THE APPLICATION OF A COMPUTER TO THE MATHEMATICAL DEFINITION OF SHIP HULL FORMS

Michael Grossman

ABSTRACT

The work described in this thesis was performed in the context of a project to improve ship design by the application of computers. The particular aspect which is treated here is the design of the hull-form.

The first part of the work describes some of the problems which arise in bringing a hull-form from the preliminary design stages to a form suitable for building. A brief description of manual methods is followed by a review of several past attempts to perform the same task numerically, using computers. The main difficulty is shown to be the wide range of slopes and curvatures of hull-forms. The drawbacks of current computer-methods are pointed out, thus indicating the reasoning behind the author's approach to the problem.

The second part of the work treats some conventional methods of hull-surface description, namely least-squares polynomials and the division of the hull into geometrically distinct regions. These methods are analysed rigourously, and the difficulties encountered in their use are indicated. Several improvements and variations are proposed, and their results are compared with those of the original methods.

The third part treats basic linear and non-linear transformations of the hull-surface into different reference systems. It is shown that many of the difficulties of curve description may be overcome by rotation and by transformation to polar co-ordinates about a uniquely defined pole. It is, however, also shown that, despite considerable improvement over the cartesian co-ordinate system, both transformations are inadequate for practical shipbuilding.

In the final part, a different concept of 'similarity' between a graphically defined form and the corresponding mathematical definition is explored. The new concept, of minimising the shortest, i.e. normal, distances from the mathematical definition to the original data is shown to lead to very powerful curve and surface-fitting methods. Further, the latter is shown to be quite close to the long-sought complete mathematical description of a hull.

The work refers almost exclusively to ships' hulls, but it is clear that hulls belong to a large class of engineering surfaces, to which the analysis is equally applicable. For illustration, the various methods treated in this work are tested by applying them to two hull-forms.

Acknowledgements

The work described in this thesis was performed under a contract to study the use of computers in ship design, placed with Imperial College jointly by the Ministry of Technology (later the Department of Trade and Industry) and the British Ship Research Association. I should like to thank the sponsors for supporting this research work, and for their permission to publish it.

Next, I would like to thank the University of London and IBM who, via the Imperial College Computing Centre, have provided most of the computing facilities needed for this work; also, the various members of the Centre, for the consistently good computing service.

Lastly, I am indebted to my supervisor, Prof. W.S. Elliott, for his constant encouragement; to J.G. Hayes of the National Physical Laboratory, for his guidance in the early stages of the work; and to K.J. MacCallum, for many stimulating discussions throughout the work.

Contents

		Page
Absract		2
Acknowledgements		3
Chapter 1	Introduction	6
	Foreword	7
	Definition of 'Fairness'	8
	Errors in the Data	11
	The Universal Hull Function	12
	Aim of this Work	13
	Outline of the Work	15
	Co-ordinate System and Notation	17
	Data used	17
	Computing Equipment and Language	17
Chapter 2	Past and Present Methods of Mathematical Hull Definition	20
	A Review of Mathematical Hull Definition Methods	21
	A Classification of Mathematical Hull Definition and Fairing Methods	22
	A Typical Division of the Hull Surface and a Geometrical Classification of the Regions	27
	The Fairing Procedure with Contemporary Computer-aided Methods	27
Chapter 3	Least-Squares Curve and Surface Fitting	31
	Introduction	32
	Curves	32
	Surfaces	50
	Discussion	60
Chapter 4	Mathematical Division of the Hull-Surface	61
	Introduction	62
	The Quadratic Method	65
	The Cubic Method	66
	The Pseudo-Curvature Method	69
	Spline Methods	71
	Results	79
	Discussion and Conclusions	86
Chapter 5	Transformations	89
	Introduction	90
	Rotation	92
	Polar Transformation	99
	Conclusions	108
Chapter 6	Parametric Curve Fitting	109
	Introduction	110
	Statement of the Problem and Method of Solution	112
	Computational Techniques	115
	Results	119
	Possible Improvements to the Algorithm	128
	Discussion and Conclusions	130

Chapter 7	Parametric Surface Fitting	132
	Introduction	133
	Statement of the Problem and a Possible Solution	135
	Computational Techniques	146
	Results	164
	A Possible Improvement	170
	Discussion and Conclusions	170
Chapter 8	Conclusions	172
	Summary	173
	Future Work and Recommendations	173
Appendix A	Data for the Experiments	177
Appendix B	Minimum Distance from a Point to a Plane Curve	193
Appendix C	Approximation to a Subset of Statistical F-Tables	196
Appendix D	The Derivation of Spline Functions	199
Appendix E	Properties of the 'Flatness' Function of Polar Transformation	203
Appendix F	Univariate Minimisation by Quadratic Interpolation, without Gradients	205
References		207
Bibliography		212

CHAPTER 1 :-

INTRODUCTION

1.1 Foreword

There are several strong reasons for defining a ship hull-form mathematically. Some of these are:

a) The simplification of hydrodynamic analysis.

b) A true mathematical definition of a hull is invariant under scaling, i.e. scaling it up, for building purposes, does not create unwanted discrepencies. This essentially puts the mathematical definition on a par with full-scale lofting.

c) Mathematical shell-plate develoment and the use of numerically controlled cutting and welding machinery ensue, with little additional effort.

d) The mathematical definition of a parent form can be distorted analytically to a new desired form. This is valuable for designing new ships which are based on successful previous designs.

e) A variety of lesser reasons; for instance, more accurate and convenient hydrostatic analysis, reduced space requirements and increased speed of lofting.

These reasons have attracted researchers to this subject for several decades, but the advent of computers brought the subject out of its infancy. The recent parallel development of computers, numerical analysis and numerically controlled tools have brought about a reformation of concepts. The reformation has been catalysed by increased competitiveness in shipbuilding, and increases in the prices of raw materials and labour. Speed of design and the minimisation of scrap are more important now than ever before.

In parallel with the development of power-ships in the last century, building regulations and traditions have also evolved. The regulations, which are mostly for safety, have constrained many aspects of ship design, and much effort has been devoted to making the corresponding computer methods conform. Tradition has also exerted a considerable, and

not always justified, influence on certain elements of ship design. Until recently, mathematical ship lofting had been such a subject.

On consideration of the literature on the mathematical definition of hull forms, much similarity of concept between different methods may be noticed. For example, most of the methods use polynomials as the lofting function, and the least-squares criterion as the measure of the similarity between the mathematical definition and the initial design; also, all the methods known to the author employ the cartesian co-ordinate system, with the occasional, localised, non-linear transformation. A particular function, goodness-of-fit criterion and co-ordinate system may be demanded theoretically for some applications, but not for hull forms.

Some of the author's earlier work falls into the category described above. The author is not, in fact, a naval architect; therefore, once the shortcomings of the earlier work were recognised, the investigation proceeded with broader applications in mind than just ships. The subsequent work is quite general, with ships serving mainly to illustrate 'difficult' engineering surfaces.

1.2 Definition of 'Fairness'

The concept of 'fairness' of ships' lines is largely subjective; so much so, in fact, that it is unusual for several naval architects and loftsmen to unanimously find a particular set of lines 'fair'. Before attempting to define hulls mathematically, which implies fairing, the concept must be defined quantitatively.

At this point, it is convenient to refer briefly to the manual, or graphical, fairing process. Normal practice is to start with a body-plan, which is drawn to a scale of the order of 1:50. It is drawn in the usual engineering convention, of two main orthogonal views - cross-sections and waterlines - with a third, confirmatory, view - buttocks;

other information, which is optional but which is often given, is that of diagonals and 'flat-of-side' curves, the former being projections unto non-principal planes, and the latter defining the region on the hull surface where the cross-section does not vary. Figure 1.1 is an illustration of a typical body-plan, with examples of all these views. The plan is often derived from a successful previous design. Data, generally known as 'offsets', is read off the plan, and checked carefully for reading and transcription errors. The offsets are then scaled up; modern practice is to scale up to 1/10th of full scale, but until about 10 years ago, full scale had been used in most European shipyards. The scaled offsets are drawn on a large, flat surface - the 'mould loft' - and it is then that the fairing, or lofting, process proper commences. This consists of smoothing the offsests by means of wooden battens, or splines, in such a way that all the principal views remain smooth simultaneously, i.e. in such a way that no unwanted bulges or hollows are created in any of the views. It is in this "lines' fairing" procedure that the subjective opinion of the loftsman is exercised.

Consideration of the graphical method and of various computer-aided fairing methods has led to the following definition, which seems to be a good compromise between mathematical rigour and acceptability to the loftsman. It is taken from Berger (Berger et al., 1966), but extended to surfaces, and consists of three conditions which the 'faired' curve or surface must satisfy:

a) The curve or surface must be continuous up to and including the the second derivative, i.e. it must possess continuity of value, slope and curvature.

b) The curve or surface must be completely free of extraneous inflexion points, while possessing those inherent in the data.

c) Deviations from the scaled offsets must be as small as possible,

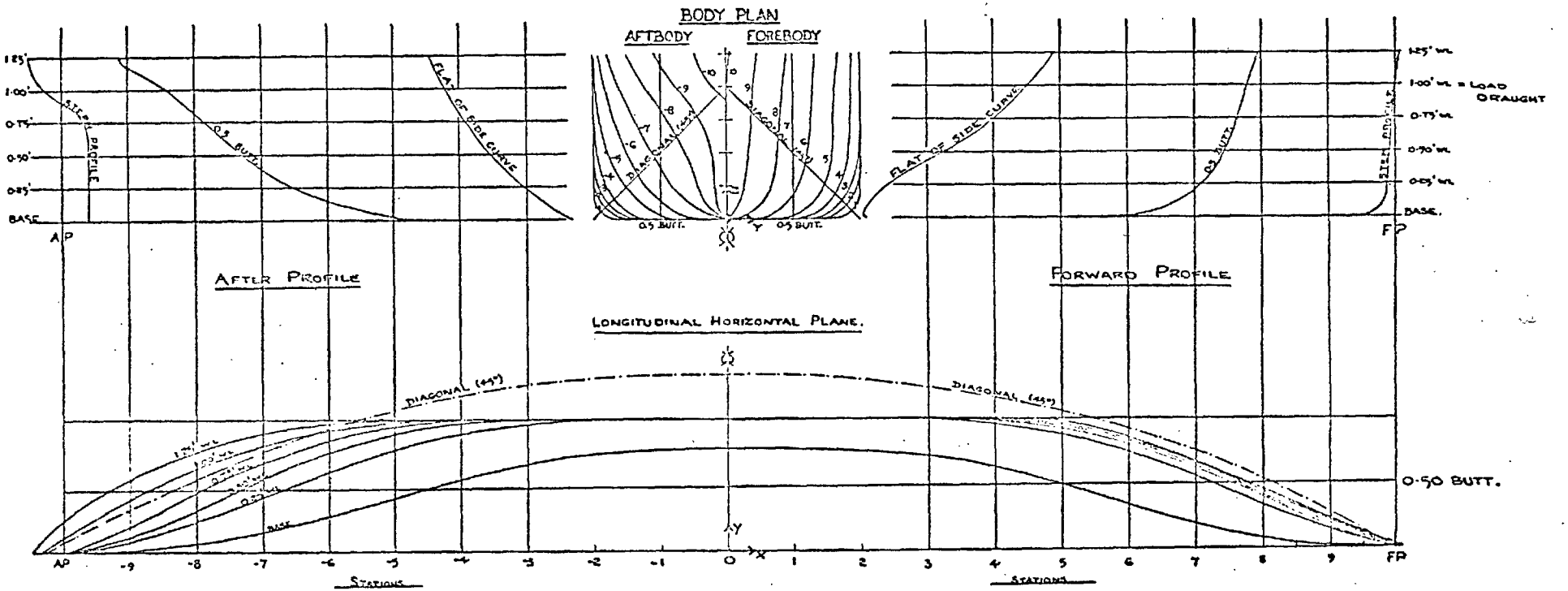


Figure 1.1 :- A typical Lines' Drawing

consistent with the satisfaction of conditions (a) and (b).

The proviso must be added that if knuckles are present, the fairness criteria shall apply only to those sections of the curve or surface that lie between the knuckles.

Condition (c), while being the easiest to quantify, has led to a serious controversy, namely: What is the maximum deviation between the scaled offsets and the mathematical definition , that should be allowed for practical fairing ? To the author's knowledge, this problem has not been satisfactorily resolved yet. At one time, it was believed that accuracies as high as 1/4" in the full scale plan were necessary. This implies that the data must be read off the 1:50 body-plan to the remarkable accuracy of half the thickness of a pencil line ! Further, it will be shown in this thesis that the value of the maximum deviation, measured in the conventional way, is not necessarily a reliable measure of the quality of the mathematical definition.

1.3 Errors in the Data

There are two sorts of errors in the data, which must be considered in any fairing system, whether it be manual or mathematical, namely:

a) The difference between the preliminary body-plan, and the corresponding faired form, which the naval architect has visualised. This difference includes the measurement error which is inherent in any measuring device, and is small, by definition.

b) 'Human' errors, such as mistakes of measurement and transcription; generally relatively large in magnitude.

The last are assumed to have been eliminated before the fairing method proper is applied. For manual fairing, these errors are removed by very careful cross-checks of the data. As this is tedious and time-consuming, it is only reasonable that computer-aided fairing methods should be preceded by computerised error-checking sequences. These do

indeed exist, and are generally simple to implement. In fact, many of the simpler errors can be rectified, at least approximately, using a slight extrapolation of the basic definition of fairing; usually, only borderline cases need be referred to the designer for confirmation. Such a feature must implicitly be included in any practical lines' fairing system.

Throughout this work, it is assumed that the data is free from gross errors, i.e. type (b), although most of the fairing methods described will be seen not to be susceptible to these errors. Whenever type (b) errors are considered, it is clear from the context.

1.4 The Universal Hull Function

The fundamental aspect of mathematical fairing is the existence of a universal function that, with different parameters, is capable of describing any hull. Earlier work in this field consisted of seeking such a function, and limited success has been achieved. However, it was soon discovered that even the simplest hull-surface consists of several regions of distinct geometrical properties, so that this universal function, if it existed, would have to be extremely complex.

There is a class of empirically derived functions, which relate some of the hull parameters, e.g. the block coefficient, to the performance required of the vessel (e.g. Watson, 1962). These functions are of considerable practical interest, but they do not convey enough information about the hull form.

It is now generally accepted that, for practical mathematical fairing, the hull surface must be divided into several parts, each to be faired separately, but in such a way that the combined surface is still continuous in value, slope and curvature. From the shipbuilder's standpoint, this is an equally valid and useful form definition.

1.5 Aim of this Work

The intention of computer-aided design is to relieve the designer from the tedium of performing monotonous calculations, and from having to use intuition to make decisions that can be deduced mathematically. In other words, a practical lines' fairing system should, as much as possible, assess its own results, and decide whether they qualify for the designer's attention.

The main drawback of systems which make their own decisions is that they are too rigid. That is to say that either because of their complex structure, or by design, they leave the user little room for experimentation.

The other, currently prevalent, approach is to simply provide the naval architect with a basic fairing package, and to leave the rest to him. This suffers from the disadvantage, that unless the user is completely familiar with the mathematics of the method, alterations to the input data in order to achieve a specific change in the results must be made on a trial and error basis.

The ideal, effective, lines' fairing system lies between the two extremes described above. Some of its most desirable properties are summarised in the following:

a) The input to the system must be simple, and in a form which is familiar to the naval architect.

b) The output from the system should be clear, preferably graphical, and, again, in a form familiar to the user.

c) If left to its own devices, the system should make competent decisions about basic aspects, and it should inform the user of these decisions. This includes the correction of obvious user errors.

d) It should be easy to override the built-in decision mechanism, or to modify it.

e) Expert knowledge of the mathematical background of the system

should not be needed to operate it successfully. Awareness of the basic principles involved is as much as should be expected of the user. This implies a clear relationship between any change to the input data and the resulting change in the output.

f) One of the unfortunate features of computer-aided design systems, which is often dismissed in theoretical works, is that practically every system is computer-dependent to some degree. It is the author's conviction that complete machine independence is some time away yet, but that any practical system should strive towards this ideal. There seems little point in developing a sophisticated lines' fairing system for a computer that will soon be obsolete. Similarly, it is unwise to dismiss the practical application of a particular fairing method because it needs bigger and faster machines than are currently available.

Each of these requirements is difficult in its own right. Some of them conflict by nature, and must be compromised upon. In particular, the need to split the hull surface into several patches, for practical fairing, has created new difficulties, while solving some of the old ones. The fairing problem is now in two distinct parts - the hull division problem, and the choice of function suitable for each part of the hull. Correspondingly, the aim of this work has been twofold:

a) To formalise hull division methods,

b) To experiment with different classes of functions, so as to minimise the number of functional forms that need be used for a satisfactory hull definition, thus also simplifying the problems of (a); or, more simply, to find such functions, that can cope with the rapid changes of slope and curvature of the typical hull form.

The aim of this work was not necessarily to produce a directly usable lines' fairing system. Nevertheless, as many as possible of the desirable qualities of such a system have been included in the experimental work.

1.6 Outline of the Work

Chapter 2 comprises a brief review of the literature on this subject, a broad classification, by their mathematical background, of a selection of methods, a typical division of the hull surface, and the fairing procedure with contemporary methods. The limitations of the various methods are discussed, as are the objections, from the naval architect's point of view, to the fairing procedure.

Chapter 3 contains a concise treatment of least-squares curve and surface-fitting by polynomials. Practical methods are described for the automatic choice of the optimum degree of polynomial for given data; also, for the correction of computational rounding error when using relatively high-order polynomials. While little of the individual ideas is new, their collective treatment and application in this way provides a foundation for the subsequent work.

Chapter 4 describes various attempts to find a general, analytical, way of splitting a hull surface into mathematically distinct patches. Several conventional and two new methods, one of basic geometry and the other using splines, are treated, and a comparative set of results given.

Using the techniques of Chapters 3 and 4, an experimental lines' fairing program has been developed by the author (Grossman, 1967), of a similar structure to that outlined by J.G. Hayes (1962). The system has not been completed, because it had such obvious deficiencies, that a totally different approach to the problem was indicated.

As a result, Chapter 5 contains experiments with basic transformations, namely rotation and transformation to polar co-ordinates. Least-squares polynomials are used to fair the transformed data in the new co-ordinate systems, using the curve-fitting method of Chapter 3, with automatic choice of degree. Similar attempts may have been made in the past, but there seems to be no evidence in the literature. Further, the automatic choice of polynomial order is thought to be a major step

forward.

By that time, it had been noticed that the deviations, in the conventional sense of the word, between the original and the faired data were often misleading as regards the quality of the results. A more realistic measure was the shortest distance from each data point to the corresponding faired value. By redefining the concept of 'deviation' suitably, a completely original, powerful, curve-fitting method was discovered - 'parametric' curve-fitting - which is described in Chapter 6. The procedure is essentially a transformation, but the data remains in the original reference system (this is an immediate advantage over transformations such as those of Chapter 5). Least-squares polynomials are again employed for the mathematical definition, but in a novel way.

The mathematical definition of a hull implies a surface, rather than a curve, description. Therefore, Chapter 7 treats the extension of the method of Chapter 6 to surfaces, i.e. the shortest distances from the data points to the faired surface are minimised. It is shown that a large part of the problem is its precise definition, and that the one used is the simplest to analyse and to implement. The parametric surface is very flexible in terms of the slopes and curvatures it can describe; so flexible, in fact, that the method can fair substantial portions of hull surfaces without subdivision. Moreover, an unusual amount of freedom is allowed in the disposition of the data points about the surface, i.e. unlike many fairing methods, the data points need not be given on a rectangular grid, nor even in any regular manner. The parametric curve and surface-fitting methods seem to fill important gaps both in the general theory of least-squares approximation, and its application to lines' fairing.

Finally, the work is summarised in Chapter 8, and a number of possible paths of further research in this field are suggested.

1.7 Co-ordinate System and Notation

Whenever reference is made specifically to ships, rather than to general surfaces, the co-ordinate system which is shown in Figure 1.2 is used.

The notation employed in the algebra is standard wherever convenient. Matrices are denoted by capitals, and vectors by overlines. When clear from the context, they are used without explicit definition. Vector spaces and norms are implicitly euclidian.

1.8 Data used

Throughout the work, except when data was constructed especially to test certain features of a method, data for analysis was taken from the two body-plans given in Appendix A. These two hulls are arbitrarily referred to as Model 1 and Model 2. Both are relatively simple forms, and have been chosen for the following reasons:

a) Model 1 has a simple bow, a compound stern profile, and a rising floor.

b) Model 2 has a bulbous bow, but a mathematically simple stern profile, and no rise-of-floor.

c) A common reason for both is that if any fairing or hull division method failed on either hull, it would undoubtedly fail on more complex forms. At the same time, these two forms provide a cross-section of the kind of data that a lines' fairing system must process.

1.9 Computing Equipment and Language

Much of the computing has been performed with the IBM-7094 Mk. II at the Department of Computing and Control of Imperial College. For some of the later work, the CDC-6600 at the University of London Computing Centre was used. Both machines have interfaces to a CALCOMP digital plotter, which has produced most of the graphical output included

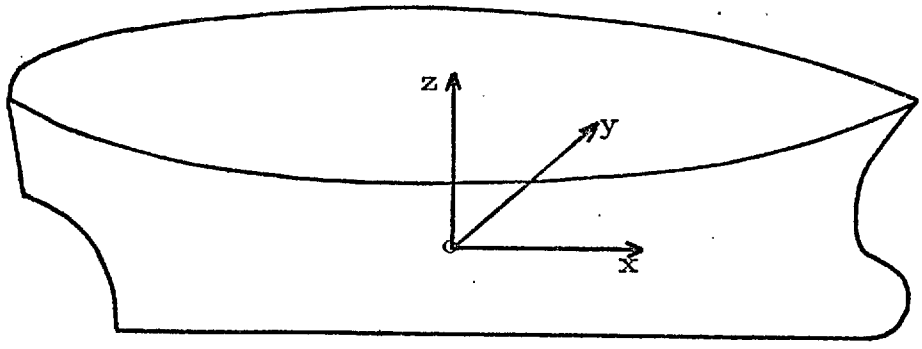


Figure 1.2 :- The Co-ordinate System

in this thesis.

The PDP-9, with a Visual Display Unit, provided by the sponsors of this work, and the UNIVAC-1108 at the National Engineering Laboratory (which also has an attached plotter) have also been used for some of the experiments.

Following the policy of machine independence, the FORTRAN-IV language has been used exclusively for computer programs. While this language has some marked deficiencies, it is implemented in the vast majority of scientific computing establishments, and seems likely to remain so in the near future. The standard version of the language has been strictly adhered to, which has permitted the use of a variety of convenient computers.

CHAPTER 2 :-

PAST AND PRESENT METHODS OF MATHEMATICAL HULL DEFINITION

2.1 A Review of Mathematical Hull Definition Methods

The research in this field has been for two pricipal aims - hydrodynamic analysis, and actual shipbuilding. The former employs quite crude approximations, and, therefore, the accuracy of the hull definition need only be of the same order as that of the approximations; the emphasis is on the simplicity of the hull function, for analysis reasons. The latter purpose requires a precise hull definition, in order to be of any practical use. It is the second category that has proved the more difficult, and that is the one of concern in this work.

The ideal solution is clearly a hull definition which is precise enough for mechanised building, and yet sufficiently simple for convenient hydrodynamic analysis. This has been borne in mind throughout the work.

It would take a large volume to enumerate and study, even superficially, all the attempts to define hull forms mathematically. Fortunately, excellent surveys exist already. A review of work in this field up to 1955 is to be found in Saunders (1957). Later efforts are reviewed by Miller and Kuo (1963) and by Pitidis-Poutous (1964). Finally, the most up-to-date study of computer methods is that of Kuo (1971). On the practical side, a good example of a modern mathematical fairing system, of which details are available, is that of Kantorowitz (1967). Commercially, AUTOKON of Norway provide a computer-aided ship design service, and SENER (Shipbuilders) of Spain have developed their own system (FORAN, 1969), to name but two. It is noted that these particular systems do not normally permit the designer any direct interaction with the computer method. One system, which has been developed with explicitly the last point in mind, is that of Horsham (1970), which employs a Time-Sharing computer terminal. A more comprehensive list of works is provided in the Bibliography.

As an illustration of the diversity of previous research in this

field, four classes of hull definition methods are described briefly, with the emphasis on the more unusual ones. Whenever relevant, reference is given to a specific utilisation of a method. Further, the main relative advantages and drawbacks of each class of methods are noted.

Mathematical fairing is often achieved by optimising some 'fairness' criterion; it is then expressed algebraically as the minimisation of the corresponding error norm. The choice of error norm usually has about as much effect on the results as the original choice of hull description function. This is shown by describing three methods which employ the same hull function, but different measures of fairness. Methods which just fit the given data are not considered.

The study of these past attempts, and of modern methods, indicates that all the lines' fairing methods , which are in regular use by ship-builders, require that the hull be split into several parts, which are faired separately. A typical hull division, for use with contemporary methods, and typical functional forms used to fair each region, are therefore included.

Finally, the complete fairing procedure with a typical computer-aided lines' fairing system is outlined.

2.2 A Classification of Mathematical Hull Definition and Fairing Methods

2.2.1 The numerical definition

This, rather basic, method is included here for completeness. As the name suggests, it consists of the straight-forward definition of a form, by as many points on the hull surface as are necessary for an accurate description, i.e. offsets, fore and aft profiles, etc. Fairing is invariably accomplished by smoothing higher order finite differences by least-squares.

Disadvantages: difficult to formulate and computerise; has few of

the desirable properties of a true mathematical definition.

2.2.2 Least-squares polynomials

This is a classical method which, because of its relative simplicity, alone accounts for over half the research in this field. As much of this thesis is concerned with various aspects of polynomial methods, only a brief summary is given here.

Initially, attempts were made to fit entire hulls with polynomials. Curve-fitting, rather than surface-fitting, was used. It was soon realised that very high order polynomials are needed to describe some parts of the typical hull form (Kerwin, 1962). Unfortunately, this is incompatible with the description of linear regions, such as the water-lines at the Parallel-Middle-Body. Moreover, waterline endings, cross-sectional curves and stem and stern profiles have been found unmanageable, because of their ranges of slopes. Consequently, the early researchers only treated the relatively simple parts of the hull, often with fifth or sixth order polynomials.

By 1960, it was generally accepted that, for practical mathematical fairing, the hull surface must be faired in parts. The parts which are suitable for polynomial definition were still faired by repeated curve-fitting; on the other hand, methods had been devised for treating 'difficult' regions (Pien, 1960; J.G. Hayes, 1962).

Advantages: simplicity of calculation and manipulation; the least-squares criterion penalises large deviations.

Disadvantages: restricted to surfaces of limited gradients; very prone to instability, i.e. unwanted oscillations between the data points; difficulty of deciding on the optimum degree of polynomial; surface-fairing by repeated curve-fairing places severe restrictions on the distribution of the given data points and on the degrees of the polynomials.

2.2.3 Mathematical splines

These will be treated more fully later in the thesis. For the time being, it suffices to say that the data is fitted by polynomials, in a piecewise manner, i.e. each polynomial applies to one interval between a pair of data points only. The coefficients of the spline equations may be calculated using 'thin-beam' theory, thus intending to simulate manual fairing. The main difference is that a curve composed of cubic splines - the most common generally, and the only ones seriously considered for lines' fairing - has discontinuities in the third derivative at the data points.

When calculating spline equations for an exact fit at the data points, an underdetermined system of linear equations (in the coefficients) is obtained. The two remaining degrees of freedom are most conveniently thought of as the curvatures, or, more precisely, the values of the second derivative, of the spline curve at the end-points. The values chosen for these control the behaviour of the fitted curve between the data points. For simple interpolation, the values are usually not critical, and may be estimated by finite differences; thin-beam theory implicitly demands that they be taken as zero, by assuming the beam to be pin-jointed at the end-points.

Method 1

The first spline method (Theilheimer and Starkweather, 1962) uses the discontinuities in the third derivative at the data points as the measure of error. This is reasonable, because these discontinuities are proportional to the forces acting at the points. The method consists of varying the estimates of the second derivative at the end-points in such a way as to minimise these discontinuities.

Disadvantages: depends heavily on the accuracy of the given data; cannot cope with steep slopes or with rapidly changing slopes.

Method 2

This method (Asker, 1962) attempts to overcome both disadvantages of the previous method. The error measure is an exact imitation of the one used for graphical fairing. An initial spline curve is fitted to the data. The points are then removed, one at a time, a new curve being fitted each time, and the difference in the dependent co-ordinate at the missing point is noted. The largest of these differences is the error 'norm'. However, the method, as published, gives no indication of how this error is minimised.

An ingenious solution is offered to the problem of steeply varying curves. Each spline is computed with respect to a local co-ordinate system, which has been rotated so as to place the relevant pair of points on the abscissa. An iterative scheme is outlined (without proof) for computing the modified splines.

Disadvantages: it is not clear how fairing is accomplished; the modified spline requires much computation, both for fitting the curve, and for calculating points on it, so as to outweigh the object of the modification; the extension to surfaces is not given, and is not obvious.

Method 3

This method (Berger et al., 1966) generates spline curves and surfaces in a completely different way. The fitted curve or surface is no longer required to pass exactly through all the data points. Instead, the basic definition of 'fairing' (Section 1.2) is expressed algebraically; in particular, the error measure is the maximum devation between the fitted function and the original points, i.e. the minimax error. Using splines as the hull definition function, the problem is shown to be one of Linear Programming (LP).

