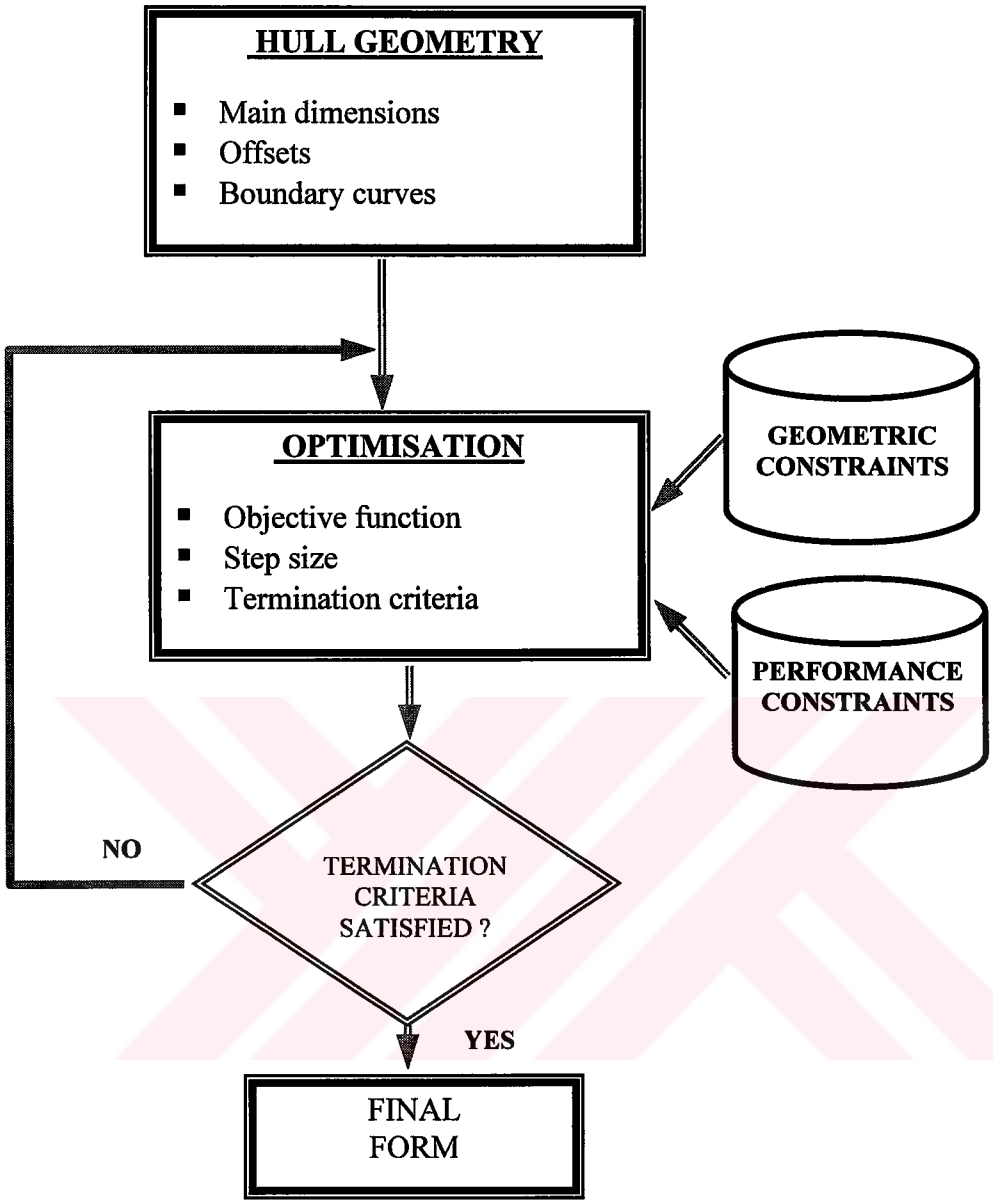


Figure 7.1. Structure of the optimisation based hull form fairing procedure.

7.1.1. Objective Function

A formal optimisation problem will require the definition of an objective function to be minimised (or maximised) which is a function of some geometric design variables, and related geometric constraints. In a fairing problem the objective function is related to the fairness of the curve or surface to be optimised. Different approaches may be adopted to represent the fairness of a curve or surface.

Fairness is a necessarily fuzzy notion based on our perception of curve or surface quality: but it is intimately related to the distribution of curvature over the form. Some of the basic properties of a mathematically fair curve or surface are

- C^2 curvature continuity
- Curvature almost piecewise linear with as few spans as possible

The curvature of a surface at a particular point in a particular tangent direction can be measured by slicing the surface with a perpendicular cutting plane and measuring the curve of intersection. The result is called the *normal section curvature* in the given direction. One of the first results in the classical differential geometry of surfaces, due to Gauss, is that the sectional curvature in the neighbourhood of a surface point is a smooth function of tangent direction and takes on its maximum and minimum values in orthogonal tangent directions. These values are referred to as the principal curvatures κ_1 and κ_2. The principal curvatures completely characterise the shape of the surface about a point, and in turn give rise to important geometric quantities, such as Gaussian curvature

$$\kappa_1\kappa_2$$

and mean curvature

$$\frac{1}{2}(\kappa_1 + \kappa_2)$$

The simplest measure of fairness of a curve is its elastic bending energy, i.e., the integral of the squared curvature with respect to arc length

$$E = \int \kappa(s)^2 \, ds = \int (c_{ss})^2 \, ds$$

where the subscript indicates differentiation with respect to the arc-length parameter s, and the squared vector in the integrand is shorthand for the dot-product of the vector with itself .

The elastic curve functional can be generalised to surfaces using principal curvatures as follows:

$$E = \int_S (\kappa_1^2 + \kappa_2^2) \, dS$$

where dS is the differential area form. This is commonly called the thin plate functional, because it approximates the strain energy of a thin elastic plate.

Moreton (1992) pointed out that minimising bending energy alone tends to concentrate curvature near the endpoints of a faired region. He uses a curve fairness functional that measures the variation of curvature over the interval, i.e.,

$$\int \left(\frac{d\kappa}{ds} \right)^2 ds$$

Minimising this functional is reported to yield curves that seek constant curvature with results that look much more *draftsman-like* than those arising from strain energy. This functional can be generalised to surfaces as follows

$$\int \left(\frac{\partial \kappa_1}{\partial e_1} \right)^2 + \left(\frac{\partial \kappa_2}{\partial e_2} \right)^2 dS$$

where e_1, e_2 are the corresponding directions of principal curvature.

7.1.2. Constraints

In a fairing problem it is necessary to apply at least some geometric constraints to the form, since the energy of an unconstrained surface can be made zero by collapsing the surface to a single point. In general two types of constraints are used in the formulation described in this section. The geometric constraints are used to control over some aspect of the form. The geometric constraints for the problem can be stated as,

- the location of points (positional constraints) which restrict the location of points on the curve or surface.
- direction of tangents (directional constraints)
- continuity constraints which require the curvature be described in a point of the curve.
- distance constraints can be used to control the deviation between the original and the modified forms
- area and volume constraints are used to control the surface are or internal volume

The second type of constraints are called the performance constraints and used to preserve or improve the performance characteristics of the original form. These performance characteristics may include but not limited to powering, intact and damaged stability, and seakeeping.

7.2. Method of Solution

Optimisation problems, in general require the best solution to some underconstrained problem. Usually, the problem contains a single function of a set of free variables that returns one real number for any position in its domain. This function is called the objective function. The solution is defined as either the minimum or maximum value possible of the objective function depending on the problem. Constraints are requirements on the values of the variables in the objective function and reduce the size of the domain.

One of the simplest forms of constrained optimisation, in terms of formulations of the problem, is linear programming. The objective function and the constraints are all linear. The space of interest is limited to some convex region of the objective function's domain. A commonly used solution method for linear programming is the *simplex method.*

If the objective function is non-linear, further non-linear mathematical programming techniques, i.e., searching should be used. Search techniques of these methods are generally performed by computing gradients of the objective function and using this information for search direction. In non-linear programming, the solution to be global minimum, a good estimation of initial search point and bounded constraints on variables are required. Also, the convexity of the domain is not assured, therefore solving techniques should avoid of bound areas rather than following edges as in linear programming. The simplest of the methods is *gradient descent*, just follows the gradient of the objective function until it intersects a local minimum or the edge of the domain. However, the method is not very efficient. *Conjugate gradient* method use more complicated functions of the objective function's gradient to determine search direction and hence less susceptible to getting involved in undesired local minima.

7.2.1. Direct Search Method of Hooke and Jeeves

Hooke and Jeeves (1961) method is one of the most widely used direct search methods. It attempts in a simple though ingenious way to find the most profitable search directions. The method is developed for non-constrained problems and based on two types of step-by-step searches alternating in turn, a *local search*, which is an unidirectional variation of each design variable resulting in the direction of steepest descent, and a *pattern move* which represents a rotation of the search direction which accelerates the search by the aid of increasing the step widths.

The search routine to minimise the object function can only proceed in a feasible space as outside that space the penalty functions are not defined.

If we consider the problem of minimising $f(x_1,x_2,\ldots,x_n)$, the general procedure, which is shown in **Figure 7.2**, can be described as follows:

- Start with an arbitrarily chosen initial base point $(b_1,b_2,\ldots,b_n)$ and step lengths $(h_1,h_2,\ldots,h_n)$ for the respective variables $(x_1,x_2,\ldots,x_n)$.
- The method proceeds by a sequence of exploratory and pattern moves. The procedure for an exploratory move about the point $(b_1,b_2,\ldots,b_n)$ is as follows:
- Evaluate $f(b_i+h_i)$. If the move from b_i to b_i+h_i is a success, replace the base point b_i by b_i+h_i. If it is a failure, evaluate $f(b_i-h_i)$. If this move is a success, replace b_i by b_i-h_i. If it is another failure, retain the original base point b_i.
- Repeat the above procedure for each variable in turn finally arriving at a new base point after (2n+1) function evaluations at most.
- If $b_{i+1}=b_i$, halve each of the step lengths h_i and return to first step. The calculations terminate when the step lengths have been reduced to some prescribed level. If $b_{i+1}\neq b_i$, make a pattern move from b_{i+1}.
- A pattern move attempts to speed up the search by using information already acquired about $f(x_1,x_2,\ldots,x_n)$. It is invariably followed by a sequence of exploratory moves, with a view to finding an improved direction of search in which to make another move. The procedure for a pattern move from b_{i+1} is as follows:

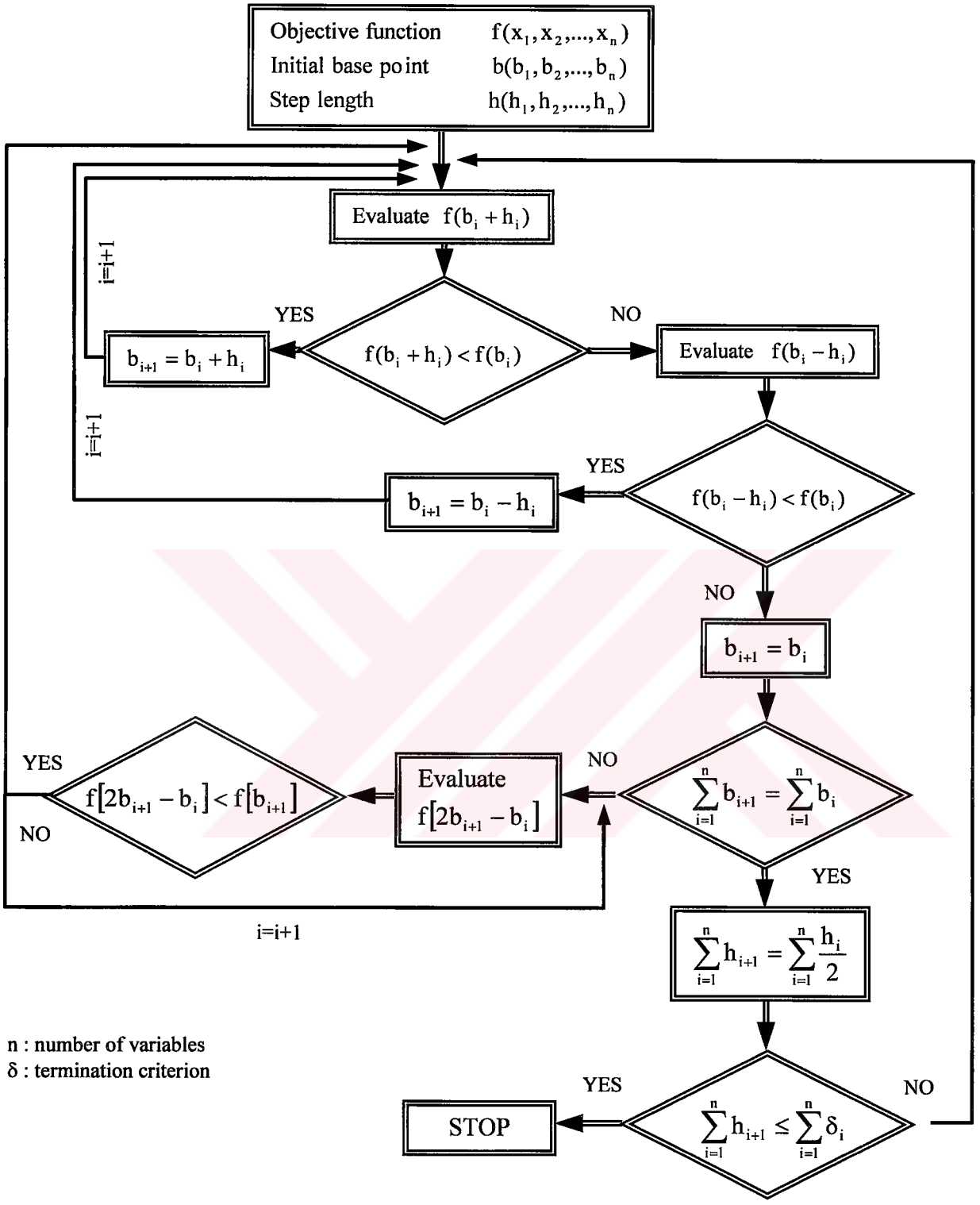


Figure 7.2. Hooke and Jeeves direct search algorithm.

- It seems sensible to move from b_{i+1} in the direction $(b_{i+1}-b_i)$, since a move in this direction has already led to a decrease in the value of f $(x_1,x_2,\ldots,x_n)$. Therefore, move from b_{i+1} to $(2b_{i+1}-b_i)$ and continue with a new sequence of exploratory moves about $(2b_{i+1}-b_i)$.

- If the lowest function value obtained during the pattern and exploratory moves of $(2b_{i+1}-b_i)$ is less than b_{i+1}, then a new base point b_{i+2} has been reached. In this case, return to $(2b_{i+1}-b_i)$ with all suffices increased by unity. Otherwise abandon the pattern move from b_{i+1} and continue with a new sequence of exploratory moves about b_{i+1}

7.3. Two Dimensional Optimisation; Curves

Before involving in 3D shape optimisation problems, consideration of two-dimensional surface entities i.e., curves will be rather beneficial for the realisation of the underlying theory. Hence, the counterpart of fairing objective for curves can be expressed as

$$E = \int \kappa^2 ds$$

which corresponds to the elastic bending energy of a physical spline. The constraints of the problem may be specified as

- positional constraints can be used to control the position of any point on the curve

- Distance constraint can be used to control the distance between the original points and the modified points on the curve

In many cases what the designer desires is a fair curve which is not much deviated from the original one. In such cases a modified objective function may be more useful. This objective function will include both the fairness and closeness terms and some weight functions for each term as follows

$$\omega_1 \int \kappa^2 ds + \omega_2 \int (y_i - y_j)^2 ds$$

where y_i and y_j represent the original and modified points respectively. ω_1 and ω_2 are the weighting functions, which are specified by the designer to determine the deviation from the original curve.

In order to illustrate the optimisation procedure some typical results are presented below. A quadratic function ($y = x^2$) was deliberately distorted in order to test the fairing procedure. The optimisation procedure is run for 11 cases in which the fairness weighting function decreases from 1.0 to 0.0, and the closeness weighting increases from 0.0 to 1.0 in regular steps of 0.1. These weighting functions indicate the relative importance of fairness and the closeness to the original curve. For example, when the weighting functions are selected as 0.5 and 0.5, both the fairness and closeness are taken as equally important. The fairness and closeness numbers, number of iterations and the value of objective function are given for each case in **Table 7.1.** Typical results are also illustrated in **Figure 7.3.**

Table 7.1. Variation of fairness and closeness functions for a distorted quadratic curve.

ω_1	ω_2	FN	CN	F	No. of Iterations
1.0	0.0	0.0001	117.1611	0.0001	1080
0.9	0.1	3.1596	38.0738	6.6510	1062
0.8	0.2	6.7657	16.2744	8.6674	772
0.7	0.3	9.4693	7.9058	9.0002	756
0.6	0.4	11.4540	4.1486	8.5319	635
0.5	0.5	12.9956	2.2397	7.6176	454
0.4	0.6	14.1842	1.2560	6.4273	284
0.3	0.7	15.2085	0.6960	5.0498	325
0.2	0.8	16.1752	0.3710	3.5319	268
0.1	0.9	17.3951	0.1611	1.8845	200
0.0	1.0	22.9610	0.0001	0.0001	61

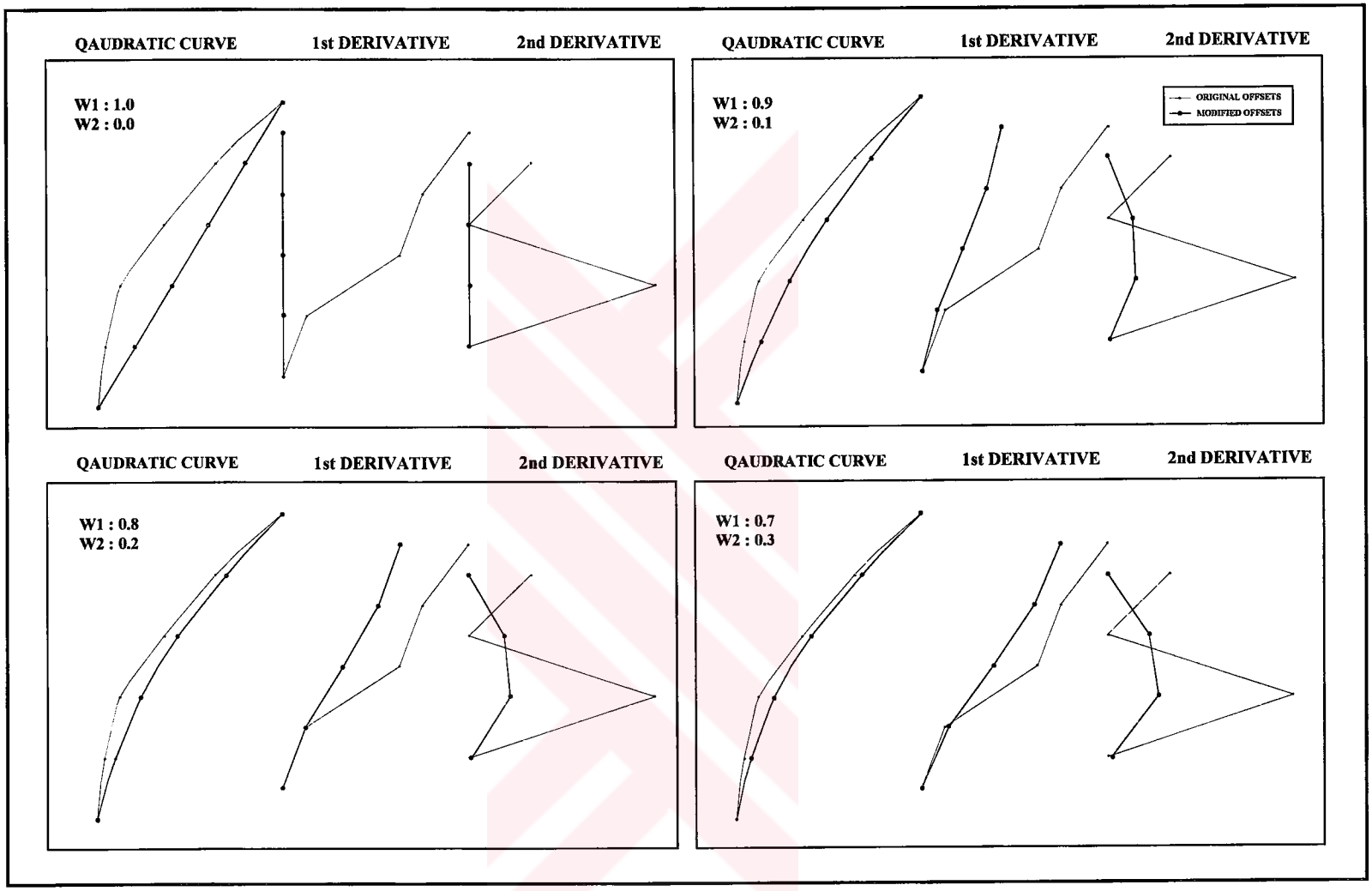


Figure 7.3.a. Effect of fairness and closeness objectives on the fairing of a distorted quadratic curve.

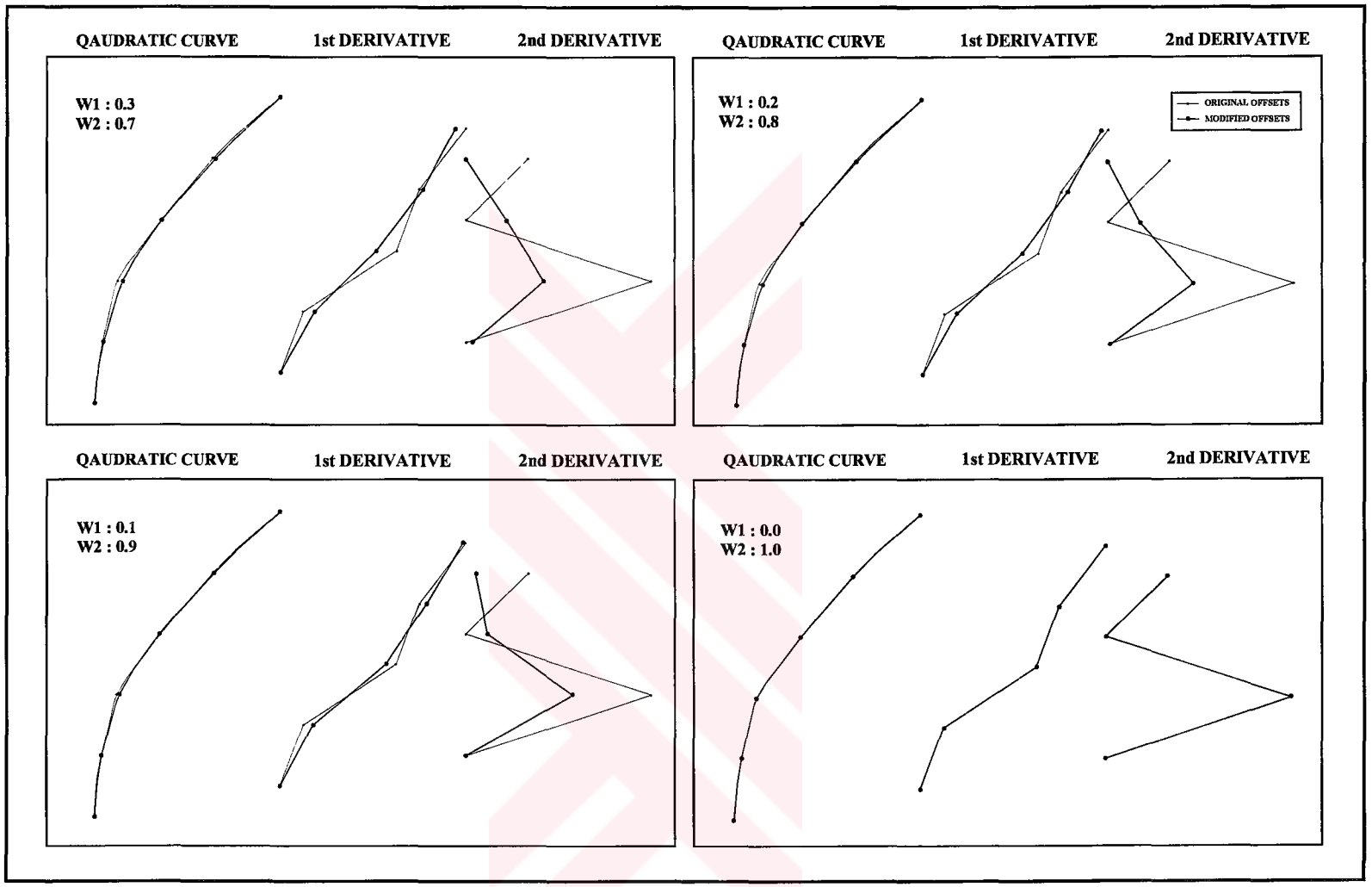


Figure 7.3.b. Effect of fairness and closeness objectives on the fairing of a distorted quadratic curve.