While the minimax is a powerful and critical measure, both theoretically and practically, it leads to an inconveniently large LP problem, as regards the number of variables and constraints. In order to ensure a

solution in a reasonable amount of computer time, Berger compromises, by redefining the error norm as the sum of the magnitudes of the deviations at all the data points.

It is shown that the extension of the method from curves to surfaces is straight-forward; also, that the matching of boundary conditions of combined surfaces presents no difficulties.

Advantages: the flexibility of the spline function means that the division of the hull surface is not critical; further constraints, perhaps involving other naval architecture considerations, can be added very simply.

Disadvantages: this method employs second degree finite differences at the data points. As these are not available at the end-points (or boundaries, in the case of surfaces), difficulties are encountered there; the method uses much computer time, even on reasonably modern computers, and using the modified error norm; the most outstanding feature of this method is the use of the minimax error norm, and the redefined norm is only marginally superior to least-squares; finally, fairly accurate data is needed (Berger gives a method for improving raw data).

2.2.4 Intrinsically fair hull forms

The alternative to fairing data which has been read off a drawing is to create an intrinsically fair form mathematically. As there is no comparison process, the definition of 'fair' no longer applies, and one must revert to the original, subjective, definition. This implies that the generation of the form is by trial and error, which, in turn, implies the use of an interactive computer system, if the process is to be attractive. One such method has been treated by MacCallum (1970), who uses a parametric surface definition (Coons, 1967). It is noted that even when 'moulding' a hull by computer in this way, the surface must be divided into several patches to achieve a satisfactory form. The

allied subject, of intrinsically-fair analytical form distortion, has been treated by Horsham (1970).

The importance of an intrinsically fair form definition cannot be overstressed. However, the relative advantages and disadvantages of this class of methods can only be discussed with reference to particular methods. The only common property is the lack of a rigourous definition of 'fairing'. If the interactive moulding is performed via a Visual Display Unit (VDU), a further drawback follows - the limited precision of VDU's means that it is difficult to assess the quality of the results.

2.3 A Typical Division of the Hull Surface and a Geometrical Classification of the Regions

A typical hull division is depicted in Figure 2.1. Table 2.1 gives details of the geometrical types of the various regions.

Conics are taken to include circular arcs as special cases. Also, not all hull forms possess all the geometrical types, e.g. a destroyer form has no Parallel-Middle-Body.

2.4 The Fairing Procedure with Contemporary Computer-Aided Methods

At present, there seems to be no method which is capable of describing even the simplest practical hull form by a single equation. Computer-aided fairing is usually performed as follows, with little variation between methods:

Step 1: Given a suitable lines' drawing, the offsets and other relevant data are read off the drawing.

Step 2: A trial division of the hull, as shown in the previous section, is made manually, often by observation.

Step 3: The initial division boundaries are usually irregular space-curves. Patch fairing methods invariably demand smooth patch boundaries, so that these must be smoothed before the patches can be faired. This

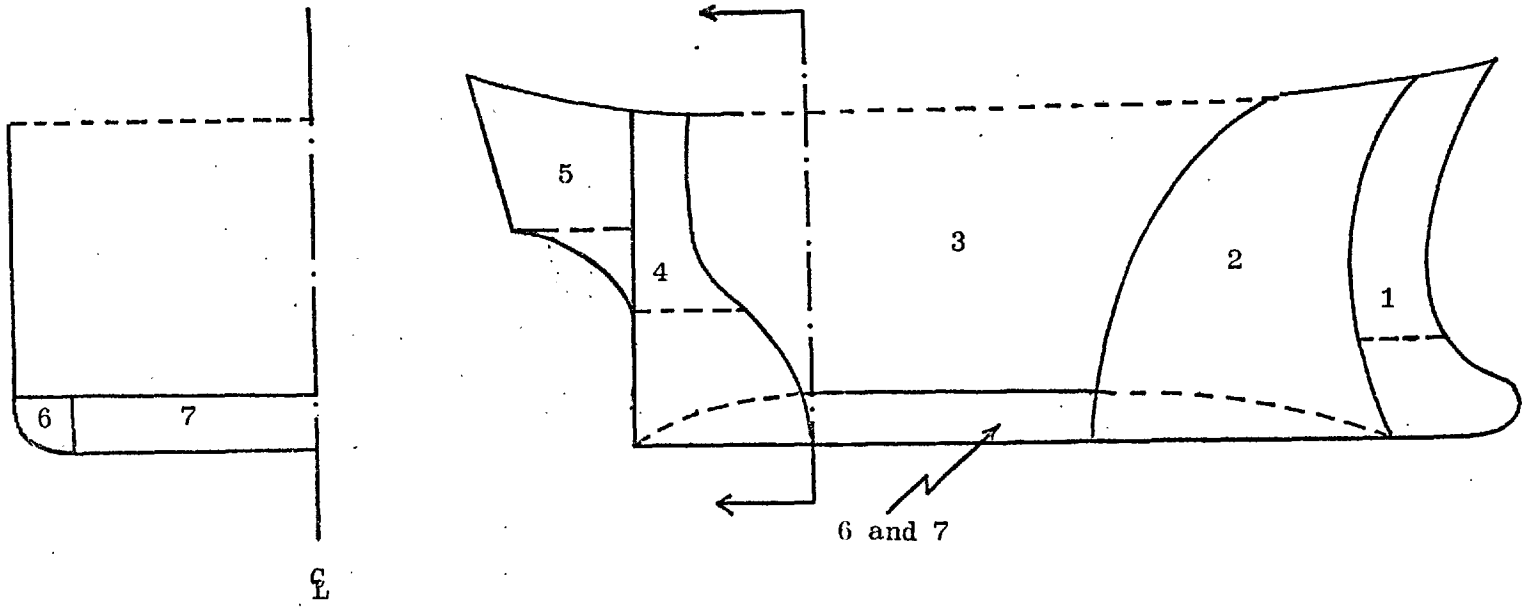


Figure 2.1:- Typical division of the hull surface

Table 2.1:-

Typical Hull Division and Patch Classification (see Figure 2.1)

Patch	Region	Fairing Function
1	Bow	Polynomial and straight-line combination for the profile; usually conic for the cross-section; sometimes subdivided further, as shown.
2	Forebody	Polynomial; usually of degree 3-5 for the waterlines, and one or two degrees lower for the cross-sections.
3	Parallel-Middle-Body	Plane, or low-order polynomial.
4	Afterbody	Similar to the Forebody, but usually the same order polynomial for both sets of curves.
5	Stern	Conic and straight-line combination for the profile; conic for the cross-section; sometimes subdivided further, as shown.
6	Bilge	Often given as conic or circular section, of known dimensions.
7	Floor	Plane, horizontal or inclined at a given slope.

is a critical and tedious phase of the procedure, because its difficulties are identical in principle to those of fairing the complete hull, i.e. the boundaries themselves contain linear and other awkward segments. The smoothing is usually performed graphically.

Step 4: Data is prepared for the computer, e.g. punched cards.

Step 5: The patches are faired separately, using a suite of computer-programs, which maintain continuity of value, slope and curvature at the patch boundaries. It is noted that when conics or straight lines are used, they are invariably fitted exactly, rather than faired.

Step 6: Generally speaking, the computer results are produced in graphical form, using a digital plotter. The size of the drawings is as large as the plotter permits - usually larger than the original body-plan; also, the waterlines and sections are plotted more densely than in the original plan, the latter often at frame stations.

Step 7: The results are examined by the designer and the loftsman. If they are satisfactory, the hull form is finalised. Otherwise, there are usually two possibilities, namely,

a) Where relevant, degrees of polynomials may be altered.

b) The patch-boundaries may be altered.

The procedure is then iterated either from Step 3 or Step 4, depending on the alterations which have been made.

The degrees of the polynomials and the positions of the patch-boundaries often bear no clear relationship to the results. Any modification to these must therefore be made on a trial and error basis. The more advanced the fairing method, the more flexible it is, i.e. the less sensitive it is to these variables. Nevertheless, the procedure is tedious and time-consuming. Several iterations are normally needed to finalise a design, which means several computer-runs, and many man-hours. Moreover, the non-systematic variation of degrees of polynomials and the artificial hull division are alien to the naval architect.

CHAPTER 3:-

LEAST SQUARES CURVE AND SURFACE FITTING

3.1 Introduction

There is a vast literature on the subject of least-squares curve and surface fitting. Because of the diversity of the accounts, a review is presented here, which contains all the theory necessary for this thesis, without elaborating on the more profound details.

In comparison with curve fitting, least-squares surface fitting has been neglected somewhat. For many applications, it has been achieved by fitting curves in orthogonal directions. This, and bivariate surface fitting proper, are both presented.

Of particular interest to lines' fairing are practical problems, such as computational rounding error and the optimal degree of polynomial for given data. Possible solutions to these problems are discussed.

3.2 Curves

3.2.1 Basic theory

An excellent summary of the theory of least squares curve fitting by polynomials is given by Forsythe (1957), together with the outline of the structure of the corresponding computer program. In this section, a brief, but in some ways more general, account is given of the former, while some of the techniques displayed in the latter are no longer necessary for the modern computer.

Given n data points, (x_i, y_i), $1 \leq i \leq n$, with x-values exact, and the y-values subject to errors ϵ_i with weights w_i, $1 \leq i \leq n$. Initially, nothing is assumed about the distribution of the errors.

It is required to approximate $\bar{y}$ by a linear combination of a set, P, of $(m + 1)$ arbitrary functions in x,

$$P = [\, p_k(x) \,,\ 0 \leq k \leq m \,],$$

in such a way as to minimise the sum of squares of weighted deviations between $\bar{y}$ and the approximation, i.e. let the approximating function be

$$f(x) = a_0 p_0(x) + a_1 p_1(x) + \ldots + a_m p_m(x) \qquad \ldots\ldots(3.1)$$

so that $$y_i = f(x_i) + e_i \quad , \quad 1 \leq i \leq n \qquad \ldots\ldots(3.2)$$

with the coefficients, $\bar{a}$, selected so that

$$S = \sum_{i=1}^{n} w_i^2 e_i^2 \qquad \ldots\ldots(3.3)$$

is a minimum. Henceforth, all summations with the index 'i' shall be taken to be over the data points, i.e. from 1 to n.

To compute $\bar{a}$, the expression for the error vector, $\bar{e}$, obtained from equations 3.1 and 3.2, is substituted in equation 3.3, which is then differentiated with respect to each 'a' ; for a turning point of S, all the derivatives must be zero. The following set of linear equations (in $\bar{a}$), known as the 'normal equations', is obtained:

$$G\,\bar{a} = T$$

where G is a $(m + 1)(m + 1)$ symmetric matrix, with elements

$$g_{kl} = \sum_i w_i^2 p_k(x_i) p_l(x_i) \quad , \quad 0 \leq k \leq m \; ; \; 0 \leq l \leq m$$

and T is the column matrix with elements

$$t_k = \sum_i w_i^2 y_i p_k(x_i) \quad , \quad 0 \leq k \leq m \, .$$

Assuming that G is of full rank, the solution is given by

$$\bar{a} = G^{-1} T$$

So far, nothing has been assumed about the functions 'p(x)'. For present purposes, let $\bar{y}$ be approximated by a polynomial.

Consider the case of $p_k(x) = x^k$, i.e.

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots\ldots + a_m x^m$$

Then
$$g_{kl} = \sum_i w_i^2 x_i^{k+l} \qquad \ldots\ldots(3.4)$$

and
$$t_k = \sum_i w_i^2 y_i x_i^k$$

It can be shown that the soluton is unique, and a minimum of S. There are two serious objections to this dorivation . The first is obvious - each time the degree of the approximating polynomiälis raised, a row and a column are added to G, and it has to be inverted again; there is no simple relationship between the two inverses. The second objection is more profound. To illustrate it, let n be large, and let $\bar{x}$ be distributed approximately uniformly in the range [0,1]. The summation of equation 3.4 can then be replaced by an integral, which simplifies to

$$g_{kl} = \frac{n}{k + 1 + 1}$$

i.e. n-times the principal minor in the Hilbert matrix

$$H = \begin{vmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \ldots\ldots \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \ldots\ldots \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \ldots\ldots \\ \ldots & \ldots & \ldots & \ldots\ldots \\ \ldots & \ldots & \ldots & \ldots\ldots \end{vmatrix}$$

Minors of the Hilbert matrix are known to be extremely ill-conditioned. For example, with $m = 7$, the ratio of the highest to lowest term in H^{-1} is about 2,500,000. Therefore, extreme computational precision is needed to get valid solutions even for moderate degrees. Moreover, the difficulty increases with the number of data points, so instead of improving, the approximation worsens.

Reverting to the general function, let the functions P be decoupled, by making them orthonormal with respect to $\bar{x}$, i.e. let them obey the condition

$$\sum_i w_i^2 \, p_k(x_i) \, p_l(x_i) = \begin{cases} 0 & \text{for } k \neq l \\ 1 & \text{for } k = l \end{cases} \qquad \ldots\ldots(3.5)$$

The matrix G then becomes a well-conditioned diagonal matrix, and the solution to the normal equations is given by

$$a_k = \sum_i w_i^2 \, y_i \, p_k(x_i), \quad 0 \leq k \leq m \qquad \ldots\ldots(3.6)$$

The corresponding minimum of S is given by

$$\hat{S}_m = \sum_i w_i^2 \left[y_i - a_0 p_0(x_i) - \ldots - a_m p_m(x_i) \right]^2$$

$$= \sum_i w_i^2 y_i^2 + a_0^2 \sum_i w_i^2 p_0(x_i)^2 + \ldots + a_m^2 \sum_i w_i^2 p_m(x_i)^2$$

$$- 2 a_0 \sum_i w_i^2 y_i \, p_0(x_i) - \ldots - 2 a_m \sum_i w_i^2 y_i \, p_m(x_i)$$

$$- 2 \sum_i w_i^2 \left[\text{cross-products of the p's} \right]$$

The last group is zero by definition; each of the summations in the intermediate group is the expression for the corresponding coefficient from equation 3.6; hence

$$\hat{S}_m = \sum_i w_i^2 y_i^2 - \sum_{k=0}^{m} a_k^2 \qquad \ldots\ldots(3.7)$$

i.e. when a new function is added to the set P , a single summation term gives the new coefficient immediately, without affecting any of the previous ones and without inverting a matrix; the resulting reduction in the sum of squares of deviations is directly available as the square of the coefficient, although rather severe rounding error is to be expected from the difference of two quantities which are generally very similar.

There remains the definition of P. For a polynomial approximating function, each p(x) must necessarily be a polynomial. It is possible to orthonormalise the vectors generated from the monomials: 1, x, x^2 , ... , x^m , by using the Gram-Schmidt procedure. A better solution is to generate orthonormal polynomials, using a three term recurrence formula suggested by Householder (1953), as follows:

Let $$b_k p_k(x) = (x - c_k) p_{k-1}(x) - d_k p_{k-2}(x) \qquad \ldots\ldots(3.8)$$

with $p_0 = 1$, $d_1 = 0$ and k = 1,2,...,m . The values of b, c and d are calculable at each stage.

To prove this formula, let it be true, and assume that a set of orthonormal polynomials has been found; then, let b, c and d be calculated.

Treating equation 3.8 as a vector relationship, multiply it by an arbitrary member of the orthonormal set, $p_j(x)$, weight and sum over the data points, to obtain the following equation:

$$b_k \sum_i w_i^2 p_j(x_i) p_k(x_i) = \sum_i w_i^2 p_j(x_i) x_i p_{k-1}(x_i)$$

$$- c_k \sum_i w_i^2 p_j(x_i) p_{k-1}(x_i)$$

$$- d_k \sum_i w_i^2 p_j(x_i) p_{k-2}(x_i) \quad \ldots\ldots(3.9)$$

Set $j = k - 1$ in equation 3.9. Using the orthonormality condition,

$$c_k = \sum_i w_i^2 x_i p_{k-1}^2(x_i) .$$

Similarly, with $j = k - 2$,

$$d_k = \sum_i w_i^2 x_i p_{k-1}(x_i) p_{k-2}(x_i) .$$

The value of b_k is such that the normality condition is maintained, and is computed from the relationship

$$b_k^2 \sum_i w_i^2 p_k^2(x_i) = 1 .$$

Finally, by setting $j = k$, the following identity is obtained

$$b_k = \sum_i w_i^2 x_i p_k(x_i) p_{k-1}(x_i)$$

whence it follows that d_k can be expressed by the alternative formula

$$d_k = b_{k-1} / b_{k-2}$$

This completes the proof. For a more powerful one, it may initially be assumed that the new polynomial is related to all the previous ones, and the relational coefficients, other than the ones derived

above, shown to be zero. The formula breaks down at $k = n - 1$, at which stage the fitted polynomial would pass exactly through all the data points.

For an arbitrary distribution of the x-values, the norms of the orthogonal vectors (i.e. the successive values of b) become either very large or very small. This can be illustrated by assuming $\bar{x}$ to be distributed approximately uniformly in the range $[-1,+1]$. The polynomials generated in that case would be similar to the Legendre polynomials, which, for that range of x, have a rapidly decreasing leading coefficient (Forsythe uses Chebyshev polynomials to show this, because they have a similar behaviour and simpler to manipulate). The leading coefficient remains constant if $\bar{x}$ is scaled to lie in $[-2,+2]$. In actual practice, the x-values are not uniformly distributed, and the leading coefficient does not necessarily swamp the others (or vice versa), as has been assumed implicitly. Nevertheless, the highest computational precision is likely with this scaling when using fixed-point arithmetic. With modern computers, which normally used floating-point arithmetic, this scaling is no longer essential, but still advisable, so that the relationship 3.8 involves quantities of similar magnitudes. It is almost imperative to at least shift the origin of x so that the mean shifted value is zero. The significance of this, in relation to the computation of polynomials, is clear from observation.

An important modification to the basic procedure was suggested by Clenshaw (1960). He shows that the orthogonal polynomials can be represented by linear combinations of Chebyshev polynomials; the entire recurrence sequence and the least squares calculations can be performed directly in terms of the Chebyshev form, with minor alterations only ($\bar{x}$ must be scaled to lie in $[-1,+1]$, and the leading

coefficient in the polynomials in the summation must be halved). The benefit from this alternative form is much lower computer storage requirement, at the sacrifice of some computational speed.

The modified procedure was adopted for the Experimental Lines' Fairing Package (Grossman, 1967), but was found unsuitable for the more general experiments attempted later.

3.2.2. Computational error analysis

Some loss of significant figures occurs in the computation of the approximating function. It is useful to know the magnitude of this loss, and, if possible, to correct it, using the available information. The following analysis is based on the work of Clenshaw and Hayes (1965).

Let C_m denote the reduction in the sum of squares of weighted deviations as the degree of the approximating polynomial is raised from $(m-1)$ to m. It is calculated from

$$C_m = -\sum_i w_i^2 [f_m(x_i) - y_i]^2 + \sum_i w_i^2 [f_{m-1}(x_i) - y_i]^2$$

$$= -\sum_i w_i^2 [f_{m-1}(x_i) + a_m p_m(x_i) - y_i]^2$$

$$+ \sum_i w_i^2 [f_{m-1}(x_i) - y_i]^2$$

$$= -2 a_m \sum_i w_i^2 p_m(x_i) [f_{m-1}(x_i) - y_i] + a_m^2 \sum_i w_i^2 p_m^2(x_i)$$

.....(3.10)

which, using the orthonormality condition and the expression for a_m (equation 3.6), reduces to

$$C_m = a_m^2 \sum_i w_i^2 p_m^2(x_i) = a_m^2 \qquad \ldots\ldots(3.11)$$

(this fact is already known from equation 3.7, but this particular derivation is used later)

Now, the computed values of $f_{m-1}(x)$, $p_m(x)$ and a_m are subject to error; let them be denoted by

$$\hat{f}_{m-1}(x) = f_{m-1}(x) + \delta f_{m-1}(x)$$

$$\hat{p}_m(x) = p_m(x) + d\delta p_m(x)$$

and

$$\hat{a}_m = a_m + \delta a_m .$$

Two estimates can now be calculated for the actual reduction in the sum of squares of weighted deviations, namely

$C_m^{(1)} = a_m^2 \sum_i w_i^2 \hat{p}_m^2(x_i)$ from equation 3.10, and, from equation 3.11,

$$C_m^{(2)} = -2\hat{a}_m \sum_i w_i^2 [p_m(x_i) + \delta p_m(x_i)][f_{m-1}(x_i) + \delta f_{m-1}(x_i)]$$

$$- \hat{a}_m^2 \sum_i w_i^2 \hat{p}_m^2(x_i) \qquad \ldots\ldots(3.12)$$

In expanding equation 3.12, it is noted that

$$\sum_i w_i^2 p_m(x_i) \delta f_{m-1}(x_i) = 0$$

by orthogonality, since $\delta f_{m-1}(x)$ is a linear combination of polynomials of degree less than m; also,

$$\sum_i w_i^2 \, \delta p_m(x_i) \, [f_{m-1}(x_i) - y_i] = 0$$

because $[f_{m-1}(x) - y]$ is the difference between a fit of degree $(m - 1)$ and an exact fit and can therefore be expressed as a linear combination of orthonormal polynomials of degrees m to $(n - 1)$, i.e.

$$f_{m-1}(x_i) - y_i = - a_m p_m(x_i) - a_{m+1} p_{m+1}(x_i) - \ldots - a_{n-1} p_{n-1}(x_i)$$

whereas $\delta p_m(x)$ is a polynomial of degree $(m - 1)$ - there is no error in the leading coefficient. Therefore, neglecting the small quantity

$$\sum_i w_i^2 \, \delta p_m(x_i) \, \delta f_{m-1}(x_i) \ ,$$

equation 3.12 simplifies to

$$C_m^{(2)} = 2 \hat{a}_m \hat{a}_m - \hat{a}_m^2 \sum_i w_i^2 \hat{p}_m^2(x_i) \ .$$

If the error in $p_m(x)$ is small, the summation in the second term is very nearly unity. Therefore, to a first approximation,

$$C_m^{(2)} = 2 \hat{a}_m \hat{a}_m - \hat{a}_m^2$$

and $$C_m^{(1)} = \hat{a}_m^2$$

whence the corrected value of the coefficient a_m can be deduced:

$$a_m = \frac{1}{2} \hat{a}_m \left[1 + C_m^{(2)} \Big/ C_m^{(1)} \right] \qquad \ldots\ldots (3.13)$$

Equation 3.13 may be used to estimate the error and to inform the user of its magnitude. A better solution is to apply the correction numerically as the computation proceeds, until the error either

exceeds a given bound, or becomes so large as to swamp all the useful information. As a safety measure, the computation should not be abandoned on the grounds of excessive rounding error of a single coefficient; it is not unusual to find coefficients that are alternately small and subject to severe error, and large and error free, e.g. when the data fitted is approximately an even function in x, and the x-values are distributed approximately symmetrically about the origin. Generally speaking, a termination of the fitting procedure because of rounding error is very rare.

3.2.3. Imposition of constraints on the fit

In the foregoing sections it has been assumed that the deviations are weighted discretely. For many applications, it is desired instead to impose certain conditions on the fit, e.g. that the fitted curve pass exactly through specified data points. A flexible way of achieving this is described by Cadwell and Williams (1961) and, more extensively, by Clenshaw and Hayes (1965).

The approximating function, f(x), is split into three parts, namely

$$f(x) = q(x)\, g(x) + r(x) \qquad \ldots\ldots(3.14)$$

where g(x) is a polynomial, ostensibly generated by the orthogonal polynomials procedure; q(x) and r(x) are the 'forcing functions'. r(x) is chosen so as to fulfil the desired requirements; q(x), the 'zeroising function', ensures that the product q(x)g(x) (and, if necessary, its derivatives) is zero at the appropriate points, i.e. that it does not 'interfere' with the values provided by r(x). For example, let $f(1) = 1$; then $q(x) = 1 - x$ and $r(x) = 1$ are the simplest forms of q(x) and r(x).

The straight forward way to implement this is to calculate a

new dependent variable, Y, from

$$Y_i = \frac{y_i - r(x_i)}{q(x_i)} = g(x_i) + \epsilon_i \quad , \quad 1 \leq i \leq n$$

(except where $q(x) = 0$, because these points contribute nothing to the sum of squares of deviations). Y is then fitted, using the values $q(x_i)$ as discrete weights, to allow for the division, i.e. if using the procedure of Section 3.2.1, $w_i = q(x_i)$. After the fit, the required function, f(x), can be recovered from equation 3.14.

Alternatively, q(x) can be included implicitly in f(x) by setting $p_0(x) = q(x)$ as the first orthogonal polynomial, and applying the procedure to $Y_i = y_i - r(x_i)$, $1 \leq i \leq n$. As $p_0(x)$ is a factor of all the succeeding polynomials, it is also a factor of their linear combination. This avoids the initial division by q(x), the weighting, and the final multiplication by q(x) to recover f(x).

Despite the superficial differences between the two methods, their formal equivalence can be proved. It is noted that the orthonormal functions, p(x), are only polynomials with the latter method if q(x) is a polynomial. This is normally not very restrictive.

Derivatives can also be imposed, using the latter method, by appropriate choice of q(x), and r(x). If a certain degree of derivative is constrained, it is implied that the lower ones are constrained, too (this is clear by observation). This may create a problem, because the lower derivatives are not necessarily given. A possible solution in such cases is to estimate the values of the derivatives; then repeatedly carry out a fit and refine the estimates, until the lowest sum of squares of deviations is found.

Further, infinite values of f(x) and/or its gradients can also be imposed, although only at the end-points of the x-interval.

for multiple singularities, the interval must be split up.

The ability to constrain a least squares fit is particularly important for lines' fairing. If the hull surface is divided into patches, continuity up to any degree can be preserved between patches.

3.2.4. Deduction of the optimum degree of polynomial by analysis of the residuals

The problem is one of fitting the given data by polynomials of progressively higher order, and deciding at which stage the best approximation to the data has been found. The concept of 'best' is vague. Common criteria for goodness-of-fit are the values of the sum of squares of residuals and the 'root-mean-square error' (RMS), which is an estimate of the variance of the errors. Unfortunately, both of these criteria can be misleading.

Here, two different solutions to this problem are described. The first is quite general, but is analytically unfounded; the second is analytically sound, but makes assumptions about the distribution of the errors and, as such, is only approximate in the general case. Despite these theoretical shortcomings, both methods have been found very efficient in a variety of applications.

a) The maximum deviation normal to the fitted curve

Consider the data and the fitted curve of figure 3.1. The deviation, ϵ, at the data point P is large, whereas the curve is a good fit to the data to most intents and purposes. It is the high value of derivative implied by the data in the neighbourhood of P which is misleading.

Let the 'true deviation', δ, be the minimum distance between the point P and the curve, i.e. the length of the normal PT dropped

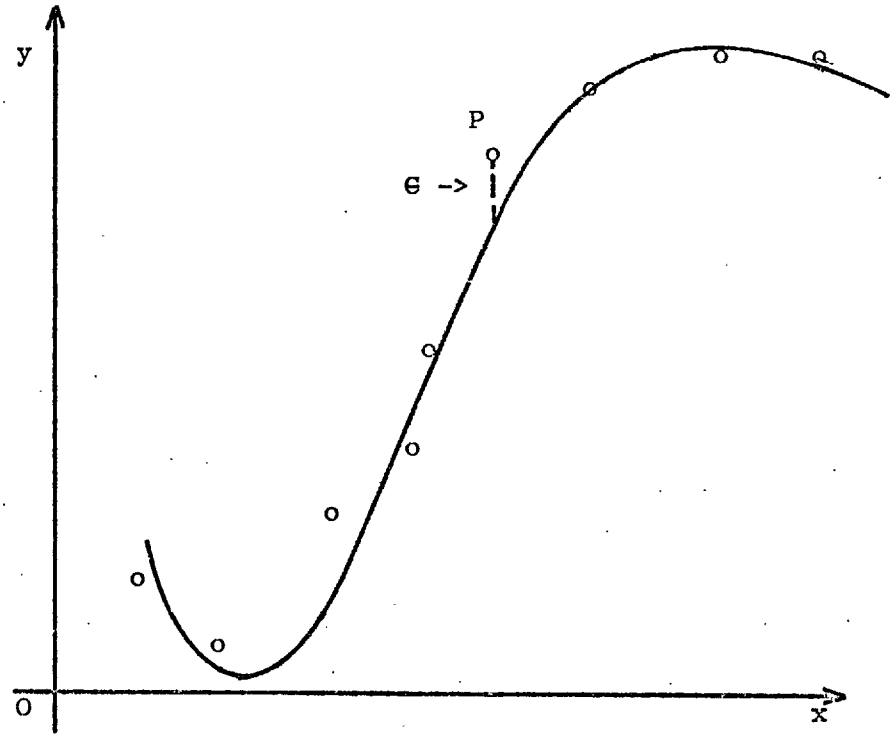


Figure 3.1 :- A typical data set, with an apparently bad fit.

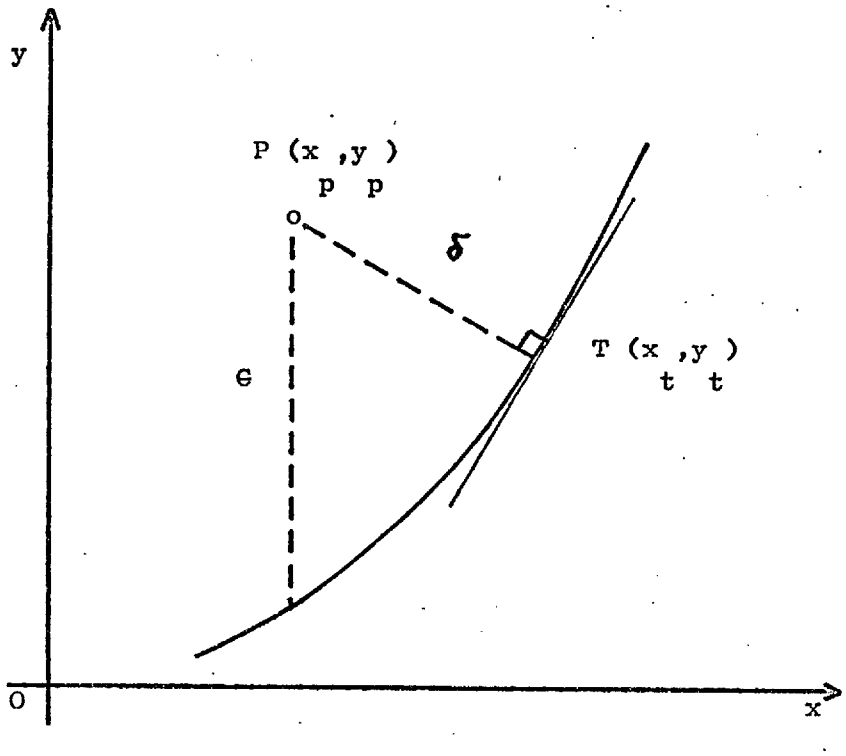


Figure 3.2 :- Enlargement of the neighbourhood of point P in Figure 3.1.

from P onto the curve at $T(x_t, y_t)$ - figure 3.2.

Define the fit as satisfactory if

$$\hat{\delta} = \max [\, \delta_i , 1 \leq i \leq n \,] \leq \beta$$

where β is a prescribed, small, positive number. $\hat{\delta}$ is recognisable as the Chebyshev, or minimax, norm of the vector of 'true deviations', $|| \bar{\delta} ||_\infty$. Its use in this context is not altogether legitimate, because the curve was fitted by minimising $|| \bar{\epsilon} ||_2$. There is, therefore, no guarantee that the condition can be met for a given value of β, regardless of the order of the fitted polynomial (although it can be proved that as $n \rightarrow \infty$, $|| \bar{\epsilon} ||_2 \rightarrow || \bar{\epsilon} ||_\infty$ - Cheney, 1966 - and hence it is likely that the condition be met for some value of β and degree of polynomial).

Leaving the theoretical objections aside for the moment, the practicality of computing $\hat{\delta}$ must be considered. The complete derivation is given in Appendix B. The calculation is a simple iterative procedure, based on Newton's method for finding a zero of a function, and combined with the Regula Falsi method to force convergence in cases where the basic procedure diverges.

Thus there seem to be no practical objections to this method of assessing the quality of a fit, and it was used in the Experimental Lines' Fairing Package (Grossman, 1967). In use, it was noted that when the points corresponding to the maximum and minimum values of δ remained unchanged when the degree of polynomial was altered, there was often a 'rogue' point in the input data.

b) The statistical significance test

This is treated by Hudson (1964, vol. II). A cursory description is given here of the principles involved, without the statistical background or the associated proofs.

So far in this chapter, the values of the dependent variable, y,

have been assumed to originate from a drawing, and to be subject to random reading errors. Let it now be postulated that there exists an exact function, $g(x)$, of which the values of y are representative. This function is to be approximated by a polynomial, $f(x)$, of degree m. The value of m is sought, which gives rise to the 'best approximation' to $g(x)$, with the actual values of the residuals playing a secondary role.

To assist the definition of the term 'best approximation', it is assumed that the measurements of y can be carried out any number of times. The best approximation is then that which minimises the variance of the fit to y, i.e. the variance of the coefficients in $f_m(x)$, over an indefinite number of 'experiments'.

The estimate of unexplained variance, σ_m^2, in one set of data is given by

$$\sigma_m^2 = S_m / \nu$$

where ν is the number of degrees of freedom, and is given by

$$\nu = n - m - 1 .$$

σ is known as the 'root-mean-square' (RMS) error.

The amount by which $f_m(x)$ will vary in repeated experiments can be shown to be

$$\mathrm{Var}\,[f_m(x)] = \sum_{k=0}^{m} p_k^2(x)\,\mathrm{Var}\,[a_k]$$

which can be proved to be

$$= \sigma^2 \sum_{k=0}^{m} b_k^2(x)$$

(the unweighted version is given, for clarity)

If the degree of $f(x)$ is increased to $(m + 1)$, the sum of squares of residuals decreases to

$$S_{m+1} = S_m - a_{m+1}^2 \qquad \text{from equation 3.7,}$$

and the new and old variances of f(x) are related by

$$\frac{\text{Var}[f_{m+1}(x)]}{\sigma^2} = \frac{\text{Var}[f_m(x)]}{\sigma^2} + b_{m+1}^2$$

So that the better fit in this case has been obtained at the expense of an increase of $\sigma^2 b_{m+1}^2$ in the variance of the fitted curve.

There normally exists a critical value, m = h, which corresponds to the first $(m + 1)$ terms in the Taylor expansion of g(x). It is this critical degree which is sought. It is postulated that

$$a_k = 0 \quad \text{for all} \quad n - 1 \geq k \geq h$$

In practice, this is not so, and the object of the following analysis is to determine a bound, a coefficient below which can be discarded to a given degree of confidence.

Until now, nothing has been assumed about the distribution of the residuals. Now assume that they have independent and identical normal distributions, with zero mean and variance σ^2 (although different variances can be accomodated by introducing weights). Then the coefficients a_k can be shown also to have a normal distribution, about a mean a_k and with a variance of σ^2. It is noted that the value of σ^2 is not known, but is estimated by σ_m^2.

Further, the important result can be derived, that

$$\left[\frac{a_j}{\sigma_m^2}\right] \quad \text{is distributed as} \quad F_{1,\nu} \quad \text{if} \quad a_j = 0$$

where the F-distribution is the standard statistical tool for measuring confidence in the identity, within experimental error, of the values

of two variances estimated independently for the same sample of normally distributed random variables.

The procedure is thus to make the hypothesis that

$$a_{m+1} = a_{m+2} = \ldots\ldots = a_{n-1} = 0 ,$$

and to test it, using the values of the F-distribution which are tabulated according to degree of confidence and the number of degrees of freedom (Pearson and Hartley, 1958).

In practice, it is possible for coefficients to be alternately zero and finite, or even large, e.g. for even functions, with the x-values distributed approximately uniformly about the origin (note the similarity with the analysis of rounding error - Section 3.2.2). It is therefore advisable to fit a polynomial of a reasonably high degree (i.e. higher than that expected for the optimum, but lower than that at which the rounding error is significant), to test the coefficients one by one, starting with the higher ones, and to discard the insignificant ones (i.e. the ones which are zero to a given degree of confidence), until a significant one is encountered. The degree, h, thus found is defined as optimal. The function f(x) can be recovered in the usual way, from the remaining $(h + 1)$ coefficients and the corresponding orthonormal polynomials.

For the procedure to be computerised, a way must be found of obtaining the F-values by computer. The table of values for each degree of confidence (four are commonly used - the 95.0, 97.5, 99.0 and 99.9 percent levels) consists of between 35 and 40 entries, which are not for regularly varying ν. In order not to have to store the entire tables in the computer and to interpolate or extrapolate for the missing values, it is usual to approximate each by a relatively simple function. Appendix C contains such

approximations, which have been derived empirically.

For ship forms, the assumptions made about the distribution of the residuals are not necessarily true, although they are likely to so. The procedure described above has been tested extensively, and found to perform very well. In some cases, an unexpectedly low order was deduced as optimal, leaving rather high residuals. Nearly always in these cases, an error was later discovered in the input data, thus confirming the validity of the experiment.

3.3 Surfaces

3.3.1. General description of the problem

Given n data points, with their associated weights,

$$(x_i, y_i, z_i, w_i), \quad 1 \leq i \leq n .$$

There are several ways of fitting a polynomial to z in terms of x and y. The ones that may be chosen depend on the distribution of the independent variables in the xy-plane, the three most important ones being

a) Data given on every intersection of a $(k \times l)$ rectangular mesh,

b) Data given on a quasi-rectangular grid, e.g. x, say, varies in a regular manner between two limits; for every value of x, x_0, y is distributed, not necessarily regularly, between the limits $f_1(x_0)$ and $f_2(x_0)$, where $f_1(x)$ and $f_2(x)$ are real functions, defined over the range of x.

c) Data distributed randomly, between irregular, finite limits.

Problem (a) is treated by Cadwell (1961), and, briefly, by Clenshaw and Hayes (1965). It is solved essentially by repeated application of the curve fitting procedure, whether this appli-

cation be analytical or numerical. Orthogonal polynomials are genrated for one of the independent variables. Their multipliers with respect to each row of z (i.e. the a's of section 3.2.1) are then fitted in terms of the second independent variable. It can be shown that the order in which the independent variables are taken is immaterial. Cadwell considers the criterion of RMS to govern the choice of degrees of x and y in the final fit. He also considers the effect of missing data points. This is overcome by a simple iterative procedure, which consists of estimating the missing points, and repeatedly fitting the augmented data and replacing the missing values by the fitted ones. It can be proved that if this process converges, then it converges to the true least squares solution. Convergence conditions are not well defined; essentially, not too many points may be missing.

Problem (b) is discussed by Berztiss (1964) and elaborated by Clenshaw and Hayes. This is an important case, because it is directly applicable to hulls, where the regularly varying parameter corresponds to the waterline heights. The solution is again essentially one of repeated curve fitting. The irregularly varying parameter, y say, is first treated by

$$Y = \frac{2y - f_1(x) - f_2(x)}{f_1(x) + f_2(x)}$$

which ensures that Y lies in the range $[-1,+1]$, as is required by the Chebyshev modification, although the important point is that the mesh is effectively 'rectangularised'. Orthogonal polynomials are generated in terms of x - the order is important in this case - and their multipliers for rows of z are fitted by polynomials in terms of Y. It is pointed out by Clenshaw and Hayes that the surface

fitted thus is not the true least squares one; for most applications, this is immaterial, because the least squares criterion is often used merely as a convenient means of obtaining a reasonable approximation to the data, and is not demanded by the theory behind the application. Clenshaw and Hayes also describe an iterative procedure to allow for missing data points.

Problem (c) was solved analytically by Weisfeld (1959). He proves that the generation of orthogonal polynomials by a three set recurrence formula is extensible to any number of dimensions. Cadwell and Williams (1961) solved the bi-variate version by an analytical application of the Gram-Schmidt orthonormalisation process to a set of monomial functions.

Initially, method (b) was considered for this work, It was tried in the Experimental Lines' Fairing Package, where the following drawbacks were discovered:

i) For the cross-fairing to be successful, the first set of curves fitted must all be fitted by polynomials of the same degree. This contradicts the objective of automatic selection of degree by computer. The latter can still be implemented, by repeatedly modifying and re-fitting the data, until the degree selection mechanism deduces the same degree for all the curves. It is very inconvenient.

ii) Undue restrictions are imposed on the data points.

Altogether, method (c) seems to be the most general and versatile. The true least squares surface is generated, with all the associated advantages. Therefore, a subset of Weisfeld's theorem, involving two independent variables only, is proved below. The procedure lends itself well to programming.

3.3.2. Bi-variate orthonormal polynomials

The form of polynomials used for curve fitting is standard, but not for surface fitting. A variety of polynomial models can be generated by orthonormal sequences. The model used here is one of the simpler extensions of the curve fitting model.

Define the mth degree polynomial as the linear combination of all the monomials $x^k y^l$, where $k + l \leq m$, i.e. let the approximating function be

$$z(x,y) = a_{00} + a_{10}x + a_{11}y + a_{20}x^2 + a_{21}xy + a_{22}y^2 + \ldots\ldots\ldots\ldots\ldots + a_{m0}x^m + a_{m1}x^{m-1}y + \ldots\ldots\ldots\ldots\ldots + a_{mm}y^m \qquad \ldots\ldots(3.15)$$

This model may be solved directly, via the normal equations. The same difficulty is encountered as in the case of curve fitting, namely severe ill-conditioning of the equations. Therefore, let the approximating function be a linear combination of of orthonormal functions, i.e. let

$$z(x,y) = a_{00}p_{00}(x,y) + a_{10}p_{10}(x,y) + a_{10}p_{10}(x,y) + \ldots\ldots\ldots\ldots\ldots + a_{m0}p_{m0}(x,y) + a_{m1}p_{m1}(x,y) + \ldots\ldots + a_{mm}p_{mm}(x,y) \qquad \ldots\ldots(3.16)$$

where

$$\sum_i w_i^2 p_{kl}(x_i,y_i)\, p_{uv}(x_i,y_i) = \begin{cases} 1 & \text{for } k = u \text{ and } l = v \\ 0 & \text{for any other indices} \end{cases}$$

As for curve fitting, it can be deduced that

$$a_{kl} = \sum_i w_i^2 z_i p_{kl}(x_i,y_i)$$

and that the minimum sum of squares of weighted deviations is given by

$$\hat{S}_m = \sum_i w_i^2 z_i^2 - \sum_{k=0}^{m} \sum_{l=0}^{k} a_{kl}^2$$

without assuming anything but orthonormality.

For a polynomial approximation to z, the orthonormal functions must also be polynomials. To assist their definition, let the following order, which corresponds to that of the monomials in equation 3.15, be induced in their generation:

$$(x^k y^l) \text{ precedes } (x^u y^v) \text{ if } k + l < u + v$$

$$\underline{\text{or}} \quad \text{if } k + l = u + v \text{ and } l < v.$$

Thus, the indices of the functions in equation 3.16 correspond to those of the monomials in equation 3.15. Now let each orthonormal function, $p_{kl}(x,y)$, be a linear combination of the monomial $x^k y^l$ and all the monomials which precede it, i.e.

$$p_{00}(x,y) = [\,1\,]$$

$$p_{10}(x,y) = [\,1\,,\, x\,]$$

$$p_{11}(x,y) = [\,1\,,\, x\,,\, y\,]$$

$$\vdots$$

$$p_{kl}(x,y) = [\,1\,,\, x\,,\, y\,,\, x^2\,,\, \ldots\,,\, x^k y^l\,]$$

It may be seen that the polynomials are in groups, each of which corresponds to a given degree, which, in turn, equals k.

It is asserted that each of these orthogonal polynomials is non-trivially (i.e. with a non-zero coefficient) related only to the polynomials of the two preceding groups, and to the preceding polynomials (if there are any) in the current group.

Proof

Assume that each polynomial is related to all the preceding ones, i.e. for $k \geq 1$ and $l \geq 1$, let

$$b_{kl}\, p_{kl}(x,y) = y\, p_{k-1,l-1}(x,y) - \sum_{v=0}^{l-1} c_{kv}\, p_{kv}(x,y)$$
$$- \sum_{v=0}^{k-1} c_{k-1,v}\, p_{k-1,v}(x,y) - \sum_{v=0}^{k-2} c_{k-2,v}\, p_{k-2,v}(x,y)$$
$$- \sum_{u=0}^{k-3} \sum_{v=0}^{u} c_{uv}\, p_{uv}(x,y) \qquad \ldots\ldots(3.17)$$

For $l = 0$, y is replaced by x in the relationship. This is left aside for the moment.

Treating equation 3.17 as a vector relationship, multiplying both sides by a weighted element of an arbitrary member of the orthonormal set, $p_{\ }(x,y)$, say, and summing over the data points:

$$b_{kl} \sum_i w_i^2 p_{rt}(x_i,y_i)\, p_{kl}(x_i,y_i) =$$

$$- \sum_i w_i^2 p_{rt}(x_i,y_i)\, y_i\, p_{k-1,l-1}(x_i,y_i)$$

$$- \sum_i w_i^2 p_{rt}(x_i,y_i) \sum_{v=0}^{l-1} c_{kv}\, p_{kv}(x_i,y_i)$$

$$- \sum_i w_i^2 p_{rt}(x_i,y_i) \sum_{v=0}^{k-1} c_{k-1,v}\, p_{k-1,v}(x_i,y_i)$$

$$- \sum_i w_i^2 p_{rt}(x_i,y_i) \sum_{v=0}^{k-2} c_{k-2,v}\, p_{k-2,v}(x_i,y_i)$$

$$- \sum_i w_i^2 p_{rt}(x_i,y_i) \sum_{u=0}^{k-3} \sum_{v=0}^{u} c_{uv}\, p_{uv}(x_i,y_i) \quad \ldots\ldots(3.18)$$

Set $r = t$ in equation 3.18. Using the orthonormality condition and the fact that the summations may be interchanged, it is obtained directly that

$$c_{kv} = \sum_i w_i^2 y_i\, p_{k-1,l-1}(x_i,y_i)\, p_{kv}(x_i,y_i), \qquad 0 < v \leq l-1$$

In a similar fashion, by setting $r = k-1$, $r = k-2$ and $r = k-3$ in equation 3.18, it is found that

$$c_{k-1,v} = \sum_i w_i^2 y_i\, p_{k-1,l-1}(x_i,y_i)\, p_{k-1,v}(x_i,y_i), \qquad 0 \leq v \leq k-1,$$

$$c_{k-2,v} = \sum_i w_i^2 y_i\, p_{k-1,l-1}(x_i,y_i)\, p_{k-2,v}(x_i,y_i), \qquad 0 \leq v \leq k-2$$

and

$$c_{uv} = \sum_i w_i^2 y_i p_{k-1,l-1}(x_i, y_i)\, p_{uv}(x_i, y_i) ,$$

$$0 \leq u \leq k-3 ; \quad 0 \leq v \leq u \qquad \ldots\ldots(3.19)$$

The value of b_{kl} can be calculated from the normality condition. Equation 3.17 can be rearranged, to give

$$y\, p_{k-1,l-1}(x,y) = b_{kl}\, p_{kl}(x,y) + [\text{ p's of lower degrees}] \quad \ldots\ldots(3.20)$$

Replacing k by $(k-3)$ in equation 3.20, and substituting for the expression $y\, p_{uv}(x,y)$ in equation 3.19, it is readily deduced, by orthogonality, that

$$c_{uv} = 0 \quad \text{for all} \quad [0 \leq u \leq k-3 ; \; 0 \leq v \leq u]$$

This part of the assertion is thus proved. The special case of $l = 0$ can be proved in exactly the same way.

The formula is valid as long as $1/2\,(m+1)(m+2) \leq n$. By reasoning analogous to curve fitting, it can be shown that the greatest computational precision is likely with the independent variables scaled to lie in the square $[-2,+2]$.

3.3.3 Rounding error

The problem of rounding error is more severe with surface fitting than with curve fitting, if for no other reason than that there are usually many more vectors to compute and normalise. It is noted that the error analysis of Section 3.2.2 relies only upon the condition of orthonormality, and may therefore be applied to surface fitting. There is one difference - for curves, each degree corresponds to a single orthonormal polynomial and its associated coefficient, but for surfaces, each degree corresponds to a whole set of polynomials and their associated coefficients. Therefore, the magnitude of the first order error and the correction for it can be derived in a

similar way to that of curve fitting, but by considering the effect of adding a single polynomial instead of raising the degree by one.

With curves, further computation is deemed pointless when two coefficients that correspond to successive degrees are swamped by their estimated rounding error, and these coefficients are excluded from the final approximation. When fitting surfaces, there are several possible tests, namely

a) All the coefficients of one degree are swamped by error - the entire set is excluded from the final approximation.

b) As (a), but applied to two successive degrees.

c) Coefficients which are swamped by their estimated rounding errors are simply excluded from the final approximation, at any time in the computation.

Test (b) is the direct equivalent of the corresponding test for curve fitting, but is unlikely in practice, except when fitting very high order polynomials.

3.3.4 Imposition of constraints on the fit

The only case considered here is that of bi-variate surface fitting. Cases of repeated curve fitting are treated by imposing two-dimensional constraints on the curves.

The imposition of constraints on surfaces is carried out in a similar way to curves. The approximating function is split up into three parts, namely

$$f(x,y) = q(x,y)\, g(x,y) + r(x,y)$$

where

$r(x,y)$ supplies the wanted values of the function (and the derivatives, if so desired) at the specified points,

$q(x,y)$ 'zeroes' the product $q(x,y)\, g(x,y)$ at these points,

and $g(x,y)$ is the polynomial fitted by the orthogonal polynomials procedure to the transformed variable

$$Z_i = \frac{z_i - r(x_i, y_i)}{q(x_i, y_i)} \quad , \quad 1 \leq i \leq n$$

with weights $w_i = q(x_i, y_i)$.

As with curve fitting, the function $q(x,y)$ can be implicitly included in $f(x,y)$ by setting the initial orthogonal polynomial to $q(x,y)$, which is much more efficient.

An important fact is that the dimensionality of the constraints is always the same as that of the space of the independent variables, i.e. with plane curves, constraints can be imposed at points; with surfaces constraints can only be imposed along lines (or curves). Thus, strictly speaking, the above method forces values of $f(x,y)$ along the contour $q(x,y) = 0$, i.e. along the projection of $q(x,y)$ on the xy-plane. Conditions at a single point can only be controlled by heavy, discrete weights at (and near) the point.

The last point is one of the advantages of surface-fitting by repeated curve-fitting for general use, but, for ship forms, inter-patch continuity would be imposed along space-curves.

3.3.5 Deduction of optimum degree of polynomial by analysis of the residuals

The methods that can be used the optimum degree of the approximating surface are similar to those available for curves - Section 3.2.4.

In practice, the first method described (checking that the maximum deviation normal to the fitted function is below a given

bound) leads to numerical difficulties. The calculation of the deviation normal to a curve already involves an iterative procedure; the equivalent procedure for a surface (other than a plane) is basically highly unstable. Whilst it can be implemented, the statistimethod offers distinct advantages.

The statistical method, which relies only on the condition of orthonormality and on the normal distribution of the errors in the data, is directly applicable to surface fitting. Nevertheless, there remains the same difficulty as for the rounding error analysis - Section 3.3.3 - namely whether to discard a single orthogonal polynomial as insignificant, a whole set, or two successive sets. By analogous reasoning, the last is the logical choice. In practice, it is unlikely to find two completely insignificant sets, but rather 'almost' insignificant ones, i.e. sets with a low proportion of statistically significant coefficients. Such cases do not lend themselves to convenient computer analysis, and are classical examples where the user's intuition is more useful than background mathematics.

3.4 Discussion

The various ideas described in this chapter have been implemented in a general program for least squares curve fitting by polynomials, and a less comprehensive surface fitting program. Although no specific results are given here, the former has been used extensively for the work described in Chapter 5 (Transformations), and a part of it in Chapter 6 (Parametric Curve Fitting). Also, a modification of the latter has been used for the work of Chapter 7 (Parametric Surface Fitting).

CHAPTER 4 :-

MATHEMATICAL DIVISION OF THE HULL SURFACE

4.1 Introduction

From the foregoing chapters, it is clear that even the simplest hulls consist of combinations of surfaces of distinct functional forms. Therefore, no universal function has been found to describe an entire hull satisfactorily. Oddly, it is the mathematically simplest form, i.e. plane surfaces, which, when joining higher order surfaces, create the greatest difficulties. This is because the higher order functional forms that describe the adjacent patches tend to oscillate about the plane surfaces. From the survey of present methods, it appears to be accepted generally that it is simpler to divide the hull into patches of different functional forms, which may then faired individually, rather than to attempt to find a universal function.

After the division, the patch boundaries should be smoothed, preferably by an analytical form, to facilitate their mathematical definition. The data defining the bounderies may itself be awkward to fit. It is found in practice that most fairing methods are sensitive to the positions of the patch boundaries. This, plus the fact that the idea of division is unnatural from the naval architect's point of view, implies the advantages to be had from a formalised approach to the division problem.

The techniques described in this chapter were devised specifically for the isolation of planar regions, in order to identify the start of the Flat-of-Bottom (FOB) and the Parallel-Middile-Body (PMB) regions of hulls. They can be applied, in modified form, to the isolation of other functional forms.

The analyses are restricted to curves, rather than surfaces, because engineering surfaces are normally defined by mutually orthogonal two-dimensional views, and the division is performed

prior to the three-dimensional fairing.

It is assumed that the slope of the linear region to be identified is known, although some of the methods described below are applicable to cases in which it is not. In the case of the FOB (line AB in Fig. 4.1), the slope, known as 'Rise-of-Floor', is part of the hull specification; the other possible straight line is the side of the hull (line CD in Fig. 4.1), which would either be vertical, or its slope would be given. For the PMB (line AB in Fig. 4.2), the slope is clearly zero. Apart from the slopes, the other information available normally is the table of offsets. For the FOB this is often not enough, and more data is usually given, by the functional form of the curved segment (BC) joining the straight lines.

Several possible solutions to the division problem are described. In principle, the curve to be divided is fitted by some function, e.g. a polynomial, in a piecewise fashion, until the piece is found that is either entirely linear or brackets a section of the right slope.

The first two methods, which use a quadratic and a cubic local fit respectively, require the knowledge of the approximate position of the breakpoint, for stability. The third, which is a pseudo-curvature method, is a means of locating the approximate position of the breakpoint. It may also be used to simultaneously check for errors in the data. The fourth method, using splines, is an elaboration of the cubic method, which obviates the need for the approximate position, and makes more efficient use of the given data.

The advantages and limitations of all the methods are described. The results, including some outright failures, are presented together, for ease of comparison.

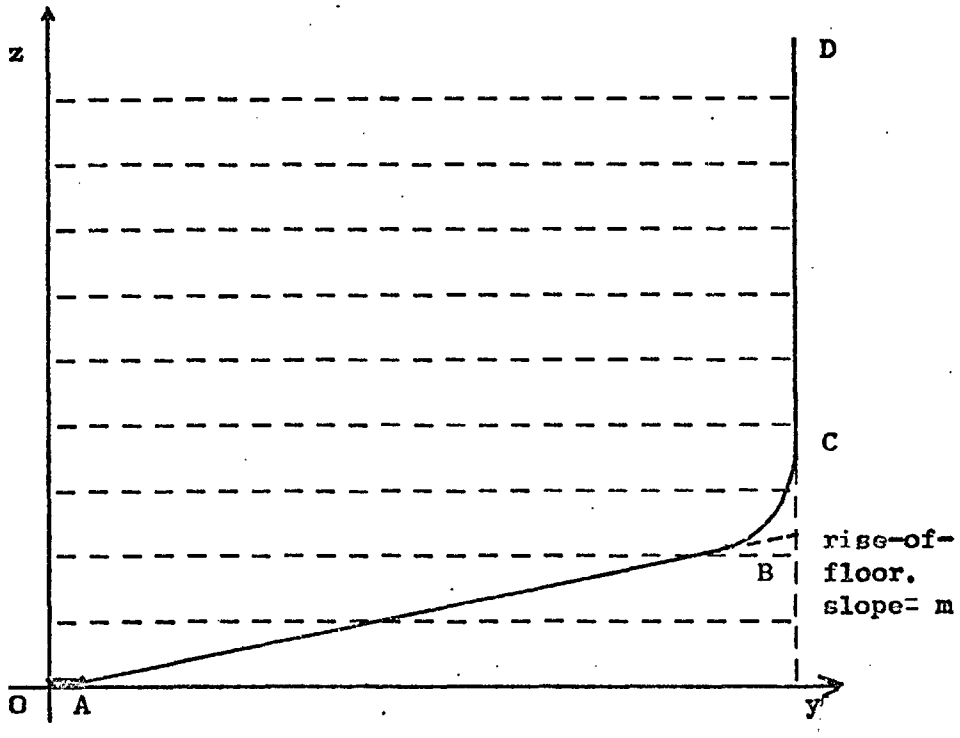


Figure 4.1 :- Flat-of-bottom and bilge analysis.

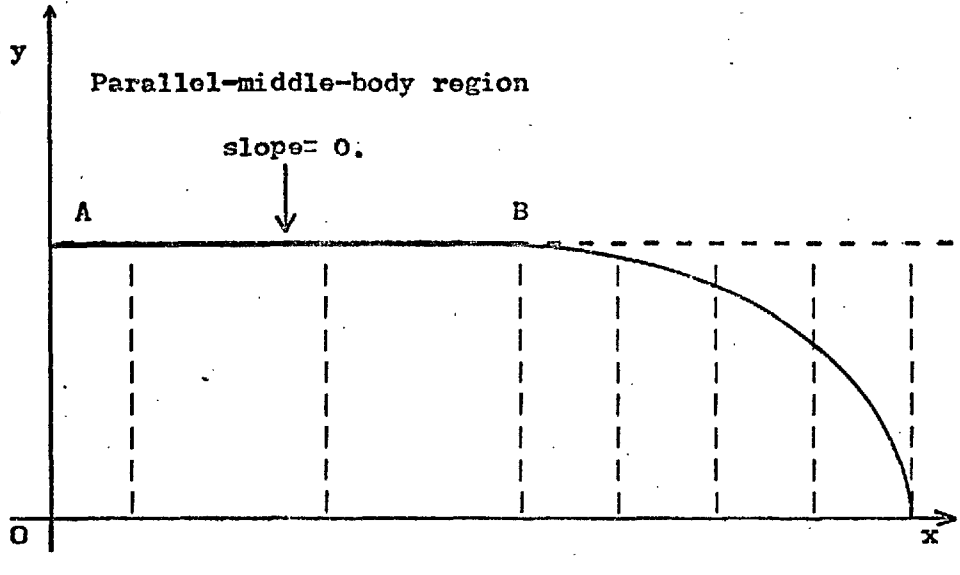


Figure 4.2 :- Parallel Middle Body analysis.

4.2 The Quadratic Method

Consider the points P, Q and R in Fig. 4.3, where P and Q are given data points, and R is the sought breakpoint.