7.4. Hull Form Optimisation

This section presents a generalised optimisation formulation to obtain ship hull forms with desired fairing properties subject to specified geometric and performance constraints. The basic elements of any engineering optimisation formulation are the optimisation variables, objective function and the constraints of the problem. Selection of these elements for the current problem is described below.

The hull form is assumed to be defined by a hull surface equation in the form $y_{ij}=y(x_i,z_j)$, and these ordinates are taken directly as optimisation variables.

There are several fairness measures, which can be used as the objective function of the problem. Details of these measures are presented in Chapter 4. The following fairness measures are considered to be suitable for a hull form fairing procedure:

Energy minimisation:

$$\int(\kappa_1^2+\kappa_2^2)dS$$

Mean curvature:

$$\int\frac{1}{2}(\kappa_1+\kappa_2)dS$$

Gaussian curvature:

$$\int\kappa_1\kappa_2 dS$$

Variation of curvature:

$$\int\left(\frac{\partial\kappa_1}{\partial e_1}\right)^2+\left(\frac{\partial\kappa_2}{\partial e_2}\right)^2 dS$$

A measure of goodness, i.e., non-linear fairness metric, is formulated at each surface point, and integrated over the entire surface to get a single number which describes the desirability of the surface shape under that metric. The search for desirable shape continues until the form which optimises this quantity while satisfying the geometric constraints specified for this case is achieved.

The problem must be geometrically constrained in order to produce realistic hull forms. Typical geometric constraints may be as follows:

- All offsets are positive and less or equal to maximum beam

$$0 \le y_{ij} \le y_{max}$$

- Known offsets as lower bound of the hull

$$y_{ij} \ge y_{ij}^{0}$$

- Known offsets as upper bound of the hull

$$y_{ij} \le y_{ij}^{0}$$

A typical application of the optimisation procedure is presented for a distorted parabolic form (Wigley form). The distorted original form and its curvature characteristics are shown in **Figure 7.4**. The objective function of the problem is selected as the minimisation of the elastic bending energy function which is defined in terms of curvatures as follows

$$\text{Minimise} \qquad \int(\kappa_1^2 + \kappa_2^2)dS$$

The energy of an unconstrained surface can be made zero by collapsing the surface to a single point. Therefore, the fairness objective must be modified by a closeness requirement which is described as follows

$$\int(y_i - y_j)^2 dS$$

where y_i and y_j represent the original and modified offset points. Then the modified objective function becomes

$$\omega_1 \int(\kappa_1^2 + \kappa_2^2)dS + \omega_2 \int(y_i - y_j)^2 dS$$

ω_1 and ω_2 are the weighting functions applied to emphasise fairness or closeness. In the typical example presented here both weighting functions are taken as 0.5.

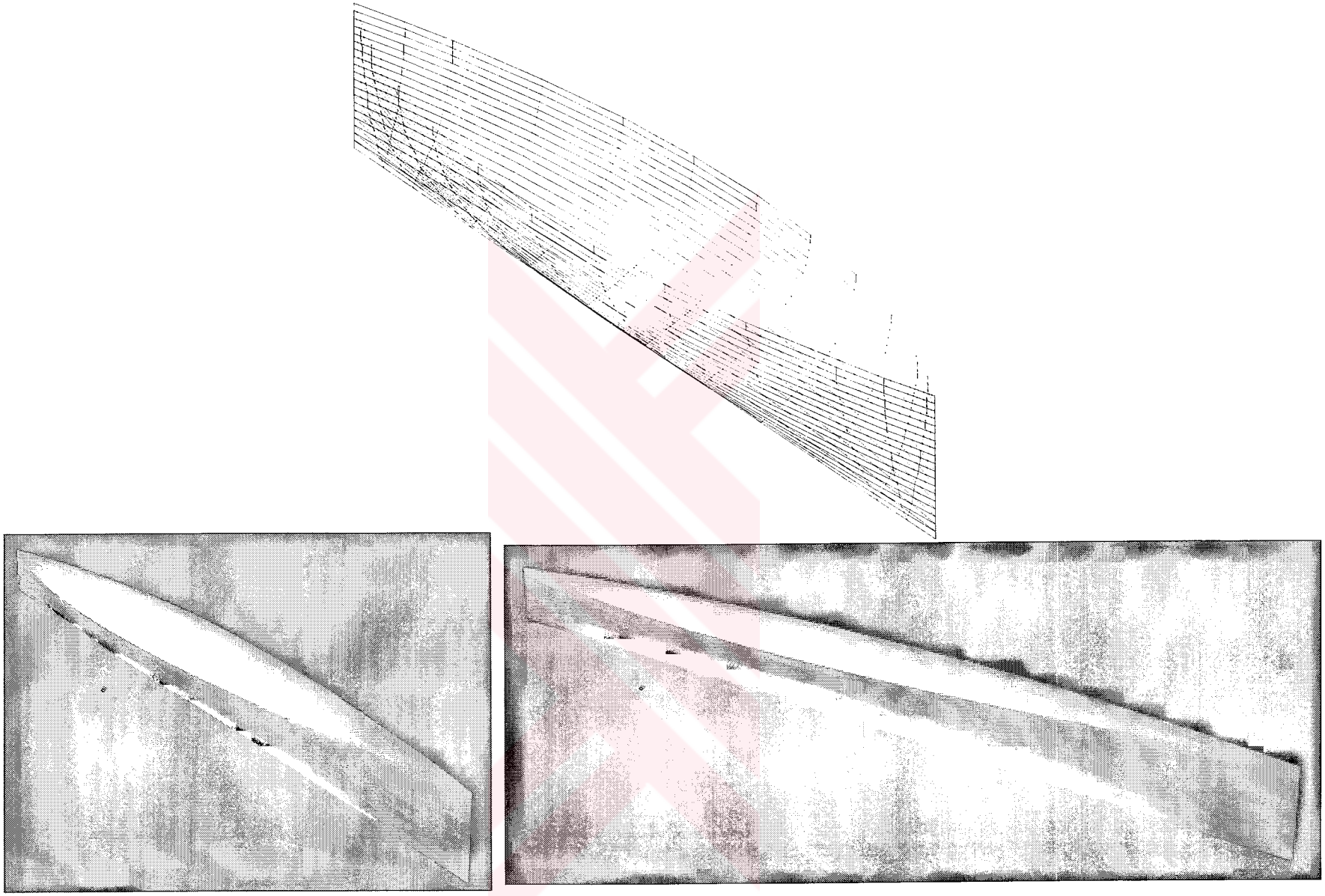


Figure 7.4.a. Original distorted Wigley form.

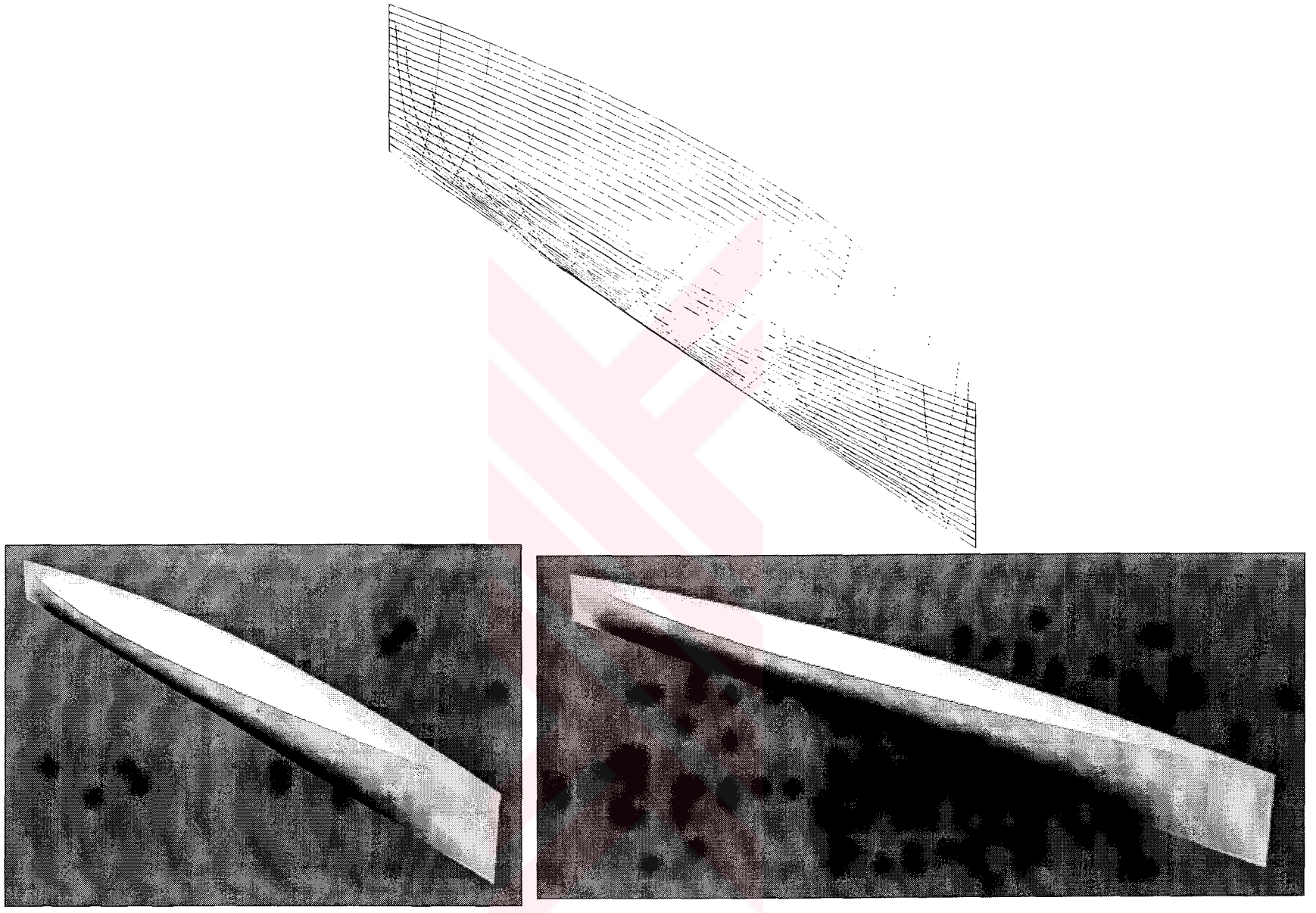


Figure 7.4.b. Optimised distorted Wigley form.

In a fairing process the modified form must be as close to the original form in order not to degrade performance characteristics of the original vessel. This can be achieved by using various geometric constraints such as closeness functions. Alternatively the designer can specify the performance requirements as a constraint of the problem. For instance the powering performance of the final form may be constrained to be not worse than that of the original form. This will require the availability of reliable numerical performance prediction tools. Once these tools are available, they can easily be implemented into the optimisation procedure.

8. APPLICATION AND COMPARISON OF DEVELOPED FAIRING PROCEDURES

In this thesis three novel computer aided fairing procedures, which make heavy use of B-splines and optimisation methods are developed and described. To evaluate the efficiency and flexibility of these tools, the procedures are applied to actual ship hull form fairing problems as presented in Chapters 5, 6, 7. However, it appears that a more realistic comparative study would require the application of all procedures on the same ship hull. This would be useful for the assessment of the relative efficiency of the methodologies. In this prospect, a high-speed displacement form is selected and fairing procedures are applied for this hull form. The lines plan of the initial hull form to be faired can be seen in **Figure 8.1.** The numerical data for the hull form is given in Appendix B. The selected form is a result of a preliminary design study and hence lacks the fairing qualities required for further detailed design calculations. Therefore, the procedures developed here, namely the forward fairing process, the inverse fairing process and the optimisation-based approach have been applied to obtain a hull form with improved fairing properties. The results of these applications are presented in the following sections.

8.1. Application of Forward Fairing Procedure for the Test Case

The forward fairing procedures are based on an iterative approach in which ship lines are represented by B-splines of suitable order. The forward fairing can be achieved either by B-spline approximation or B-spline fitting. Both procedures are applied for the current problem to produce final forms with improved fairing characteristics after a few iterations. It can be seen from **Table 8.1** that smaller fairness numbers can be obtained by B-spline approximation, and the deviation problem is overcome by selecting relatively low number of iterations, also low orders should be preferred in order to minimise deviation from the original offsets. In this case, after three iterations a final form as shown in **Figure 8.2** is obtained.

As can be seen from **Figure 8.2**, the final form has a smaller fairness number, which indicates fairer lines as can be observed visually. However, there is a penalty involved in the process, which is characterised by the closeness number, i.e., the

maximum deviation between the original and faired offset points. As can be seen, this number increases with increasing fairness. In this case the process is terminated after 3 iterations in order to stay within specified deviation limits. Geometric properties of the original and final forms with those generated during the iterative process of fairing are also presented in **Table 8.1**. For the cases where a higher degree of fairness is required the final form may not satisfy the closeness requirement and a geometric variation may be applied to achieve the original form coefficients.

Table 8.1. Forward fairing of the high-speed hull form using B-spline approximation.

	PARENT FORM	ITERATION NO. 1	ITERATION NO. 2	ITERATION NO. 3	FINAL FORM
L (m)	42.000	42.000	42.000	42.000	42.000
B (m)	7.000	7.000	7.000	7.000	7.000
T (m)	2.300	2.300	2.300	2.300	2.300
C_{WP}	0.7844	0.7801	0.7792	0.7784	0.7763
C_B	0.3602	0.3572	0.3538	0.3509	0.3477
C_M	0.4852	0.4822	0.4805	0.4783	0.4766
LCB (%)	4.9015	0.4792	0.4519	0.4398	4.2469
$\nabla(m^3)$	239.03	236.81	233.77	231.05	229.06
FN	1.087	0.985	0.762	0.448	0.307
CN	0.000	2.851	3.972	5.438	6.856

8.2. Application of Inverse Fairing Procedure for the Test Case

This process is based on the assumption that the fairness characteristics of a curve or surface is indicated by its curvature plot, hence the original curve or surface can be modified by representing its curvature plot by B-splines of suitable order. The original offset points are obtained from the curvature plot in an inverse manner. The order of the B-spline representation determines the degree of fairness. In general a compromise solution must be sought in order to preserve the original form characteristics. The inverse fairing procedures have successfully been applied to realistic ship forms as well as a mathematical form, in the previous chapters. The experience has shown that cubic and quartic B-splines provide the best results.

The inverse fairing process is applied to the fairing of the high-speed form which is accepted as the test case of this study. The second derivatives, which are assumed to represent the curvature surface greatly, exaggerate the wrinkles and deformations on

FN=1.087

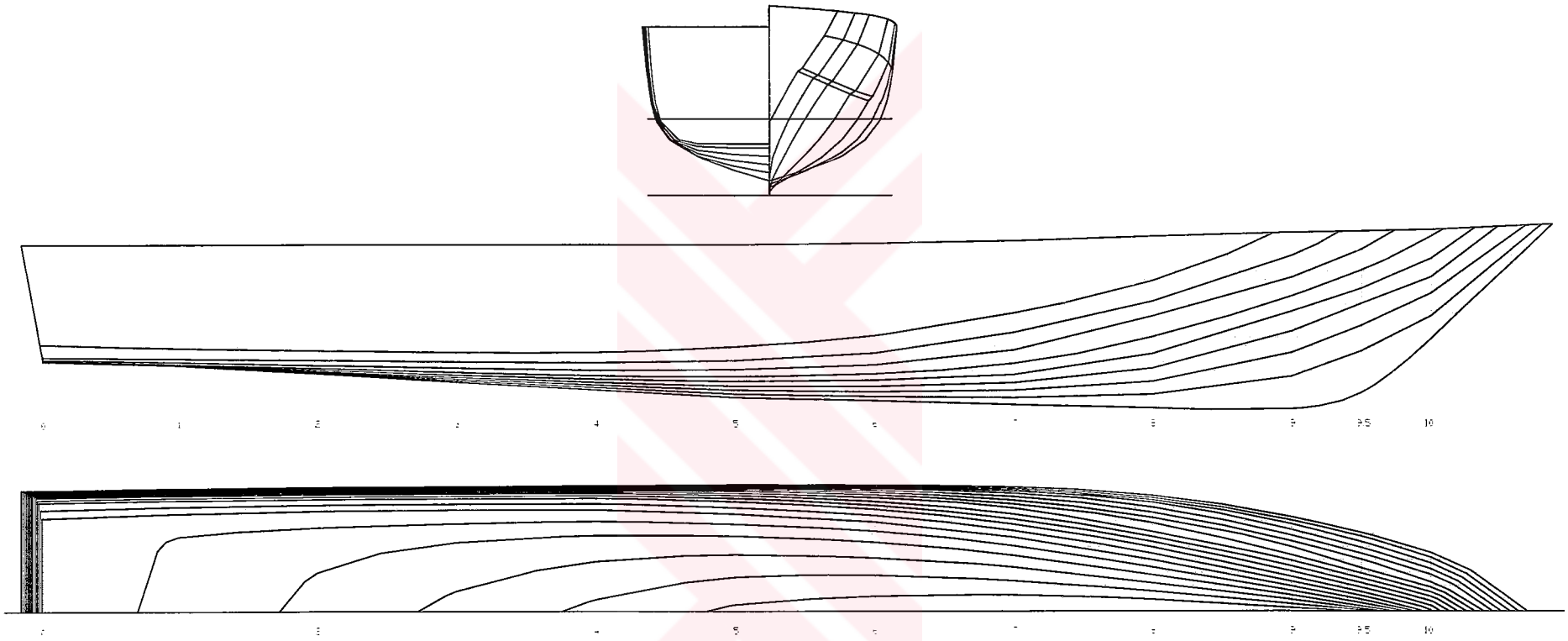


Figure 8.1. Parent high-speed hull form.

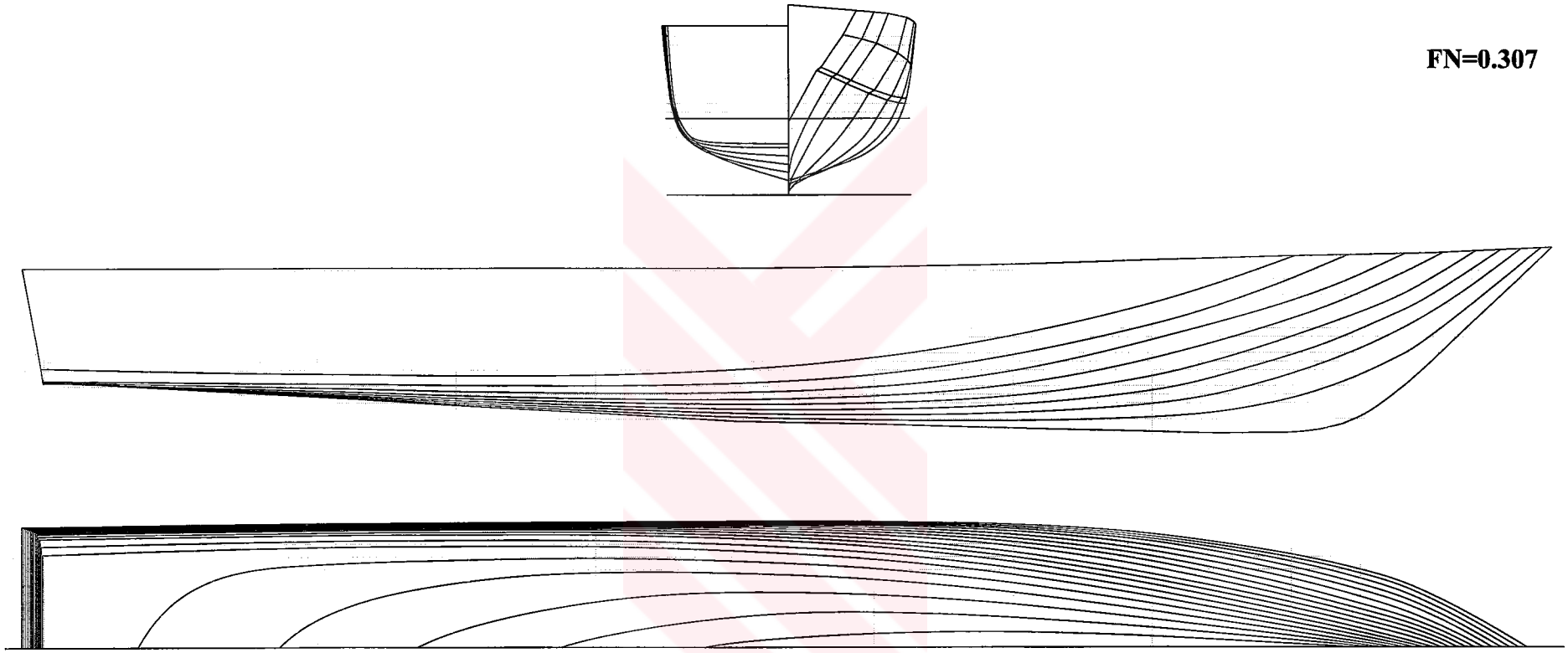


Figure 8.2. Final high-speed hull form.

the hull surface. This is shown in **Figure 8.3** where the second derivatives in length (u), and depth (w) directions are illustrated for the high-speed hull. As can be seen in the figure, the second derivatives in the (w) direction is far more obvious indicator of fairness compared with the hull surface.

The principal of inverse fairing procedure is to represent the curvature plot by a suitable B-spline curve (or surface) and obtain the original offsets by double integration from the modified curvature coordinates. For a surface this process must be carried out in u and w parametric directions in a simultaneous manner.

The selection of the suitable order of B-spline surface is again crucial and a compromise between the fairness and closeness characteristics must be achieved. For the test case, the inverse fairing is achieved by using cubic B-splines in both u and w directions, however it is also possible to select a B-spline surface with different orders in u and w parametric directions.

The results of the application of the inverse fairing procedure to the high-speed form are shown in **Figures 8.4.** These figures indicate that the curvature plot has a smooth surface resulting in a form with better fairing characteristics.