Let the region QPR be described by the parabola

$$y = a_o + a_1 x + a_2 x^2 \qquad \ldots\ldots (4.1)$$

Then, substituting the co-ordinates of points P and Q in equation 4.1, one obtains

$$y_p = a_o + a_1 x_p + a_2 x_p^2 \qquad \ldots\ldots (4.2)$$

and

$$y_q = a_o + a_1 x_q + a_2 x_q^2 \qquad \ldots\ldots (4.3)$$

Let the (known) straight line SR be given by

$$y = m x + b$$

Then, at the (unknown) point R, the parabola and the straight line intersect. Therefore

$$m x_r + b = a_o + a_1 x_r + a_2 x_r^2 \qquad \ldots\ldots (4.4)$$

Also, assuming continuity of first derivative, SR is tangential to the parabola at R. By differentiating equation 4.1 and equating the derivative to m, one obtains

$$a_1 + 2 a_2 x_r = m \qquad \ldots\ldots (4.5)$$

Either by solving these equations, or from the geometric properties of the parabola, the following expression is obtained

$$\left[\frac{x_p - x_r}{x_q - x_r}\right]^2 = \frac{y_p - m x_p - b}{y_q - m x_q - b} = \beta^2 \quad , \text{ say}$$

Then, if β is real, the two roots are given by

$$x_r = \frac{\pm\beta x_q - x_p}{\pm\beta - 1}$$

If an approximation to the breakpoint is provided, or, at least, its position relative to x_p and x_q is specified, the appropriate root may be chosen, assuming the two roots are sufficiently far apart for the correct one to be easily distinguishable. The meaning of a complex β is that there is no parabola that satisfies the conditions - a very unlikely event, except through errors in data.

The low order of the parabola makes it somewhat lacking in flexibility. It is therefore quite simple to construct failing cases for this method. Also, the analysis involves only the data points next to the breakpoint, thus making rather inefficient use of the given data, and requiring an auxiliary method to find those points.

4.3 The Cubic Method

Fitting a parabola to the segment PQR (Fig. 4.3) implies a discontinuity in the second derivative at the breakpoint, R. For second derivative continuity at the joint, a cubic must be fitted. This is possible, using just the two neighbouring data points, as before, the extra data demanded by the cubic fit being provided by the second derivative continuity.

Let the cubic be of form

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \qquad \ldots\ldots (4.6)$$

Let the (known) straight line be

$$y = m x + b \quad , \text{ as before.}$$

Then, at points P and Q,

$$y_p = a_o + a_1 x_p + a_2 x_p^2 + a_3 x_p^3 \qquad \ldots\ldots (4.7)$$

$$y_q = a_o + a_1 x_q + a_2 x_q^2 + a_3 x_q^3 \qquad \ldots\ldots (4.8)$$

From functional continuity at point R,

$$m x_r + b = a_o + a_1 x_r + a_2 x_r^2 + a_3 x_r^3 \qquad \ldots\ldots (4.9)$$

From continuity of first derivative at point R,

$$m = 2 a_2 x_r + 3 a_3 x_r^2 \qquad \ldots\ldots (4.10)$$

From continuity of second derivative at point R,

$$0 = 2 a_2 + 6 a_3 x_r \qquad \ldots\ldots (4.11)$$

Solving these equations for x_r or making use of the geometrical properties of cubics, leads to the following equation in x_r

$$\left[\frac{x_p - x_r}{x_q - x_r}\right]^3 = \frac{y_p - m x_p - b}{y_q - m x_q - b} = \beta^3 \quad , \text{ say.}$$

Then, taking the (only) real root,

$$x_r = \frac{\beta x_q - x_p}{\beta - 1}$$

The cubic is clearly a more powerful method than the quadratic. There is no ambiguity about the solution of the equation, and the second derivative continuity is of practical importance. However, like the quadratic, it is a purely localised method, and, therefore, the data points adjacent to the breakpoint must be found before it can be applied.

A method will now be described, that locates the segment containing the breakpoint.

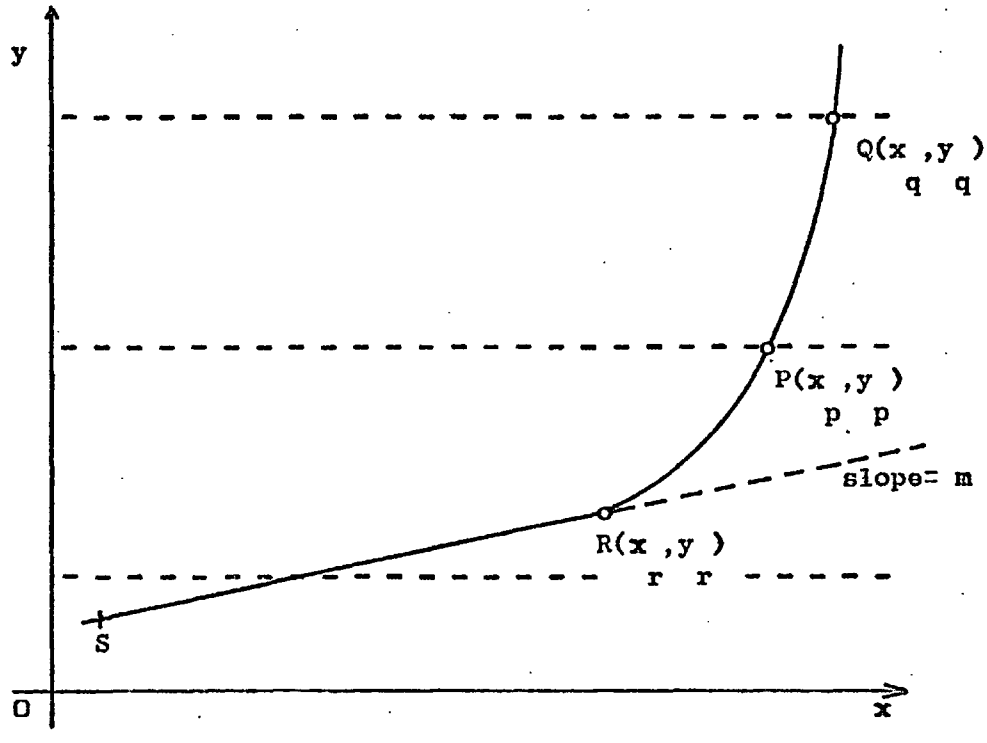


Figure 4.3 :- Detection of the starting point of a linear region by fitting quadratic or cubic.

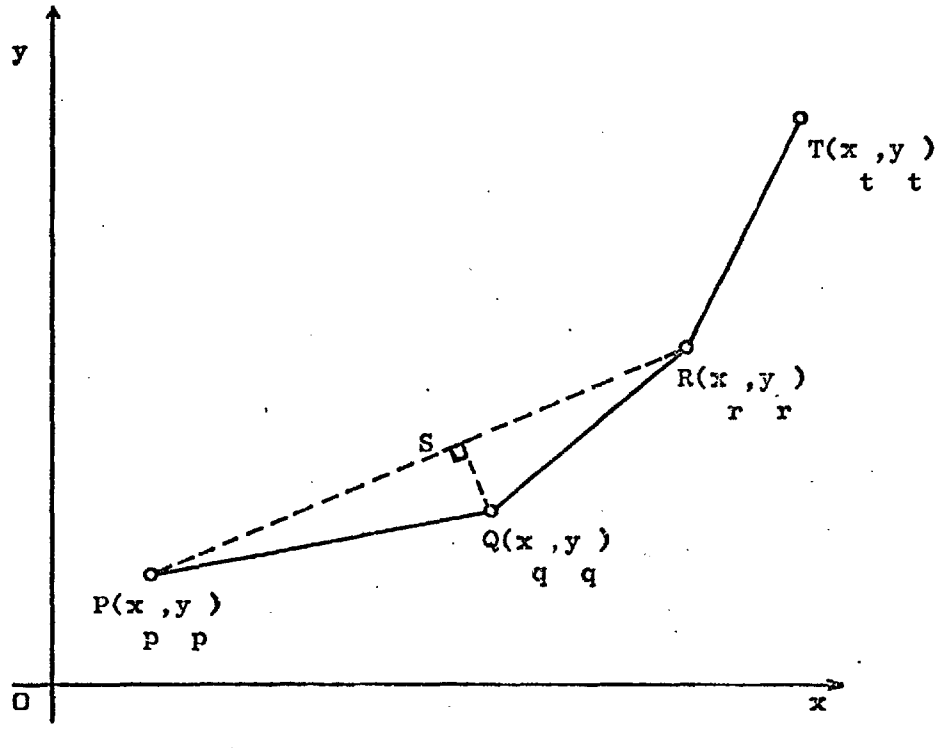


Figure 4.4 :- Pseudo-curvature analysis.

4.4 The Pseudo-Curvature Method

Both foregoing division methods require the two data points adjacent to the breakpoint on the curved side. If a straight line can be identified, the entering or leaving a linear region can be detected. Refering to Fig. 4.4, there are many ways of deciding whether the points P, Q and R lie on a straight line, vis.

a) Compute area of triangle PQR, given by

$$A = \frac{1}{2} \det \begin{vmatrix} 1 & y_p & x_p \\ 1 & y_q & x_q \\ 1 & y_r & x_r \end{vmatrix}$$

If the points are on a straight line, the area is zero.

b) Estimate the second derivative at Q, using the standard finite difference formula

$$\frac{\Delta^2 y}{\Delta^2 x} = \frac{(y_q - y_p)(x_r - x_p) - (y_r - y_p)(x_q - x_p)}{(x_q - x_p)(x_r - x_p)(x_r - x_q)}$$

In the straight line region, the second derivative is zero.

c) Estimate the curvature at Q by finite differences, i.e. compute

$$\varrho = \frac{\tan^{-1}\left[\frac{y_q - y_p}{x_q - x_p}\right] - \tan^{-1}\left[\frac{y_r - y_q}{x_r - x_q}\right]}{\sqrt{(x_q - x_p)^2 + (y_q - y_p)^2 + (x_r - x_q)^2 + (y_r - y_q)^2}}$$

If P, Q and R lie in a straight line, $\varrho = 0$.

Method (a) is simple and sound, but informs neither of the overall behaviour of the data, nor of its correctness. Method (b)

is an improvement in both respects, and has been used by Berger et al., (1966) to detect errors in data. It is as simple to compute as (a), but is orientation dependent. Method (c) is orientation independent, but is more complicated. It also uses a trigonometric function, making it more sensitive to computatational rounding errors. The method finally selected has the combined advantages of all three, without some of the drawbacks.

Define 'pseudo-curvature' at point Q by the height of the triangle PQR, normalised with respect to its base, i.e.

$$\rho^{*} = \frac{QS}{RP}$$

$$= \frac{(x_p - x_q)(y_p - y_r) - (x_p - x_r)(y_p - y_q)}{(x_r - x_p)^2 + (y_r - y_p)^2}$$

ρ^{*} has properties similar to those of ρ. The magnitude of ρ^{*} indicates the deviation of the data points from a straight line; changes in the sign of ρ^{*} indicate inflexions, the number and positions of which permit errors to be detected. It is simple to compute, except at the end-points, where all second order methods fail, anyway. The points next to the breakpoint are found by taking the points in successive triplets (which, refering to Fig. 4.4, means that having processed the set PQR, one of the points, say P, is discarded and point T is included, so that the new triangle to be processed is QRT), computing the value of ρ^{*} for each triplet, and checking for a change of state from linear to non-linear or vice-versa. The method does, however, have its limitations, as follows:-

a) There is an infinity of triangles with apexes at three specific points, which all give rise to the same value of ρ^{*} (Fig.

4.5). For meaningful results, the angles $\hat{P}$ and $\hat{R}$ should be non-obtuse. This limitation is not serious, beacuse the check for acuteness is simple, and even if it is not carried out, any undesirable behaviour of the data should be noticed when the next point is considered.

b) It is extremely unlikely that a value of ρ^* of exactly zero would be obtained when processing real data, especially in view of possible data errors. Therefore, some small, positive parameter, ϵ, say, must be chosen, and a region 'accepted' as linear if $|\rho^*| < \epsilon$. The value of ϵ must be a compromise between the accuracy required and that which can reasonably be expected from the data.

c) The values computed for ρ^* depend on the relative scaling of the variables. Therefore, the value of ϵ is not universal, and must be selected according to the data. Usually, $\epsilon = 0$ [0.001].

d) The definition of ρ^* implies that a linear region must span at least three points in order for it to be detected.

Despite these limitations, the simplicity of the method made the investigation worthwhile.

4.5 Spline Methods

Splines are essentially a piecewise fit by cubics. Their application to lines' fairing is an important subject in its own right (Berger et al., 1966 ; Theilheimer and Starkweather, 1961). Here they are considered merely as a tool to identify linear regions.

Given the data points (x_i , y_i), $1 \leq i \leq n$.

Let them be fitted exactly by a set of $(n - 1)$ cubics, each of which is valid between one pair of points only, in such a way that continuity of up to the second derivative is maintained at the points.

Consider the ith interval, between (x_i, y_i) and (x_{i+1}, y_{i+1}). Then, for $x_i \leq x \leq x_{i+1}$, let y be given by

$$y = C_{i,1}(x_{i+1} - x)^3 + C_{i,2}(x - x_i)^3 + C_{i,3}(x_{i+1} - x) + C_{i,4}(x - x_i) \quad \ldots\ldots (4.12)$$

where C is the $(n - 1) * 4$ matrix of spline coefficients. The derivation of C is given in Appendix D. The following points are noted:-

a) Unless specified, the values of the second derivative at the end-points must be estimated or taken to be zero (note similarity in this respect with other methods of this chapter).

b) The splines are orientation dependent. That is, if the data happens to represent a regular function, the spline fit is a valid approximation to the function only if the values of the first derivative implied by the data are small. Computationally, this is manifested by terms in $\frac{1}{(x_{i+1} - x_i)}$ in the derivation of C.

c) The data should be one-valued, i.e. the x values should increase or decrease monotonically.

The obvious way to locate the start of linear segments (or, for that matter, any segment of a given slope) is to fit the data with splines; then, by differentiating the spline in each interval, the interval containing the breakpoint can be found. The derivative of the corresponding spline, which is a quadratic, can then be solved for the exact position of the breakpoint.

The straight-forward implementation of this is prevented by property (b), particularly for ship lines. One way to overcome this problem is to generate (n - 1) co-ordinate systems, with their

origins at the first (n - 1) points and the straight line between each pair of points taken as the corresponding abscissa, and to fit the splines with respect to these (Asker, 1962). As this is a somewhat inefficient process, a better way is to use 'parametric splines' (APT, 1967), as follows:-

Let $(u_i, 1 \leq i \leq n)$ be a vector of monotonically increasing or decreasing values of some parameter.

Let the sets of x values and y values, each be fitted by splines in terms of u, i.e. for each interval $u_i \leq u \leq u_{i+1}$

$$x = C^{(x)}_{i,1}(u_{i+1} - u)^3 + C^{(x)}_{i,2}(u - u_i)^3 + C^{(x)}_{i,3}(u_{i+1} - u) + C^{(x)}_{i,4}(u - u_i) \quad \ldots\ldots (4.13)$$

and

$$y = C^{(y)}_{i,1}(u_{i+1} - u)^3 + C^{(y)}_{i,2}(u - u_i)^3 + C^{(y)}_{i,3}(u_{i+1} - u) + C^{(y)}_{i,4}(u - u_i) \quad \ldots\ldots (4.14)$$

where the superscripts denote the particular set of coefficients.

Then $$\frac{dy}{dx} = \frac{dy}{du} \Big/ \frac{dx}{du}$$

and any value of $\frac{dy}{dx}$ implied by the data can be obtained by relative scaling of $\frac{dx}{du}$ and $\frac{dy}{du}$, whilst keeping the last two within easily manageable limits.

In practice, a convenient way to generate the parameter vector vector is to compute the accumulative linearised arc-length between the data points, i.e.

Let $u_1 = 0$

and $$u_i = u_{i-1} + \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

for $2 \leq i \leq n$.

Then, in each interval, taking extreme cases of horizontal and vertical intervals, it may be seen that the finite difference estimate of the mean derivative in the ith interval behaves well, that is

$$0 \leq \left[\frac{\Delta y}{\Delta u}\right]_i = \frac{|y_i - y_{i-1}|}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \leq 1$$

and similarly for x. If necessary, u may be scaled down further, to control the parametric gradients.

Assume that the interval containing the breakpoint is known - let it be the ith interval. The exact value, (x_r, y_r), must be computed indirectly, by computing the corresponding value of u, u_r, first. The last may be found in three distinct ways :-

a) By computing the point of intersection of the ith spline with the given straight line, i.e. with $y_r = m x_r + b$.

Substituting for x_r and y_r, in terms of u_r, from equations 4.13 and 4.14, rearranging and simplifying, the following equation is obtained for u_r :-

$$a_3 u_r^3 + a_2 u_r^2 + a_1 u_r + a_0 = 0 \quad \ldots\ldots (4.15)$$

where it is convenient to define p, q, v and w as

$$p = C_{i,1}^{(y)} - m\, C_{i,1}^{(x)} \quad ; \quad q = C_{i,2}^{(y)} - m\, C_{i,2}^{(x)}$$

$$v = C_{i,3}^{(y)} - m\, C_{i,3}^{(x)} \quad ; \quad w = C_{i,4}^{(y)} - m\, C_{i,4}^{(x)}$$

and $a_3 = q - p$

$$a_2 = 3 (p u_{i+1} - q u_i)$$

$$a_1 = 3 (q u_i^2 - p u_{i+1}^2) + w - v$$

$$a_0 = p u_{i+1}^3 - q u_i^3 + v u_{i+1} - w u_i - b$$

Equation 4.15 can have one or three real roots, none or all of which may be inside the ith interval.

b) By computing the position of the point at which the ith spline has the given slope, m.

$$\text{Given} \quad \left.\frac{dy}{dx}\right|_{u_r} = m \qquad \ldots\ldots (4.16)$$

By differentiating equations 4.13 and 4.14, substituting in equation 4.16, rearranging and simplifying, or by observing that equation 4.16 is the derivative of equation 4.15, the following quadratic is obtained for u_r :-

$$3 a_3 u_r^2 + 2 a_2 u_r + a_1 = 0 \qquad \ldots\ldots (4.17)$$

where a_1, a_2 and a_3 are as defined above.

Geometrically, the roots of equation 4.17 are the positions of the maximum and minimum distances of the spline from the straight line. There may be no real roots, coincident roots or two distinct roots, none or any of which may be in the permissible range.

c) By computing the position of the point at which the ith spline has zero second derivative (i.e. second derivative continuity with the straight line).

Given $\left.\dfrac{d^2y}{dx^2}\right|_{u_r} = 0$

Now,
$$\frac{d^2y}{dx^2} = \frac{\dfrac{dx}{du}\dfrac{d^2y}{du^2} - \dfrac{dy}{du}\dfrac{d^2x}{du^2}}{\left[\dfrac{dx}{du}\right]^3} \qquad \ldots\ldots (4.18)$$

Hence, if $\left.\dfrac{dx}{du}\right|_{u_r} \neq 0$, u_r is the root of

$$\frac{dx}{du}\frac{d^2y}{du^2} = \frac{dy}{du}\frac{d^2x}{du^2} \qquad \ldots\ldots (4.19)$$

By differentiating equations 4.13 and 4.14 once and twice, substituting in equation 4.19, rearranging and simplifying, the following quadratic is obtained for u_r :-

$$b_2 u_r^2 + b_1 u_r + b_0 = 0 \qquad \ldots\ldots (4.20)$$

where it is again convenient to define p and q as

$$p = C^{(x)}_{i,4} - C^{(x)}_{i,3} \quad \text{and} \quad q = C^{(y)}_{i,4} - C^{(y)}_{i,3} ,$$

and b_2 , b_1 and b_0 are given by

$$b_2 = 3\,[u_{i+1} - u_i]\,[C^{(x)}_{i,2} C^{(y)}_{i,1} - C^{(x)}_{i,1} C^{(y)}_{i,2}]$$

$$b_1 = p\,[C^{(y)}_{i,1} - C^{(y)}_{i,2}] - q\,[C^{(x)}_{i,1} - C^{(x)}_{i,2}] - b_2\,[u_{i+1} + u_i]$$

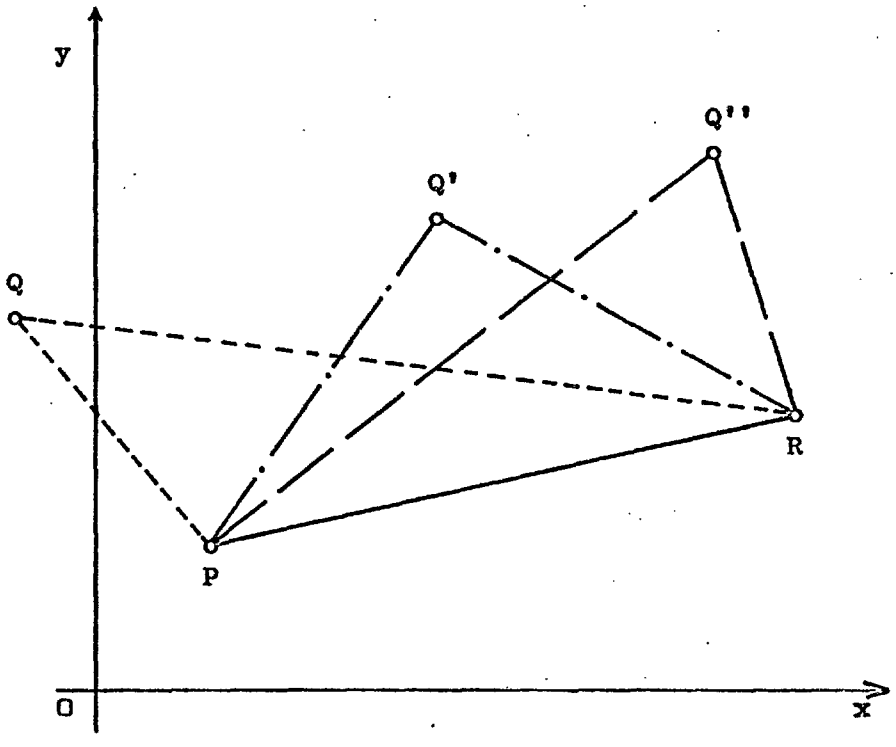


Figure 4.5 :- Illustration that an infinity of triangles give the same value of pseudo-curvature.

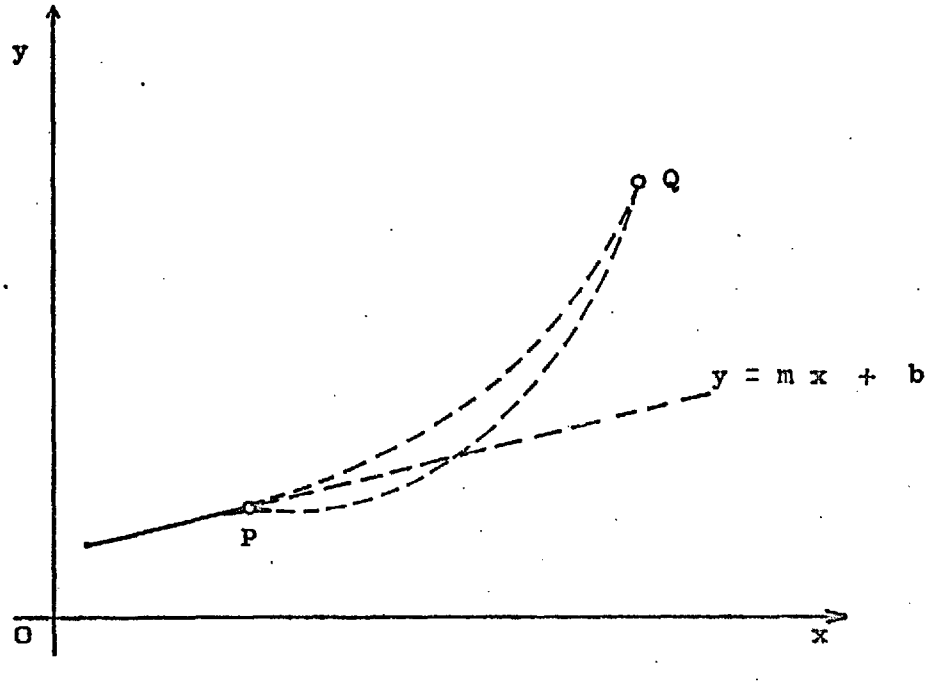


Figure 4.6 :- Possible behaviours of the spline in the interval containing the breakpoint.

$$\text{and } b_0 = q\,[C^{(x)}_{i,1}\,u_{i+1} - C^{(x)}_{i,2}\,u_i] - p\,[C^{(y)}_{i,1}\,u_{i+1} - C^{(y)}_{i,2}\,u_i]$$

$$+ b_2\,u_i\,u_{i+1}$$

Here, again, there is the possibility of none, one or two valid solutions.

Theoretically, either method may also be used to find the interval containing the breakpoint. Given the information about the straight line, and having fitted the data points with splines, either equation 4.15, 4.17 or 4.20 may be solved in each interval and the roots, if real, checked for being inside the interval, until the correct root is found. In practice, similar problems arise with all three methods. They are best illustrated pictorially. Assume that the breakpoint is not exactly at one of the given data points. Then, refering to Fig. 4.6, the spline in the relevant interval cannot describe both a curved and a linear region, and must therefore oscillate about the given straight line. Hence each of three methods can, produce more than one theoretically valid solution. Moreover, method (a) fails in cases where the spline functions do not intersect the straight line, and method (c) may detect an inflexion point which is totally unrelated to the sought linear region. Consequently, it appears that only method (b) may be applied in a mechanical fashion, especially if there is no prior knowledge about the breakpoint.

The analysis of method (c) is illuminating about the general properties of parametric splines. It may be seen from equation (4.18) that they are extremely flexible, inasmuch as they have an infinite range of curvature. Each spline has two inflexions, and, of course, the basic reason for their choice in this context is their ability

to describe infinite y/x slopes. Later in the work, these properties were used as the basis for a complete fairing technique.

4.6 Results

The methods described in this chapter were applied to data extracted from Appendix A. The description of data used is presented in Table 4.1, and the results in Tables 4.2 and 4.3, using the following notation:-

a) x_r = co-ordinate of breakpoint. Sometimes, this is not in the conventional x-direction, but the units are consistent for each set of data, and are usually a percentage of the corresponding hull dimension.

b) The breakpoints given in Table 4.1 are visual estimates.

c) For waterline data, only the afterbody breakpoint was computed.

d) The letter 'F' as an entry indicates a failure of the particular method for the corresponding data. Note: some data were used, that have no linear region. In such cases, 'F' is the correct result.

e) The code 'N.A.' as an entry indicates that the particular method was not applicable to the corresponding data, e.g. a slope dependent method when the slope was not given.

f) The value used for ϵ in the 'pseudo-curvature' method was 0.01. This was found to be an acceptable compromise for most data.

g) For testing purposes, the 'quadratic' and the 'cubic' methods were applied to the interval which visually contained the breakpoint (as given in Table 4.1).

h) The 'Agreement' column is an overall qualitative assessment of the agreement between several methods and the visually correct breakpoint.

Table 4.1 :-

List of data sets used for testing hull division methods

a) Model 1

Ident. No.	Description of Data	No. of Points	Straight line Equation	Breakpoint	
				Interval	x_r
1	Section at Station 1 1/2	11	.201 x - .141	1	0.7
2	Section at Station 2	11	.201 x - .141	1	0.7
3	Section at Station 2 1/2	11	.201 x - .141	1	10.1
4	Section at Station 3	11	.201 x - .141	1	16.8
5	Section at Station 4	11	.201 x - .141	2	40.5
6	Section at Station 5	11	.201 x - .141	3	54.7
7	Section at Station 6	11	.201 x - .141	2	40.6
8	Section at Station 7	11	.201 x - .141	1	21.3
9	Section at Station 7 1/2	11	.201 x - .141	1	14.2
10	Section at Station 8	11	.201 x -.141	1	6.5
11	Section at Station 8 1/2	11	.201 x - .141	1	0.7
12	Stern Profile	13	-35.714 x - 126.071	7	1.2
13	Bow Profile	15	none	----	-----
14	60 p.c. Waterline	18	y = 100.0	----	-----
15	80 p.c. Waterline	19	y = 100.0	9	45.0
16	100 p.c. Waterline	23	y = 100.0	11	38.0
17	120 p.c. Waterline	23	y = 100.0	11	35.0
18	140 p.c. Waterline	23	y = 100.0	11	32.0
19	160 p.c. Waterline	23	y = 100.0	9	25.0
20	Bulwarks Curve	25	y = 100.0	10	22.0

Table 4.1 (continued)

b) Model 2

Ident. No.	Description of Data	No. of Points	Straight line Equation	Breakpoint	
				Interval	x r
21	Section at Station 1/2	11	y = 0	1	2.8
22	Section at Station 1	12	y = 0	1	8.2
23	Section at Station 1 1/2	12	y = 0	1	23.1
24	Section at Station 2	12	y = 0	1	42.4
25	Section at Station 2 1/2	12	y = 0	1	63.3
26	Section at Station 3	12	y = 0	1	78.9
27	Section at Station 3 1/2	12	y = 0	1	86.0
28	Section at Station 4	12	y = 0	1	86.0
29	Section at Station 8	12	y = 0	1	82.5
30	Section at Station 8 1/2	12	y = 0	1	71.4
31	Section at Station 9	12	y = 0	1	44.3
32	Section at Station 9 1/2	12	y = 0	1	12.7
33	Section at Station 9 3/4	12	y = 0	1	4.2
34	Section at Station 10	11	y = 0	1	0
35	Waterline B	20	y = 100.0	8	37.0
36	Waterline C	20	y = 100.0	7	34.0
37	Waterline D	20	y = 100.0	6	27.0
38	Waterline E	20	y = 100.0	6	26.0
39	Waterline F	21	y = 100.0	7	25.5
40	Waterline G	21	y = 100.0	6	23.0

Table 4.2:-

Computed Position of Breakpoint, using the Pseudo-curvature the Quadratic and the Cubic Methods.

a) Model 1

Ident. No. of Data	Interval No. by Pseudo-Curvature	x_r by Quadratic	x_r by Cubic	Agreement
1	2 +	12.6	-26.0	very bad
2	F	-2.3	-11.5	very bad
3	F	1.9	-8.5	very bad
4	F	26.9	-5.7	very bad
5	2 +	35.3	25.3	fair
6	2 +	58.5	47.1	good
7	7 (F)	36.6	28.1	good
8	6 (F)	16.8	11.0	fair
9	5 (F)	9.8	2.8	fair
10	5 (F)	3.2	-4.0	bad
11	5 (F)	-1.0	-7.3	bad
12	5 +	-6.9	-6.9	bad
13	3 +	N.A.	N.A.	good (*)
14	4 + (F)	48.8	55.2	very good
15	2 + (F)	43.8	47.4	very good
16	6 +	40.2	46.4	good
17	2 + (F)	38.0	43.0	good
18	11 +	28.2	40.4	good
19	9 +	30.9	37.5	fair
20	9 +	17.7	70.5	bad

(*) Point of inflexion found.

Table 4.2 (continued):-

b) Model 2

Ident. No. of Data	Interval No. by Pseudo-Curvature	x by r Quadratic	x by r Cubic	Agreement
21	F	1.8	-2.1	bad
22	5 +	7.4	-2.9	bad
23	10 (F)	22.9	9.4	fair
24	8 (F)	44.9	32.0	fair
25	7 (F)	66.2	56.4	very good
26	6 (F)	80.8	74.0	very good
27	4 (F)	89.2	84.8	very good
28	4 (F)	93.9	91.3	good
29	5 (F)	87.5	82.8	very good
30	5 (F)	73.8	66.5	very good
31	5 + (F)	46.8	36.5	very good
32	9 (F)	14.9	5.3	good
33	F	4.9	-2.7	fair
34	F	3.3	-1.4	fair
35	8 +	36.7	38.3	very good
36	8 +	33.3	35.8	very good
37	7 +	29.9	33.3	good
38	7 +	27.8	30.1	very good
39	7 +	26.6	28.2	very good
40	7 +	24.6	27.9	very good

Trblo 4.3 :-

Computed Position of Breakpoint, using the Spline Methods

a) Model 1

Ident. No. of Data	Method (a)		Method (b)		Method (c)		Agreement
	Interval No.	x_r	Interval No.	x_r	Interval No.	x_r	
1	1	0.7	F	-----	8 (F)	72.4	fair
2	1	0.7	F	-----	1	0.7	very good
3	1	0.7	F	-----	1	0.7	bad
4	1	0.7	F	-----	1	0.7	bad
5	1	0.7	F	-----	2	25.5	bad
6	3	50.6	2	42.0	2	31.2	fair
7	2	36.0	1	25.1	1	0.7	bad
8	1	18.9	1	11.3	1	0.7	fair
9	1	7.5	1	4.6	1	0.7	bad
10	1	0.7	F	-----	1	0.7	bad
11	1	0.7	F	-----	1	0.7	very good
12	8	-6.8 (1)	6	3.5	10	-7.9	bad
13	N.A.	-----	N.A.	-----	6	-0.1 (2)	good
14	F	----- (1)	8	47.1	1	50.0 (3)	very good
15	9	46.1 (1)	9	46.2	3	9.0	good
16	12	45.0	12	45.1	2	-1.6	fair
17	11	40.0	12	45.5	3	22.7	fair
18	F	----- (1)	12	46.0	3	0.1	very bad
19	10	27.5	10	27.0	1	-0.5	good
20	10	25.0	8	12.3	1	-0.9	bad

Table 4.3 (continued)

b) Model 2

Ident. No. of Data	Method (a)		Method (b)		Method (c)		Agreement
	Interval No.	x r	Interval No.	x r	Interval No.	x r	
21	1	2.8	F	-----	2	10.9	fair
22	2	8.2	1	4.8	1	0	fair
23	2	23.1	1	13.4	1	0	fair
24	2	42.4	1	24.5	1	0	bad
25	2	63.3	1	36.6	1	0	bad
26	2	78.9	1	46.5	1	0	bad
27	1	86.0	1	51.5	1	0	bad
28	2	86.0	1	49.9	1	0	bad
29	2	82.5	1	48.2	1	0	bad
30	2	71.4	1	41.8	1	0	bad
31	2	44.3 (1)	1	25.6	1	0	bad
32	1	6.4	1	7.3	1	0	very bad
33	2	4.2	1	2.4	1	0	fair
34	1	0	F	-----	1	0 (3)	very good
35	8	38.1 (1)	8	37.7	2	9.9	good
36	10	55.0 (1)	8	37.0	2	8.9	bad
37	6	30.0	1	2.5	2	6.2	bad
38	10	55.3 (1)	7	31.9	1	2.5	very bad
39	7	26.1 (1)	7	27.6	1	0	very good
40	6	25.0	7	26.9	1	0	very good

(1) Average of two valid roots.

(2) Inflexion found.

(3) Point of maximum breadth of waterline found.

i) A '+' accompanying an entry indicates that more than one linear region was found. Only the main one was investigated in such cases.

4.7 Discussion and Conclusions

The results obtained using the methods described in this chapter were altogether unsatisfactory. Some of the methods were found to work some of the time, for simple data, but there was to much variation in the quality of the results to even consider the automatic application of any of these methods.

The pseudo-curvature method showed up errors in the input data, as expected. This, possibly excessive, sensitivity throws some doubt on the usefulness of the method for practical fairing.

The quadratic and the cubic methods performed well for waterlines, where the data is eminently suitable. As for the cross-sectional data, the curved parts are usually conics, and cannot be approximated well by low order polynomials, which is confirmed by the results. Surprisingly, there was little difference between the two methods in the quality of the results.

The three spline methods were found to be more reliable, inasmuch as they either failed completely or produced valid, although not always acceptable, solutions. The problem anticipated, i.e. oscillations about the solution, was found to be quite severe, mainly because of the flexibility of the parametric splines.

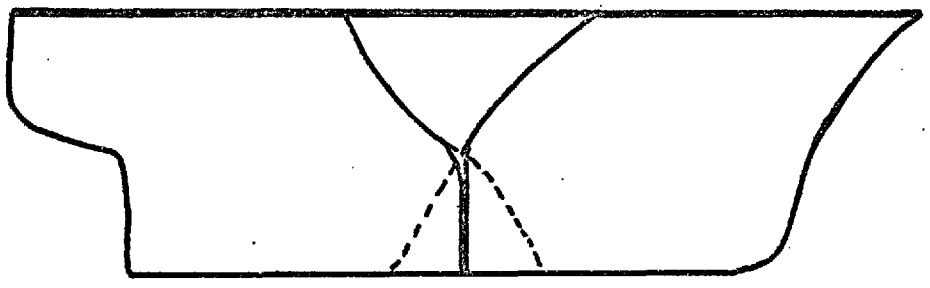
Two further problems were found, which are common to all hull division methods, as follows

a) All the methods depend on accurate input data. This somewhat contradicts the original aim - that of fairing 'out' errors. Usually, the division boundaries would be smoothed after the division, which

reduces the effect of the initial errors, but does not eliminate it.

b) Data, such as the 60 % waterline of Model 1, which have no linear region, may still have to be split, so as to separate the forebody from the afterbody, if the hull has any PMB. Any division algorithm may cause the two breakpoints to overlap, as illustrated in Fig. 4.7. Such a situation is clearly not practical and must be checked.

In conclusion, it was found that the most satisfactory way of separating patches is to estimate the division point manually, i.e. visually, and to improve the estimate using either the quadratic or the cubic method. The pseudo-curvature method may still be applied initially to detect errors in data. The designer's supervision is necessary in all cases. Assuming that it is necessary to divide the hull, a much preferable solution would be to use fairing methods which are insensitive to the exact division boundaries. Experiments with such methods will now be described.



----- Hypohtetical patch boundary.

_____ Practical patch boundary.

Figure 4.7:- Hypothetical overlap of hull surface patch boundaries.

CHAPTER 5 :-

TRANSFORMATIONS

5.1 Introduction

In the previous chapters it was shown that the problem of hull division is not easily soluble. The curves that prove to be the most difficult are those which are made up of several segments, each of a different mathematical type. This chapter is concerned mainly with the analysis of such curves, and the development of practical fitting methods.

Consider the typical stern profile in Fig. 5.1 and the typical parallel-middle-body section in Fig. 5.2. Points lifted from either drawing are unsuitable for direct polynomial fitting. Fig. 5.1 is simply badly oriented, and could probably be fitted with line AO as the abscissa. Fig. 5.2 , in addition to being badly oriented, has straight line regions, making it unsuitable for polynomials even if rotated.

The first transformation attempted was the elementary one of rotation. It is shown that, by rotating the co-ordinate axes of raw data, it is possible to fit a selection of curves with least-squares polynomial methods.

The second transformation considered was that to polar co-ordinates. This is a more interesting transformation, because in general, providing the pole is chosen well, straight line segments are eliminated, and 'difficult' slopes in the cartesian system become manageable in the polar system.

In order to assess the improvement obtained by the transformations, the original and the transformed data had to be fitted to some common standard. The simplest standard is the degree of the fitted polynomials, but there is generally no obvious guide as to what that should be, especially in view of possible gross errors in the data. A better standard consists of extracting the maximum information from

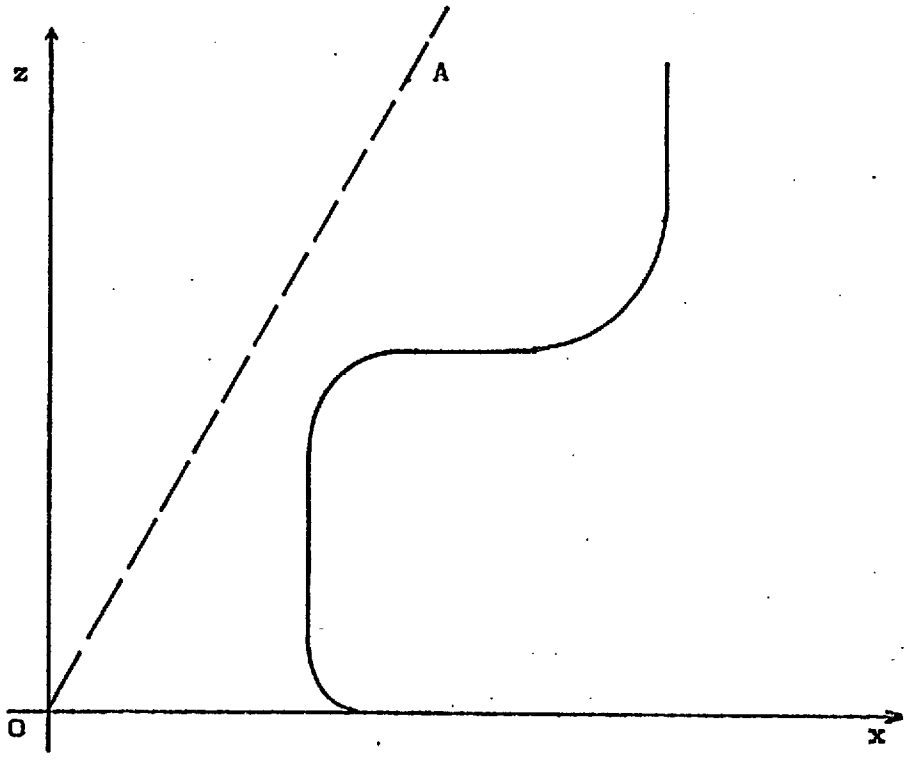


Figure 5.1 :- A typical stern profile, less the propeller boss.

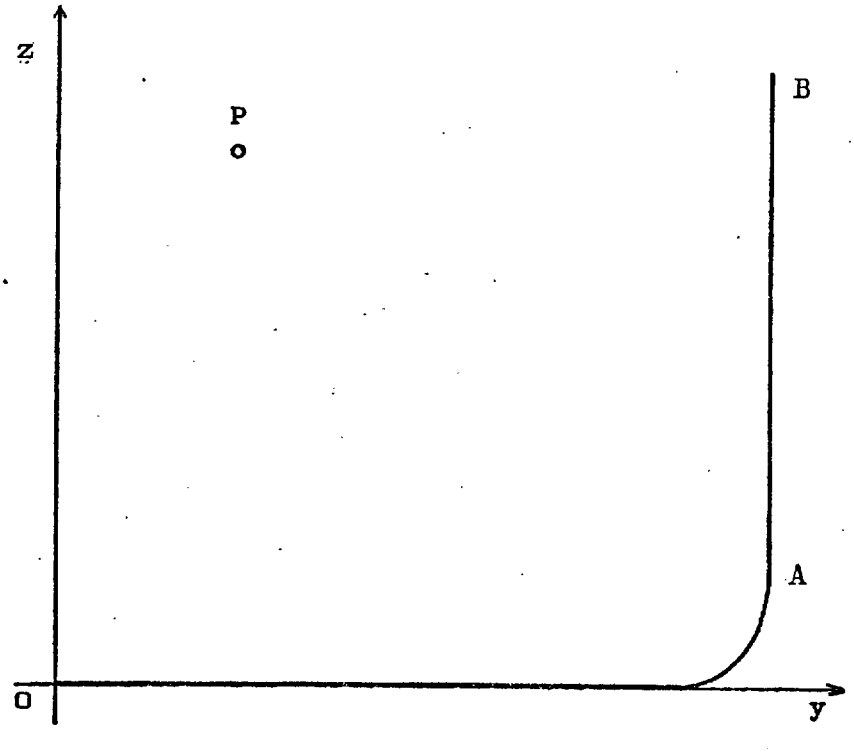


Figure 5.2 :- A typical Parallel Middle Body sectional curve.

the data in each case. Such a standard was avaliable in the form of the curve-fitting package of Chapter 3, in which the optimum degree of a polynomial is decided automatically, by analysing the residual deviations for each degree. Using this package, it also turned out to be quite simple to detect 'human' errors in data, and thus correct them.

It is shown, for both transformetions, that the imposition of end-conditions, e.g. an exact fit at the end-points, presents no difficulty.

It is further shown that both transformations, whilst achieving considerable improvements over direct poynomial fitting, have limitations, which prevent them from being used in a mechanical, computer like manner.

5.2 Rotation

5.2.1 Description of problem

Generally speaking, data points implying slopes greater than about ± 60 deg. (e.g. the neighbourhood of point B in Fig. 5.1, depending on the scaling of the variables) cannot be fitted by polynomials with any reliable accuracy. It is required to find that angle of rotation that will make the implied slopes optimal (defined below) for polynomials.

Given : the data points $(y_i, z_i, 1 \leq i \leq n)$. The points must be assumed to be ordered and reasonably accurate, to make the following analysis meaningful.

The data is to be rotated by Φ to a new set of reference axes, (y', z'), where they become $(y'_i, z'_i, 1 \leq i \leq n)$. There are several criteria that may be chosen for the optimality of Φ, but they all have in common the necessity to calculate the initial implied deriva-

tives of the data.

5.2.2 Estimates of slopes and preliminary analysis

The derivatives may be calculated using finite difference methods, but high order methods would be unstable for the same reasons that the data is to be rotated, and also unnecessary, since the data contains errors. Therefore, let the slope at each data point (or, more strictly speaking, between each pair of points) be estimated by the following first order formula:

$$\theta_i = \tan^{-1}\left[\frac{z_{i+1} - z_i}{y_{i+1} - y_i}\right] \qquad 1 \leq i \leq n-1$$

One point that arises immediately is that of multi-valued data, i.e. data that has, for at least one value of y, more than one value of z. For instance, the presence of line AB in Fig. 5.2 makes it multi-valued in that particular co-ordinate system. Least-squares fitting tends to 'average' out the multiple z values for such data, because no account is taken of the order of the points. Whereas this condition may be overcome easily for the data of Fig. 5.2, simply by rotating it, as shown in Fig. 5.6, it is clear that some data will be multi-valued in any cartesian reference system. The ability to detect this is a by-product of derivative estimation, as follows:-

Consider the ith point. The slope estimate is θ_i.

The straight forward way of checking for a multi-valued point is to compute $\Delta\theta = \theta_i - \theta_1$.

Then, if $|\Delta\theta| \geq \pi$, the data is multi-valued about any cartesian system.

In practice, problems arise with θ changing quadrants. It is therefore simpler to compute the angle spanned by the points, as follows:-

Let $$\beta = \sum_{i=1}^{n-2} \Delta\theta_i$$

where

$$\Delta\theta_i = \theta_{i+1} - \theta_i \qquad \text{and, from Fig. 5.3,}$$

$$= \tan^{-1}\left[\frac{(y_{i+1} - y_i)(z_{i+2} - z_{i+1}) - (y_{i+2} - y_{i+1})(z_{i+1} - z_i)}{(y_{i+2} - y_{i+1})(y_{i+1} - y_i) - (z_{i+2} - z_{i+1})(z_{i+1} - z_i)}\right]$$

The check is then whether $|\beta| \geq \pi$.

This was included in the rotation experiments, as a preliminary test for the suitability of data.

5.2.3 Optimum angle of rotation

Assuming that a data set has passed the preliminary check, it is now required to rotate it to the orientation best suited to polynomial fitting. Although there is no unique solution to this, a simple and effective condition is that the maximum rotated slope be minimised, that is, the value of Φ is used, that minimises

$$\theta = \max_i \left[(|\theta_i - \Phi|) , 1 \leq i \leq n \right]$$

The value of Φ may be deduced either intuitively, or by using elementary minimax techniques. Consider a typical graph of θ vs. Φ in Fig. 5.4. The envelope of the maximum of θ is shown shaded. The minimum value on the envelope is at the intersection of the lines representing $(\Phi - \theta_{min})$ and $(\theta_{max} - \Phi)$, where θ_{min} and θ_{max} are the minimum and the maximum estimated slopes respectively, thus leading to the result of

$$\Phi = (\theta_{max} + \theta_{min}) / 2$$

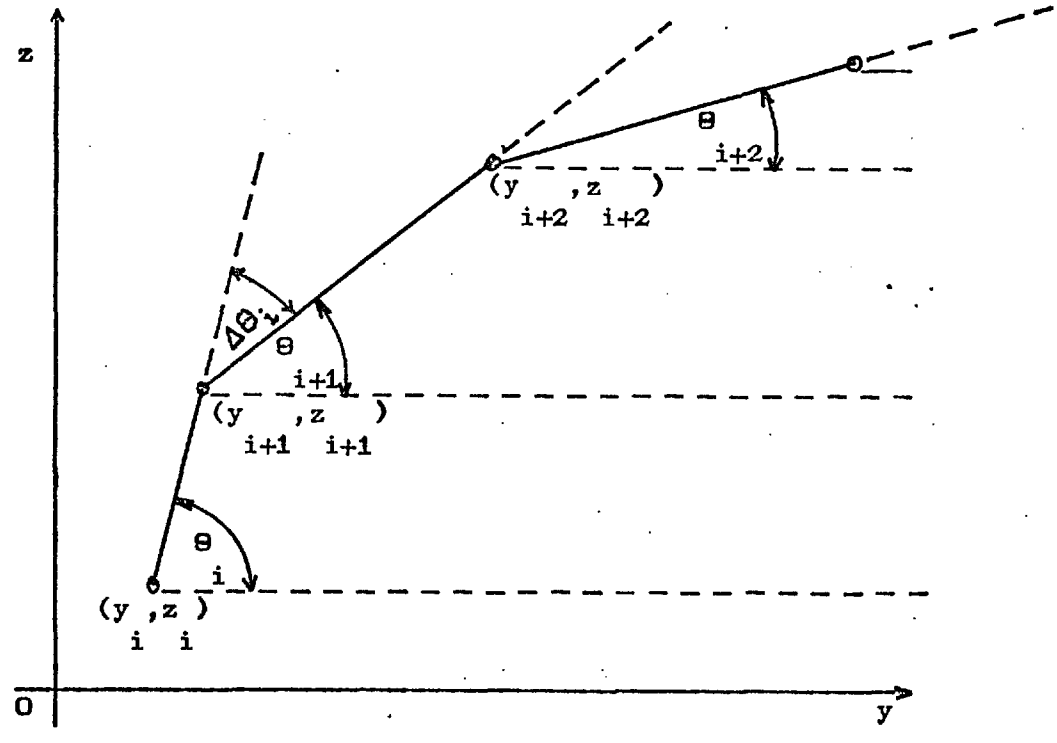


Figure 5.3 :- First order estimate for slopes, and check for multi-valued data.

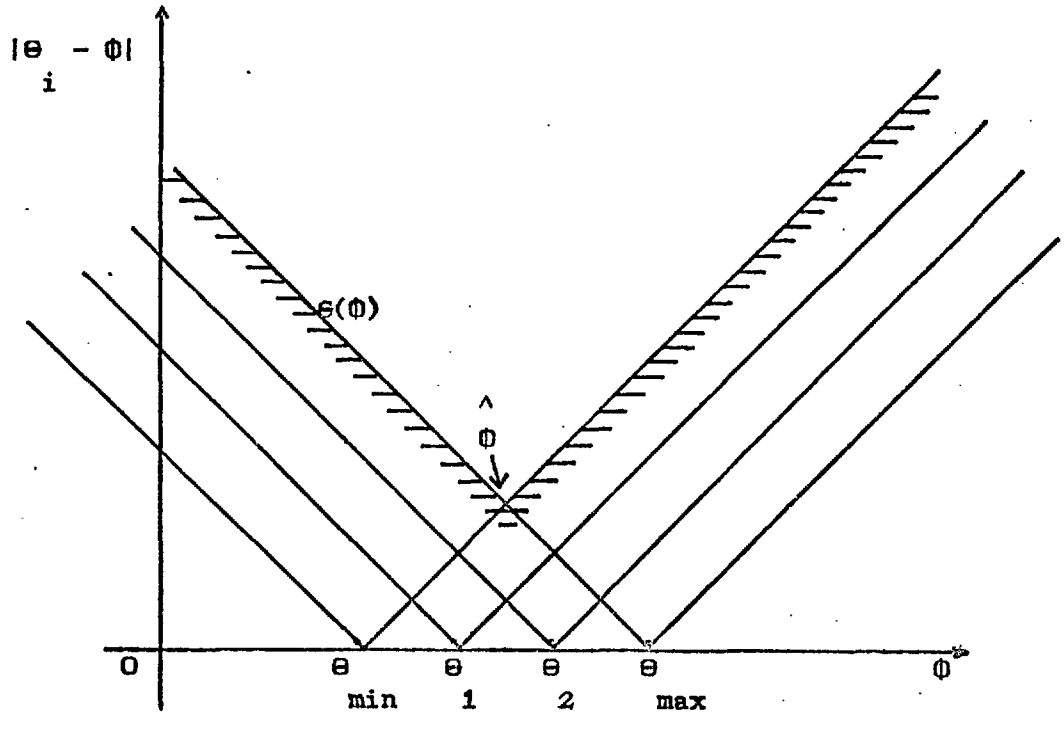


Figure 5.4 :- Computation of optimum angle of rotation.

which gives rise to a maximum slope, in the rotated co-ordinate system, of $(\theta_{max} - \theta_{min}) / 2$ at two points.

5.2.4 Rotation matrix and precautions

Having calculated the value of Φ, the data may be rotated to the new orientation using the standard 2D rotation matrix, bearing in mind that Φ is considered as subtracted from the slopes, vis.

$$\begin{vmatrix} y' \\ z' \end{vmatrix} = \begin{vmatrix} \cos(\Phi) & \sin(\Phi) \\ -\sin(\Phi) & \cos(\Phi) \end{vmatrix} \begin{vmatrix} y \\ z \end{vmatrix}$$

All slope dependent processing may now be performed.

Before rotation, the new maximum slope should be checked for being 'reasonable'. This, as before, depends on the scaling of the variables, but should not, in general, exceed ± 60 deg.

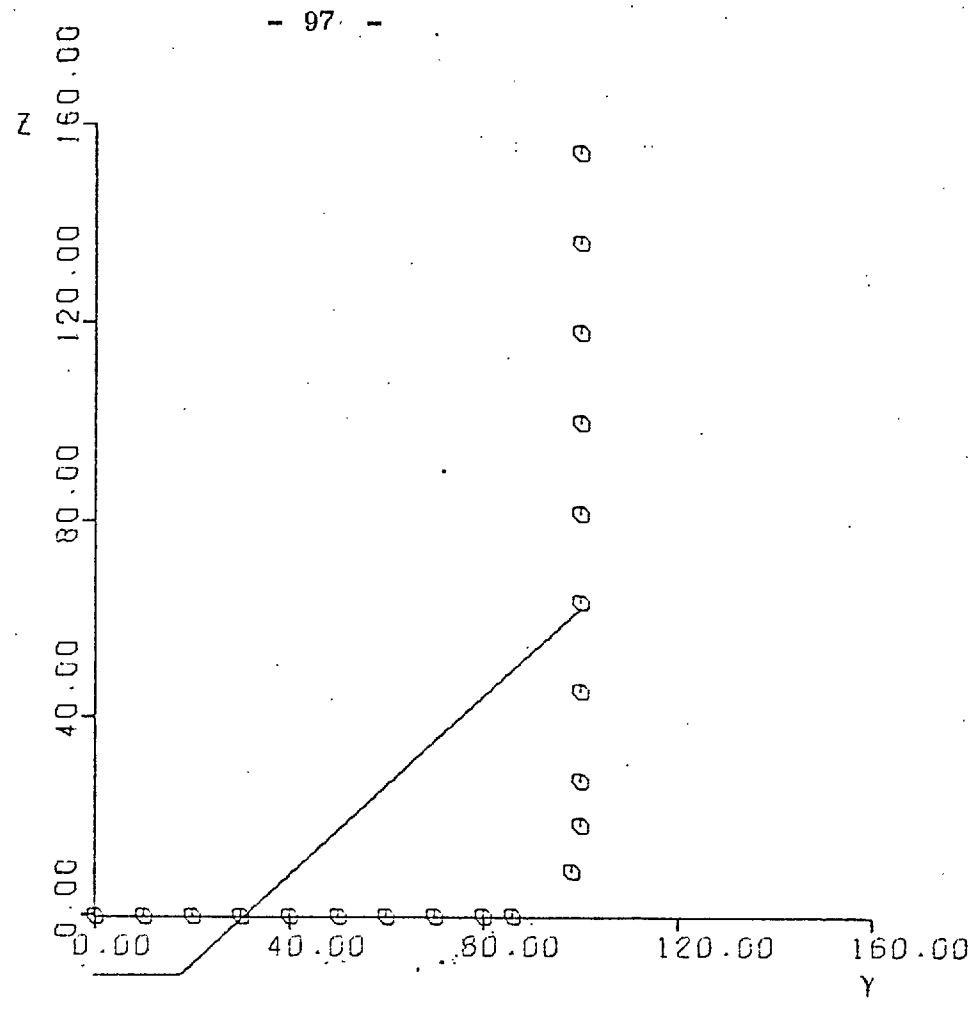
5.2.5 Imposition of end-conditions

Rotation is a linear transformation. Therefore any values imposed upon points and/or derivatives of the polynomial fitted to the original data must simply be rotated in the same way as the data, using the above matrix, and then applied to the transformed data.

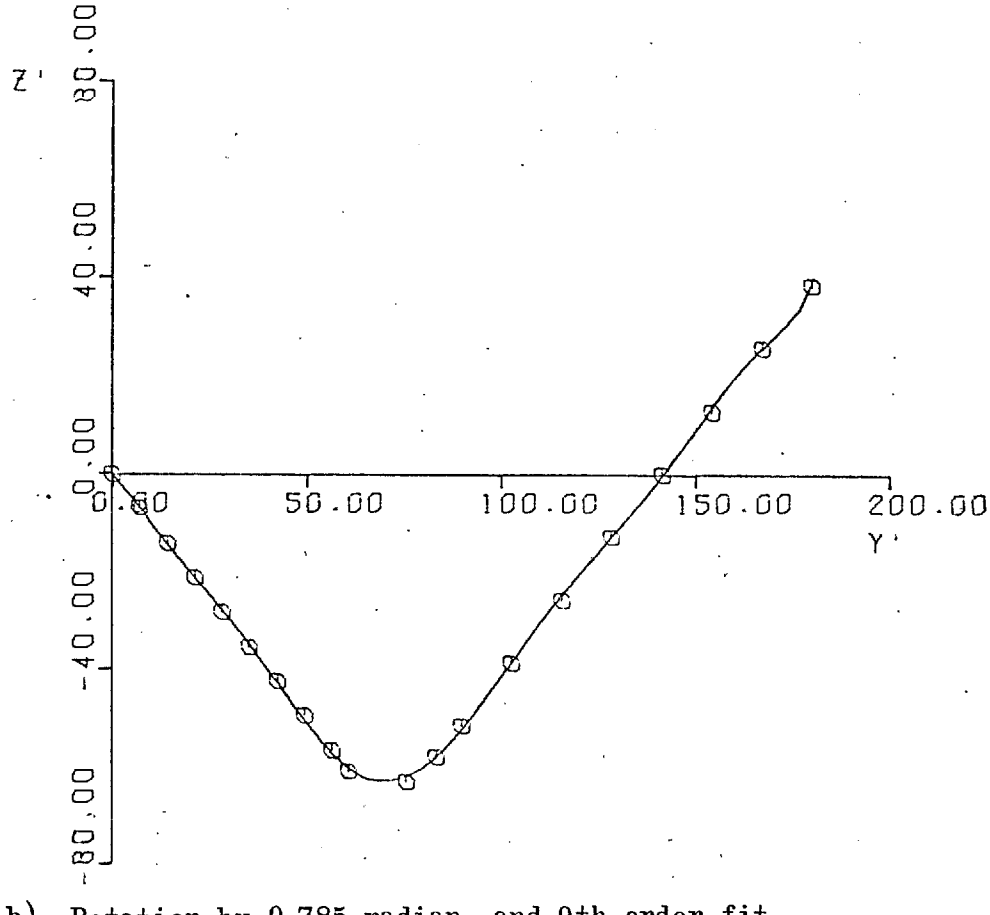
5.2.6. Results

The method has been tested with a representative selecion of data. The rotation itself hardly necessitated the use of a computer. Polynomials were fitted using the curve-fitting package of Chapter 3, with automatic choice of degree.