8.3. Application of the Optimisation Approach for the Test Case

In Chapter 7, the fairing of ship hull forms problem is formulated as a non-linear optimisation problem in which the objective function is a fairness functional to be optimised subject to appropriate geometric and performance constraints. Various fairness functionals are available including energy minimising, mean curvature, Gaussian curvature and the variation of curvature.

The application of the optimisation procedure is presented for the high speed form. The initial form and its shaded image are shown in **Figure 8.5**. The objective function of the problem is selected as the minimisation of the elastic bending energy function which is defined in terms of curvatures and a closeness requirement which is described as follows:

$$\omega_1 \int(\kappa_1^2 + \kappa_2^2)dS + \omega_2 \int(y_i - y_j)^2 dS$$

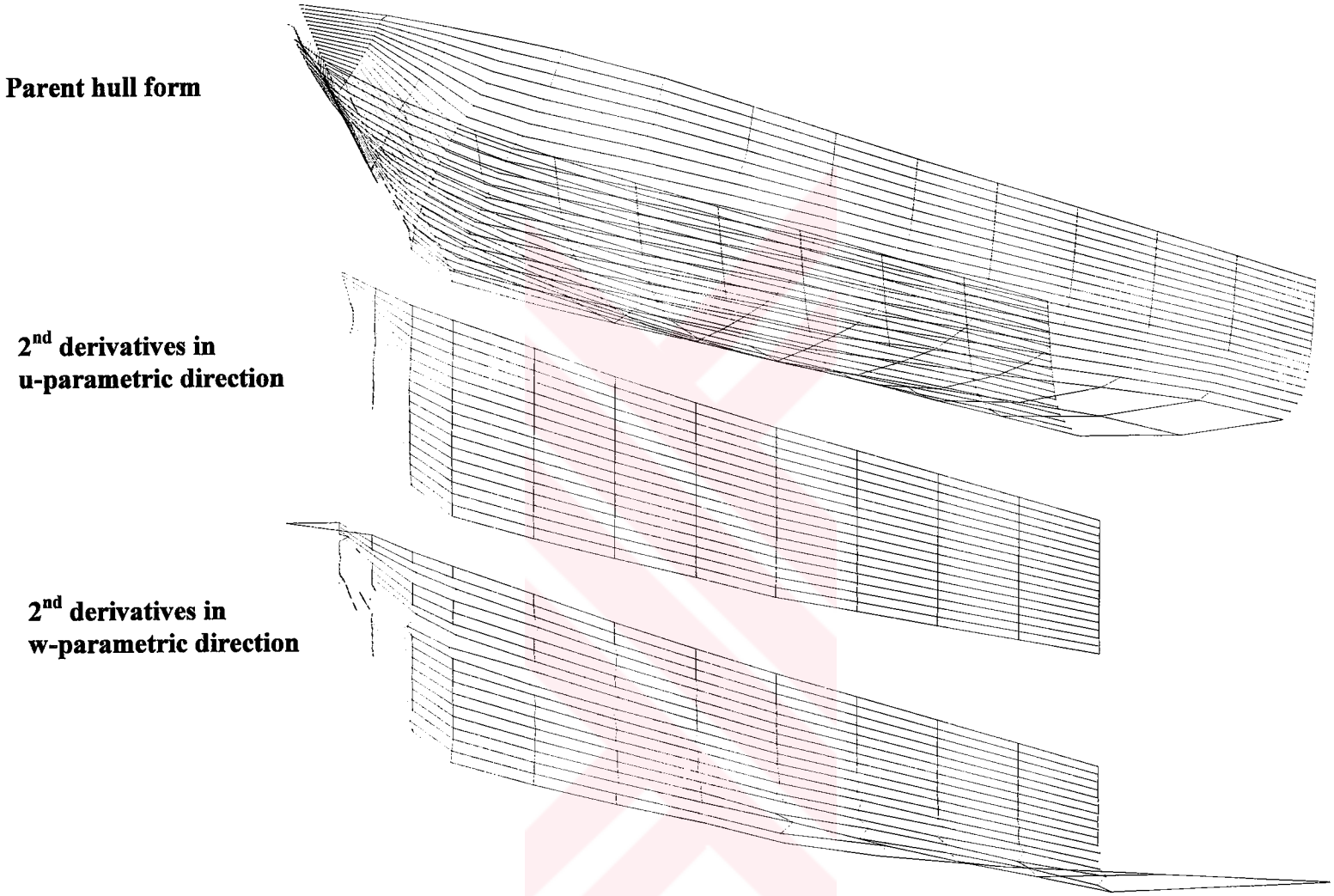


Figure 8.3. High-speed hull form and parametric second derivatives.

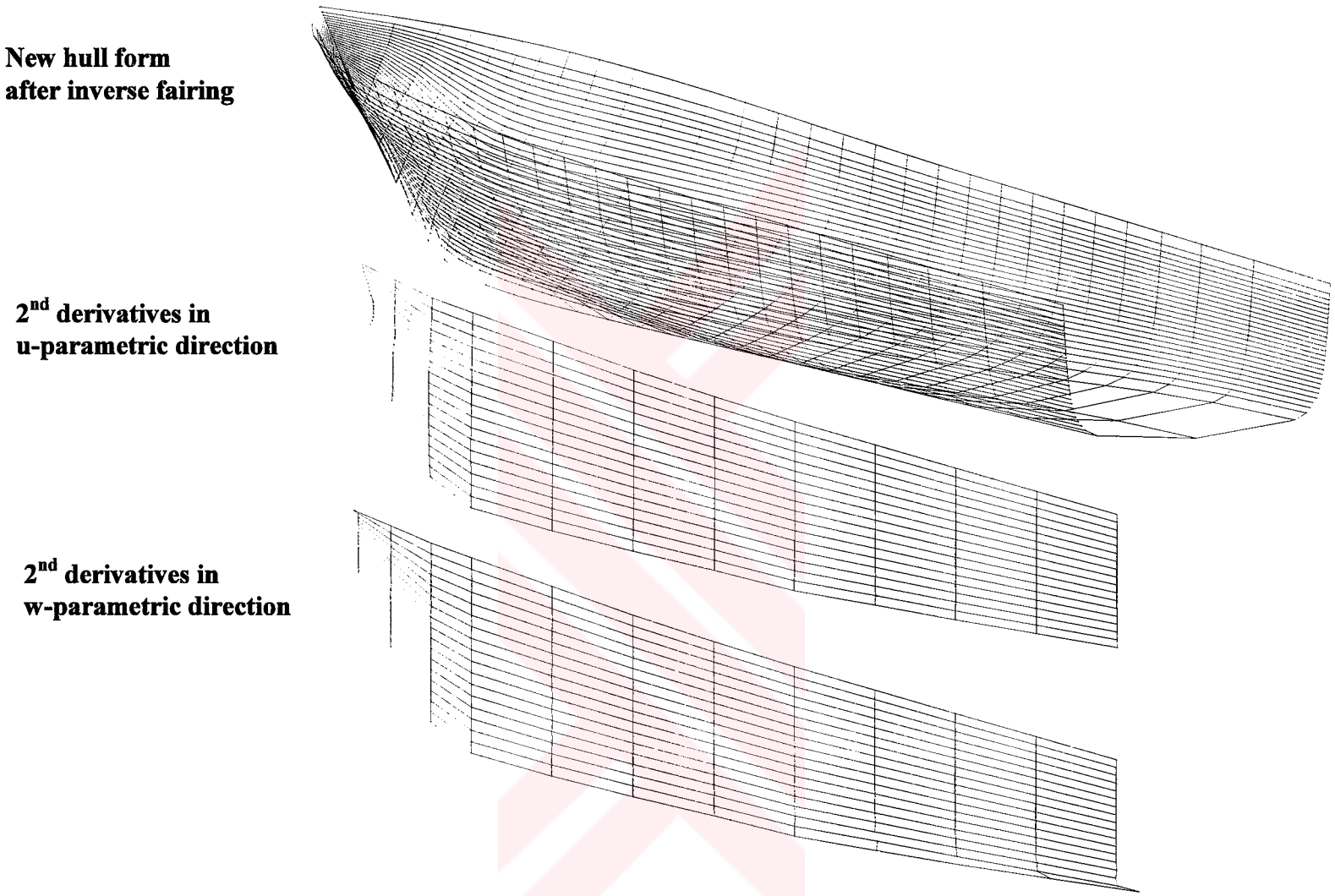


Figure 8.4. High-speed hull form and parametric derivatives after inverse fairing.

The optimisation process will also require a set of geometric constraints to ensure that the final form has satisfactory form properties. In this case the geometric constraints are selected as follows:

- $y_{ij} \geq 0$, i.e., all the offsets must be positive
- $y_{ij} \leq B/2$, i.e., the offsets must be less or equal to zero

The results of the optimisation procedure is shown in **Figures 8.6.**

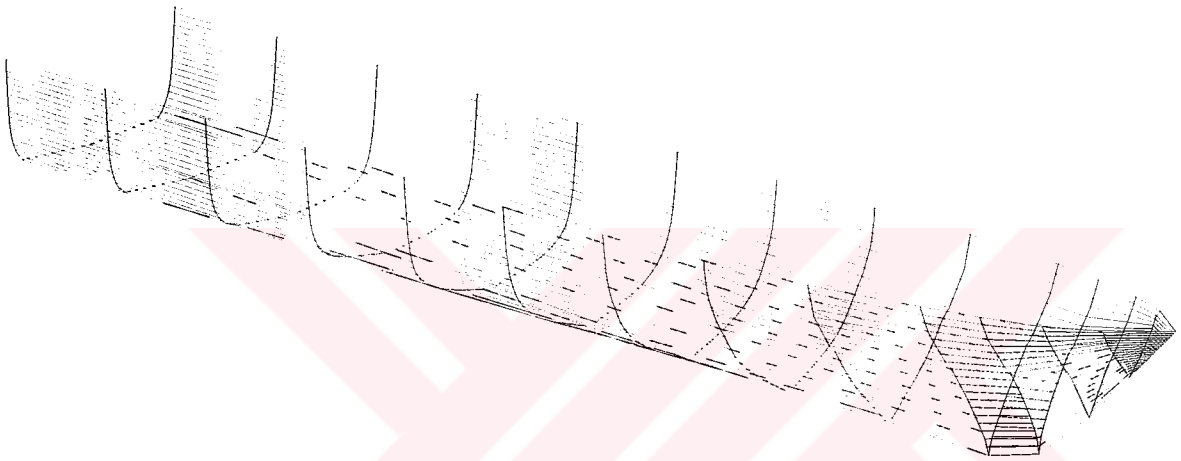


Figure 8.5.a. 3D mesh view of initial high-speed hull form.

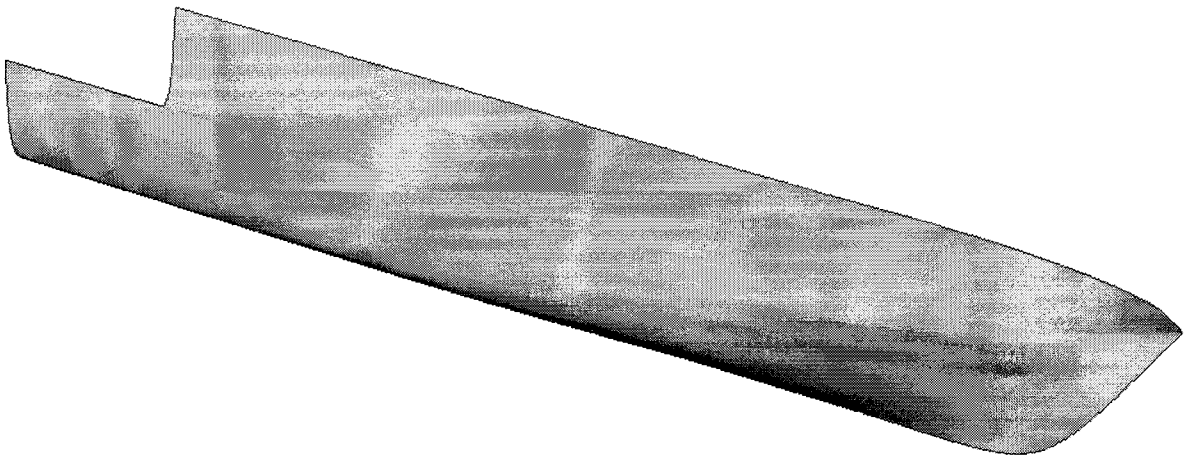


Figure 8.5.b. Shaded view of initial high-speed hull form.

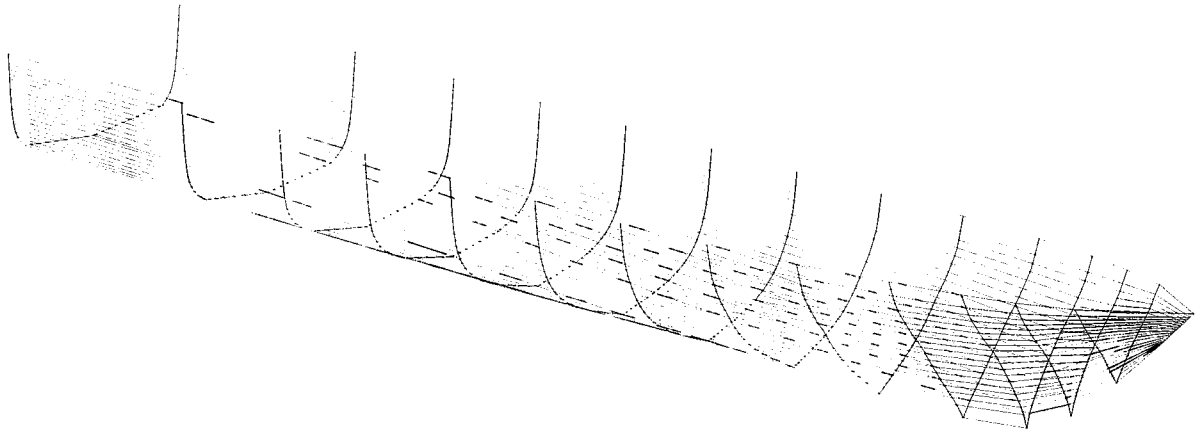


Figure 8.6.a. 3D mesh view of optimised high-speed hull form.

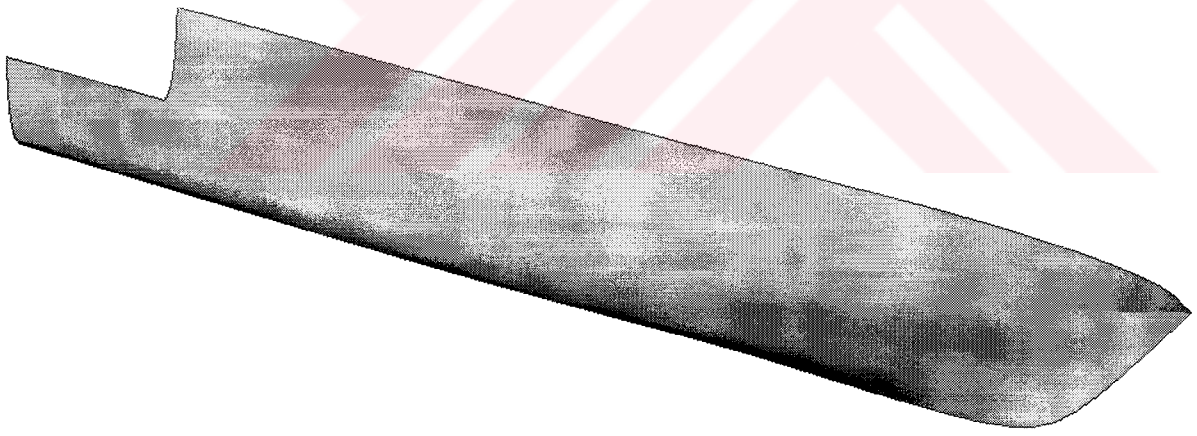


Figure 8.6.b. Shaded view of optimised high-speed hull form.

8.4. Comparative Assessment of Alternative Methodologies

In the previous sections, a typical high-speed hull form, with preliminary fairing properties, has been selected as a test case and three different fairing procedures, namely the forward fairing, the inverse fairing and the optimisation based fairing procedures have been applied to improve poor fairing characteristics. The results clearly indicate that the procedures, introduced and described in this study, manage to produce an alternative feasible hull form design with better fairing properties compared with the original hull form. However, the methodologies may differ in terms of flexibility, accuracy, human interaction and computational cost. In general, the following trends have been observed in the applications:

- The forward fairing process produces an alternative design only after few iterations. It can be seen from **Figure 8.2** that the final form closely represents the original design. This similarity is also proved by the low and acceptable degree of change in displacement and basic form parameters that are shown in **Table 8.1.** The method is flexible and easy to use, with no special experience needed. However, user intervention may be needed to terminate the iterations.

- The inverse fairing process also produces hull forms with improved fairness characteristics as the basics of the method stems from the fact that with improved curvature characteristics, it is possible to generate high quality surfaces in terms of fairness. The improved curvature characteristics of the selected hull (see **Figure 8.4**) approve the method as a useful and easy-to-use-fairing tool.

- The optimisation approach emerges as a rapidly evolving fairing tool. High quality surfaces can be obtained if the constraints of the fairing problem are clearly defined. The applications clearly indicate the flexibility and applicability of the method in real hull form fairing problems. The final hull form is obtained with no human intervention, which makes the process desirable in terms of automation, however there is no control over the outcome of the process.

The advantages and drawbacks of the developed procedures are summarised in **Table 8.2.**

Table 8.2. Comparison of the developed fairing methodologies.

	Forward Fairing	Inverse Fairing	Optimisation Approach
Computer storage (Input)	• Offset points • Fore, aft and deck profiles • Degree of waterlines, body lines	• Offset points • Fore, aft and deck profiles • Degree of the surface	• Offset points • Fore, aft and deck profiles • Objective function • Geometric constraints • Performance constraints • Termination criteria
Computing difficulty	Medium	Medium-low	Medium-high
User intervention	Minimal	Minimal	Minimal
Termination criteria	• Percentage change in form parameters • User defined maximum deviation	Automatic	Defined as input
Local control	No local control	No local control	High local control
Shape control handles	• Location of offset points • Number of defining vertex points • Degree of waterlines and body lines • Iteration number	• Location of offset points • Number of offset points • Degree of the surface	• Location of offset points • Number of offset points • Solution method • Objective function • Geometric constraints • Performance constraints • Termination criteria
Ability to represent corners, breaks, etc.	Defining multiple(k-1) polygon vertices or knot values at that point	Defining multiple(k-1) polygon vertices or knot values at that point	Defining as geometric constraints

9. CONCLUSIONS

The concept of a fair curve or surface is very important for designers of products involving free form curves and surfaces. Ship forms are typical examples for this type of products. Clearly, if the surface of a hull form is not fair, a fairing process should be carried out in order to obtain the required degree of fairness. The conventional solution of the problem is based on physical splines and experienced draftsman who will reduce the three-dimensional fairing problem into several two dimensional sub-problems and solve it in an iterative manner. Although, this method has been used successfully for many years, the designer clearly needs less time consuming and robust methods.

In this thesis three novel computer aided fairing procedures, which make heavy use of numerical splines and optimisation techniques are developed and described. The research conducted has yielded computational design tools for improving fairing characteristics of ship hull forms. The efficiency and flexibility of these tools are proved by applications to actual ship hull form fairing problems. The fairing procedures developed in this study are organised in three main groups:

Forward Fairing Procedures: Two distinct procedures based on B-spline approximation and B-spline fitting are developed and applied for fairing of two-dimensional ship lines. Three-dimensional fairing is obtained by simultaneous fairing of two-dimensional ship lines on three orthogonal planes, namely, the body plan waterlines and buttocks. The objective in both procedures is to minimise the rate of change of curvature while the deviation from original offset points should remain within specified limits. The objective is characterised by fairness and closeness numbers, which are introduced to evaluate the new forms generated throughout the fairing process. Although both numbers should be optimised simultaneously, this is not possible due to their diverse effects, hence the designer must seek a compromise solution by changing the order of B-spline curves, or by changing the number of iterations.

Both procedures are applied for fairing the lines of a distorted mathematical hull form and the parent hull of a semi-displacement hull form series. These procedures are shown to produce fair final forms after a few iterations. In general, smaller fairness numbers can be obtained by B-spline approximation but there may be

unacceptable deviations from the original offset points. This excessive deviation problem may be overcome by applying linear distortion techniques to preserve original hull form characteristics without losing already obtained fairness characteristics or by applying alternative B-spline fitting process. This alternative process could produce very close forms with specified degree of fairness. Moreover, the fairness and closeness of the final form can easily be controlled by selecting suitable number of polygon vertices.

Inverse Fairing Procedures: This process is based on the assumption that the fairness characteristics of a curve or surface is indicated by its curvature plot, hence the original curve or surface can be modified by representing its curvature plot by B-splines of suitable order. The original offset points are obtained from the curvature plot in an inverse manner. The order of the B-spline representation determines the degree of fairness. In general a compromise solution must be sought in order to preserve the original form characteristics. The inverse fairing procedures have successfully been applied to realistic ship forms as well as a mathematical form. The experience has shown that cubic and quartic B-splines provide the best results.

Fairing as a Non-Linear Optimisation Problem: As a further contribution, the fairing of ship hull forms problem is formulated as a non-linear optimisation problem in which the objective function is a fairness functional to be optimised subject to appropriate geometric and performance constraints. Various fairness functionals are available including energy minimising, mean curvature, Gaussian curvature and the variation of curvature. The applications indicate that these fairness functionals based on geometric properties of curvature yield quite similar results, producing high quality surfaces at significant computational cost.

In order to produce realistic hull forms both geometric and performance constraints should be used. The geometric constraints typically include positions and directions while the performance constraints may include volume, surface area, powering, cost, etc. Typical applications indicate that, provided that the designer can specify the objectives and constraints of the problem clearly, feasible and fair hull forms can be obtained after a reasonable number of iterations.

The main conclusions of the study are as follows:

- The need to develop automated fairing procedures has been apparent to ship designers with the advent of new computer aided hull form design procedures and developments in mathematical spline techniques. Three distinct novel

procedures have been developed and presented in this thesis whereby fairing of a given hull form can be improved within the constraints of the design problem.

- Curves and surfaces computed using non-linear numerical optimisation techniques were until recently too computationally expensive to be practical for general use. However, with wide availability of high computer processing power, optimisation techniques have been both practical and attractive. The applications in this thesis demonstrate optimisation as a superior and efficient tool in hull form fairing, and we can conclude that this process emerges as a rapid evolving and promising tool in hull form design environment.

- The discussions in this thesis have been concerned with the improvement of fairing characteristics of hull forms. In a real design problem, the designer must evolve a hull configuration, which satisfies many requirements, fairing being one of these. Therefore, major performance characteristics such as powering, cost and seakeeping must be a part of the fairing procedure.

- Applications, which include both mathematical and real ship hull forms have demonstrated the flexibility of the procedures.

The research program should now be expanded to include other problems in the following areas:

- Incorporation of the fairing procedures developed into a larger computer aided ship design system where all major design considerations are taken into account simultaneously.

- Further development of the optimisation based fairing procedures to include multi-objective decision making approaches.

- Generalisation of the fairing procedures for multi-hulls, developable surfaces and other hull form types.

In conclusion, it is believed that this study has succeeded in clearly describing a contribution to the fairing design effort, taking place in the Department of Ocean Engineering at ITU.

REFERENCES

Asker B., 1962. The Spline Curve, A Smooth Interpolating Function Used in Numerical Design of Ship Lines, BIT, **2**, 76-82.

Atkins D.A., Berger S.A., Webster W.C., Tapia R., 1966. Mathematical Ship Lofting , *Journal of Ship Research,* **Vol. 10, No. 4**, 203-222.

Bailey D., 1976. The NPL High Speed Round Bilge Displacement Hull Series, *Maritime Technology Monograph*, RINA, **4**.

Barnhill R.E. and Riesenfeld R.F. (eds), 1974. Computer Aided Geometric Design, Academic Press, New York.

Bartels R.H., Beatty J.C. and Barsky B.A., 1987. An Introduction to Splines for Use in Computer Graphics and Geometric Modelling, Morgan Kaufmann Publishers, California.

Benson F.W., 1940. Mathematical Ship Lines, *Transactions of RINA*, 129-151.

Bezier P., 1966. Definition Numerique des Courbes et Surfaces I, Automatisme, **XI**, 625-632.

Birkhoff G.D., 1933. Aesthetic Measure, Harvard University Press.

Bu-Qing S. and Ding-Yuan L., 1989. Computational Geometry, Curve and Surface Modelling, Academic Press, San Diego.

Calkins D.E., Thedoracatos V.E., Aguilor G.D. and Bryant D.M., 1989. Small Craft Hull Form Surface Definition in a High Level Computer Graphics Design Environment, *Transactions of. SNAME*, **Vol. 97**.

Chapman F.H. de., 1760. A Treatise on Shipbuilding, Translated into English by James Inman in 1820, Cambridge and London.

Chen S.E. and Parent R.E., 1989. Shape Averaging and Its Applications to Industrial Design, IEEE Computer Graphics and Applications, January, 47-54.

Cline A.K., 1974. Scalar and Planar Valued Curve fitting Using Splines Under Tension, *Communications of the ACM*, **17,** 218-220.

Coons S.A., 1964. Surfaces for Computer Aided Design, MIT, Project **MAC-TR-41**.

Coons W., 1974. Surface Patches and B-Spline Curves, In Computer Aided Geometric Design, R.E. Barnhill and R.F. Riesenfeld (eds.), Academic Press, New York, 1-16.

Cox M.G., 1971. The Numerical Evaluation of B-Splines, *National Physical Laboratory*, DNAC 4.

Cox M.G., 1972. The Numerical Evaluation of B-Splines, *Journal of the Institute of Mathematics and its Applications*, **Vol. 10, No. 2**, 134-149.

de Boor C., 1972. On Calculation with B-Splines., *Journal of Approximation Theory*, **Vol. 6, No. 1**, 50-62.

de Casteljau P., 1959. Courbes et Surfaces a Poles, Citroen, Paris.

Dill J.C., 1981. An Application of Colour Graphics to the Display of Surface Curvature, *Computer Graphics*, **Vol. 15**.

Farin G., Rein G. Sapidis N. Worsey A.J., 1987. Fairing Cubic B-Spline Curves, *Computer Aided Geometric Design,* 91-103.

Farin G.E. and Sapidis N., 1989. Curvature and the Fairness of Curves and Surfaces, *IEEE Computer Graphics and Applications,* 52-57.

Farin G., 1993. Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide, Third Edition , Academic Press, San Diego.

Faux I.D. and Pratt M.J., 1981. Computational Geometry for Design and Manufacture, Ellis Horwood, West Sussex, England.

Ferguson J.C., 1964. Multivariate Curve Interpolation, *Journal of ACM,* **Vol. 11, No 2**, 221-228.

Foley T.A., 1987. Interpolation with Interval and Point Tension Controls Using Cubic ν-Splines , ACM *Transactions on Mathematical Software, 13,* 68-96.

Forrest A.R., 1968. Curves and Surfaces for Computer Aided Design, *Ph.D. Thesis,* Cambridge.

Gordon W. and Riesenfeld R., 1974. B-Spline Curves and Surfaces, In Computer Aided Geometric Design, R.E. Barnhill and R.F. Riesenfeld (eds.), Academic Press, New York, 95-126.

Hagen H., 1985. Geometric Spline Curves, *Computer Aided Geometric Design*, **2,** 223-227.

Holladay J.C., 1957. Smoothest Curve Approximation, *Math Tables Aids Computations*, **11**.

Hooke R. and Jeeves T.A., 1961. Direct Search Solution of Numerical and Statistical Problems, *Journal of the Association for Computing Machines,* **Vol. 8,** 212-229.

Hoschek J. and Lasser D., 1993. Fundamentals of Computer Aided Geometric Design, L.L. Schumaker (trans.), A K Peters, Massachusetts.

Kerwin J., 1960. Polynomial Surface Representation of Arbitrary Ship Forms, *Journal of Ship Research*, **4**, 12-20.

Lackenby H., 1950. On the Systematic Geometrical Variation of Ship Forms, *Transactions of INA,* **Vol. 92**, London.

Lyche T. and Schumaker L.L. (eds.), 1992. Mathematical Methods in Computer Aided Geometric Design, Academic Press.

Meier H, and Nowacki H., 1987. Interpolating Curves with Gradual Changes in Curvature, *Computer Aided Geometric Design,* 297-305.

Miller N.S. and Kuo C., 1963. The Mathematical Fairing of Ship Lines, *European Shipbuilding, RINA,* **Vol. 12, No. 4**, 72-85.

Moor D.I., 1970. Effects on Performance in Still water and Waves of Some Geometric Changes to the Form of Large Twin-Screw Ship, *Trans. SNAME,* **Vol. 98**.

Moreton H.P., 1992. Minimum Curvature Variation Curves, Networks, and Surfaces for Fair Free-Form Shape Design, *Ph.D. Thesis,* Computer Science Department, University of California, Berkeley.

Narlı E., 1995. Fairing of Two-Dimensional Ship Lines, *M.Sc. Thesis,* Istanbul Technical University.

Narlı E., 1997. Development of Fair Hull Forms by Using B-Spline Techniques, *Proceedings of the Eighth Congress of the International Maritime Association of Mediterranean,* **Vol. 1**, 2-9 November, Istanbul-Turkey.

Narlı E. and Sarıöz K., 1998. Fairing of High Speed Displacement Hull Forms by B-spline Approximation and Fitting, *Naval Engineers Journal,* **Vol. 110, No. 2**, 35-47.

Narlı E., Kükner A. and Sarıöz K., 1999. Variation and Distortion of Fishing Vessel Forms, *Proceedings of International Conference on Technics & Technology in Fishing Vessels,* 15-17 May, Ancona –Italy.

Nielson G., 1974. Some Piecewise Polynomial Alternatives to Splines Under Tension, In Computer Aided Geometric Design, R.E. Barnhill and R.F. Riesenfeld (eds.), Academic Press, New York, 209-235.

Nowacki H., Dingyuan L. and Xinmin L., 1990. Fairing Bezier Curves with Constraints, *Computer Aided Geometric Design,* **7**, 43-55.

Nowacki H., Bloor M.I.G. and Oleksiewicz B., 1995. Computational Geometry for Ships, World Scientific Publishing, Singapore.

Nystrom J.W., 1863. Journal of Franklin Institute, **XLVI.**

Overhauser A.W., 1968. Analytic Definition of Curves and Surfaces by Parabolic Blending, Tech. Report, Ford Motor Company Scientific Lab., **SL68-40.**

Pattulo R.N.M. and Thomson G.R., 1965. The BSRA Trawler Series (Part I), *Transactions of RINA*, **Vol. 107**.

Pien P.C., 1960. Mathematical Ship Surface, David Taylor Model Basin Report 1398.

Pigounakis K., Sapidis N. and Kaklis P., 1996. Fairing Spatial B-Spline Curves, *Journal of Ship Research*, **Vol. 40, No. 4**, 351-367.

Pilcher D., 1974. Smooth Parametric Surfaces, In Computer Aided Geometric Design, R.E. Barnhill and R.F. Riesenfeld (eds.), Academic Press, New York, 237-253.

Pottmann H., 1990. Smooth Curves Under Tension, *Computer Aided Design,* **22,** 241-245.

Reese D., 1985. Fairing of Ship Lines and Surfaces, *Computer Applications in the Automation of Shipyard Operation and Ship Design V,* **11,** 395-399.

Rabien U., 1996. Ship geometry Modelling, *Schiffsetechnik, Ship Technology research,* **Vol. 43,** 115-123.

Riesenfeld R.F., 1973. Application of B-Spline Approximation to Geometric Problems of Computer-Aided Design, *Ph.D. Thesis,* Computer Science Department, Syracuse University, N.Y., Available as Tech. Report No. UTEC-CSc-73-126, University of Utah.

Rogers D.F. and Adams J.A., 1990. Mathematical Elements for Computer Graphics, Second Edition, McGraw-Hill Publishing.

Roulier J.A., Rando T. and Piper B., 1991. Fairness and Monotone Curvature, In Approximation Theory and Functional Analysis, C.K. Chui (ed.), Academic Press, New York.

Roulier J.A. and Rando T., 1994. Measures of Fairness for Curves and Surfaces, In Designing Fair Curves and Surfaces: Shape Quality in Geometric Modelling and Computer Aided Design, N. Sapidis (ed.), SIAM, Philadelphia, 75-122.

Roseman D.P. (ed.), 1987. The MARAD Systematic Series of Full Form Ship Models, SNAME, Jersey City.

Rosingh W. and Berghaus J., 1959. Mathematical Shipform, *International Shipbuilding Progress,* **Vol. 6.**

Sapidis N. and Farin G., 1990. Automatic Fairing Algorithm for B-Spline Curves, *Computer Aided Design,* **22.**

Schoenberg I.J., 1946. Contribution to the Problem of Approximation of Equidistant Data by Analytic Functions, *Quarterly Applied Mathematics*, **Vol. 4, No. 1**, 45-99.

Schweikert D.G., 1966. An Interpolation Curve Using a Spline in Tension, *Journal of Mathematics and Physics*, **45**, 312-317.

Söding H., 1967. Entwurf von Schiffsformen mit dem Rechner, *Schiff und Hafen.*.

Taylor D.W., 1915. Calculation for Ship's Forms and the Light Thrown by Model Experiments upon Resistance, Propeller and Rolling of Ships, *Transactions of International Engineering Congress*, San Francisco.

Taylor F., 1963. Computer Applications to Shipbuilding, *Transactions of RINA*, **105**.

Theilheimer F., 1957. Universe Calculations of Ship Lines, *Journal of American Society of Naval Engineers,* May.

Theilheimer F. and Starkweather W., 1961. The Fairing of Ship Lines on a High Speed Computer, *Mathematics of Computation,* **15.**

Versprille K.J., 1975. Computer Aided Design Applications of the Rational B-Spline Form, *Ph.D. Thesis,* Department of Computer Science, Syracuse University.

von Kerczek C. and Tuck E.O., 1969. The Representation of Ship Hulls by Conformal Mapping Functions, *Journal of Ship Research,* **Vol. 13, No 4.**

Yamaguchi F., 1988. Curves and Surfaces in Computer Aided Geometric Design, Springer-Verlag, Berlin Heidelberg.

Watanabe K., 1946. Mathematical Ships' Lines, *Journal of Zosen Kyokai,* November, **Vol. 77**.

Welch W. and Witkin A., 1992. Variational Surface Modelling, In *Proceedings of SIGGRAPH '92 Computer Graphics,* **Vol. 26, No. 2**, 157-166.

Welch W., 1995. Serious Putty, Topological Design for Variational Curves and Surfaces, *Ph.D. Thesis,* Carnegie Mellon University.

Weinblum G., 1934. Exact Waterlines and Sectional Area Curves, *Schiffbau*, **120**, 135-142.

Wesselink J.W., 1996. Variational Modelling of Curves and Surfaces, *Ph.D. Thesis,* Eindhoven Technical University.

Wigley W.C.S., 1934. A Comparison of Experiment and Calculated Wave Profiles and Wave Resistance for a Form Having Parabolic Waterlines, Proceedings of Royal, Series A, **Vol. 144**.

Williams A., 1964. Mathematical Representation of Ordinary Ship Forms, *Schiff und Hafen,* **Vol. 16, No. 51,** 414-433.

APPENDIX A - TEST CURVE DEFINITIONS

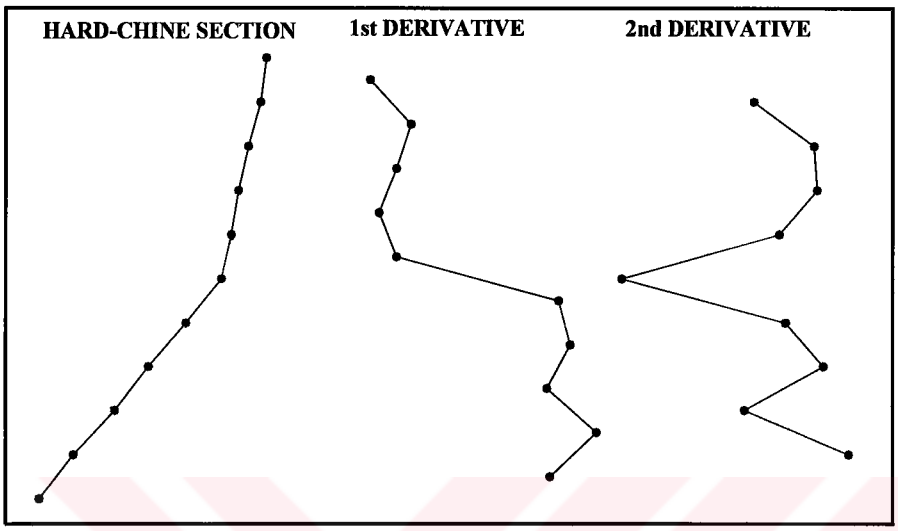


Figure A.1. Hard-chine section in Figure 3.5.

Table A.1. Offsets, first and second derivative values of hard-chine section.

Depth (m)	Half-Breadth	1st Derivative	2nd Derivative
0.00	0.00		
0.10	0.75	7.50	16.00
0.20	1.66	9.10	-17.00
0.30	2.40	7.40	8.00
0.40	3.22	8.20	-4.00
0.50	4.00	7.80	-56.00
0.60	4.22	2.20	-6.00
0.70	4.38	1.60	6.00
0.80	4.60	2.20	5.00
0.90	4.87	2.70	-14.00
1.00	5.00	1.30	

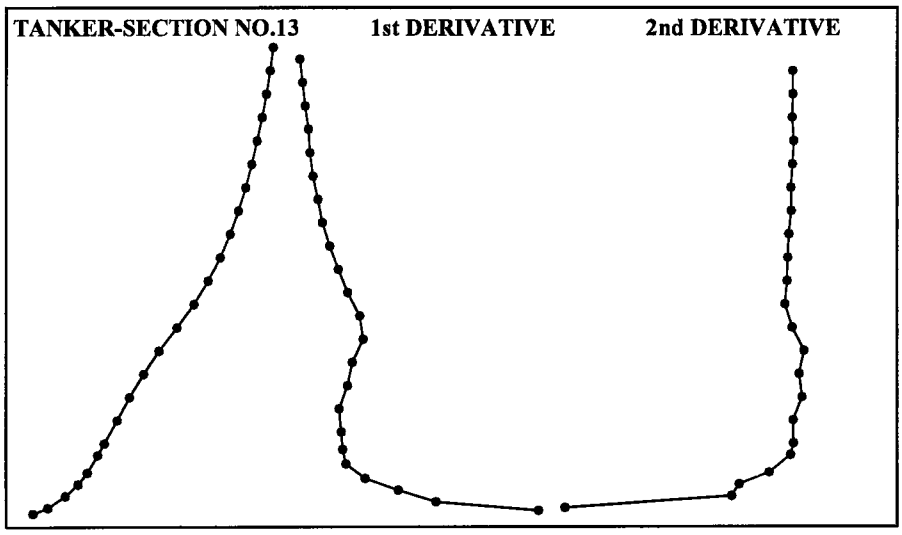


Figure A.2. Tanker section No. 13.

Table A.2. Offsets, first and second derivative values of Tanker section No.13.

Depth (m)	Half-Breadth	1st Derivative	2nd Derivative
0.00	1.306		
0.10	1.641	3.3500	-9.1000
0.30	2.038	1.9850	-2.5000
0.50	2.335	1.4850	-2.2000
0.70	2.544	1.0450	-1.0200
1.00	2.781	0.7900	-0.1800
1.20	2.930	0.7450	-0.0583
1.60	3.221	0.7275	-0.0750
2.00	3.500	0.6975	0.2812
2.40	3.824	0.8100	0.1625
2.80	4.174	0.8750	0.3562
3.20	4.581	1.0175	-0.1125
3.60	4.970	0.9725	-0.4000
4.00	5.295	0.8125	-0.3125
4.40	5.570	0.6875	-0.2875
4.80	5.799	0.5725	-0.2483
5.20	5.989	0.4750	-0.1500
5.60	6.155	0.4150	-0.1625
6.00	6.295	0.3500	-0.1000
6.40	6.419	0.3100	-0.0500
6.80	6.535	0.2900	-0.1125
7.20	6.633	0.2450	-0.0875
7.60	6.717	0.2100	-0.0937
8.00	6.786	0.1725	

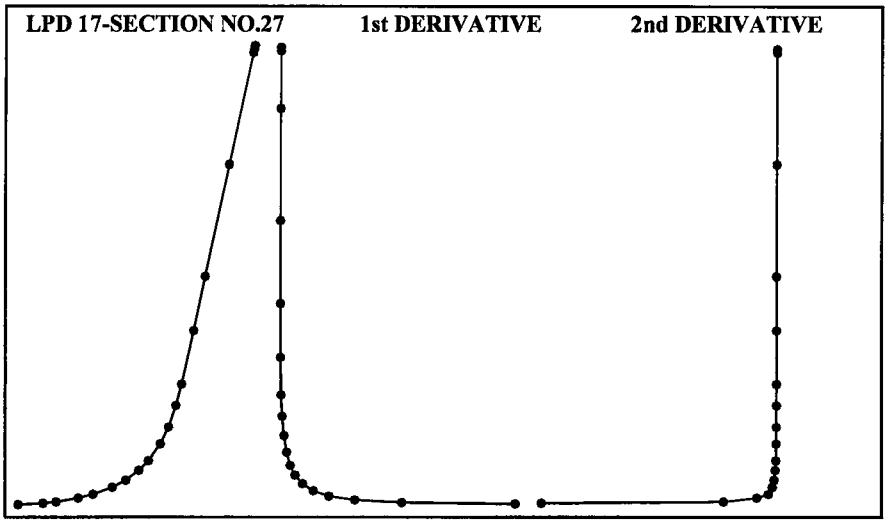


Figure A.3. LPD 17 – Section No. 27.

Table A.3. Offsets, first and second derivative values of LPD 17-Section No.27.

Depth (m)	Half-Breadth	1st Derivative	2nd Derivative
-0.0140	10.1104	14.9471	-171.1158
0.0276	10.7322	7.8201	-38.5113
0.0693	11.0583	4.8432	-14.4009
0.1822	11.6051	3.2181	-5.9961
0.2950	11.9681	2.2368	-3.1698
0.5095	12.4479	1.5566	-1.8047
0.7242	12.7821	1.0977	-1.0273
1.0181	13.1047	0.7958	-0.5531
1.3119	13.3385	0.5738	-0.3369
1.8208	13.6305	0.4024	-0.2070
2.3292	13.8351	0.2828	-0.1042
2.9767	14.0182	0.2153	-0.0260
3.6236	14.1575	0.1859	-0.0050
5.2349	14.4571	0.1778	-0.0005
6.8462	14.7436	0.1767	-0.0003
10.2187	15.3394	0.1757	0.0015
13.5911	15.9319	0.1783	0.0049
13.6943	15.9503	0.1788	
13.8000	15.9692		

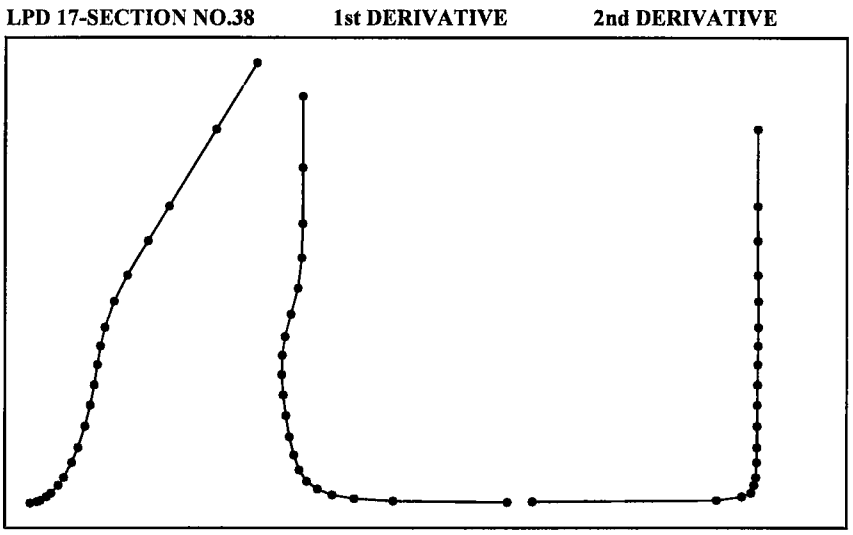


Figure A.4. LPD 17 – Section No. 38.

Table A.4. Offsets, first and second derivative values of LPD 17-Section No.38.

Depth (m)	Half-Breadth	1st Derivative	2nd Derivative
-0.0139	0.7462		
0.0424	1.0579	5.5364	-48.0305
0.0990	1.2178	2.8251	-8.7336
0.2546	1.5132	1.8985	-3.3575
0.4101	1.7272	1.3762	-1.4145
0.7447	2.0717	1.0296	-0.7662
1.0788	2.3301	0.7734	-0.3694
1.7189	2.7100	0.5935	-0.1946
2.3586	3.0100	0.4690	-0.1416
3.2647	3.3358	0.3596	-0.0918
4.1705	3.5862	0.2764	-0.0738
5.0327	3.7638	0.2112	-0.0427
5.8944	3.9186	0.1744	0.0166
6.6911	4.0685	0.1882	0.0824
7.4870	4.2705	0.2538	0.1467
8.6050	4.7112	0.3942	0.1506
9.7236	5.3405	0.5626	0.0683
11.2117	6.3102	0.6516	0.0131
12.7002	7.3091	0.6711	0.0006
15.9946	9.5244	0.6724	-0.0003
18.8500	11.4415	0.6714	

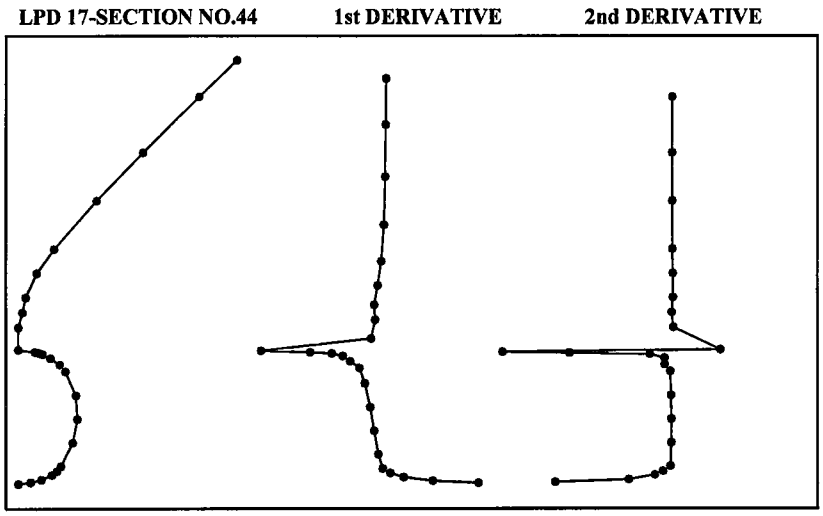


Figure A.5. LPD 17 – Section No. 44.