A marked improvement in the quality of fit was observed in all cases. Two typical cases are shown in Figs. 5.5 and 5.6, with their untreated counterparts, for comparison.



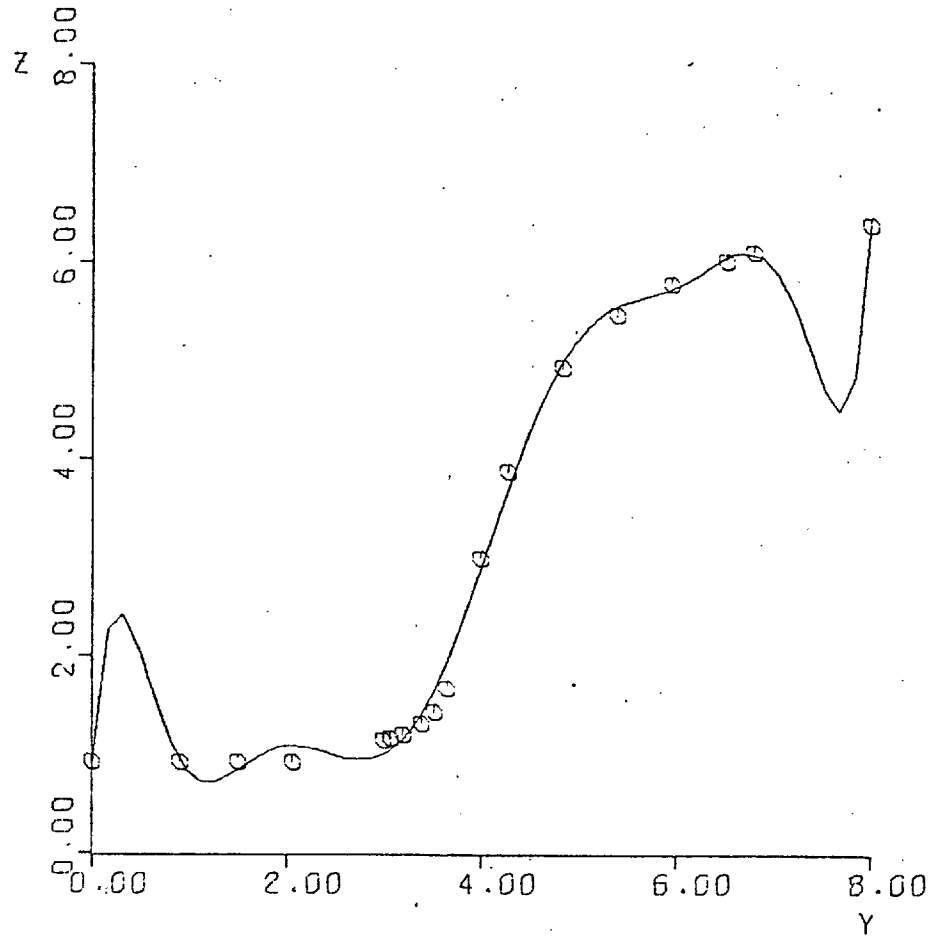
a) Original points and 1st order (statistically optimal*) fit.



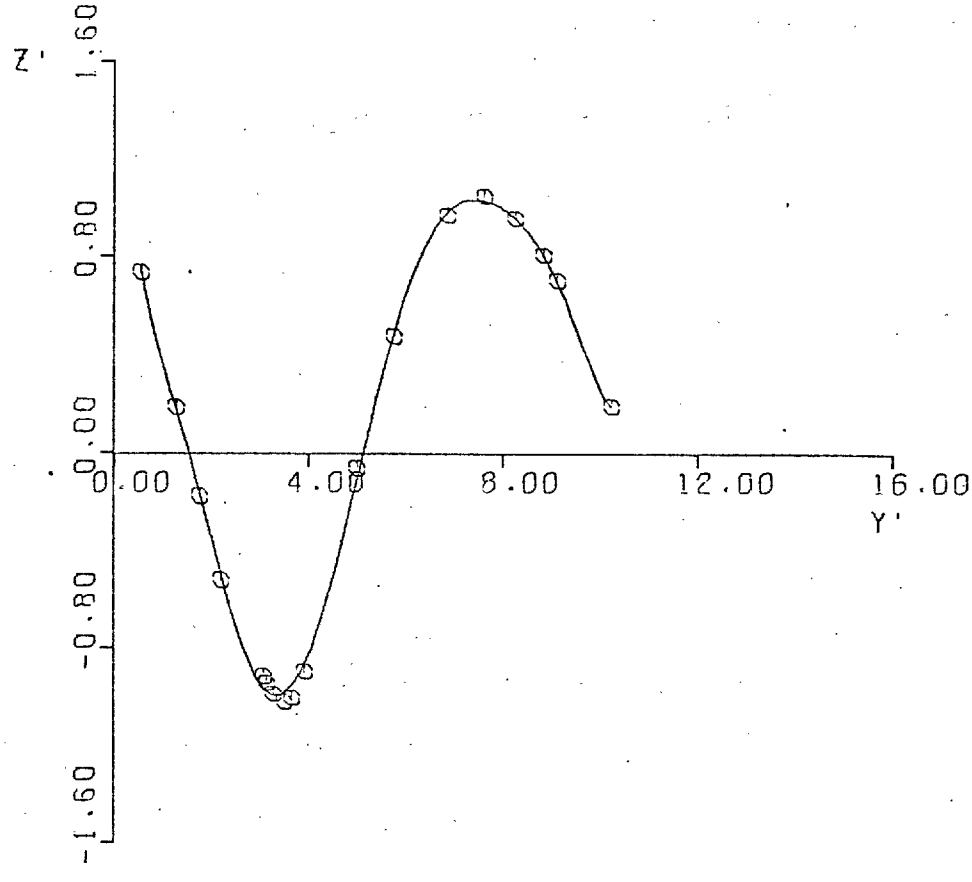
b) Rotation by 0.785 radian, and 9th order fit.

Figure 5.5:- Effect of rotation on PMB section of Model 2.

* The next statistically significant degree is 12.



a) Original points and 9th order (statistically optimal) fit.



b) Rotation by 0.656 radian, and 8th order fit.

Figure 5.6:- Effect of rotation on the stern profile of Model 1.

It is clear, even from just the two cases shown, that despite the improvement due to rotation, the resulting fit to the data is considerably worse than that tolerable in practice.

5.3 Polar Transformation

5.3.1 Description of problem

Consider the data points $(y_i, z_i, 1 \leq i \leq n)$ in Fig. 5.7. For conciseness, 'curve' in this section refers to the figure generated by the straight line segments joining the points.

Let point P be at (y_p, z_p).

Let the points be transformed to their polar representation with P as a pole, i.e. to $(r_i, \theta_i, 1 \leq i \leq n)$, where

$$\theta_i = \tan^{-1}\left[\frac{z_i - z_p}{y_i - y_p}\right] \qquad \ldots\ldots (5.1)$$

and

$$r_i = \sqrt{(y_i - y_p)^2 + (z_i - z_p)^2} \qquad \ldots\ldots (5.2)$$

The data may then be smoothed by fitting a polynomial to r in terms of θ. The approximate effect of this transformation on data such as in Fig. 5.2 is shown in Fig. 5.8. Even this rough sketch suffices to show up the advantage of the polar representation.

5.3.2 Choice of pole

In the preceding discussion, point P appears to have been chosen at random. Clearly some points are more suitable for a pole than others. Therefore, let the optimum pole position be defined as that which, when used in the transformation, produces the 'flattest' r/θ

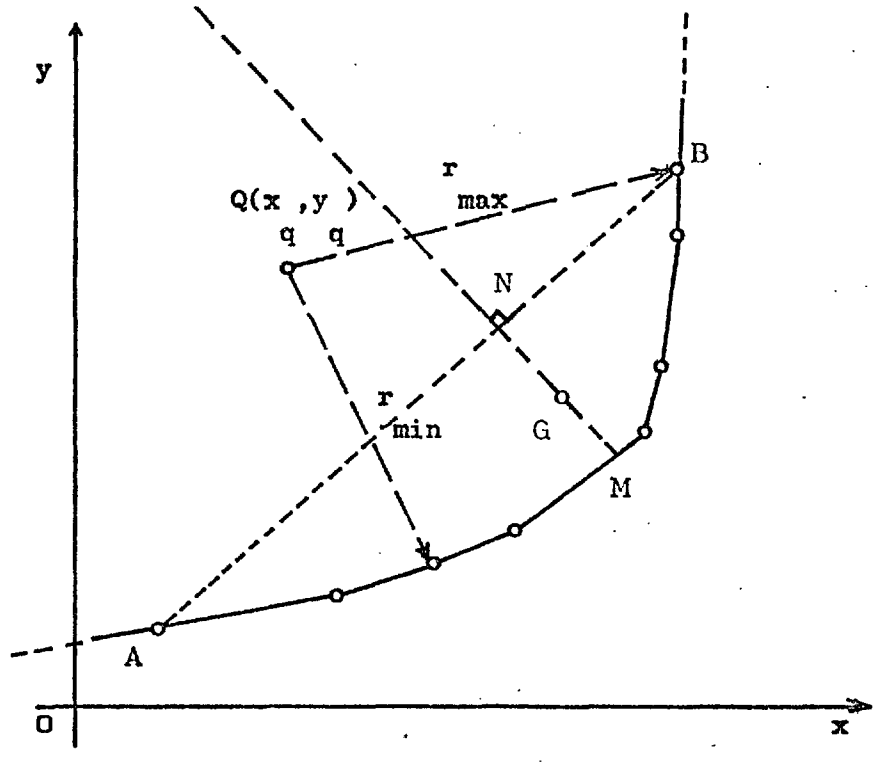


Figure 5.7 :- Typical data, with the points joined by straight line segments.

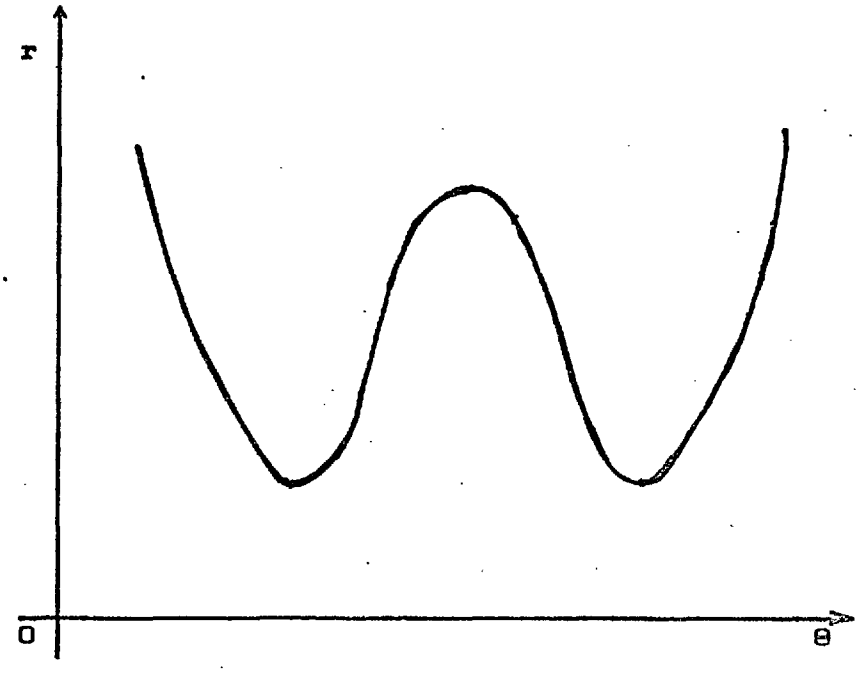


Figure 5.8 :- Approximate effect of the polar transformation on the data of figure 5.2, with pole at point P.

curve. To complete the definition, a measure of 'flatness' must be introduced.

Several metrics are possible, e.g.

a) $(r_{max} - r_{min})$ where $r_{max} = \max [r_i]$

and $r_{min} = \min [r_i]$ $\qquad 1 \leq i \leq n$

b) $\sum_{i=1}^{n} (r_i - \bar{r})^2$ where $\bar{r} = \frac{1}{n} \sum_{i=1}^{n} r_i$

i.e. similar to least-squares.

c) $\sum_{i=1}^{n} |r_i - \bar{r}|$

The norms (a) and (c) are similar in nature. (b) suffers from the disadvantage that although the variance of r may be minimal, sharp peaks that might be introduced by the transformation are not prevented. For most reasonably smooth data, the choice of norm would be immaterial, but as a numerically iterative, rather than analytical, procedure must be used to minimise any of the three, the highest order norm, l_∞, or the minimax norm, should be used, i.e. norm (a).

5.3.3 Computation of position of pole

Let point $Q(y_q, z_q)$ (Fig. 5.7) be a trial point for P.
Define the 'flatness' function

$$\Phi(y_q, z_q) = r_{max} - r_{min}$$

where the values of r are calculated from equation 5.2, with respect to point Q.

The function Φ has some interesting properties, discussed in Appendix E. The ones relevant to this section are

a) Φ is non-analytic, non-negative, and continuous, with, in general, discontinuous first and higher derivatives, and, the more important property, found empirically,

b) If the curve generated by the data points is convex in the range AB, then Φ has a unique minimum in the convex half-space of the curve, half-space in this context being defined as the area enclosed by the two end-segments of the curve extended to infinity. That minimum is the optimum position of P.

It has been found empirically, from other characteristics of Φ, that P lies approximately on the line MGN, where G is the centroid of the data points, and GN is perpendicular to AB. This fact is used to obtain a first appproximation to (y_p, z_p), by a linear search and parabolic interpolation (Appendix F) along MGN. The values are refined by a general purpose minimisation procedure, which does not use derivatives (Powell, 1965).

The above does not necessarily apply to non-convex data. More than one minimum may exist, or none. However, instead of devising special techniques for such data, it has been found simpler to impose limits on the area of search for P, and abandon the search if the first approximation is not found within these limits.

5.3.4 Imposition of end-conditions

The simplest imposed end-conditions are simply that one or both end points have particular z values. For definiteness, consider one end-point, say at $y = y_1$, and let the required z value be z_1.

Let the equivalent polar co-ordinates be θ_1 and r_1.

It is clear that point (y_1, z_1) need simply be transformed like any other data point, using equations 5.1 and 5.2.

Other possible end-conditions include the imposition of certain values of end-derivatives. For definiteness, again, consider the imposition of

$$\left.\frac{dz}{dy}\right|_{y_1} = z'_1$$

Using equations 5.1 and 5.2 and differentiating, it can be shown that the equivalent value of derivative in the polar system is given by

$$\left.\frac{dr}{d\theta}\right|_{\theta_1} = \frac{1 + z'_1 \tan(\theta_1)}{-1 + z'_1 \cot(\theta_1)}$$

where $\tan(\theta_1)$ is obtained from equation 5.1.

Similar analyses may be performed for the second and higher derivatives, if required, so that no difficulty should be experienced in a practical implementation of this transformation.

5.3.5 Results and observations

a) Computation of (y_p, z_p)

The algorithm has been tested with both real and artificial data. The first approximation was found to be excellent in most cases, including some non-convex ones. Failures were obtained for data of very low curvature, data with more than one multi-valued point, and data which was roughly symmetrically 'S' shaped, all for fairly obvious reasons. These failures immediately limited the use of the method .

b) Transformation and polynomial fit

The transformation itself is straight forward, but a further limitation of the method was discovered during the testing stages. Some non-convex data, for which optimum poles could be found, were found to have multi-valued polar equivalents about these particular

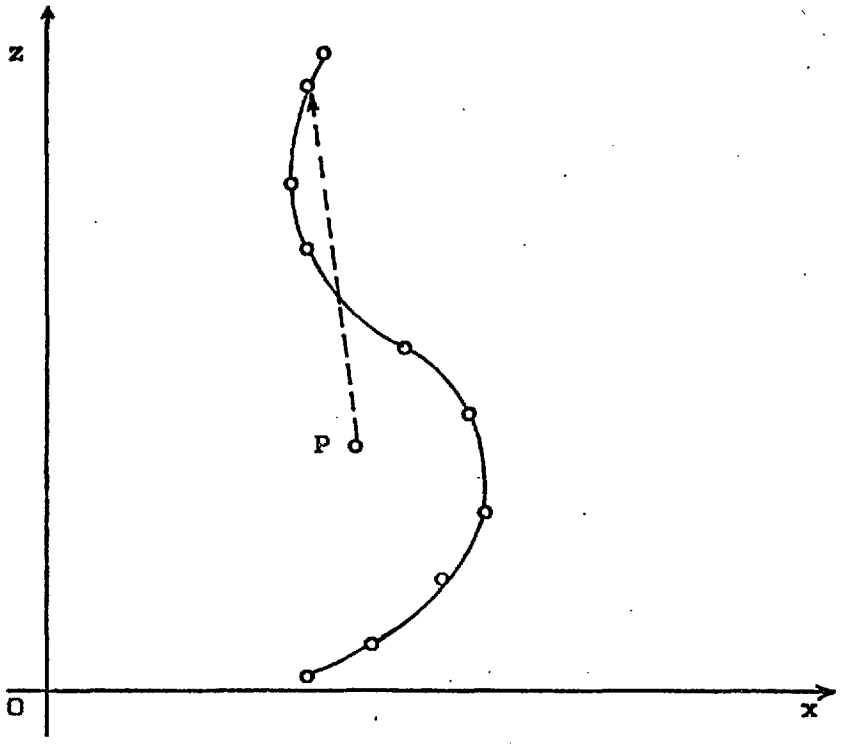


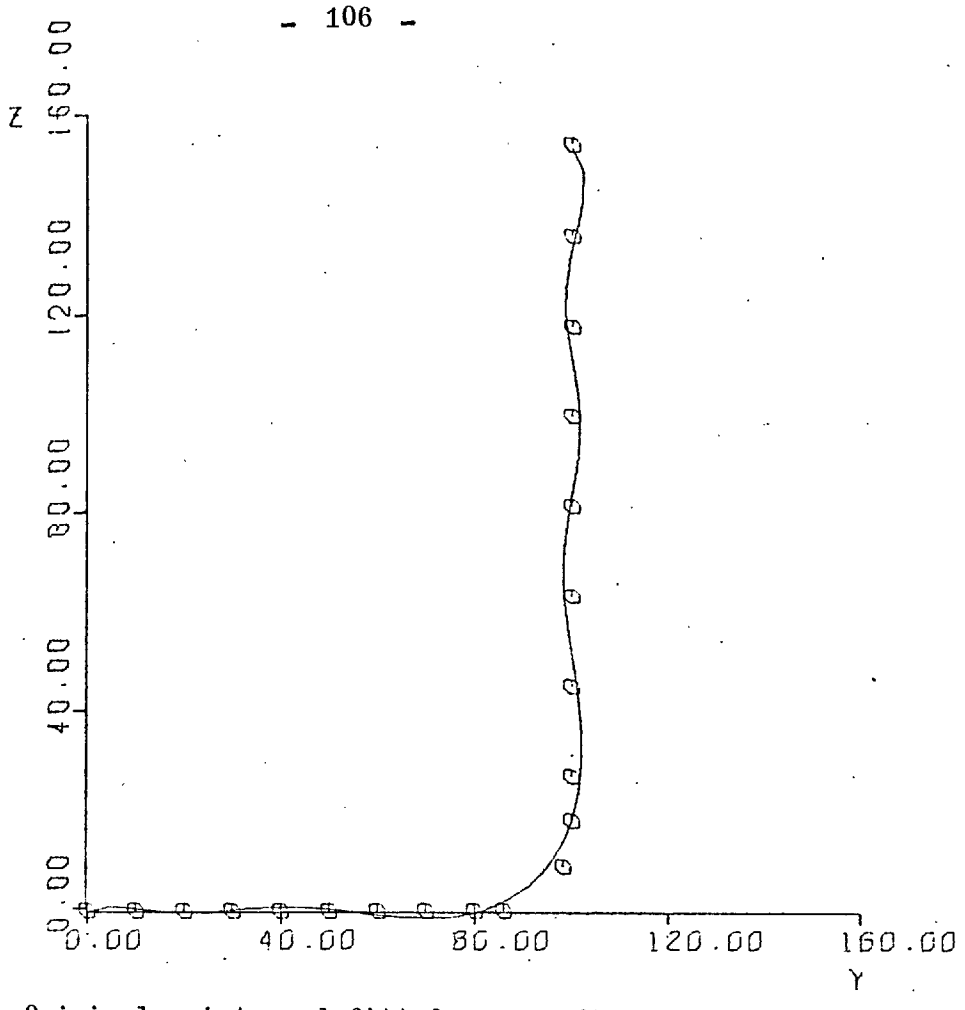
Figure 5.9 :- Non-convex data, for which an optimum pole can be found, but which transforms to multi-valued co-ordinates about the pole P.

poles (Fig. 5.9). This is clearly undesirable, but no satisfactory solution has been found to this problem. However, the condition is detectable during the transformation, by checking that θ increases or decreases monotonically when transforming the points in their specified order.

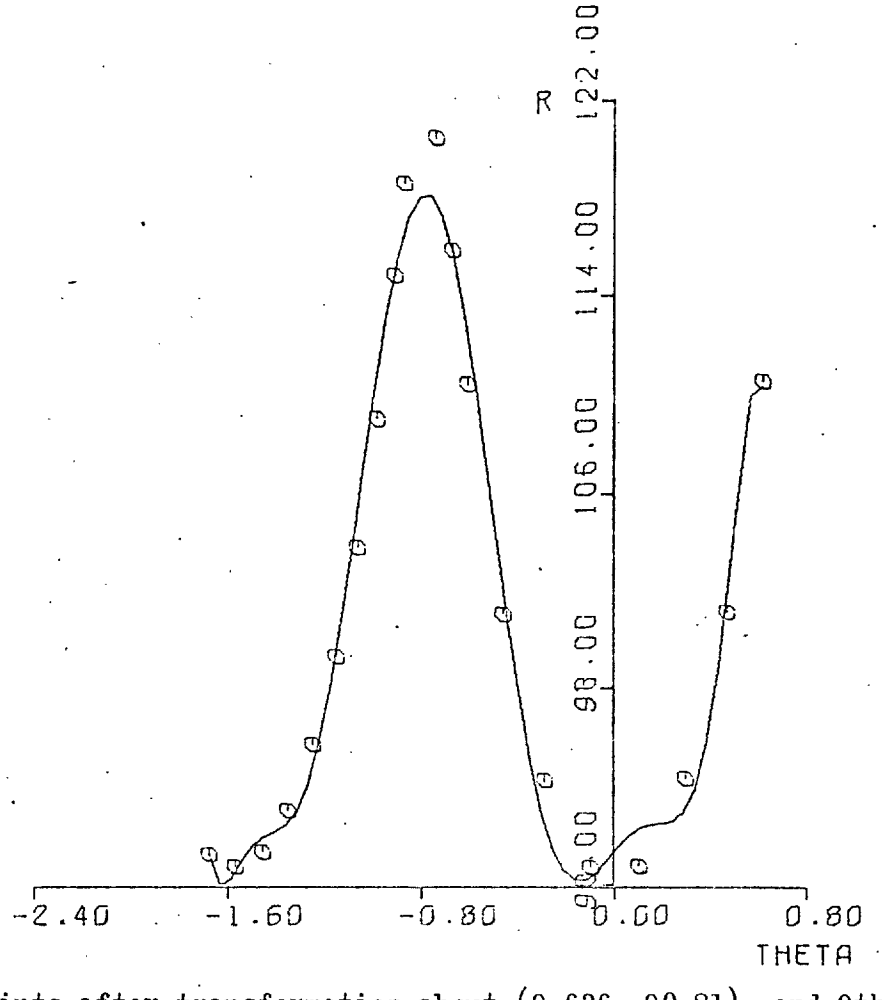
Results from typical data are shown in Figs. 5.10 and 5.11. These and others confirmed the validity of the experiment. The degrees of the fitted polynomials were usually relatively high, and might have been higher if permitted by the curve-fitting package, as was indicated by fairly large residual deviations in the polar representation. Despite this, the fit in the cartesian system, after the reverse transformation, was usually very good, particularly visually. The reason for this is that the residual deviations in the polar representation are implicitly in the direction of r, and are therefore shared between the original y and z. In some cases, the position of the pole was such that the worst deviations were practically along the fitted curve, thus producing an apparently exact fit.

c) Oscillations between points.

This problem occurs with practically every form of curve fitting by polynomials as the power is raised progressively in attempts to improve the fit. In this experiment, the r/θ curves were found to be exceptionally well behaved and free of this problem. The reverse transformation, however, did occasionally generate unwanted oscillations, as may be seen in Fig. 5.10.

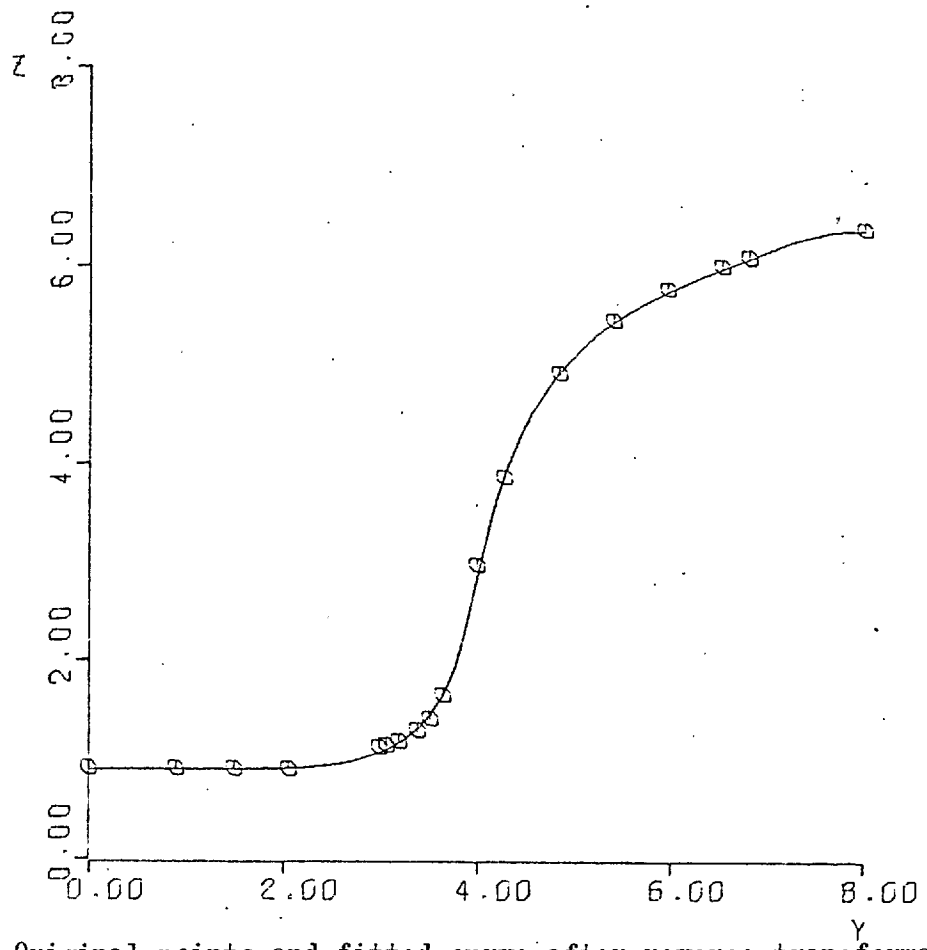


a) Original points and fitted curve after reverse transformation.