Table A.5. Offsets, first and second derivative values of LPD 17-Section No.44.

Depth (m)	Half-Breadth	1st Derivative	2nd Derivative
0.0228	0.0027	4.5034	-20.5528
0.0959	0.3319	2.5807	-7.5895
0.2099	0.6261	1.3603	-3.0285
0.4175	0.9085	0.7979	-1.5985
0.5813	1.0392	0.4867	-0.2897
0.8069	1.1490	0.3020	-0.1695
1.8565	1.4660	0.1242	-0.1559
2.9057	1.5963	-0.0409	-0.2100
3.9742	1.5526	-0.2654	-0.3346
5.0433	1.2689	-0.4947	-1.3118
5.3451	1.1196	-0.8808	-1.3398
5.6319	0.8670	-1.1910	-3.9099
5.8083	0.6569	-1.6540	-18.0245
5.8687	0.5570	-2.5651	-29.7863
5.9094	0.4526	-4.6278	8.4859
6.0072	0.0000	0.0000	0.2082
7.0001	0.0000	0.1725	-0.0581
7.6639	0.1145	0.1339	0.1660
8.3277	0.2034	0.2783	0.1427
9.4038	0.5029	0.4319	0.0665
10.1809	0.9681	0.5386	0.0217
12.6103	2.1149	0.5849	0.0101
14.7404	3.3607	0.6082	0.0088
17.2175	4.8672	0.6263	
18.8500	5.8896		

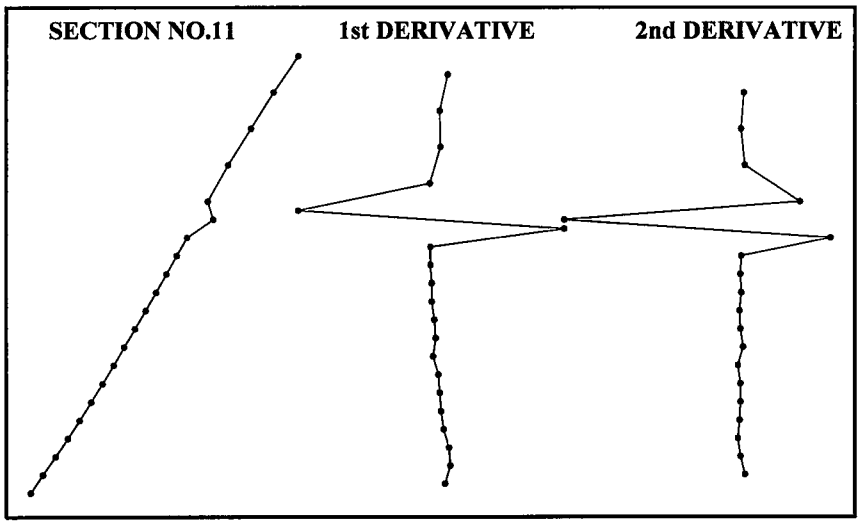


Figure A.6. NPL– Section No.11.

Table A.6. Offsets, first and second derivative values of NPL-Section No.11.

Depth (m)	Half-Breadth	1st Derivative	2nd Derivative
0.0000	0.0034		
0.0100	0.0110	0.7600	4.0000
0.0200	0.0190	0.8000	-1.0000
0.0300	0.0269	0.7900	-4.0000
0.0400	0.0344	0.7500	-2.0000
0.0500	0.0417	0.7300	-1.0000
0.0600	0.0489	0.7200	-1.0000
0.0700	0.0560	0.7100	-4.0000
0.0800	0.0627	0.6700	2.0000
0.0900	0.0696	0.6900	-1.0000
0.1000	0.0764	0.6800	-2.0000
0.1100	0.0830	0.6600	0 .0000
0.1200	0.0896	0.6600	-1.0000
0.1300	0.0961	0.6500	0.0000
0.1400	0.1026	0.6500	101.0000
0.1500	0.1192	1.6600	-201.0000
0.1600	0.1157	-0.3500	66.3333
0.1800	0.1286	0.6450	4.0000
0.2000	0.1431	0.7250	-0.2500
0.2200	0.1575	0.7200	3.0000
0.2400	0.1731	0.7800	

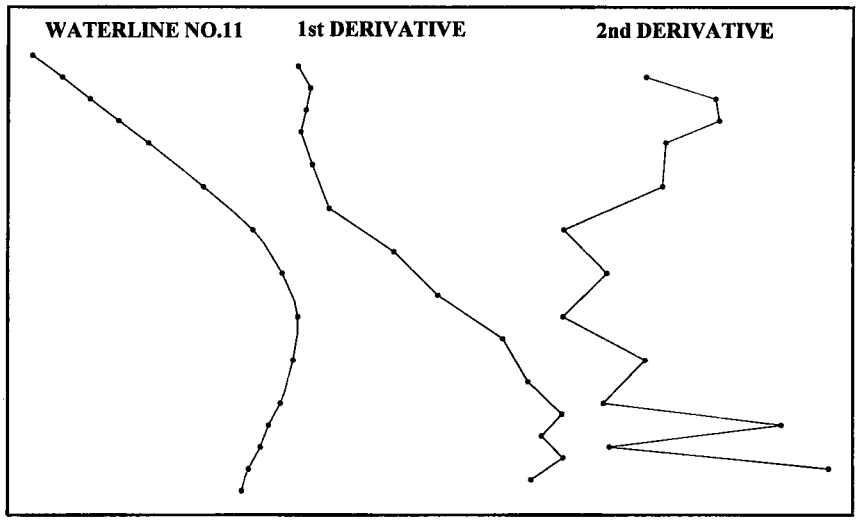


Figure A.7. NPL – Waterline No. 11

Table A.7. Offsets, first and second derivative values of NPL-Waterline No.11.

Station Length (m)	Half-Breadth (m)	1st Derivative	2nd Derivative
0.0000	0.1505		
0.1270	0.1554	0.0386	
0.2540	0.1639	0.0669	0.2232
0.3810	0.1700	0.0480	-0.1488
0.5080	0.1784	0.0661	0.1426
0.7620	0.1875	0.0358	-0.1591
1.0160	0.1909	0.0134	-0.0884
1.2700	0.1796	-0.0445	-0.2279
1.5240	0.1584	-0.0835	-0.1535
1.7780	0.1226	-0.1409	-0.2263
2.0320	0.0830	-0.1559	-0.0589
2.1590	0.0619	-0.1661	-0.0537
2.2860	0.0414	-0.1614	0.0372
2.4130	0.0214	-0.1575	0.0310
2.5400	0.0000	-0.1685	-0.0868

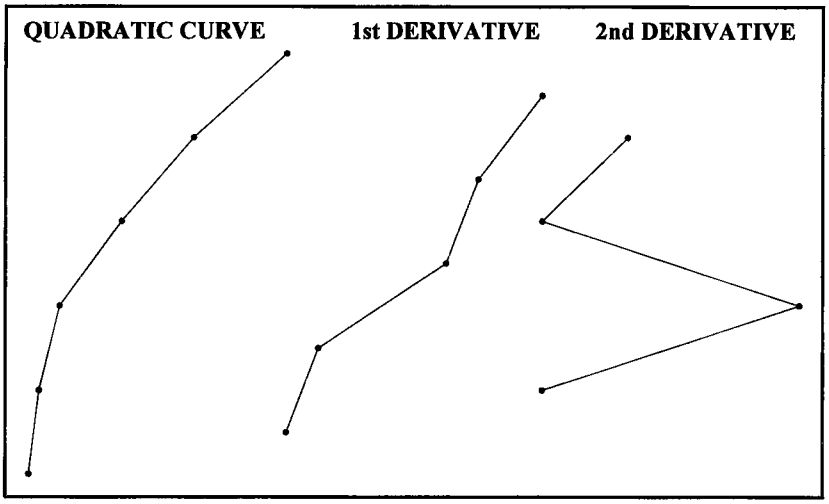


Figure A.8. Deliberately distorted quadratic curve.

Table A.8. Offsets, first and second derivative values of quadratic curve.

Depth (m)	Half-Breadth	1st Derivative	2nd Derivative
0.0000	0.0000		
1.0000	1.0000	1.0000	
2.0000	3.0000	2.0000	1.0000
3.0000	9.0000	6.0000	4.0000
4.0000	16.0000	7.0000	1.0000
5.0000	25.0000	9.0000	2.0000

APPENDIX B - TEST HULL SURFACE DEFINITIONS AND RESULTS

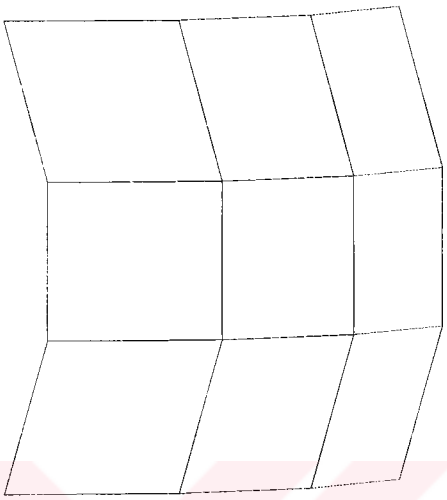


Figure B.1. Defining control net in Figure 3.10.

Table B.1. Control net points of the test surface illustrated in Figure 3.10.

x	y	z
-15.00	0.00	15.00
-15.00	5.00	5.00
-15.00	5.00	-5.00
-15.00	0.00	-15.00
-5.00	5.00	15.00
-5.00	10.00	5.00
-5.00	10.00	-5.00
-5.00	5.00	-15.00
5.00	5.00	15.00
5.00	10.00	5.00
5.00	10.00	-5.00
5.00	5.00	-15.00
15.00	0.00	15.00
15.00	5.00	5.00
15.00	5.00	-5.00
15.00	0.00	-15.00

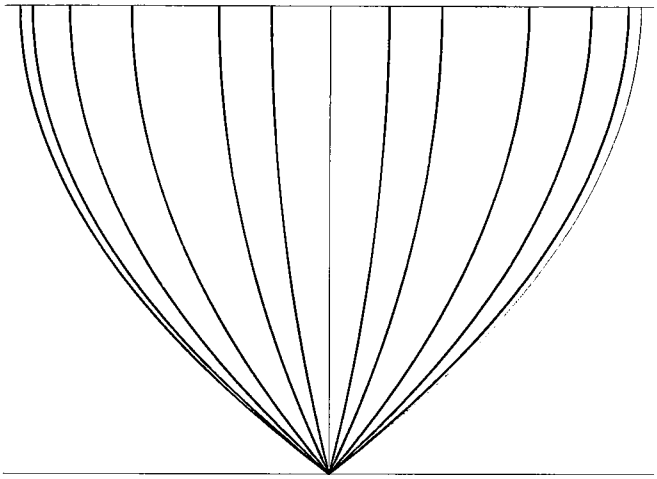


Figure B.2. Mathematical Wigley hull form body plan.

Table B.2. Wigley hull form offsets obtained from three different fairing procedures. (Forward, inverse and optimisation)

Station X	WL Z (m)	Original Y (m)	Distorted Y (m)	Forward Fairing (1) Y (m)	Forward Fairing (2) Y (m)	Inverse Fairing Y (m)	Optimisation Approach Y(m)
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.1053	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.1579	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2105	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2632	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3158	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3684	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.4211	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.4737	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.5263	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.5789	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6316	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6842	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.7368	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.7895	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.8421	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.8947	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.9474	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

1/2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0156	0.0111	0.0128	0.0137	0.0138	0.0147
	0.1053	0.0303	0.0322	0.0259	0.0275	0.0276	0.0292
	0.1579	0.0442	0.0433	0.0389	0.0414	0.0415	0.0433
	0.2105	0.0573	0.0544	0.0517	0.0551	0.0553	0.0567
	0.2632	0.0695	0.0655	0.0642	0.0682	0.0687	0.0693
	0.3158	0.0808	0.0866	0.0757	0.0804	0.0811	0.0809
	0.3684	0.0914	0.0977	0.0860	0.0912	0.0920	0.0914
	0.4211	0.1011	0.1088	0.0947	0.1006	0.1013	0.1010
	0.4737	0.1099	0.1099	0.1016	0.1086	0.1091	0.1095
	0.5263	0.1179	0.1110	0.1075	0.1156	0.1159	0.1171
	0.5789	0.1251	0.1221	0.1132	0.1219	0.1221	0.1240
	0.6316	0.1314	0.1332	0.1189	0.1279	0.1280	0.1303
	0.6842	0.1368	0.1343	0.1243	0.1335	0.1337	0.1360
	0.7368	0.1415	0.1454	0.1292	0.1387	0.1391	0.1411
	0.7895	0.1453	0.1465	0.1335	0.1432	0.1437	0.1456
	0.8421	0.1482	0.1476	0.1371	0.1467	0.1474	0.1494
	0.8947	0.1503	0.1587	0.1396	0.1491	0.1498	0.1526
	0.9474	0.1516	0.1598	0.1403	0.1505	0.1509	0.1558
	1.0000	0.1520	0.1509	0.1398	0.1514	0.1515	0.1589
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0295	0.0215	0.0247	0.0263	0.0264	0.0279
	0.1053	0.0574	0.0524	0.0498	0.0526	0.0527	0.0554
	0.1579	0.0838	0.0838	0.0746	0.0787	0.0790	0.0822
	0.2105	0.1085	0.1045	0.0989	0.1043	0.1049	0.1078
	0.2632	0.1316	0.1356	0.1223	0.1288	0.1297	0.1320
	0.3158	0.1532	0.1562	0.1438	0.1516	0.1527	0.1542
	0.3684	0.1731	0.1771	0.1633	0.1720	0.1733	0.1744
	0.4211	0.1915	0.1985	0.1803	0.1900	0.1912	0.1925
	0.4737	0.2082	0.2092	0.1940	0.2056	0.2066	0.2085
	0.5263	0.2234	0.2204	0.2059	0.2193	0.2200	0.2227
	0.5789	0.2369	0.2319	0.2171	0.2317	0.2321	0.2354
	0.6316	0.2498	0.2428	0.2279	0.2431	0.2435	0.2467
	0.6842	0.2593	0.2533	0.2378	0.2536	0.2541	0.2569
	0.7368	0.2681	0.2641	0.2468	0.2631	0.2638	0.2658
	0.7895	0.2752	0.2752	0.2547	0.2713	0.2722	0.2731
	0.8421	0.2808	0.2868	0.2612	0.2777	0.2787	0.2793
	0.8947	0.2848	0.2878	0.2658	0.2823	0.2832	0.2846
	0.9474	0.2872	0.2882	0.2675	0.2852	0.2857	0.2896
	1.0000	0.2880	0.2890	0.2674	0.2873	0.2873	0.2948
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0525	0.0515	0.0455	0.0477	0.0480	0.0499
	0.1053	0.1021	0.1021	0.0909	0.0951	0.0957	0.0993
	0.1579	0.1489	0.1439	0.1349	0.1412	0.1422	0.1472

	0.2105	0.1929	0.1949	0.1768	0.1855	0.1867	0.1931
	0.2632	0.2340	0.2350	0.2164	0.2275	0.2289	0.2362
	0.3158	0.2723	0.2763	0.2534	0.2666	0.2681	0.2758
	0.3684	0.3078	0.3078	0.2876	0.3026	0.3043	0.3116
	0.4211	0.3404	0.3484	0.3185	0.3354	0.3371	0.3438
	0.4737	0.3702	0.3792	0.3450	0.3649	0.3665	0.3726
	0.5263	0.3971	0.3901	0.3680	0.3913	0.3927	0.3981
	0.5789	0.4212	0.4212	0.3890	0.4147	0.4160	0.4212
2	0.6316	0.4425	0.4425	0.4085	0.4354	0.4366	0.4419
	0.6842	0.4609	0.4639	0.4256	0.4535	0.4547	0.4598
	0.7368	0.4765	0.4745	0.4402	0.4691	0.4704	0.4751
	0.7895	0.4893	0.4853	0.4527	0.4823	0.4836	0.4877
	0.8421	0.4992	0.4962	0.4631	0.4930	0.4943	0.4981
	0.8947	0.5063	0.5073	0.4709	0.5012	0.5023	0.5070
	0.9474	0.5106	0.5186	0.4752	0.5075	0.5081	0.5154
	1.0000	0.5120	0.5100	0.4770	0.5129	0.5130	0.5237
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0689	0.0619	0.0611	0.0630	0.0636	0.0649
	0.1053	0.1340	0.1320	0.1218	0.1255	0.1265	0.1291
	0.1579	0.1955	0.1935	0.1801	0.1858	0.1872	0.1914
	0.2105	0.2532	0.2542	0.2350	0.2434	0.2450	0.2510
	0.2632	0.3071	0.3051	0.2865	0.2979	0.2996	0.3072
	0.3158	0.3574	0.3564	0.3352	0.3487	0.3506	0.3591
	0.3684	0.4039	0.4079	0.3808	0.3959	0.3979	0.4066
	0.4211	0.4468	0.4488	0.4218	0.4392	0.4413	0.4496
3	0.4737	0.4859	0.4899	0.4574	0.4784	0.4805	0.4884
	0.5263	0.5212	0.5202	0.4889	0.5136	0.5155	0.5228
	0.5789	0.5529	0.5519	0.5179	0.5448	0.5467	0.5536
	0.6316	0.5808	0.5828	0.5442	0.5724	0.5742	0.5810
	0.6842	0.6050	0.6030	0.5670	0.5962	0.5980	0.6049
	0.7368	0.6255	0.6245	0.5863	0.6165	0.6182	0.6249
	0.7895	0.6422	0.6452	0.6024	0.6335	0.6351	0.6412
	0.8421	0.6552	0.6562	0.6153	0.6473	0.6487	0.6541
	0.8947	0.6646	0.6676	0.6253	0.6583	0.6595	0.6647
	0.9474	0.6701	0.6781	0.6316	0.6672	0.6679	0.6743
	1.0000	0.6720	0.6790	0.6356	0.6753	0.6755	0.6837
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0787	0.0717	0.0705	0.0720	0.0726	0.0739
	0.1053	0.1532	0.1522	0.1406	0.1432	0.1444	0.1469
	0.1579	0.2234	0.2234	0.2077	0.2121	0.2137	0.2175
4	0.2105	0.2893	0.2843	0.2703	0.2780	0.2798	0.2852
	0.2632	0.3510	0.3550	0.3292	0.3404	0.3424	0.3490
	0.3158	0.4085	0.4065	0.3854	0.3988	0.4010	0.4080
	0.3684	0.4617	0.4677	0.4383	0.4527	0.4551	0.4621

	0.4211	0.5106	0.5186	0.4855	0.5019	0.5044	0.5115
	0.4737	0.5553	0.5593	0.5262	0.5463	0.5486	0.5558
	0.5263	0.5957	0.5907	0.5625	0.5862	0.5882	0.5959
	0.5789	0.6318	0.6318	0.5962	0.6218	0.6238	0.6314
	0.6316	0.6638	0.6628	0.6268	0.6536	0.6556	0.6624
	0.6842	0.6914	0.6934	0.6533	0.6813	0.6834	0.6886
	0.7368	0.7148	0.7148	0.6755	0.7050	0.7071	0.7108
	0.7895	0.7340	0.7350	0.6939	0.7248	0.7267	0.7290
	0.8421	0.7489	0.7469	0.7085	0.7408	0.7425	0.7441
	0.8947	0.7595	0.7575	0.7193	0.7536	0.7548	0.7570
	0.9474	0.7659	0.7689	0.7267	0.7639	0.7646	0.7684
	1.0000	0.7680	0.7690	0.7321	0.7734	0.7736	0.7796
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0820	0.0810	0.0737	0.0751	0.0758	0.0786
	0.1053	0.1596	0.1526	0.1469	0.1494	0.1507	0.1561
	0.1579	0.2327	0.2337	0.2170	0.2211	0.2228	0.2309
	0.2105	0.3014	0.3044	0.2824	0.2897	0.2915	0.3019
	0.2632	0.3657	0.3657	0.3438	0.3546	0.3566	0.3684
	0.3158	0.4255	0.4265	0.4024	0.4152	0.4175	0.4300
	0.3684	0.4809	0.4879	0.4577	0.4713	0.4738	0.4861
	0.4211	0.5319	0.5389	0.5069	0.5225	0.5250	0.5365
5	0.4737	0.5784	0.5794	0.5494	0.5687	0.5710	0.5816
	0.5263	0.6205	0.6205	0.5874	0.6102	0.6123	0.6221
	0.5789	0.6582	0.6512	0.6225	0.6475	0.6495	0.6585
	0.6316	0.6914	0.6924	0.6545	0.6808	0.6828	0.6907
	0.6842	0.7202	0.7232	0.6823	0.7099	0.7120	0.7191
	0.7368	0.7446	0.7446	0.7055	0.7348	0.7369	0.7437
	0.7895	0.7645	0.7655	0.7247	0.7554	0.7575	0.7641
	0.8421	0.7801	0.7861	0.7399	0.7718	0.7737	0.7812
	0.8947	0.7911	0.7971	0.7511	0.7848	0.7861	0.7956
	0.9474	0.7978	0.7988	0.7588	0.7950	0.7958	0.8093
	1.0000	0.8000	0.8090	0.7648	0.8043	0.8045	0.8232
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0787	0.0717	0.0705	0.0721	0.0728	0.0739
	0.1053	0.1532	0.1522	0.1406	0.1435	0.1447	0.1469
	0.1579	0.2234	0.2234	0.2077	0.2123	0.2139	0.2175
	0.2105	0.2893	0.2843	0.2703	0.2781	0.2799	0.2852
6	0.2632	0.3510	0.3550	0.3291	0.3403	0.3423	0.3490
	0.3158	0.4085	0.4065	0.3851	0.3985	0.4006	0.4080
	0.3684	0.4617	0.4677	0.4380	0.4524	0.4547	0.4621
	0.4211	0.5106	0.5186	0.4855	0.5018	0.5042	0.5115
	0.4737	0.5553	0.5593	0.5269	0.5464	0.5487	0.5558
	0.5263	0.5957	0.5907	0.5635	0.5864	0.5885	0.5959
	0.5789	0.6318	0.6318	0.5967	0.6222	0.6242	0.6314

	0.6316	0.6638	0.6628	0.6270	0.6540	0.6560	0.6624
	0.6842	0.6914	0.6934	0.6533	0.6817	0.6837	0.6886
	0.7368	0.7148	0.7148	0.6755	0.7053	0.7074	0.7108
	0.7895	0.7340	0.7350	0.6939	0.7250	0.7269	0.7290
	0.8421	0.7489	0.7469	0.7084	0.7407	0.7424	0.7441
	0.8947	0.7595	0.7575	0.7193	0.7531	0.7544	0.7570
	0.9474	0.7659	0.7689	0.7269	0.7630	0.7637	0.7684
	1.0000	0.7680	0.7690	0.7326	0.7718	0.7720	0.7796
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0689	0.0619	0.0614	0.0627	0.0632	0.0650
	0.1053	0.1340	0.1320	0.1223	0.1249	0.1258	0.1292
	0.1579	0.1955	0.1935	0.1804	0.1852	0.1865	0.1914
	0.2105	0.2532	0.2542	0.2351	0.2430	0.2445	0.2512
	0.2632	0.3071	0.3051	0.2863	0.2978	0.2995	0.3076
	0.3158	0.3574	0.3564	0.3346	0.3491	0.3510	0.3597
	0.3684	0.4039	0.4079	0.3800	0.3968	0.3987	0.4076
	0.4211	0.4468	0.4488	0.4216	0.4405	0.4427	0.4510
	0.4737	0.4859	0.4899	0.4584	0.4799	0.4821	0.4899
	0.5263	0.5212	0.5292	0.4905	0.5150	0.5171	0.5249
	0.5789	0.5529	0.5519	0.5189	0.5458	0.5478	0.5564
	0.6316	0.5808	0.5828	0.5446	0.5728	0.5746	0.5839
	0.6842	0.6050	0.6030	0.5670	0.5962	0.5978	0.6071
	0.7368	0.6255	0.6245	0.5862	0.6163	0.6178	0.6261
	0.7895	0.6422	0.6452	0.6023	0.6333	0.6347	0.6412
	0.8421	0.6552	0.6562	0.6152	0.6476	0.6488	0.6537
	0.8947	0.6646	0.6676	0.6253	0.6595	0.6604	0.6643
	0.9474	0.6701	0.6781	0.6321	0.6696	0.6701	0.6734
	1.0000	0.6720	0.6790	0.6368	0.6790	0.6791	0.6824
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0525	0.0515	0.0467	0.0481	0.0484	0.0493
	0.1053	0.1021	0.1021	0.0927	0.0957	0.0964	0.0980
	0.1579	0.1489	0.1439	0.1363	0.1417	0.1428	0.1452
	0.2105	0.1929	0.1944	0.1775	0.1857	0.1870	0.1904
	0.2632	0.2340	0.2350	0.2162	0.2274	0.2288	0.2332
	0.3158	0.2723	0.2763	0.2524	0.2662	0.2677	0.2732
	0.3684	0.3078	0.3078	0.2860	0.3022	0.3037	0.3099
	0.4211	0.3404	0.3484	0.3170	0.3353	0.3369	0.3432
	0.4737	0.3702	0.3792	0.3445	0.3652	0.3668	0.3726
	0.5263	0.3971	0.3901	0.3687	0.3918	0.3934	0.3984
	0.5789	0.4212	0.4212	0.3899	0.4152	0.4167	0.4212
	0.6316	0.4425	0.4425	0.4091	0.4358	0.4371	0.4411
	0.6842	0.4609	0.4629	0.4260	0.4537	0.4549	0.4583
	0.7368	0.4765	0.4735	0.4404	0.4692	0.4703	0.4729
	0.7895	0.4893	0.4843	0.4528	0.4824	0.4835	0.4852

	0.8421	0.4992	0.4952	0.4630	0.4934	0.4945	0.4955
	0.8947	0.5063	0.5063	0.4709	0.5023	0.5032	0.5047
	0.9474	0.5106	0.5176	0.4755	0.5096	0.5101	0.5137
	1.0000	0.5120	0.5180	0.4780	0.5162	0.5163	0.5229
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0295	0.0295	0.0272	0.0279	0.0283	0.0270
	0.1053	0.0574	0.0584	0.0536	0.0553	0.0560	0.0536
	0.1579	0.0838	0.0877	0.0778	0.0814	0.0822	0.0796
	0.2105	0.1085	0.1065	0.1003	0.1058	0.1067	0.1047
	0.2632	0.1316	0.1356	0.1217	0.1287	0.1295	0.1287
	0.3158	0.1532	0.1552	0.1418	0.1497	0.1506	0.1511
	0.3684	0.1731	0.1741	0.1602	0.1690	0.1699	0.1717
	0.4211	0.1915	0.1935	0.1768	0.1867	0.1875	0.1904
	0.4737	0.2082	0.2022	0.1917	0.2029	0.2035	0.2071
	0.5263	0.2234	0.2214	0.2054	0.2177	0.2183	0.2222
	0.5789	0.2369	0.2329	0.2178	0.2312	0.2318	0.2355
	0.6316	0.2498	0.2438	0.2291	0.2436	0.2442	0.2470
	0.6842	0.2593	0.2543	0.2390	0.2546	0.2553	0.2567
	0.7368	0.2681	0.2651	0.2478	0.2642	0.2650	0.2649
	0.7895	0.2752	0.2762	0.2554	0.2720	0.2730	0.2722
	0.8421	0.2808	0.2878	0.2616	0.2779	0.2790	0.2786
	0.8947	0.2848	0.2888	0.2657	0.2816	0.2826	0.2846
	0.9474	0.2872	0.2892	0.2668	0.2834	0.2840	0.2906
	1.0000	0.2880	0.2800	0.2662	0.2845	0.2844	0.2966
9 ½	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0156	0.0196	0.0146	0.0149	0.0152	0.0146
	0.1053	0.0303	0.0383	0.0286	0.0296	0.0300	0.0290
	0.1579	0.0442	0.0472	0.0411	0.0434	0.0438	0.0429
	0.2105	0.0573	0.0556	0.0526	0.0562	0.0567	0.0561
	0.2632	0.0695	0.0645	0.0635	0.0681	0.0686	0.0684
	0.3158	0.0808	0.0838	0.0740	0.0790	0.0795	0.0799
	0.3684	0.0914	0.0924	0.0837	0.0890	0.0894	0.0904
	0.4211	0.1011	0.1021	0.0922	0.0981	0.0985	0.1002
	0.4737	0.1099	0.1039	0.0998	0.1065	0.1067	0.1092
	0.5263	0.1179	0.1149	0.1071	0.1142	0.1145	0.1176
	0.5789	0.1251	0.1251	0.1138	0.1215	0.1217	0.1253
	0.6316	0.1314	0.1364	0.1199	0.1282	0.1285	0.1323
	0.6842	0.1368	0.1378	0.1253	0.1343	0.1346	0.1385
	0.7368	0.1415	0.1485	0.1301	0.1395	0.1400	0.1437
	0.7895	0.1453	0.1493	0.1342	0.1438	0.1444	0.1478
	0.8421	0.1482	0.1482	0.1374	0.1468	0.1475	0.1511
	0.8947	0.1503	0.1573	0.1395	0.1484	0.1491	0.1539
	0.9474	0.1516	0.1566	0.1398	0.1488	0.1492	0.1565
	1.0000	0.1520	0.1550	0.1390	0.1487	0.1488	0.1592

	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.0526	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.1053	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.1579	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2105	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2632	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3158	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3684	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.4211	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.4737	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.5263	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.5789	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6316	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6842	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.7368	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.7895	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.8421	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.8947	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.9474	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

where

Station - **X** : Station numbers

WL - **Z** : Waterline depths

Original - **Y** : Half-breadth values of original mathematical form

Forward Fairing (1) - **Y** : Half-breadth values obtained from B-spline approximation

Forward Fairing (2) - **Y** : Half-breadth values obtained from B-spline fitting process

Inverse Fairing - **Y** : Half-breadth values obtained from inverse fairing process

Optimisation Approach -**Y** : Half-breadth values obtained from optimisation process

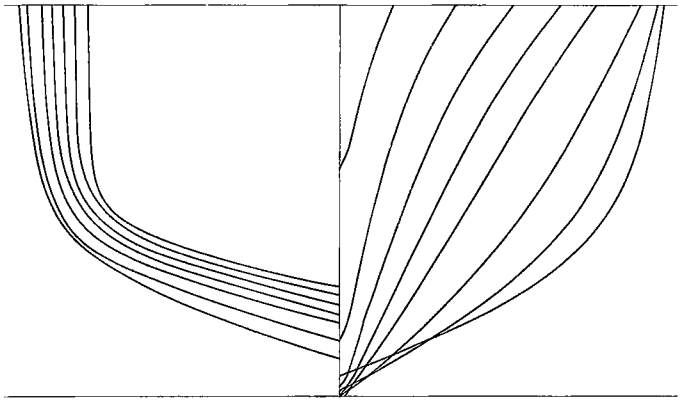


Figure B.3. NPL hull form body plan.

Table B.3. NPL form offsets obtained from two alternative forward fairing procedures.

Station No.	Waterline Depths Z (m)	Original NPL Form Offsets Y (m)	Forward Fairing (1) Y (m)	Forward Fairing (2) Y (m)
0	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0000	0.0000	0.0000
	0.0300	0.0000	0.0000	0.0000
	0.0400	0.0000	0.0000	0.0000
	0.0500	0.0000	0.0000	0.0000
	0.0600	0.0000	0.0000	0.0000
	0.0700	0.0197	0.0140	0.0151
	0.0800	0.0652	0.0579	0.0608
	0.0900	0.1026	0.0949	0.0983
	0.1000	0.1316	0.1234	0.1267
	0.1100	0.1505	0.1431	0.1460
	0.1200	0.1597	0.1552	0.1572
	0.1300	0.1645	0.1620	0.1631
	0.1400	0.1666	0.1654	0.1660
	0.1500	0.1678	0.1672	0.1675
	0.1600	0.1684	0.1680	0.1682
	0.1800	0.1691	0.1689	0.1690
	0.2000	0.1695	0.1694	0.1695
	0.2200	0.1699	0.1699	0.1699
	0.2400	0.1703	0.1703	0.1703
	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0000	0.0000	0.0000
	0.0300	0.0000	0.0000	0.0000

	0.0400	0.0000	0.0000	0.0000
	0.0500	0.0000	0.0000	0.0000
	0.0600	0.0000	0.0000	0.0000
	0.0700	0.0400	0.0347	0.0373
	0.0800	0.0824	0.0750	0.0786
	0.0900	0.1172	0.1084	0.1122
	0.1000	0.1405	0.1336	0.1368
	0.1100	0.1554	0.1510	0.1533
1/2	0.1200	0.1646	0.1620	0.1636
	0.1300	0.1705	0.1685	0.1697
	0.1400	0.1739	0.1722	0.1732
	0.1500	0.1761	0.1743	0.1752
	0.1600	0.1772	0.1754	0.1763
	0.1800	0.1780	0.1766	0.1773
	0.2000	0.1782	0.1774	0.1780
	0.2200	0.1787	0.1780	0.1785
	0.2400	0.1792	0.1785	0.1791
	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0000	0.0000	0.0000
	0.0300	0.0000	0.0000	0.0000
	0.0400	0.0000	0.0000	0.0000
	0.0500	0.0000	0.0000	0.0000
	0.0600	0.0189	0.0150	0.0161
	0.0700	0.0596	0.0551	0.0580
	0.0800	0.0984	0.0916	0.0954
	0.0900	0.1317	0.1215	0.1254
1	0.1000	0.1510	0.1437	0.1467
	0.1100	0.1639	0.1589	0.1608
	0.1200	0.1724	0.1687	0.1700
	0.1300	0.1776	0.1749	0.1760
	0.1400	0.1806	0.1788	0.1798
	0.1500	0.1824	0.1811	0.1822
	0.1600	0.1836	0.1824	0.1835
	0.1800	0.1849	0.1839	0.1849
	0.2000	0.1856	0.1850	0.1858
	0.2200	0.1864	0.1858	0.1865
	0.2400	0.1870	0.1865	0.1872
	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0000	0.0000	0.0000
	0.0300	0.0000	0.0000	0.0000
	0.0400	0.0000	0.0000	0.0000

1 ½	0.0500	0.0000	0.0000	0.0000
	0.0600	0.0443	0.0367	0.0397
	0.0700	0.0794	0.0736	0.0774
	0.0800	0.1158	0.1066	0.1113
	0.0900	0.1413	0.1333	0.1379
	0.1000	0.1587	0.1528	0.1565
	0.1100	0.1700	0.1662	0.1686
	0.1200	0.1773	0.1750	0.1767
	0.1300	0.1825	0.1808	0.1822
	0.1400	0.1862	0.1846	0.1859
	0.1500	0.1886	0.1870	0.1884
	0.1600	0.1903	0.1885	0.1899
	0.1800	0.1918	0.1904	0.1917
	0.2000	0.1929	0.1917	0.1929
	0.2200	0.1937	0.1927	0.1938
	0.2400	0.1946	0.1936	0.1946
2	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0000	0.0000	0.0000
	0.0300	0.0000	0.0000	0.0000
	0.0400	0.0000	0.0000	0.0000
	0.0500	0.0224	0.0176	0.0202
	0.0600	0.0629	0.0541	0.0598
	0.0700	0.0985	0.0885	0.0949
	0.0800	0.1306	0.1186	0.1257
	0.0900	0.1535	0.1426	0.1494
	0.1000	0.1689	0.1601	0.1657
	0.1100	0.1784	0.1720	0.1763
	0.1200	0.1849	0.1800	0.1833
	0.1300	0.1892	0.1854	0.1880
	0.1400	0.1922	0.1891	0.1914
	0.1500	0.1943	0.1916	0.1937
	0.1600	0.1959	0.1932	0.1953
	0.1800	0.1983	0.1954	0.1976
	0.2000	0.1996	0.1970	0.1992
	0.2200	0.2006	0.1983	0.2003
	0.2400	0.2015	0.1994	0.2013
	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0000	0.0000	0.0000
	0.0300	0.0000	0.0000	0.0000
	0.0400	0.0231	0.0172	0.0222
	0.0500	0.0584	0.0492	0.0583

3	0.0600	0.0923	0.0808	0.0910
	0.0700	0.1215	0.1098	0.1214
	0.0800	0.1487	0.1346	0.1462
	0.0900	0.1683	0.1543	0.1650
	0.1000	0.1802	0.1688	0.1781
	0.1100	0.1875	0.1789	0.1867
	0.1200	0.1926	0.1859	0.1925
	0.1300	0.1967	0.1909	0.1965
	0.1400	0.1995	0.1945	0.1995
	0.1500	0.2019	0.1972	0.2018
	0.1600	0.2036	0.1991	0.2035
	0.1800	0.2065	0.2020	0.2062
	0.2000	0.2084	0.2043	0.2083
	0.2200	0.2100	0.2062	0.2099
	0.2400	0.2113	0.2080	0.2113
4	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0000	0.0000	0.0000
	0.0300	0.0253	0.0164	0.0255
	0.0400	0.0582	0.0435	0.0569
	0.0500	0.0880	0.0714	0.0858
	0.0600	0.1152	0.0977	0.1127
	0.0700	0.1374	0.1211	0.1352
	0.0800	0.1567	0.1411	0.1544
	0.0900	0.1722	0.1572	0.1698
	0.1000	0.1833	0.1695	0.1811
	0.1100	0.1909	0.1786	0.1891
	0.1200	0.1962	0.1854	0.1949
	0.1300	0.2003	0.1905	0.1993
	0.1400	0.2032	0.1944	0.2026
	0.1500	0.2054	0.1976	0.2051
	0.1600	0.2072	0.2000	0.2071
	0.1800	0.2104	0.2040	0.2102
	0.2000	0.2130	0.2074	0.2127
	0.2200	0.2149	0.2103	0.2148
	0.2400	0.2166	0.2130	0.2168
	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0223	0.0124	0.0209
	0.0300	0.0480	0.0344	0.0472
	0.0400	0.0742	0.0578	0.0727
	0.0500	0.0973	0.0805	0.0962
	0.0600	0.1173	0.1013	0.1159

5	0.0700	0.1348	0.1197	0.1329
	0.0800	0.1491	0.1356	0.1479
	0.0900	0.1612	0.1489	0.1606
	0.1000	0.1713	0.1597	0.1711
	0.1100	0.1796	0.1684	0.1796
	0.1200	0.1865	0.1754	0.1864
	0.1300	0.1920	0.1811	0.1919
	0.1400	0.1966	0.1858	0.1964
	0.1500	0.2004	0.1898	0.2000
	0.1600	0.2039	0.1932	0.2030
	0.1800	0.2077	0.1989	0.2078
	0.2000	0.2124	0.2041	0.2119
	0.2200	0.2157	0.2087	0.2153
	0.2400	0.2188	0.2132	0.2185
6	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0148	0.0072	0.0132
	0.0200	0.0340	0.0228	0.0322
	0.0300	0.0525	0.0407	0.0505
	0.0400	0.0700	0.0587	0.0693
	0.0500	0.0863	0.0759	0.0857
	0.0600	0.1017	0.0916	0.1012
	0.0700	0.1157	0.1058	0.1155
	0.0800	0.1279	0.1184	0.1275
	0.0900	0.1391	0.1295	0.1384
	0.1000	0.1490	0.1391	0.1483
	0.1100	0.1584	0.1474	0.1572
	0.1200	0.1656	0.1547	0.1650
	0.1300	0.1721	0.1610	0.1718
	0.1400	0.1780	0.1666	0.1778
	0.1500	0.1831	.01716	0.1831
	0.1600	0.1879	0.1761	0.1877
	0.1800	0.1957	0.1843	0.1956
	0.2000	0.2029	0.1920	0.2027
	0.2200	0.2088	0.1994	0.2090
	0.2400	0.2145	0.2066	0.2149
	0.0000	0.0034	0.0000	0.0000
	0.0100	0.0164	0.0110	0.0151
	0.0200	0.0294	0.0233	0.0293
	0.0300	0.0418	0.0363	0.0432
	0.0400	0.0540	0.0490	0.0540
	0.0500	0.0650	0.0611	0.0650
	0.0600	0.0760	0.0724	0.0755
	0.0700	0.0864	0.0829	0.0856

7	0.0800	0.0960	0.0925	0.0955
	0.0900	0.1054	0.1015	0.1048
	0.1000	0.1141	0.1097	0.1136
	0.1100	0.1226	0.1173	0.1219
	0.1200	0.1303	0.1243	0.1297
	0.1300	0.1378	0.1308	0.1371
	0.1400	0.1447	0.1369	0.1442
	0.1500	0.1515	0.1426	0.1508
	0.1600	0.1576	0.1481	0.1571
	0.1800	0.1694	0.1587	0.1690
	0.2000	0.1813	0.1693	0.1805
	0.2200	0.1922	0.1799	0.1917
	0.2400	0.2031	0.1905	0.2027
8	0.0000	0.0034	0.0034	0.0034
	0.0100	0.0110	0.0095	0.0112
	0.0200	0.0190	0.0173	0.0197
	0.0300	0.0269	0.0254	0.0275
	0.0400	0.0344	0.0332	0.0346
	0.0500	0.0417	0.0408	0.0416
	0.0600	0.0489	0.0481	0.0487
	0.0700	0.0560	0.0552	0.0557
	0.0800	0.0627	0.0620	0.0626
	0.0900	0.0696	0.0686	0.0695
	0.1000	0.0764	0.0750	0.0763
	0.1100	0.0830	0.0813	0.0829
	0.1200	0.0896	0.0873	0.0894
	0.1300	0.0961	0.0933	0.0959
	0.1400	0.1026	0.0992	0.1024
	0.1500	0.1192	0.1050	0.1089
	0.1600	0.1157	0.1109	0.1154
	0.1800	0.1286	0.1231	0.1288
	0.2000	0.1431	0.1358	0.1429
	0.2200	0.1575	0.1492	0.1577
	0.2400	0.1731	0.1629	0.1729
	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0084	0.0061	0.0072
	0.0200	0.0137	0.0120	0.0131
	0.0300	0.0190	0.0180	0.0186
	0.0400	0.0246	0.0238	0.0245
	0.0500	0.0297	0.0294	0.0301
	0.0600	0.0350	0.0350	0.0356
	0.0700	0.0404	0.0404	0.0411
	0.0800	0.0455	0.0458	0.0464

8 ½	0.0900	0.0510	0.0511	0.0519
	0.1000	0.0564	0.0563	0.0573
	0.1100	0.0619	0.0616	0.0628
	0.1200	0.0672	0.0668	0.0683
	0.1300	0.0726	0.0721	0.0738
	0.1400	0.0784	0.0774	0.0794
	0.1500	0.0841	0.0829	0.0851
	0.1600	0.0901	0.0885	0.0910
	0.1800	0.1022	0.1004	0.1038
	0.2000	0.1165	0.1132	0.1177
	0.2200	0.1317	0.1271	0.1329
	0.2400	0.1490	0.1414	0.1488
9	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0043	0.0013	0.0016
	0.0200	0.0077	0.0050	0.0055
	0.0300	0.0113	0.0090	0.0091
	0.0400	0.0147	0.0130	0.0134
	0.0500	0.0174	0.0170	0.0175
	0.0600	0.0220	0.0210	0.0215
	0.0700	0.0256	0.0249	0.0254
	0.0800	0.0293	0.0288	0.0293
	0.0900	0.0333	0.0327	0.0333
	0.1000	0.0372	0.0366	0.0374
	0.1100	0.0414	0.0407	0.0416
	0.1200	0.0455	0.0447	0.0459
	0.1300	0.0497	0.0490	0.0502
	0.1400	0.0543	0.0533	0.0546
	0.1500	0.0577	0.0579	0.0592
	0.1600	0.0638	0.0628	0.0643
	0.1800	0.0740	0.0732	0.0754
	0.2000	0.0865	0.0847	0.0880
	0.2200	0.1004	0.0974	0.1025
	0.2400	0.1170	0.1107	0.1179
	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0000	0.0000	0.0000
	0.0300	0.0000	0.0000	0.0000
	0.0400	0.0044	0.0011	0.0011
	0.0500	0.0068	0.0035	0.0037
	0.0600	0.0093	0.0060	0.0061
	0.0700	0.0116	0.0084	0.0085
	0.0800	0.0140	0.0108	0.0111
	0.0900	0.0164	0.0133	0.0136

9 ½	0.1000	0.0189	0.0158	0.0162
	0.1100	0.0214	0.0185	0.0191
	0.1200	0.0242	0.0212	0.0220
	0.1300	0.0280	0.0242	0.0251
	0.1400	0.0299	0.0273	0.0282
	0.1500	0.0333	0.0307	0.0314
	0.1600	0.0369	0.0344	0.0351
	0.1800	0.0444	0.0423	0.0435
	0.2000	0.0537	0.0512	0.0535
	0.2200	0.0645	0.0613	0.0655
	0.2400	0.0781	0.0720	0.0785
10	0.0000	0.0000	0.0000	0.0000
	0.0100	0.0000	0.0000	0.0000
	0.0200	0.0000	0.0000	0.0000
	0.0300	0.0000	0.0000	0.0000
	0.0400	0.0000	0.0000	0.0000
	0.0500	0.0000	0.0000	0.0000
	0.0600	0.0000	0.0000	0.0000
	0.0700	0.0000	0.0000	0.0000
	0.0800	0.0000	0.0000	0.0000
	0.0900	0.0000	0.0000	0.0000
	0.1000	0.0000	0.0000	0.0000
	0.1100	0.0000	0.0000	0.0000
	0.1200	0.0000	0.0000	0.0000
	0.1300	0.0000	0.0000	0.0000
	0.1400	0.0000	0.0000	0.0000
	0.1500	0.0056	0.0015	0.0015
	0.1600	0.0082	0.0035	0.0036
	0.1800	0.0134	0.0080	0.0083
	0.2000	0.0196	0.0132	0.0139
	0.2200	0.0268	0.0196	0.0216
	0.2400	0.0360	0.0263	0.0300

where

Original - Y :Half-breadth values of original NPL hull form offsets

Forward Fairing (1) - Y :Half-breadth values obtained from B-spline approximation

Forward Fairing (2) - Y :Half-breadth values obtained from B-spline fitting process

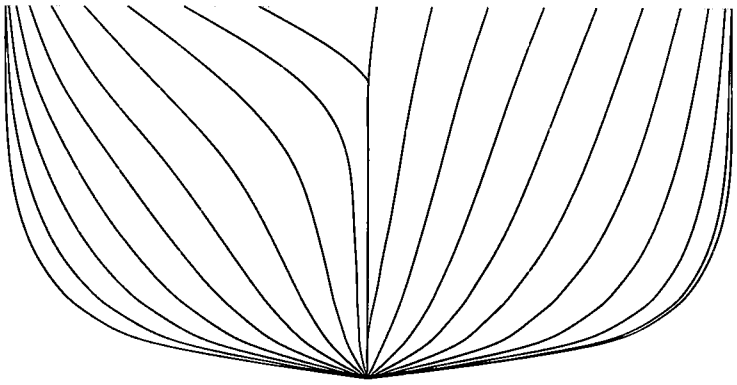


Figure B.4. Parent trawler form.