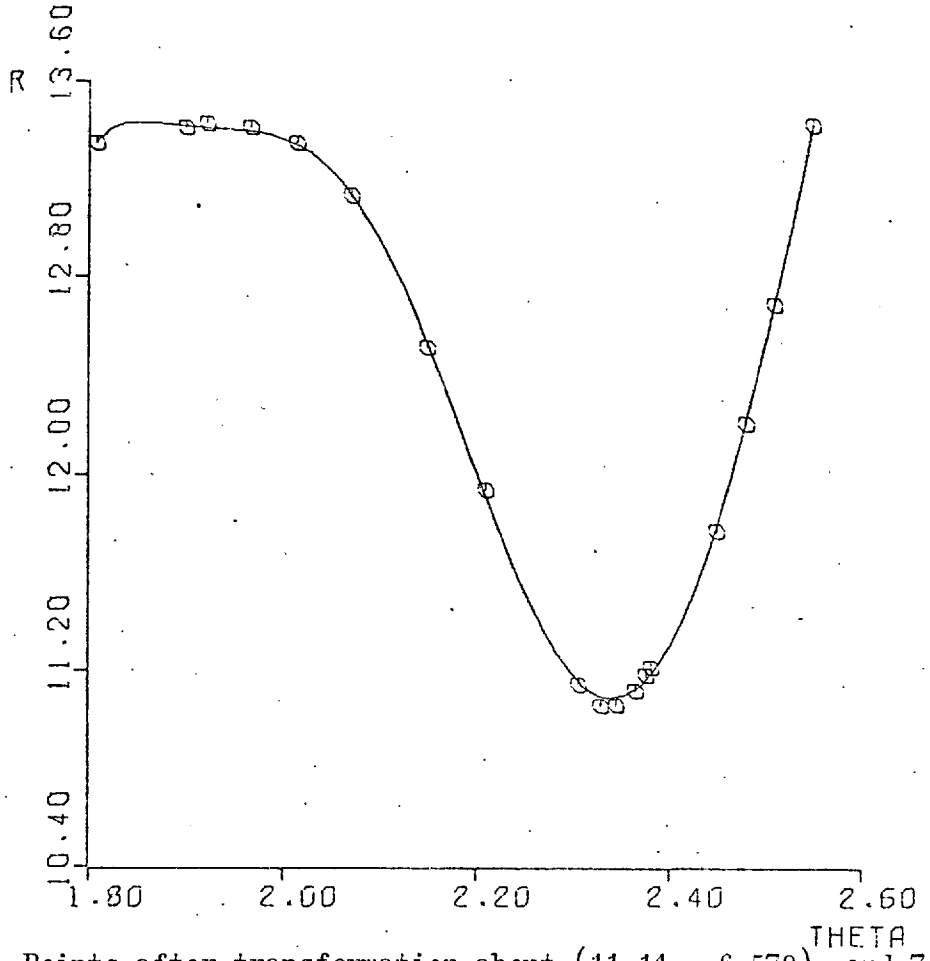


b) Points after transformation about (9.636, 90.81), and 9th order fit.

Figure 5.10:- Effect of polar transformation on PMB section of Model 2.



a) Original points and fitted curve after reverse transformation.



b) Points after transformation about (11.14, -6.579), and 7th order fit.

Figure 5.11:- Effect of polar transformation on the stern profile of Model 1.

5.4 Conclusions

Both transformations were found to produce substantial improvements in the quality of fit over direct fitting methods. The polar transformation was generally much more effective.

Despite the improvements, the results obtained were not good enough to warrant the use of either transformation as the basis for an automatic, or even semi-automatic, hull-fairing method.

Another disadvantage of both transformations is their dependence on the accuracy of the data. Algorithms have been devised to identify and correct, or reject, bad data (Berger et al., 1966) and, in practice, one such algorithm would be applied; also the curve-fitting package helped to detect bad data. Nevertheless, both transformations were found too sensitive to errors in data.

One important point, common to most transformations, emerged from this experiment - the directions of the residual deviations after the fit. The standard least-squares polynomial model assumes the errors to be in the values of the ordinate only. When the polynomial is fitted to transformed data, the residuals are implicitly in the values of the ordinate in the transformed space, i.e. generally shared between the original abscissa and ordinate. Although this alters the mathematical model of the fit, the effect on the fitted curve (or surface), after the reverse transformation, may often be beneficial. This discovery led to the postulation of a different mathematical model for curve and surface fitting, described in the following chapters, which takes the errors in the data to be in all the co-ordinate directions.

CHAPTER 6:-

PARAMETRIC CURVE FITTING (*)

(*) An abridged version of this chapter appeared in the Computer Journal in Vol. 14, May 1971, under the same title.

6.1 Introduction

In most of the foregoing work, and also in most contemporary line's fairing methods, it is assumed that the errors in the data are in one of the co-ordinate directions only, namely in the dependent variable. In Chapter 5, it was shown that there is much to be gained by assuming the error to be distributed between the co-ordinate variables. This approach may be formalised, by defining the distribution and its associated weighting pattern.

For this work, the division of errors may be justified by recalling the fact that data is read off a body-plan. If the same measuring device is used for both ordinate and abscissa, there is no reason to suppose that measurements in one direction are more accurate; for a measuring device of given precision, the absolute error in measurement, in any direction, can be taken as proportional to the scaling factor of 'real units'/'units of measurement'. This is used to deduce the relative distribution of error between abscissa and ordinate. Individual weights may be assigned to data points in the usual way, i.e. by a subjective estimate of their accuracy.

The problem of least squares curve fitting when both ordinate and abscissa are subject to error has been of interest for some time A recent successful algorithm for for solving it is described by O'Neill (O'Neill et al., 1969). Given a set of experimental data

$$(x_i, y_i),\ 1 \leq i \leq n,$$

with x_i and y_i subject to error with weights w_{xi} and w_{yi}, it is required to calculate points (x'_i, y'_i) that lie on a continuous curve so as to minimise the sum of squares of weighted perpendicular distances from the points (x_i, y_i) to the curve, i.e. to minimise

$$S = \sum_{i=1}^{n} \left[\theta_x^2 w_{xi}^2 (x_i - x'_i)^2 + \theta_y^2 w_{yi}^2 (y_i - y'_i)^2 \right]$$

where the overall ratio of error in y to error in x is given by θ_x / θ_y, which also governs the angle between the line from each data point to the corresponding point on the generated curve, and the tangent to the curve at that point.

In O'Neill's work, y' is derived as a polynomial in x', and the discrete values x'_i corresponding to (x_i, y_i) are calculated by an iterative procedure. The idea can be generalised by introducing a third variable, u, and functions f_x and f_y, called the parametric functions, such that

$$x' = f_x(u)$$

$$y' = f_y(u) \quad .$$

For this work, polynomials, because they are very convenient, have been used as parametric functions, i.e.

$$f_x(u) = \sum_{r=0}^{k_x} a_{xr} u^r$$

and

$$f_y(u) = \sum_{r=0}^{k_y} a_{yr} u^r$$

Also, because the data sets to be processed have a natural ordering, it was further stipulated that the order of the data points be maintained. This definition makes the parametric curve orientation-independent, which enables the fitting of multi-valued data, such as the typical hull cross-section often is, and even loops.

The parameters to be calculated are the n-vector $\bar{u}$ and the

polynomial coefficients, i.e. $(n + k_x + k_y + 2)$ unknowns. In this chapter it is shown that there are several solutions, in the sense that they produce a minimum of S. An iterative technique is described, without a proof, which generally converges to the best solution in the least squares sense, without using derivatives. Although polynomials are used throughout the analysis, most of it is general, and can easily be applied to other parametric functions.

A certain similarity of functional form may be noticed here between the parametric curve described above, and that used by MacCallum (1970), although the method of generation is completely different. This is intentional, because the lines' fairing work was complementary to MacCallum's work on curve and surface definition.

6.2 Statement of the Problem and Method of Solution

6.2.1 The objective function

The function to be minimised is given by

$$S(\bar{a}_x, \bar{a}_y, \bar{u}) = \theta_x^2 \sum_{i=1}^{n} w_{xi}^2 \left[x_i - \sum_{r=0}^{k_x} a_{xr} u_i^r \right]^2 + \theta_y^2 \sum_{i=1}^{n} w_{yi}^2 \left[y_i - \sum_{r=0}^{k_y} a_{yr} u_i^r \right]^2 \quad \ldots\ldots (6.1)$$

Let $k = \max [k_x , k_y]$

Then S is a polynomial of order $(2k + 2)$ in $(n + k_x + k_y + 2)$ independent variables. As such, S has at most $k_x k_y$ real minima.

Intuitively, equation (6.1) appears to be underdetermined; if $\bar{u}$ is linearly transformed, compensatory changes in $\bar{a}_x$ and $\bar{a}_y$ would still lead to the same solution. By applying the Newton-

Raphson technique to the variables near a minimum of S, this property is clearly demonstrated; the resulting system of equations has $(n + k_x + k_y + 2)$ unknowns, and only $(n + k_x + k_y)$ equations. The two variables eliminated from the system were u_1 and u_n, for reasons discussed below. There are algorithms for computing local minima of functions such as S, but they are rather inefficient for $n > 20$ (e.g. Powell, 1965).

6.2.2 The constraints

Ordinary least squares fitting takes no account of the order of the independent variable. The reason for fitting x and y in terms of u is that a possibly multi-valued, ordered data set is separated into two independent, single valued sets. The last property ensures that the order is not lost by the least squares fit, if the parametric functions are themselves single valued by nature, e.g. polynomials. To obtain correct separation for all data, some order must be induced in $\bar{u}$. The natural choice was made, that $u_i \leq u_{i+1}$, $1 \leq i \leq n - 1$, with u_1 and u_n fixed. Because of the properties of linear transformation, it is only the distribution of u in the range u_1 to u_n that affects the value of S. The values of u_1 and u_n are arbitrary; they were chosen as $u_1 = -2$ and $u_n = +2$ for computational precision (Section 6.3.1).

6.2.3 Non-linear programming

The constraints on $\bar{u}$ change the nature of the problem from a standard minimisation one to a non linear programming problem, with a non-convex objective function and linear constraints. There are many techniques and programs available for the solution of such problems, but none of them take advantage of the relative simplicity of this particular problem.

6.2.4 A possible algorithm

Upon further consideration of the problem, it appears that it may be split into two simple parts. Given an initial u-vector, it is a relatively trivial matter to compute the functions f_x and f_y using least squares; and vice versa - given the functions f_x and f_y, then, subject to certain conditions (to be discussed below), a new approximation to $\bar{u}$ that results in a reduced value of S, may be computed by local minimisation of S with respect to each u_i in turn. For this to be successful, the components of $\bar{u}$ should be 'reasonably' independent. Loosely defined, this means that the u-values should be ones that arise due to the nature of the data, rather than being forced by the constraints, i.e. the ultimate u-values should be well away from violating the constraints. Complete independence between the components of $\bar{u}$ is always sufficient (because the objective function reduces to a sum of independent functions, which may be minimised individually), but was found not to be strictly necessary (Section 6.3.2).

The algorithm therefore consists of two processes - each major iteration consists of (n - 2) minor iterations for the new u-values (u_1 and u_n being held fixed), and a least squares fit to calculate the new polynomials which must, by definition, reduce S further or leave it unaltered. The major iteration is repeated until convergence is reached.

This is a rather simple approach to a complex problem, but in combination with a good first approximation to $\bar{u}$, it has given some interesting results.

6.3 Computational Techniques

6.3.1 Least-squares derivation of the polynomials

The method used was that of orthogonal polynomials, which is treated in Chapter 3. It is shown there that the highest degree of computational precision is obtained if the independent variable lies in the range $[-2, +2]$, which dictated the values of u_1 and u_n.

In the same chapter, methods are described for discrete weighting of data points, and also for the imposition of desired end-conditions. These were also implemented.

6.3.2 Iteration for u-vector

The contribution at the ith point to the sum of squares of deviations is given by

$$s(u_i) = \theta_{x\,xi}^2 w_{x\,i}^2 [x_i - f_x(u_i)]^2 + \theta_{y\,yi}^2 w_{y\,i}^2 [y_i - f_y(u_i)]^2$$

It is a $(2k)$th order polynomial in u_i. A new value u'_i is to be found, such that $s(u'_i) \leq s(u_i)$, subject to $u_{i-1} \leq u'_i \leq u_{i+1}$. Figs. 6.1 to 6.5 show typical s/u behaviour, in order of decreasing frequency of occurence. It was convenient to write the program so that the local minimisation starts at u_i. Hence, at the ith point, in the jth iteration, the value of u_{i-1} is from the current iteration, and that of u_{i+1} is from the $(j-1)$th iteration. Fig. 6.5 depicts the problem that was encountered by O'Neill - that of several local minima - but is less likely to occur, since it depends on the turning points of two functions, rather than one. Behaviour as in Figs. 6.3 and 6.4 is more common, because it can be caused by inflexions in f_x and f_y, and by certain combinations of the derivatives of f_x and f_y. Fig. 6.1 depicts the most desirable behaviour; it is the geometrical representation of independence between succe-

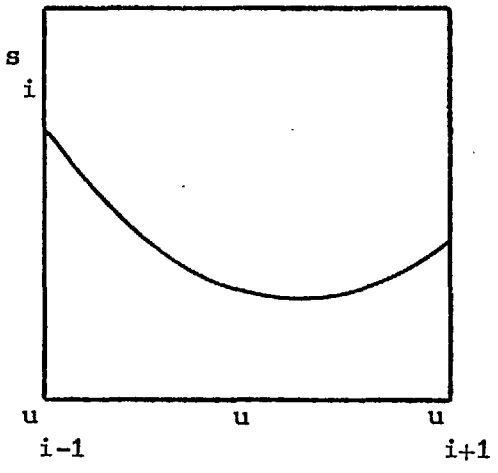


Figure 6.1

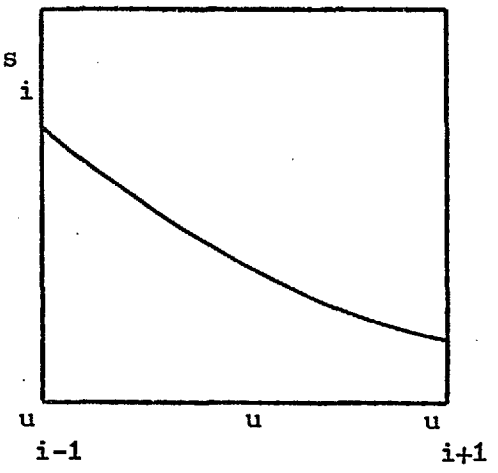


Figure 6.2

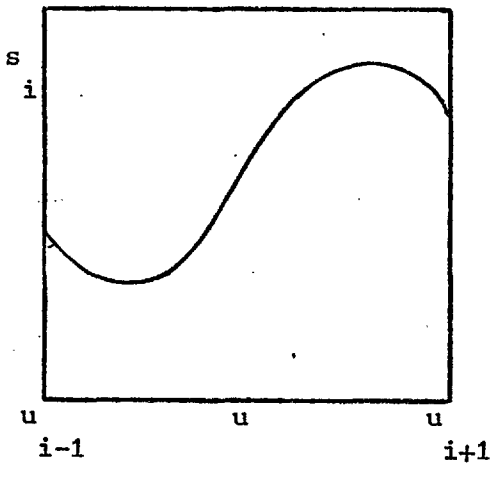


Figure 6.3

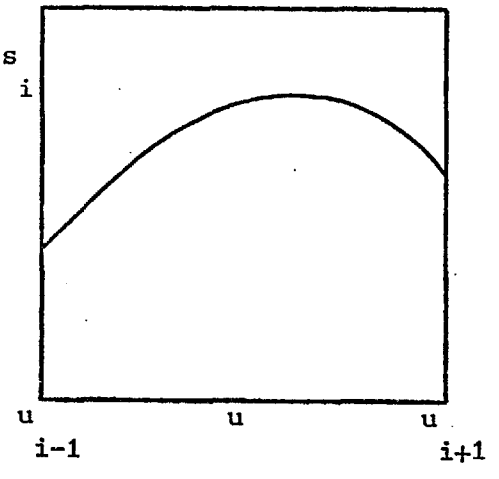


Figure 6.4

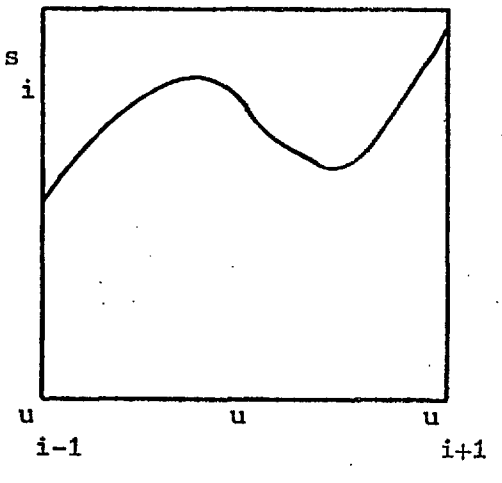


Figure 6.5

Figures 6.1 - 6.5 :-

Typical behaviour of the ith contribution to the sum of squares of deviations, with varying u_i.

ssive u values. Fig. 6.2 is a case that occurs frequently in the early iterations. It represents the prevention, by a constraint, of a complete local minimisation.

There are many ways, of varying sophistication, of locating the minimum of s. The method finally adopted - a simple, constant step search, without using gradients - has been found to be more than adequate. The position of the minimum is bracketed by moving down-hill from u_i, in steps of $(u_{i+1} - u_{i-1})/20$. The position is then refined by parabolic interpolation (Appendix F), and the process repeated with a step-size 1/100th of the original. As expected, this procedure has been found a little inefficient in the first one or two iterations, but this is more than offset by the simplicity of the program. The average number of calls to the function that evaluates the polynomial s(u) - the major computing time consumer in the entire algorithm - is 6 to 8 per point.

No satisfactory solution has been found for problematic s/u curves. Generally speaking, successive u values tended to move in the same direction, that is when $u_{i-1} \longrightarrow u_i$, u_i tended to increase, too. Cases such as Figs. 6.3 to 6.4 were rarely found, and, in any case, there cannot be more than k such points in a set of data.

6.3.3 The initial approximation to $\bar{u}$

There are two 'natural' choices for the initial values of u_i, $2 \leq i \leq n-1$, as follows:

a) Equally spaced, i.e $u_i = u_1 + i\,h$, where

$$h = \frac{u_n - u_1}{n - 1} .$$

b) Linear approximation to arc-length, i.e. joining the data

points by straight line segments, and relating u to the lengths of these segments as

$$u_i = u_{i-1} + m\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

where m is a constant, such as to make $u_n = 2$. (a) is the simpler of the two, but (b) may be justified qualitatively, as follows:

Given the two functions $f_x(u)$ and $f_y(u)$, and their derivatives, $f'_x(u)$ and $f'_y(y)$, the arc length σ is related to u by

$$\frac{d\sigma}{du} = [f'_x(u)^2 + f'_y(y)^2]^{1/2}$$

i.e.

$$\Delta\sigma \approx \Delta u\ [f'_x(u)^2 + f'_y(y)^2]^{1/2}$$

and

$$\Delta u \approx \Delta\sigma\ [f'_x(u)^2 + f'_y(u)^2]^{-1/2} \qquad \ldots\ldots (6.2)$$

(any text on elementary Differential Geometry, e.g. Eisenhart, 1960).

If the functions are a good approximation to the data points, the constant m can be thought of as an 'a priori' approximation to the mean (over n points) of the bracketed expression in equation (6.2). In words, the concentration of components of u is directly related to the concentration of the data points.

In practice, it has been found that starting from either (a) or (b) led to the same solution, but, as hoped, (b) <u>always</u> gave a much better starting point.

In an attempt to locate all the minima of S, other initial approximations have been tried for several sets of data. These approximations were logarithmic, with values of u concentrated at either end of the range. The minima obtained in these cases were different from the ones found using (a) or (b); they were

in every instance worse, both in the initial and the final values of the objective function.

Method (b) was finally adopted as the best first approximation. Results obtained indicate strongly that it normally leads to the global minimum of S.

6.3.4 Convergence

The question of convergence is rarely a simple one. The obvious way is to compute the derivatives of the objective function with respect to the variables, and stop iterating when their values are sufficiently small. When the constraints are involved, the projection of the gradient vector unto the constrained sub-space must be considered instead. However, for this algorithm, the calculation of derivatives would detract from the simplicity of the program, and would slow it down considerably. Moreover, as only one constraint is considered at a time, gradient projection has no meaning. Because of the high speed of the program, it was decided to permit it to iterate until the limit of precision of the machine is reached (between 8 and 9 decimal places in single precision). The initial sum of squares of deviations S_0, is calculated; iteration stops when the reduction in S in two successive iterations is less than $S_0 * 10^{-8}$, or when a prescribed running time or number of iterations have been completed, whichever occurs earlier.

6.4 Results

6.4.1 Data

A typical set of data for which this method was developed is shown in Fig. 6.6, with the solution superimposed. Many data sets of this kind were successfully tackled by the program, using

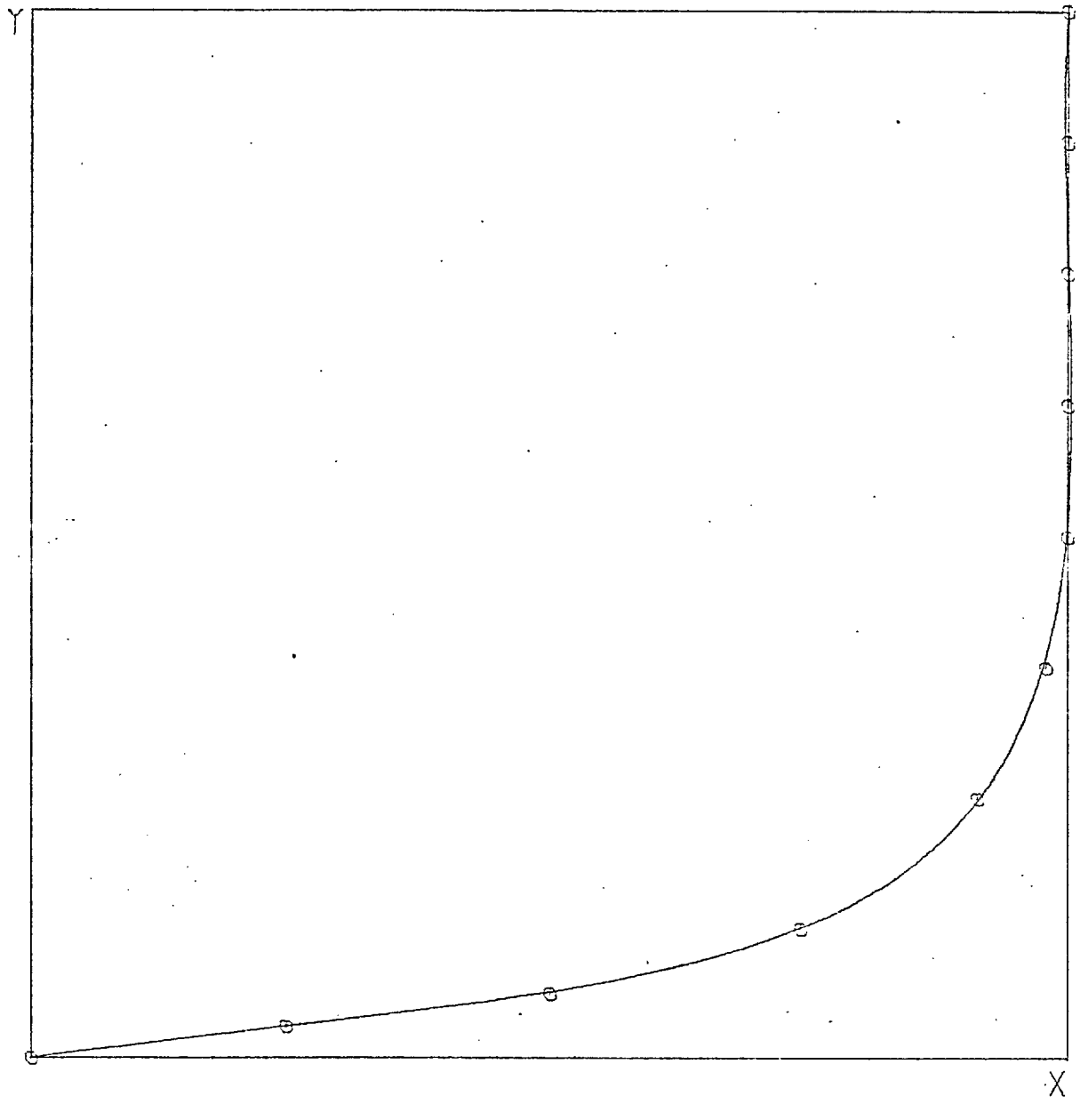


Figure 6.6:- A typical cross-sectional curve of a ship's hull (quartic fit).

cubics and quartics for parametric polynomials. Typical x/u and y/u curves generated are shown in Figs. 6.7 and 6.8.

As an experiment, the program was applied to data that had been generated by by parametric quartics. The computed quartics were found to be identical to the generating ones - Fig. 6.9 - within the precision of the computer.

Theoretically, the technique should be able to tackle a loop in the data. Data was generated using the equation

$$x^2 = y^2 \left[\frac{0.5 + y}{0.5 - y} \right]$$

and was fitted with parametric quartics - Fig. 6.10.

These examples illustrate the great flexibility of the algorithm; given enough degrees of freedom in the polynomials, most practical engineering curves could be fitted. Unwanted oscillations between the points were found to be much less severe than with a y/x fit, probably because low order parametric polynomials permit a quality of fit only attainable by very high order polynomials of y/x.

The ultimate test was the application of this algorithm to the offsets of Models 1 and 2. The results are shown in Figs. 6.11 and 6.12.

6.4.2 Effects of the degrees of the parametric polynomials

For simplicity, the same degree of polynomial was used for both x and y; this was usually between 3 and 5.

The interesting fact has been noticed, that for many data sets the degree of polynomial had little numerical and no visual effect on the fit. This occured often when the fit was good, which indicates that parametric fitting is less sensitive to 'noise' in the data - a phenomenon discussed in Chapter 3 - than standard methods.

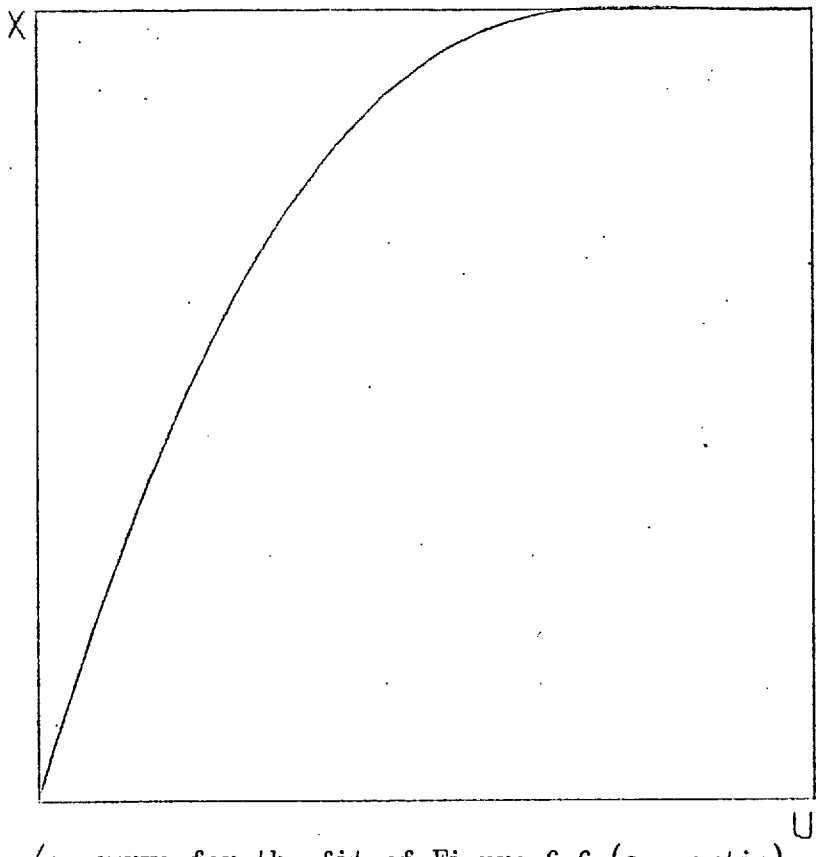
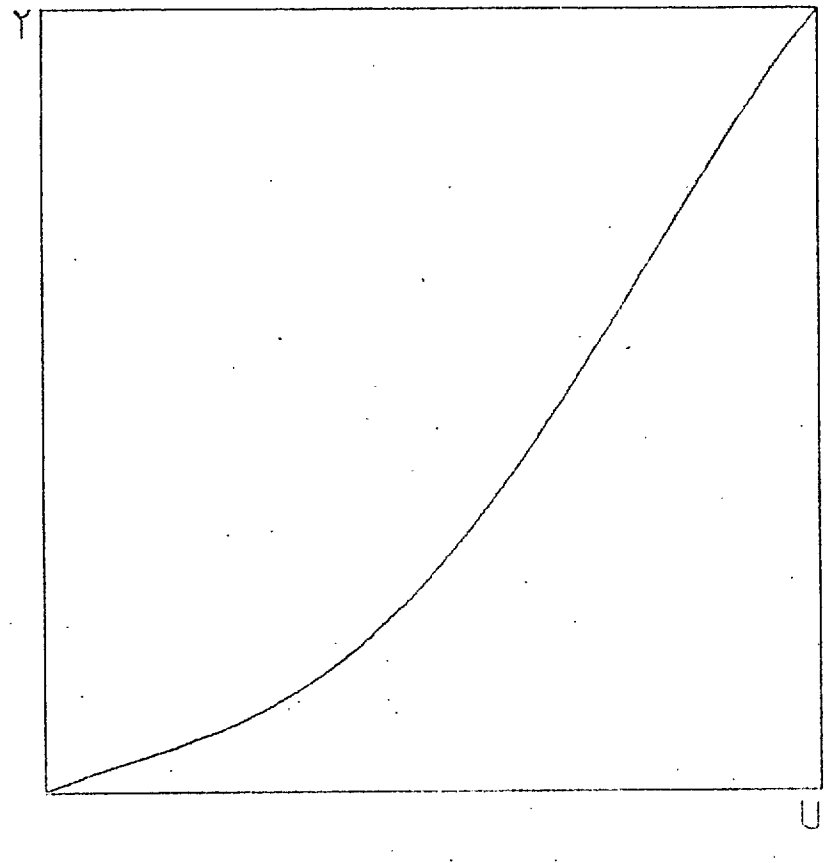


Figure 6.7:- x/u curve for the fit of Figure 6.6 (a quartic).