Table B.4. Trawler form offsets obtained from different distortion techniques.

Station X	WL Z (m)	Parent Y (m)	Swing Y (m)	1-C_P Y (m)	Lackenby Y (m)	Moor Y (m)	Non -Linear Y (m)
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3770	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.5030	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6285	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.7545	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.0060	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.5085	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	2.0115	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	2.5150	0.9195	0.9195	0.9195	0.9195	0.9195	0.9195
1/2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.0690	0.0822	0.0943	0.0622	0.0708	0.0690
	0.3770	0.0755	0.0930	0.1094	0.0668	0.0775	0.0755
	0.5030	0.0870	0.1086	0.1290	0.0764	0.0893	0.0870
	0.6285	0.0895	0.1145	0.1384	0.0775	0.0919	0.0895
	0.7545	0.0950	0.1229	0.1497	0.0817	0.0975	0.0950
	1.0060	0.1150	0.1493	0.1824	0.0987	0.1180	0.1150
	1.5085	0.1790	0.2340	0.2868	0.1530	0.1837	0.1790
	2.0115	0.5740	0.6672	0.7480	0.5240	0.5892	0.5740
	2.5150	1.5525	1.6325	1.6976	1.5065	1.5942	1.5525
1	0.0000	0.0000	0.0000	0.0000	0.0000	0 .0000	0.0000
	0.2515	0.1790	0.2252	0.2584	0.1599	0.1841	0.1790
	0.3770	0.2295	0.2938	0.3392	0.2025	0.2361	0.2295
	0.5030	0.2790	0.3582	0.4135	0.2454	0.2871	0.2790
	0.6285	0.3170	0.4102	0.4746	0.2771	0.3262	0.3170
	0.7545	0.3525	0.4588	0.5323	0.3070	0.3628	0.3525

	1.0060	0.4325	0.5633	0.6533	0.3764	0.4451	0.4325
	1.5085	0.6735	0.8615	0.9854	0.5896	0.6930	0.6735
	2.0115	1.2260	1.4311	1.5605	1.1307	1.2598	1.2260
	2.5150	2.0255	2.1553	2.2360	1.9644	2.0786	2.0255
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.5385	0.7023	0.8130	0.4876	0.5482	0.5385
	0.3770	0.7010	0.9008	1.0332	0.6375	0.7136	0.7010
	0.5030	0.8415	1.0710	1.2209	0.7676	0.8565	0.8415
	0.6285	0.9615	1.2155	1.3786	0.8787	0.9786	0.9615
	0.7545	1.0740	1.3442	1.5138	0.9841	1.0930	1.0740
	1.0060	1.2920	1.5867	1.7640	1.1901	1.3146	1.2920
	1.5085	1.7560	2.0527	2.2155	1.6449	1.7858	1.7560
	2.0115	2.2730	2.5016	2.6150	2.1803	2.3099	2.2730
	2.5150	2.6670	2.7986	2.8616	2.6124	2.7087	2.6670
3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	1.0660	1.3223	1.5304	0.9987	1.0803	1.0660
	0.3770	1.3310	1.6264	1.8604	1.2524	1.3480	1.3310
	0.5030	1.5505	1.8664	2.1067	1.4645	1.5694	1.5505
	0.6285	1.7270	2.0441	2.2766	1.6377	1.7469	1.7270
	0.7545	1.8695	2.1875	2.4142	1.7790	1.8902	1.8695
	1.0060	2.1185	2.4144	2.6146	2.0306	2.1396	2.1185
	1.5085	2.5110	2.7296	2.8615	2.4412	2.5309	2.5110
	2.0115	2.8005	2.9242	2.9900	2.7587	2.8175	2.8005
	2.5150	2.9590	3.0173	3.0437	2.9379	2.9737	2.9590
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	1.6320	1.8900	2.0244	1.5861	1.6418	1.6320
	0.3770	1.9720	2.2467	2.3870	1.9219	1.9827	1.9720
	0.5030	2.2170	2.4763	2.6050	2.1678	2.2275	2.2170
	0.6285	2.3815	2.6255	2.7461	2.3348	2.3915	2.3815
	0.7545	2.5135	2.7360	2.8433	2.4695	2.5229	2.5135
	1.0060	2.6995	2.8842	2.9719	2.6619	2.7076	2.6995
	1.5085	2.9130	3.0156	3.0618	2.8905	2.9181	2.9130
	2.0115	3.0120	3.0454	3.0565	3.0027	3.0145	3.0120
	2.5150	3.0500	3.0499	3.0462	3.0478	3.0511	3.0500
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	2.0255	1.9741	2.0130	2.0255	2.0255	2.0255
	0.3770	2.3895	2.3415	2.3797	2.3895	2.3895	2.3895
	0.5030	2.6065	2.5638	2.5987	2.6065	2.6065	2.6065
	0.6285	2.7450	2.6988	2.7351	2.7450	2.7450	2.7450
	0.7545	2.8385	2.7920	2.8275	2.8385	2.8385	2.8385
	1.0060	2.9620	2.9112	2.9475	2.9620	2.9620	2.9620
	1.5085	3.0500	3.0078	3.0365	3.0500	3.0500	3.0500
	2.0115	3.0500	3.0311	3.0451	3.0500	3.0500	3.0500

	2.5150	3.0500	3.0494	3.0536	3.0500	3.0500	3.0500
6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	1.7720	1.4994	1.6496	1.8915	1.8290	1.7720
	0.3770	2.1210	1.8001	1.9788	2.2542	2.1863	2.1210
	0.5030	2.3490	2.0191	2.2044	2.4808	2.4159	2.3490
	0.6285	2.4870	2.1613	2.3447	2.6168	2.5541	2.4870
	0.7545	2.5915	2.2797	2.4559	2.7147	2.6569	2.5915
	1.0060	2.7270	2.4431	2.6039	2.8397	2.7893	2.7270
	1.5085	2.8715	2.6483	2.7768	2.9559	2.9245	2.8715
	2.0115	2.9470	2.7765	2.8776	3.0026	2.9906	2.9470
	2.5150	2.9940	2.8727	2.9457	3.0320	3.0313	2.9940
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	1.1180	0.9023	1.0123	1.4157	1.2983	1.1180
	0.3770	1.3460	1.0966	1.2230	1.6999	1.5600	1.3460
	0.5030	1.5390	1.2706	1.4071	1.9141	1.7708	1.5390
	0.6285	1.6790	1.4018	1.5435	2.0569	1.9184	1.6790
	0.7545	1.8065	1.5251	1.6700	2.1787	2.0493	1.8065
	1.0060	1.9995	1.7209	1.8659	2.3502	2.2414	1.9995
	1.5085	2.2665	2.0032	2.1426	2.5719	2.5000	2.2665
	2.0115	2.4550	2.2199	2.3456	2.7144	2.6749	2.4550
	2.5150	2.6190	2.4182	2.5270	2.8261	2.8211	2.6190
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.5315	0.4393	0.4833	0.7985	0.6551	0.5315
	0.3770	0.6765	0.5707	0.6215	0.9785	0.8288	0.6765
	0.5030	0.8065	0.6857	0.7439	1.1420	0.9855	0.8065
	0.6285	0.9065	0.7749	0.8384	1.2665	1.1058	0.9065
	0.7545	1.0040	0.8622	0.9307	1.3850	1.2229	1.0040
	1.0060	1.1715	1.0135	1.0903	1.5775	1.4230	1.1715
	1.5085	1.4360	1.2629	1.3475	1.8610	1.7354	1.4360
	2.0115	1.6745	1.4945	1.5834	2.0885	2.0136	1.6745
	2.5150	1.9140	1.7363	1.8247	2.3015	2.2899	1.9140
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.1455	0.1126	0.1267	0.2891	0.1687	0.1455
	0.3770	0.2135	0.1710	0.1893	0.3929	0.2562	0.2135
	0.5030	0.2740	0.2264	0.2468	0.4795	0.3390	0.2740
	0.6285	0.3270	0.2763	0.2980	0.5493	0.4135	0.3270
	0.7545	0.3750	0.3203	0.3437	0.6169	0.4793	0.3750
	1.0060	0.4645	0.4053	0.4305	0.7346	0.6064	0.4645
	1.5085	0.6345	0.5654	0.5948	0.9478	0.8456	0.6345
	2.0115	0.8020	0.7245	0.7574	1.1522	1.0834	0.8020
	2.5150	1.0030	0.9156	0.9529	1.3840	1.3692	1.0030
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.0000	0.0000	0.0000	0.0601	0.0000	0.0000

9 ½	0.3770	0.0195	0.0117	0.0141	0.1023	0.0178	0.0195
	0.5030	0.0530	0.0428	0.0460	0.1500	0.0659	0.0530
	0.6285	0.0870	0.0746	0.0786	0.1950	0.1152	0.0870
	0.7545	0.1135	0.0992	0.1039	0.2329	0.1533	0.1135
	1.0060	0.1755	0.1575	0.1636	0.3116	0.2437	0.1755
	1.5085	0.2815	0.2565	0.2650	0.4551	0.3969	0.2815
	2.0115	0.3915	0.3595	0.3705	0.6001	0.5564	0.3915
	2.5150	0.5300	0.4923	0.5054	0.7731	0.7622	0.5300
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3770	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.5030	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6285	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.7545	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.0060	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.5085	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	2.0115	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	2.5150	0.0675	0.0675	0.0675	0.0675	0.0675	0.0675

where

Station - **X** : Station numbers

WL - **Z** : Waterline depths

Parent - **Y** : Half-breadth values of BSRA trawler parent form

Swing - **Y** : Half-breadth values obtained by swinging the sectional area curve

1-C_P - **Y** : Half-breadth values obtained by one-minus prismatic method

Lackenby - **Y** : Half-breadth values obtained by Lackenby's method

Moor - **Y** : Half-breadth values obtained by Moor's method

Non-Linear -**Y** : Half-breadth values obtained by using a non-linear distortion method

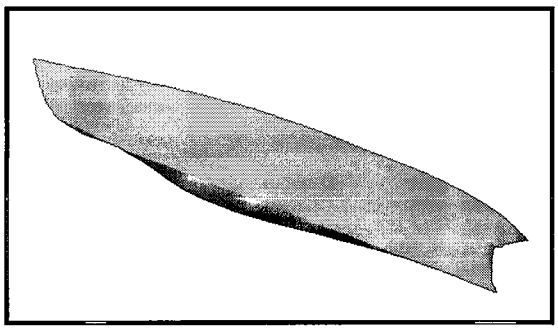


Figure B.5. Parent trawler form.

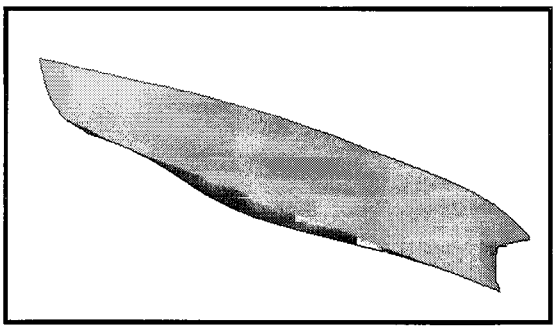


Figure B.6. Trawler form. (Swing)

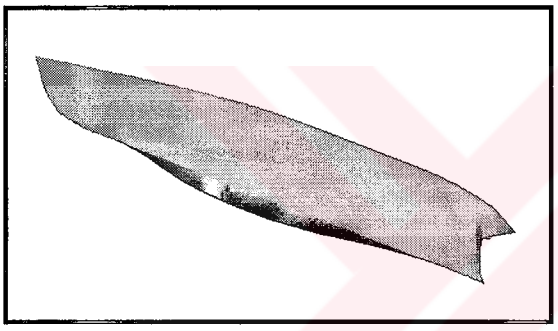


Figure B.7. Trawler form. (1-C_P)

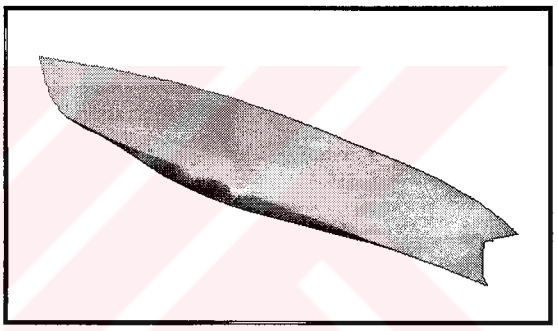


Figure B.8. Trawler form. (Lackenby)

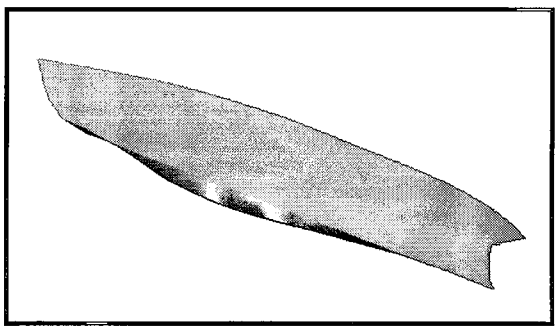


Figure B.9. Trawler form. (Moor)

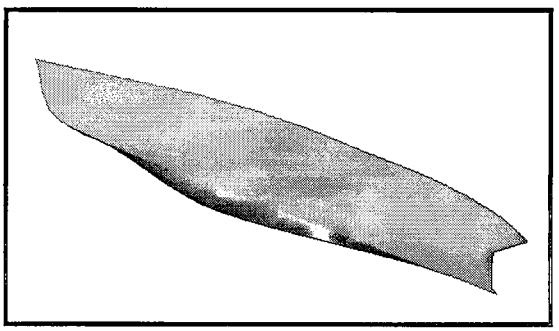


Figure B.10. Trawler form.(Non-linear)

Table B.5. Offset tables of parent, variant and final trawler forms obtained from shape averaging method.

Station X	WL Z (m)	Parent Y (m)	Variant 1 Y (m)	Variant 2 Y (m)	Variant 3 Y (m)	Variant 4 Y (m)	Final Y (m)
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3770	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.5030	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6285	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.7545	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.0060	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.5085	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	2.0115	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	2.5150	0.9195	0.9195	0.9195	0.9195	0.9195	0.9195
1/2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.0690	0.0909	0.1128	0.1346	0.1565	0.1784
	0.3770	0.0755	0.1061	0.1368	0.1674	0.1981	0.2287
	0.5030	0.0870	0.1252	0.1634	0.2016	0.2398	0.2780
	0.6285	0.0895	0.1348	0.1800	0.2253	0.2705	0.3158
	0.7545	0.0950	0.1462	0.1974	0.2487	0.2999	0.3511
	1.0060	0.1150	0.1782	0.2413	0.3045	0.3676	0.4308
	1.5085	0.1790	0.2774	0.3758	0.4742	0.5726	0.6710
	2.0115	0.5740	0.7039	0.8337	0.9636	1.0934	1.2233
	2.5150	1.5525	1.6468	1.7410	1.8353	1.9295	2.0238
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.1790	0.2252	0.2715	0.3177	0.3640	0.4102
	0.3770	0.2295	0.2914	0.3533	0.4153	0.4772	0.5391
	0.5030	0.2790	0.3536	0.4283	0.5029	0.5776	0.6522
	0.6285	0.3170	0.4033	0.4895	0.5758	0.6620	0.7483
	0.7545	0.3525	0.4501	0.5477	0.6452	0.7428	0.8404
	1.0060	0.4325	0.5506	0.6688	0.7869	0.9051	1.0232
	1.5085	0.6735	0.8294	0.9853	1.1412	1.2971	1.4530
	2.0115	1.2260	1.3834	1.5408	1.6982	1.8556	2.0130
	2.5150	2.0255	2.1231	2.2208	2.3184	2.4161	2.5137
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.5385	0.6181	0.6977	0.7772	0.8568	0.9364
	0.3770	0.7010	0.7966	0.8923	0.9879	1.0836	1.1792
	0.5030	0.8415	0.9499	1.0584	1.1668	1.2753	1.3837
	0.6285	0.9615	1.0797	1.1980	1.3162	1.4345	1.5527
	0.7545	1.0740	1.1977	1.3214	1.4451	1.5688	1.6925
	1.0060	1.2920	1.4226	1.5532	1.6838	1.8144	1.9450
	1.5085	1.7560	1.8790	2.0020	2.1251	2.2481	2.3711
	2.0115	2.2730	2.3616	2.4501	2.5387	2.6272	2.7158
	2.5150	2.6670	2.7168	2.7665	2.8163	2.8660	2.9158

3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	1.0660	1.1369	1.2078	1.2786	1.3495	1.4204
	0.3770	1.3310	1.4123	1.4936	1.5749	1.6562	1.7375
	0.5030	1.5505	1.6367	1.7230	1.8092	1.8955	1.9817
	0.6285	1.7270	1.8129	1.8988	1.9846	2.0705	2.1564
	0.7545	1.8695	1.9552	2.0409	2.1265	2.2122	2.2979
	1.0060	2.1185	2.1974	2.2762	2.3551	2.4339	2.5128
	1.5085	2.5110	2.5680	2.6251	2.6821	2.7392	2.7962
	2.0115	2.8005	2.8321	2.8638	2.8954	2.9271	2.9587
	2.5150	2.9590	2.9736	2.9882	3.0027	3.0173	3.0319
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	1.6320	1.6690	1.7060	1.7431	1.7801	1.8171
	0.3770	1.9720	2.0116	2.0512	2.0909	2.1305	2.1701
	0.5030	2.2170	2.2547	2.2924	2.3302	2.3679	2.4056
	0.6285	2.3815	2.4171	2.4526	2.4882	2.5237	2.5593
	0.7545	2.5135	2.5462	2.5789	2.6116	2.6443	2.6770
	1.0060	2.6995	2.7268	2.7542	2.7815	2.8089	2.8362
	1.5085	2.9130	2.9285	2.9440	2.9595	2.9750	2.9905
	2.0115	3.0120	3.0174	3.0228	3.0281	3.0335	3.0389
	2.5150	3.0500	3.0503	3.0506	3.0508	3.0511	3.0514
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	2.0255	2.0255	2.0255	2.0255	2.0255	2.0255
	0.3770	2.3895	2.3895	2.3895	2.3895	2.3895	2.3895
	0.5030	2.6065	2.6065	2.6065	2.6065	2.6065	2.6065
	0.6285	2.7450	2.7450	2.7450	2.7450	2.7450	2.7450
	0.7545	2.8385	2.8385	2.8385	2.8385	2.8385	2.8385
	1.0060	2.9620	2.9620	2.9620	2.9620	2.9620	2.9620
	1.5085	3.0500	3.0500	3.0500	3.0500	3.0500	3.0500
	2.0115	3.0500	3.0500	3.0500	3.0500	3.0500	3.0500
	2.5150	3.0500	3.0500	3.0500	3.0500	3.0500	3.0500
6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	1.7720	1.8047	1.8375	1.8702	1.9030	1.9357
	0.3770	2.1210	2.1571	2.1932	2.2292	2.2653	2.3014
	0.5030	2.3490	2.3844	2.4198	2.4552	2.4906	2.5260
	0.6285	2.4870	2.5219	2.5568	2.5916	2.6265	2.6614
	0.7545	2.5915	2.6245	2.6576	2.6906	2.7237	2.7567
	1.0060	2.7270	2.7572	2.7875	2.8177	2.8480	2.8782
	1.5085	2.8715	2.8940	2.9164	2.9389	2.9613	2.9838
	2.0115	2.9470	2.9613	2.9756	2.9899	3.0042	3.0185
	2.5150	2.9940	3.0036	3.0132	3.0229	3.0325	3.0421
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	1.1180	1.2065	1.2950	1.3836	1.4721	1.5606
	0.3770	1.3460	1.4514	1.5568	1.6622	1.7676	1.8730
	0.5030	1.5390	1.6502	1.7614	1.8726	1.9838	2.0950
	0.6285	1.6790	1.7905	1.9020	2.0135	2.1250	2.2365

	0.7545	1.8065	1.9156	2.0248	2.1339	2.2431	2.3522
	1.0060	1.9995	2.1015	2.2035	2.3054	2.4074	2.5094
	1.5085	2.2665	2.3536	2.4407	2.5277	2.6148	2.7019
	2.0115	2.4550	2.5278	2.6007	2.6735	2.7464	2.8192
	2.5150	2.6190	2.6760	2.7330	2.7901	2.8471	2.9041
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.5315	0.6231	0.7147	0.8063	0.8979	0.9895
	0.3770	0.6765	0.7805	0.8846	0.9886	1.0927	1.1967
	0.5030	0.8065	0.9210	1.0354	1.1499	1.2643	1.3788
	0.6285	0.9065	1.0281	1.1496	1.2712	1.3927	1.5143
	0.7545	1.0040	1.1313	1.2585	1.3858	1.5130	1.6403
	1.0060	1.1715	1.3045	1.4375	1.5704	1.7034	1.8364
	1.5085	1.4360	1.5717	1.7074	1.8432	1.9789	2.1146
	2.0115	1.6745	1.8037	1.9330	2.0622	2.1915	2.3207
	2.5150	1.9140	2.0323	2.1507	2.2690	2.3874	2.5057
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.1455	0.2001	0.2548	0.3094	0.3641	0.4187
	0.3770	0.2135	0.2801	0.3468	0.4134	0.4801	0.5467
	0.5030	0.2740	0.3508	0.4277	0.5045	0.5814	0.6582
	0.6285	0.3270	0.4106	0.4942	0.5779	0.6615	0.7451
	0.7545	0.3750	0.4660	0.5570	0.6481	0.7391	0.8301
	1.0060	0.4645	0.5671	0.6697	0.7723	0.8749	0.9775
	1.5085	0.6345	0.7523	0.8700	0.9878	1.1055	1.2233
	2.0115	0.8020	0.9321	1.0623	1.1924	1.3226	1.4527
	2.5150	1.0030	1.1413	1.2796	1.4178	1.5561	1.6944
9 ½	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.0000	0.0321	0.0642	0.0962	0.1283	0.1604
	0.3770	0.0195	0.0621	0.1047	0.1473	0.1899	0.2325
	0.5030	0.0530	0.1015	0.1499	0.1984	0.2468	0.2953
	0.6285	0.0870	0.1395	0.1920	0.2446	0.2971	0.3496
	0.7545	0.1135	0.1707	0.2278	0.2850	0.3421	0.3993
	1.0060	0.1755	0.2385	0.3015	0.3646	0.4276	0.4906
	1.5085	0.2815	0.3581	0.4347	0.5114	0.5880	0.6646
	2.0115	0.3915	0.4803	0.5690	0.6578	0.7465	0.8353
	2.5150	0.5300	0.6320	0.7341	0.8361	0.9382	1.0402
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.2515	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.3770	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.5030	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.6285	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	0.7545	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.0060	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	1.5085	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	2.0115	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	2.5150	0.0675	0.0675	0.0675	0.0675	0.0675	0.0675

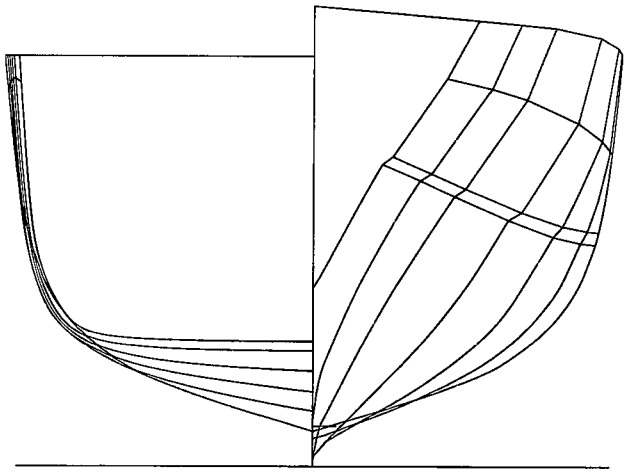


Figure B.11. Typical high-speed craft used in applications in Chapter 8.

Table B.6. Parent and final hull form offsets with computed first and second derivatives.

Waterline No.	x (m)	Parent y (m)	Faired y (m)	Parent 1st der.	Faired 1st der.	Parent 2nd der.	Faired 2nd der.
1	20.1285	0.0000	0.0000	0.2257	0.1561	-0.0651	-0.0361
	21.0000	0.1967	0.1360	0.0605	0.0645	-0.0123	-0.0124
	25.2000	0.4507	0.4068	0.0090	0.0124	-0.0100	-0.0105
	29.4000	0.4884	0.4590	-0.0331	-0.0315	-0.0064	-0.0053
	33.6000	0.3496	0.3266	-0.0600	-0.0539	0.0021	-0.0005
	37.8000	0.0975	0.1004	-0.0538	-0.0554		
	39.6131	0.0000	0.0000				
2	15.7774	0.0000	0.0000	0.3435	0.2701	-0.0696	-0.0460
	16.8000	0.3513	0.2762	0.1619	0.1499	-0.0352	-0.0274
	21.0000	1.0311	0.9057	0.0138	0.0346	-0.0130	-0.0172
	25.2000	1.0892	1.0513	-0.0407	-0.0376	-0.0085	-0.0084
	29.4000	0.9183	0.8933	-0.0764	-0.0729	-0.0035	-0.0037
	33.6000	0.5973	0.5871	-0.0910	-0.0883	0.0005	-0.0008
	37.8000	0.2152	0.2161	-0.0895	-0.0907	-0.0108	-0.0045
	39.9000	0.0272	0.0256	-0.1023	-0.0960		
	40.1661	0.0000	0.0000				
3	11.4321	0.0000	0.0000	0.5245	0.4053	-0.1164	-0.0719
	12.6000	0.6126	0.4733	0.2122	0.2122	-0.0389	-0.0339
	16.8000	1.5039	1.3647	0.0487	0.0698	-0.0173	-0.0211
	21.0000	1.7086	1.6577	-.0239	-0.0188	-0.0100	-0.0115
	25.2000	1.6082	1.5786	-.0661	-0.0673	-0.0117	-0.0096
	29.4000	1.3305	1.2960	-.1153	-0.1078	-0.0010	-0.0030
	33.6000	0.8461	0.8432	-.1194	-0.1203	-0.0035	-0.0016
	37.8000	0.3445	0.3379	-.1304	-0.1253	0.0134	0.0049
	39.9000	0.0707	0.0747	-.1121	-0.1186		
	40.5303	0.0000	0.0000				

4	7.2521 8.4000 12.6000 16.8000 21.0000 25.2000 29.4000 33.6000 37.8000 39.9000 40.8538	0.0000 1.1685 2.0814 2.2881 2.2668 2.0800 1.6802 1.0943 0.4765 0.1401 0.0000	0.0000 0.8189 1.9381 2.2500 2.2392 2.0445 1.6492 1.0890 0.4686 0.1437 0.0000	1.0179 0.2174 0.0492 -.0051 -.0445 -.0952 -.1395 -.1471 -.1602 -.1469	0.7134 0.2665 0.0743 -0.0026 -0.0463 -0.0941 -0.1334 -0.1477 -0.1547 -0.1507	-0.2994 -0.0400 -0.0129 -0.0094 -0.0121 -0.0106 -0.0018 -0.0042 0.0087	-0.1671 -0.0458 -0.0183 -0.0104 -0.0114 -0.0093 -0.0034 -0.0022 0.0026
5	2.9757 4.2000 8.4000 12.6000 16.8000 21.0000 25.2000 29.4000 33.6000 37.8000 39.9000 41.1772	0.0000 2.2503 2.5221 2.6474 2.7126 2.6263 2.4148 1.9700 1.3253 0.6118 0.2395 0.0000	0.0000 1.4489 2.4923 2.6374 2.6873 2.6054 2.3759 1.9367 1.3138 0.6073 0.2357 0.0000	1.8381 0.0647 0.0298 0.0155 -0.0205 -0.0504 -0.1059 -0.1535 -0.1699 -0.1773 -0.1875	1.1835 0.2484 0.0345 0.0119 -0.0195 -0.0546 -0.1046 -0.1483 -0.1682 -0.1770 -0.1845	-0.6539 -0.0083 -0.0034 -0.0086 -0.0071 -0.0132 -0.0113 -0.0039 -0.0023 -0.0060	-0.3448 -0.0509 -0.0054 -0.0075 -0.0084 -0.0119 -0.0104 -0.0047 -0.0028 -0.0045
6	0.0911 0.1248 4.2000 8.4000 12.6000 16.8000 21.0000 25.2000 29.4000 33.6000 37.8000 39.9000 41.4762	2.8087 2.8087 2.9462 3.0222 3.0615 3.0629 2.9343 2.6911 2.2313 1.5440 0.7498 0.3435 0.0000	2.8087 2.8097 2.9331 3.0161 3.0552 3.0412 2.9152 2.6550 2.1933 1.5262 0.7471 0.3338 0.0000	0.0000 0.0337 0.0181 0.0094 0.0003 -0.0306 -0.0579 -0.1095 -0.1636 -0.1891 -0.1934 -0.2179	0.0275 0.0303 0.0198 0.0093 -0.0033 -0.0300 -0.0620 -0.1099 -0.1589 -0.1855 -0.1968 -0.2117	0.0164 -0.0038 -0.0021 -0.0021 -0.0074 -0.0065 -0.0123 -0.0129 -0.0061 -0.0014 -0.0133	0.0013 -0.0025 -0.0025 -0.0030 -0.0063 -0.0076 -0.0114 -0.0117 -0.0063 -0.0036 -0.0081

7	0.0420	3.0576	3.0576	0.0000	0.0175	0.0117	0.0019
	0.1248	3.0576	3.0590	0.0242	0.0214	-0.0029	-0.0018
	4.2000	3.1564	3.1462	0.0122	0.0138	-0.0011	-0.0016
	8.4000	3.2074	3.2041	0.0074	0.0072	-0.0014	-0.0023
	12.6000	3.2384	3.2342	0.0014	-0.0026	-0.0071	-0.0061
	16.8000	3.2442	3.2234	-0.0284	-0.0281	-0.0067	-0.0078
	21.0000	3.1251	3.1053	-0.0566	-0.0607	-0.0126	-0.0117
	25.2000	2.8875	2.8504	-0.1095	-0.1099	-0.0132	-0.0123
	29.4000	2.4277	2.3887	-0.1650	-0.1616	-0.0084	-0.0080
	33.6000	1.7348	1.7101	-0.2002	-0.1953	-0.0023	-0.0058
	37.8000	0.8941	0.8897	-0.2075	-0.2135	-0.0199	-0.0122
	39.9000	0.4583	0.4412	-0.2469	-0.2377		
	41.7562	0.0000	0.0000				
8	-0.0063	3.2815	3.2815	0.0291	0.0202	-0.0062	-0.0027
	0.1248	3.2853	3.2841	0.0161	0.0146	-0.0021	-0.0014
	4.2000	3.3508	3.3435	0.0074	0.0089	-0.0003	-0.0008
	8.4000	3.3817	3.3807	0.0059	0.0055	-0.0010	-0.0018
	12.6000	3.4065	3.4037	0.0019	-0.0020	-0.0065	-0.0057
	16.8000	3.4144	3.3954	-0.0253	-0.0258	-0.0072	-0.0080
	21.0000	3.3082	3.2871	-0.0555	-0.0592	-0.0126	-0.0119
	25.2000	3.0753	3.0384	-0.1083	-0.1090	-0.0136	-0.0130
	29.4000	2.6206	2.5805	-0.1655	-0.1635	-0.0108	-0.0097
	33.6000	1.9255	1.8938	-0.2107	-0.2044	-0.0033	-0.0073
	37.8000	1.0405	1.0354	-0.2212	-0.2272	-0.0199	-0.0157
	39.9000	0.5761	0.5582	-0.2630	-0.2601	-0.4499	-0.1074
	42.0000	0 0237	0.0120	-0.7426	-0.3746		
	42.0320	0.0000	0.0000				
9	-0.0497	3.3589	3.3589	0.0290	0.0200	-0.0065	-0.0029
	0.1248	3.3639	3.3624	0.0151	0.0139	-0.0019	-0.0013
	4.2000	3.4257	3.4189	0.0071	0.0085	-0.0003	-0.0007
	8.4000	3.4554	3.4547	0.0060	0.0055	-0.0010	-0.0015
	12.6000	3.4806	3.4778	0.0020	-0.0008	-0.0049	-0.0047
	16.8000	3.4888	3.4743	-0.0187	-0.0207	-0.0077	-0.0079
	21.0000	3.4102	3.3875	-0.0512	-0.0539	-0.0116	-0.0115
	25.2000	3.1952	3.1612	-0.0997	-0.1024	-0.0153	-0.0140
	29.4000	2.7762	2.7313	-0.1640	-0.1611	-0.0112	-0.0106
	33.6000	2.0875	2.0547	-0.2109	-0.2056	-0.0070	-0.0088
	37.8000	1.2017	1.1910	-0.2331	-0.2334	-0.0129	-0.0200
	39.9000	0.7123	0.7007	-0.2601	-0.2754	-0.2632	-0.1244
	42.0000	0.1660	0.1225	-0.5746	-0.4240		
	42.2889	0.0000	0.0000				

10	-0.0930 0.1248 4.2000 8.4000 12.6000 16.8000 21.0000 25.2000 29.4000 33.6000 37.8000 39.9000 42.0000 42.5458	3.4363 3.4364 3.4937 3.5201 3.5440 3.5559 3.5022 3.3104 2.9277 2.2410 1.3582 0.8488 0.3026 0.0000	3.4363 3.4382 3.4872 3.5197 3.5420 3.5449 3.4792 3.2786 2.8770 2.2083 1.3425 0.8413 0.2406 0.0000	0.0003 0.0141 0.0063 0.0057 0.0028 -0.0128 -0.0457 -0.0911 -0.1635 -0.2102 -0.2426 -0.2601 -0.5544	0.0087 0.0120 0.0077 0.0053 0.0007 -0.0157 -0.0478 -0.0956 -0.1592 -0.2061 -0.2387 -0.2860 -0.4408	0.0064 -0.0019 -0.0001 -0.0007 -.00037 -0.0078 -0.0108 -0.0172 -0.0111 -0.0103 -0.0083 -0.2225	0.0016 -0.0010 -0.0006 -0.0011 -0.0039 -0.0076 -0.0114 -0.0151 -0.0112 -0.0103 -0.0226 -0.1170
11	-0.1421 0.1248 4.2000 8.4000 12.6000 16.8000 21.0000 25.2000 29.4000 33.6000 37.8000 39.9000 42.0000 42.8059	3.4728 3.4728 3.5285 3.5564 3.5776 3.5923 3.5649 3.4314 3.1104 2.3917 1.5240 0.9937 0.4381 0.0000	3.4728 3.4749 3.5226 3.5553 3.5765 3.5853 3.5472 3.4001 3.0556 2.4130 1.5097 0.9886 0.3635 0.0000	0.0000 0.0137 0.0066 0.0050 0.0035 -0.0065 -0.0318 -0.0764 -0.1711 -0.2066 -0.2525 -0.2646 -0.5436	0.0079 0.0117 0.0078 0.0050 0.0021 -0.0091 -0.0350 -0.0820 -0.1530 -0.2151 -0.2482 -0.2976 -0.4511	0.0063 -0.0017 -0.0004 -0.0004 -0.0024 -0.0060 -0.0106 -0.0225 -0.0085 -0.0146 -0.0058 -0.1920	0.0018 -0.0009 -0.0007 -0.0007 -0.0027 -0.0062 -0.0112 -0.0169 -0.0148 -0.0105 -0.0236 -0.1056
12	-0.1913 0.1248 4.2000 8.4000 12.6000 16.8000 21.0000 25.2000 29.4000 33.6000 37.8000 39.9000 42.0000 43.0661	3.5088 3.5088 3.5628 3.5924 3.6107 3.6283 3.6191 3.5179 3.2355 2.6569 1.6899 1.1424 0.5736 0.0000	3.5088 3.5111 3.5576 3.5905 3.6106 3.6238 3.6037 3.4877 3.1861 2.5922 1.6751 1.1381 0.4903 0.0000	0.0000 0.0133 0.0070 0.0044 0.0042 -0.0022 -0.0241 -0.0672 -0.1377 -0.2302 -0.2607 -0.2709 -0.5380	0.0074 0.0114 0.0078 0.0048 0.0031 -0.0048 -0.0276 -0.0718 -0.1414 -0.2184 -0.2557 -0.3084 -0.4599	0.0060 -0.0015 -0.0006 0.0000 -0.0015 -0.0052 -0.0103 -0.0168 -0.0220 -0.0097 -0.0048 -0.1687	0.0018 -0.0009 -0.0007 -0.0004 -0.0019 -0.0054 -0.0105 -0.0166 -0.0183 -0.0119 -0.0251 -0.0957

13	-0.2384 0.1248 4.2000 8.4000 12.6000 16.8000 21.0000 25.2000 29.4000 33.6000 37.8000 39.9000 42.0000 43.3311	3.5356 3.5356 3.5916 3.6192 3.6394 3.6550 3.6582 3.5832 3.3486 2.8153 1.9358 1.3011 0.7190 0.0000	3.5356 3.5383 3.5856 3.6180 3.6386 3.6530 3.6452 3.5566 3.2988 2.7492 1.8579 1.3015 0.6265 0.0000	0.0000 0.0138 0.0066 0.0048 0.0037 0.0007 -0.0179 -0.0559 -0.1270 -0.2094 -0.3023 -0.2772 -0.5402	0.0074 0.0116 0.0077 0.0049 0.0034 -0.0019 -0.0211 -0.0614 -0.1309 -0.2122 -0.2649 -0.3214 -0.4707	0.0062 -0.0017 -0.0004 -0.0003 -0.0007 -0.0044 -0.0090 -0.0169 -.00196 -0.0295 0.0119 -0.1533	0.0019 -0.0009 -0.0007 -0.0004 -0.0013 -0.0046 -0.0096 -0.0165 -0.0194 -0.0167 -0.0269 -0.0870
14	-0.2852 0.1248 4.2000 8.4000 12.6000 16.8000 21.0000 25.2000 29.4000 33.6000 37.8000 39.9000 42.0000 43.5971	3.5605 3.5605 3.6193 3.6441 3.6671 3.6799 3.6941 3.6440 3.4592 2.9729 2.1073 1.5589 0.8755 0.0000	3.5605 3.5636 3.6121 3.6438 3.6654 3.6801 3.6834 3.6216 3.4089 2.9076 2.0668 1.5137 0.7816 0.0000	0.0000 0.0144 0.0059 0.0055 0.0031 0.0034 -0.0119 -0.0440 -0.1158 -0.2061 -0.2611 -0.3254 -0.5482	0.0075 0.0119 0.0075 0.0051 0.0035 0.0008 -0.0147 -0.0506 -0.1194 -0.2002 -0.2634 -0.3487 -0.4894	0.0064 -0.0021 -0.0001 -0.0006 0.0001 -0.0036 -0.0076 -0.0171 -0.0215 -0.0175 -0.0306 -0.1205	0.0020 -0.0011 -0.0006 -0.0004 -0.0007 -0.0037 -0.0085 -0.0164 -0.0192 -0.0201 -0.0406 -0.0761
15	-0.3328 0.1248 4.2000 8.4000 12.6000 16.8000 21.0000 25.2000 29.4000 33.6000 37.8000 39.9000 42.0000 43.8604	3.5802 3.5800 3.6405 3.6652 3.6891 3.7026 3.7221 3.6869 3.5659 3.1292 2.2814 1.7350 1.1038 0.0000	3.5802 3.5835 3.6330 3.6651 3.6874 3.7036 3.7130 3.6726 3.5133 3.0607 2.2531 1.7178 0.9816 0.0000	-0.0003 0.0148 0.0059 0.0057 0.0032 0.0046 -0.0084 -0.0288 -0.1040 -0.2019 -0.2602 -0.3006 -0.5933	0.0073 0.0121 0.0076 0.0053 0.0039 0.0022 -0.0096 -0.0379 -0.1078 -0.1923 -0.2549 -0.3506 -0.5276	0.0067 -0.0022 0.0000 -0.0006 0.0003 -0.0031 -0.0049 -0.0179 -0.0233 -0.0185 -0.0192 -0.1478	0.0021 -0.0011 -0.0006 -0.0003 -0.0004 -0.0028 -0.0067 -0.0166 -0.0201 -0.0199 -0.0456 -0.0894

16	-0.3808	3.5972	3.5972	-0.0007	0.0070	0.0069	0.0023
	0.1248	3.5969	3.6008	0.0151	0.0123	-0.0022	-0.0011
	4.2000	3.6586	3.6510	0.0062	0.0079	-0.0001	-0.0006
	8.4000	3.6844	3.6841	0.0057	0.0055	-0.0005	-0.0003
	12.6000	3.7083	3.7070	0.0038	0.0043	0.0003	-0.0003
	16.8000	3.7243	3.7253	0.0052	0.0029	-0.0029	-0.0025
	21.0000	3.7462	3.7376	-0.0070	-0.0078	-0.0040	-0.0053
	25.2000	3.7167	3.7050	-0.0236	-0.0301	-0.0132	-0.0142
	29.4000	3.6174	3.5786	-0.0792	-0.0897	-0.0283	-0.0227
	33.6000	3.2850	3.2016	-0.1982	-0.1852	-0.0188	-0.0215
	37.8000	2.4525	2.4237	-0.2575	-0.2530	-0.0215	-0.0495
	39.9000	1.9118	1.8925	-0.3026	-0.3569	-0.1415	-0.0860
	42.0000	1.2762	1.1429	-0.6013	-0.5385		
	44.1224	0.0000	0.0000				
17	-0.4288	3.6143	3.6143	-0.0011	0.0067	0.0071	0.0025
	0.1248	3.6137	3.6180	0.0154	0.0125	-0.0022	-0.0011
	4.2000	3.6766	3.6690	0.0064	0.0081	-0.0002	-0.0006
	8.4000	3.7037	3.7031	0.0057	0.0056	-0.0003	-0.0002
	12.6000	3.7276	3.7266	0.0044	0.0048	0.0003	-0.0003
	16.8000	3.7459	3.7469	0.0058	0.0036	-0.0027	-0.0024
	21.0000	3.7702	3.7622	-0.0057	-0.0062	-0.0035	-0.0049
	25.2000	3.7464	3.7360	-0.0205	-0.0269	-0.0126	-0.0130
	29.4000	3.6602	3.6231	-0.0735	-0.0813	-0.0238	-0.0202
	33.6000	3.3514	3.2816	-0.1733	-0.1661	-0.0259	-0.0254
	37.8000	2.6236	2.5840	-0.2548	-0.2461	-0.0238	-0.0555
	39.9000	2.0885	2.0672	-0.3047	-0.3626	-0.1351	-0.0826
	42.0000	1.4486	1.3058	-0.6075	-0.5477		
	44.3843	0.0000	0.0000				
18	-0.4768	3.6313	3.6313	-0.0013	0.0065	0.0073	0.0027
	0.1248	3.6305	3.6352	0.0157	0.0127	-0.0022	-0.0011
	4.2000	3.6946	3.6871	0.0067	0.0084	-0.0002	-0.0006
	8.4000	3.7229	3.7222	0.0057	0.0057	-0.0002	-0.0001
	12.6000	3.7468	3.7463	0.0049	0.0053	0.0003	-0.0002
	16.8000	3.7676	3.7685	0.0064	0.0043	-0.0025	-0.0022
	21.0000	3.7942	3.7868	-0.0043	-0.0047	-0.0031	-0.0045
	25.2000	3.7761	3.7669	-0.0174	-0.0237	-0.0121	-0.0123
	29.4000	3.7030	3.6673	-0.0683	-0.0754	-0.0223	-0.0187
	33.6000	3.4160	3.3506	-0.1618	-0.1541	-0.0216	-0.0237
	37.8000	2.7364	2.7033	-0.2300	-0.2287	-0.0339	-0.0626
	39.9000	2.2535	2.2231	-0.3012	-0.3602	-0.1312	-0.0818
	42.0000	1.6209	1.4667	-0.6125	-0.5543		
	44.6463	0.0000	0.0000				

19	-0.5248	3.6484	3.6484	-0.0016	0.0063	0.0075	0.0028
	0.1248	3.6473	3.6525	0.0160	0.0129	-0.0022	-0.0010
	4.2000	3.7127	3.7051	0.0070	0.0086	-0.0003	-0.0006
	8.4000	3.7421	3.7412	0.0057	0.0059	0.0000	0.0000
	12.6000	3.7660	3.7659	0.0055	0.0058	0.0003	-0.0002
	16.8000	3.7892	3.7902	0.0069	0.0050	-0.0024	-0.0020
	21.0000	3.8183	3.8114	-0.0030	-0.0032	-0.0027	-0.0041
	25.2000	3.8058	3.7979	-0.0143	-0.0205	-0.0116	-0.0118
	29.4000	3.7457	3.7116	-0.0631	-0.0700	-0.0215	-0.0182
	33.6000	3.4806	3.4174	-0.1535	-0.1466	-0.0225	-0.0234
	37.8000	2.8361	2.8017	-0.2243	-0.2204	-0.0292	-0.0626
	39.9000	2.3651	2.3389	-0.2856	-0.3518	-0.1283	-0.0792
	42.0000	1.7652	1.6001	-0.6070	-0.5502		
	44.9083	0.0000	0.0000				

where

Waterline No. : Waterline numbers

X : Longitudinal distance of the waterlines

Parent - Y : Half-breadth values of the corresponding waterlines for the parent high speed semi displacement form

Faired - Y : Half-breadth values of the corresponding waterlines obtained by using forward fairing procedure

Parent 1st der. : First derivative values of the parent hull form

Faired 1st der. : First derivative values of the faired hull form

Parent 2nd der. : Second derivative values of the parent hull form

Faired 2nd der. : Second derivative values of the faired hull form

RESUME

Ebru Narlı was born in April 11, 1972 in İstanbul. She graduated from İstanbul Technical University, Faculty of Naval Architecture and Ocean Engineering as a Naval Architect and Ocean Engineer in 1993. She received her M.Sc. degree in Ocean Engineering in the same Faculty in 1995. She is currently employed as a Research Assistant in the Faculty of Naval Architecture and Ocean Engineering at ITU.