Figure 6.8:- y/u curve for the fit of Figure 6.6 (a quartic).



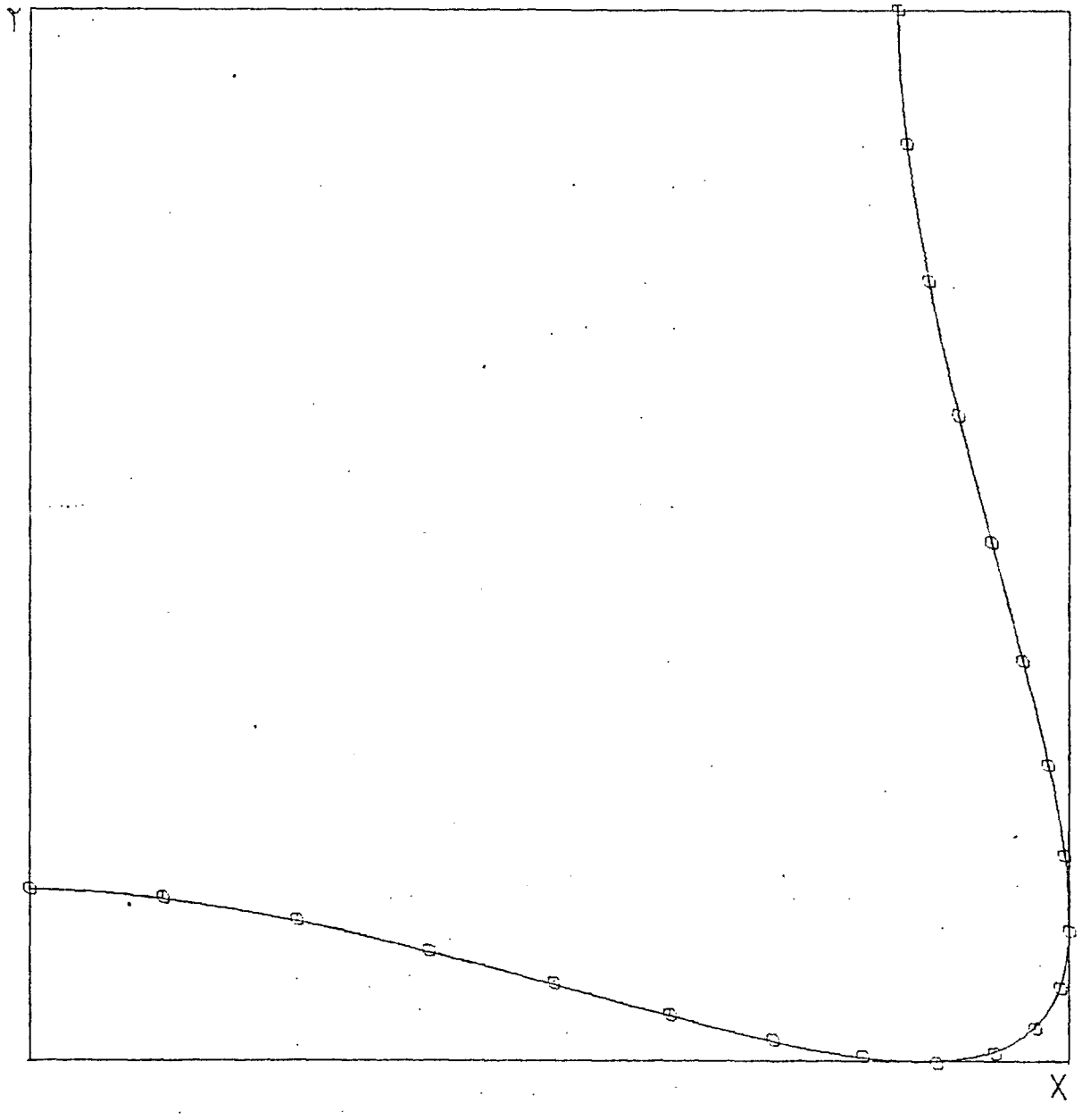


Figure 6.9:- Symmetric parametric quartics. The quartic fit converged to the generating functions.

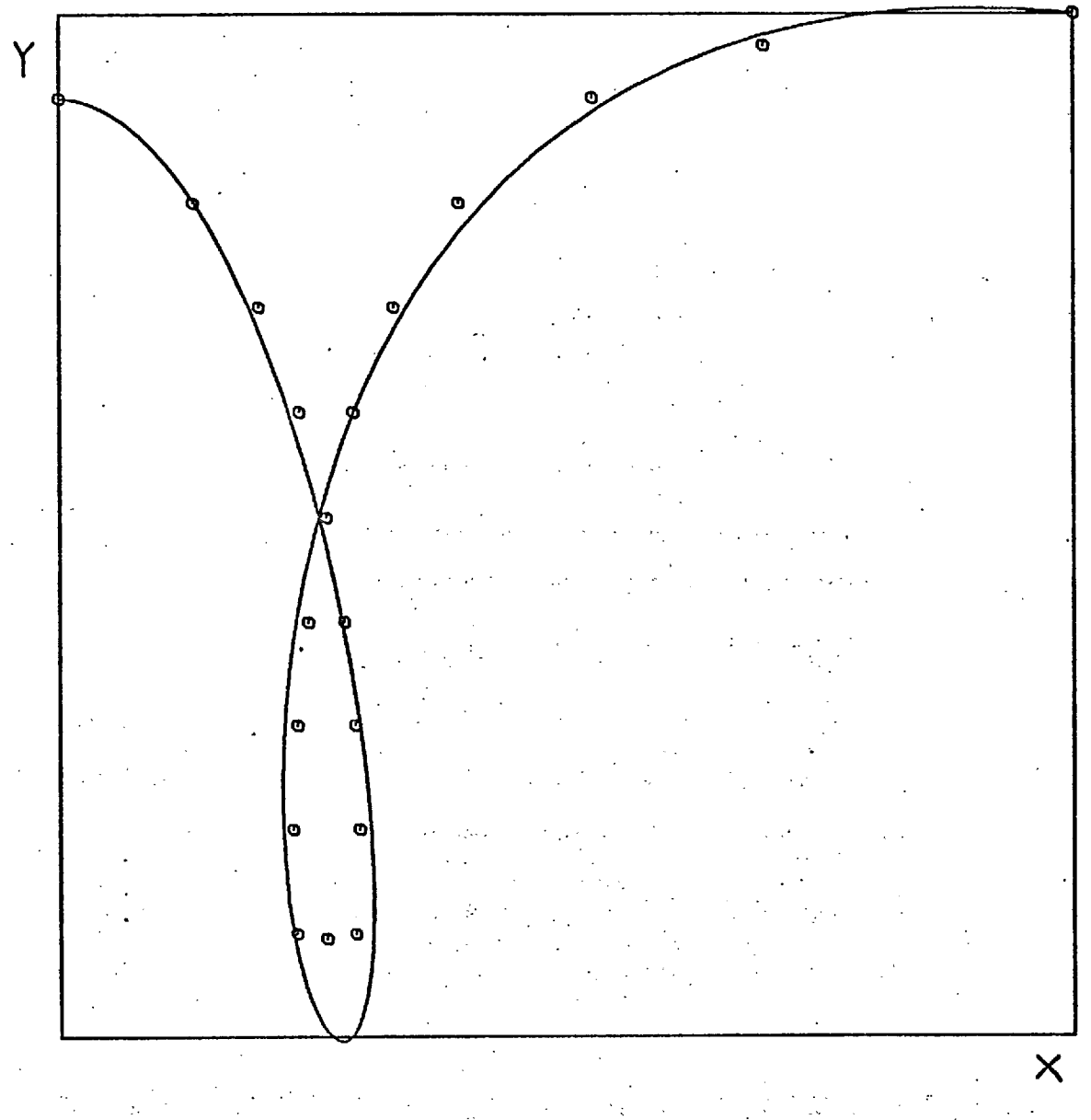


Figure 6.10:- Example of a loop, or multi-valued data, generated from $x^2 = y^2 (0.5 + y) / (0.5 - y)$ for $-1.2 \leq x \leq 3.36$; quartic fit, with exact fit at the end-points.

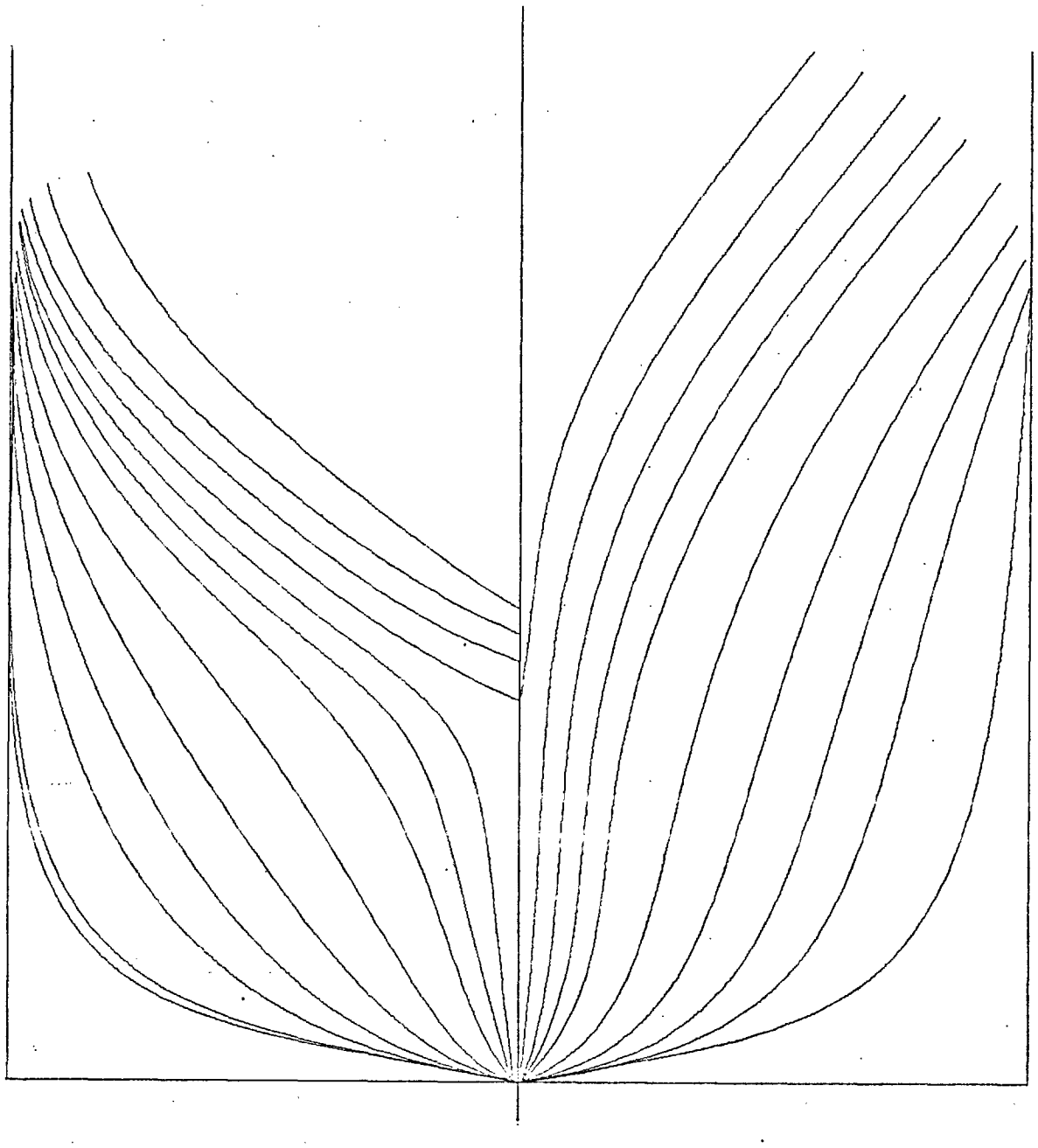


Figure 6.11:- Quartic fit to the cross-sectional curves of Model 1, after a maximum of 2000 iterations; the maximum deviation= 0.48 $^{o}/_{o}$, at station 3/4.

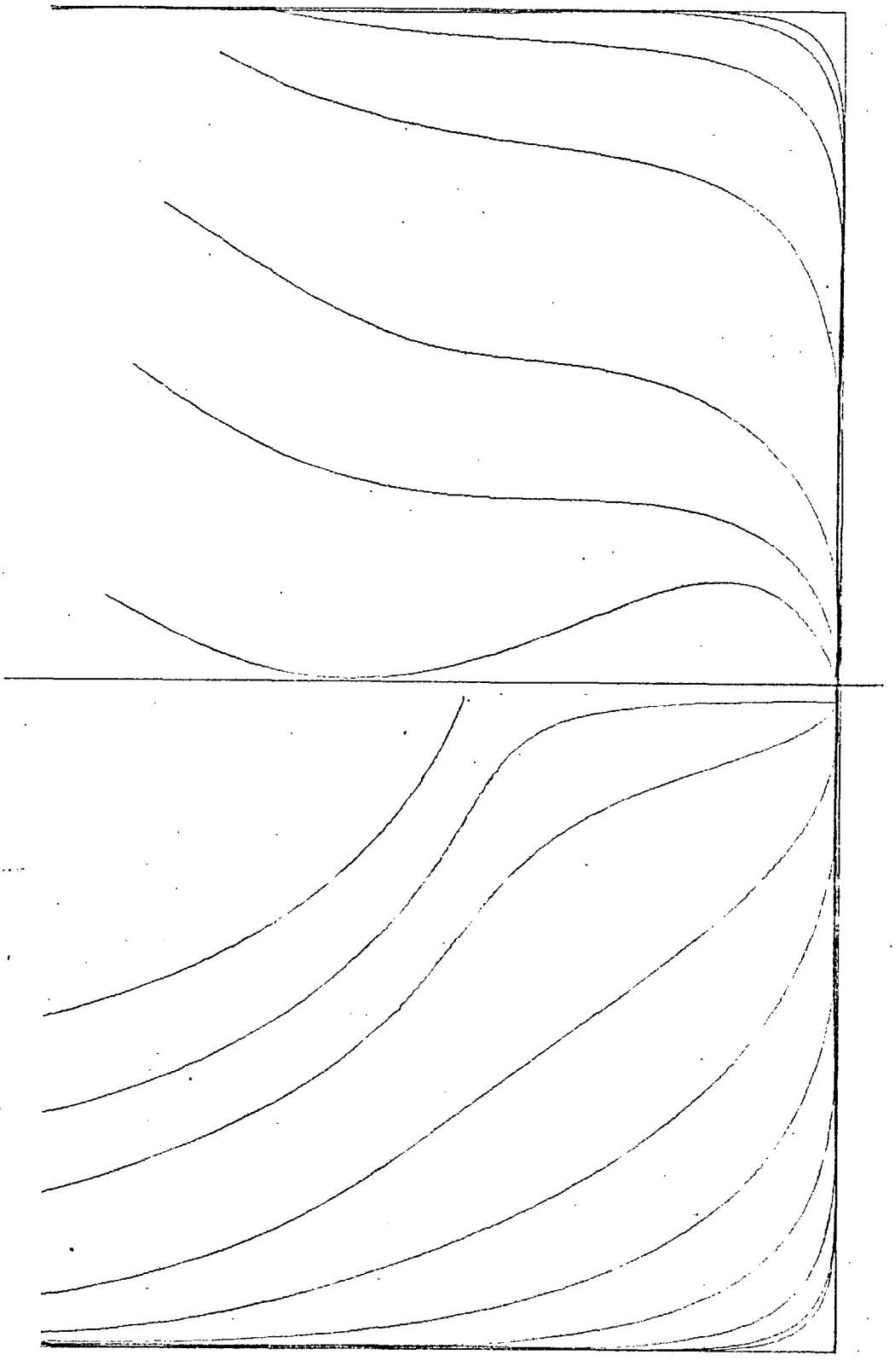


Figure 6.12:- Quartic fit to the cross-sectional curves of Model 2, with three points defining the Flat-of-Bottom on each curve, after a maximum of 2000 iterations; the maximum deviation= 0.69 $^{o}/_{o}$, at station 9.

In no case did a reasonable increase in polynomial order cause unwanted oscillations between the data points.

6.4.3 Effects of weighting

As expected, neither discrete weighting nor the imposition of end-conditions - Fig. 6.10 - affected the performance of the algorithm. For some data, e.g. Fig. 6.6, the free fit was so close, that imposed end-conditions had no material effect on the result.

6.4.4 Number of iterations and processing time for convergence

The number of iterations required was found to be unpredictable. In some cases, both initial and final fit were bad, resulting in extremely slow convergence. In other cases, practically complete convergence was attained within 3 to 4 iterations.

The aspect of computer processing time been neglected in previous chapters, because the time was negligible for all the experiments described this far. It had to be considered for this algorithm, because of the indeterminate number of iterations that may be required. This aspect, too, showed the advantages of a problem oriented approach over general methods. The computing time was found to be approximately proportional to the number of points (as compared with proportionality to at least the square of the number of points for general non-linear programming methods), the first iteration usually taking longer than the others. The times on the IBM 7094 II were typically 2.3 mS/point/iteration for cubics, and 2.7 mS/point/iteration for quartics., i.e. very reasonable, from the point of view of usability, for the average data set.

6.4.5 Validation of the results

No mathematical proof has been found that a program based on

this algorithm should converge to a valid solution, global or otherwise. In practice, however, no data has been found for which it failed. As a check on the results and the convergence criterion, the problem was set up as a non-linear programming problem. This was then solved by a general problem solving package, which is based on a variant of the Method of Feasible Directions (Benbow and Whitehead, 1968), using results from this algorithm as the initial feasible solution vector. Several sets of data were tried, and in no case was there a further reduction in the objective function. This, combined with relative slowness of the general problem solver, confirmed the validity of the simplified approach.

6.4.6 Artificially found minima

As mentioned in Section 6.3.3, various first approximations to $\bar{u}$ have been tried, in an attempt to locate all the constrained minima of S. In some cases, up to three additional minima were found (verified by the general problem solver - Section 6.4.5). As expected, the additional solutions were always worse than the ones obtained using the arc-lengths as the first approximation. It has not been found possible to prove that it would be so in every case.

6.5 Possible Improvements to the Algorithm

6.5.1 The least-squares fit

It is possible to include the coefficients of the parametric functions as variables in the local minimisation procedure. This means performing a least squares fit each time a u-value is moved, so that the functions remain the true least squares ones; this clearly eliminates the least squares fit at the end of the major iteration.

At first sight it seems impractical, but, on scrutiny, it turns out that only a few intermediate sums need be stored by the fitting program, updated in parallel with u, and a few divisions result in the new polynomials. This is somewhat similar to O'Neill's method, with constraints applied.

Intuitively, the same results should be obtained as with the basic algorithm, but in fewer iterations. This was borne out in a number of tests, but it was also found that the increase in computation time per iteration roughly offset the reduction in the number of iterations needed. Therefore, effectively nothing was gained from the added complexity of the program, and the idea was abandoned.

6.5.2 The local minimisation

In the local minimisation procedure, a group of u-values may persistently tend to violate their mutual constraints. Because of the sequential operation of the algorithm, this means that none of the interacting u-values are allowed to change. A possible improvement to the algorithm is to detect this situation, and to then treat the group as a single value (which can change), which is essentially what it has reduced to. At the end of each major iteration, an attempt must be made to 'release' the group, i.e. to separate it into individual values, to check whether the problem persists.

The implementation of this improvement is far more difficult than that of the first. Up to (n - 1) points may form a cluster, and, of course, there may be several clusters. It is possible to generate data to show this, but whenever this problem occured with real data, it was due to the bad or errorneous ordering of the

points. Therefore, from a practical point of view, the value of this change to the algorithm is dubious, and it was not implemented.

6.5.3 A formal approach to the problem

In recent years, there has been tremendous development in the field of non-linear optimisation, both constrained and unconstrained. It should now be possible to develop an efficient algorithm, with a sound mathematical basis, but which still makes use of the special properties of this particular problem.

6.6 Discussion and conclusions

6.6.1 The mathematics of the algorithm

The two questions that arise immediately from the results are

a) Can convergence to a minimum be guaranteed ? and

b) If so, can it be proved to be the global minimum in every case ?

No direct mathematical proof has been found for (a), but some of the results have been validated by a method for which convergence to a minimum can be proved; also, no data has been found, for which the program failed to converge.

The answer to (b) is much more difficult. Unconditional convergence to the global optimum of a non-convex function is extremely difficult to prove, and such a proof exists for very few methods indeed, if any. It can, therefore, only be stated that, based upon experience, the procedure described above tends to converge to the global minimum.

If any of the above statements is untrue, it should be possible to disprove them, or, at least, to construct contradictory data. There is, however, no doubt as to the efficacy of the algorithm for most practical data.

6.6.2. The naval architecture aspect

Parametric curve fitting goes a long way towards a mathematical description of a complete hull, as may be seen from the representative data that has been fitted. The definition of 'fairing', as given in Chapter 1, is just about complied with. However, even parametric polynomials cannot describe a line which consists of a curved and a linear segment, without some oscillations about the linear part. Therefore, for practical fairing, the hull should undoubtedly be split into patches. The important point is that fewer patches are needed, and the positions and shapes of their boundaries are not critical.

The next point that must be considered is the fairing of the hull as a 3-D surface, rather than a collection of curves. There are two alternatives. The first is to cross-fair the coefficients of the parametric polynomials which describe the curves, in a similar way to that described in Chapter 3 in connection with ordinary polynomials. The problem of different degrees of polynomials of different curves does not arise, because the relative insensitivity, to degree, of the parametric fit means that all the curves can safely be faired with as high an order as is necessary for the worst data. On the other hand, the choice of independent variable for the cross-fairing is not clear. The second alternative is to extend the technique to 3-D, i.e. to develop a parametric surface fitting method. This is considered in the next chapter.

Failing the complete hull description, this method generates ideal analytical curves for other naval architecture applications, such as the calculation of displacement, longitudinal strength, etc., when straight forward interpolation is no longer sufficiently accurate.

CHAPTER 7:-

PARAMETRIC SURFACE FITTING

7.1 Introduction

In the last chapter, a versatile curve fairing method was described. Although the Parametric Curve Fitting (PCF) algorithm can be used to fair surfaces in a similar way to ordinary polynomials, i.e. by repeated curve-fairing, a better solution is to extend the method to surfaces.

Given data points (x_i, y_i, z_i), $1 \leq i \leq n$, subject to errors with weights w_{xi}, w_{yi} and w_{zi}, it is required to calculate the points (x'_i, y'_i, z'_i) that lie on a continuous surface, so as to minimise the sum of squares of weighted normal distances from the points (x_i, y_i, z_i) to the surface, i.e. to minimise

$$S = \sum_{i=1}^{n} \left[\theta_x^2 w_{xi}^2 (x_i - x'_i)^2 + \theta_y^2 w_{yi}^2 (y_i - y'_i)^2 + \theta_z^2 w_{zi}^2 (z_i - z'_i)^2 \right]$$

where $[\theta_x, \theta_y, \theta_z]$ are parameters that control the direction cosines of the deviation vector, and can be used to allow for the overall errors in x, y and z being of different orders of magnitude.

For this work, two variables, u and v, called the 'parameters', are introduced. The values of x', y' and z' are derived from functions in u and v, called the 'parametric' functions, i.e.

$$x' = f_x(u,v)$$

$$y' = f_y(u,v)$$

$$z' = f_z(u,v)$$

where f_x, f_y and f_z are of known functional form.

The parametric surface has certain properties which make it highly suitable for fitting general data. It is orientation independent; also, because of the relationships between the physical surface

gradients and the gradients of the parametric functions, a wide range of slopes and curvatures implied by the data can be accomodated.

It was shown by Coons (1967) that, given the desired properties of a parametric surface at its corners and boundaries, the surface euqations, in canonical form, can be deduced. Although the method of deducing the surface equations in this work is completely different from Coons' method, the ease of generation and manipulation of polynomials, and the convenience of comparing results, virtually dictated the use of polynomials for the parametric functions.

The form of bi-variate polynomials, of degree m, used most commonly for surface fitting is

$$f(u,v) = \sum_{q=0}^{m} \sum_{r=0}^{q} a_{qr} u^{q-r} v^{r} \qquad \ldots\ldots (7.1)$$

which is treated fully in Chapter 3. However, by analysis of Coons' surfaces, it can be deduced that their true mathematical model is the double-power series

$$f(u,v) = \sum_{q=0}^{m} \sum_{r=0}^{m} a_{qr} u^{q} v^{r} \qquad \ldots\ldots (7.2)$$

for each of the dependent variables (MacCallum, 1970). As there is a growing interest in Coons' patches for moulding engineering surfaces, the latter form has been chosen for this work, rather than (7.1). Moreover, although the present method of computing the equations is different from Coons' method, the corners and boundaries still play an important part.

In this section, it is shown to be necessary to restrict the parameters to a particular region in the parameter plane. The boundaries of this region are the projections of the parametric surface onto the parameter plane. As the surface boundaries are not known

beforehand, they are approximated by the corresponding points in the input data. This means that the form of storing and handling input data must attach special importance to the corners and boundaries. This, in turn, assumes an implicit ordering of the data points; that is to say that the surface must not simply pass near the data points. as in an ordinary least squares fit, but it must follow the points in a given sequence. This is a direct extension of the PCF definition, and, as for PCF, it precludes the use of the method for scattered data.

Further, the concept of 'order' of data points is defined. It is shown that the preservation of this order necessitates the imposition of certain interdependent constraints on the parameter values, resulting in an inconveniently large non-linear minimisation problem.

The method of solution is conceptually identical to that of PCF. Instead of solving the problem as one large Non-linear Programming (NLP) problem, it is divided into far simpler, local, NLP problems, which are then solved serially. Although polynomials are used exclusively in the analysis, most of it is quite general, and applies equally to other parametric functions.

7.2 Statement of the Problem and a Possible Solution

7.2.1 The objective function

For simplicity, and without loss of generality, let the three parametric polynomials be of the same degree, m. Then the given data points are described by

$$x_i = \sum_{q=0}^{m} \sum_{r=0}^{m} a_{qr} u_i^q v_i^r + \epsilon_{xi} ,$$

$$y_i = \sum_{q=0}^{m} \sum_{r=0}^{m} b_{qr} u_i^q v_i^r + \epsilon_{yi} ,$$

and $$z_i = \sum_{q=0}^{m} \sum_{r=0}^{m} c_{qr} u_i^q v_i^r + \epsilon_{zi} , \qquad 1 \leq i \leq n .$$

where u_i and v_i are the parameter values corresponding to the data point (x_i, y_i, z_i), and ϵ_{xi}, ϵ_{yi} and ϵ_{zi} are the deviations of the fitted surface from the data point in the three co-ordinate directions.

Then the least squares parametric surface is found by calculating the sets of coefficients a, b and c, and the [uv] matrix that minimise

$$S(u,v) = \sum_{i=1}^{n} \left[\theta_{x_i} w_{xi}^2 \epsilon_{xi}^2 + \theta_{y_i} w_{yi}^2 \epsilon_{yi}^2 + \theta_{z_i} w_{zi}^2 \epsilon_{zi}^2 \right] \quad \ldots (7.3)$$

whilst maintaining the original surface topology.

The function S is a polynomial of order (4 m) in (2 n) independent variables (assuming implicitly that the polynomial coefficients are dependent on u and v). With polynomials of degree greater than one, the function S is non-convex, with a large number of minima. The imposition of constraints on the variables causes an unpredictable reduction in the number of permissible minima, but the probability of attaining the global minimum of S is low, regardless of the minimisation method used.

With other types of parametric functions, the situation is similar. It is only for simple, monotonic functions, that any method can be guaranteed to converge to the global minimum.

7.2.2 The organisation of data

The method used to store and manipulate the data in the computer

must fulfil the following three requirements:

a) The topology of the data should be represented, and preserved during the fitting procedure.

b) The data description should be similar to that of a Coons' Patch, i.e. considerable emphasis should be placed on the corners and the boundaries of the fitted surface.

c) The preparation of the data for input should be flexible and as simple as possible.

The structure chosen fulfils all the requirements, and has the additional advantage of being very simple to program. Each data point is represented by a non-zero entry at a node on a rectangular, planar planar topological grid. Normally, the physical boundaries of the data surface should coincide with the topological boundaries, which, for convenience of analysis, should not have any vacant nodes. Away from the boundaries, it is desirable to be able to leave nodes vacant, and this is therefore permitted, subject to the proviso that there be at least two points per grid line, for otherwise, the 'line' is not defined. Figure 7.1 is an example of a topological data grid.

Let the grid be of size $(k \times l)$. It is assumed, without loss of generality, that the lowest row and column indices are 1.

Consider the $(n \times 6)$ matrix, in which the rth row contains the information pertaining to the rth data point (i.e. the columns are the n-vectors of x, y, z, u, v and w).

Define the discrete mapping function T(i, j) by

$$(\bar{x}, \bar{y}, \bar{z}, \bar{u}, \bar{v}, \bar{w}) =$$

$$[x_T, y_T, z_T, u_T, v_T, w_T : 1 \leq T(i,j) \leq n, \quad 1 \leq i \leq k, \quad 1 \leq j \leq k] \qquad \ldots\ldots (7.4)$$

i.e. T(i, j) maps the topological co-ordinates (i, j) into physical co-ordinates, provided that the node (i, j) is not a null entry. From

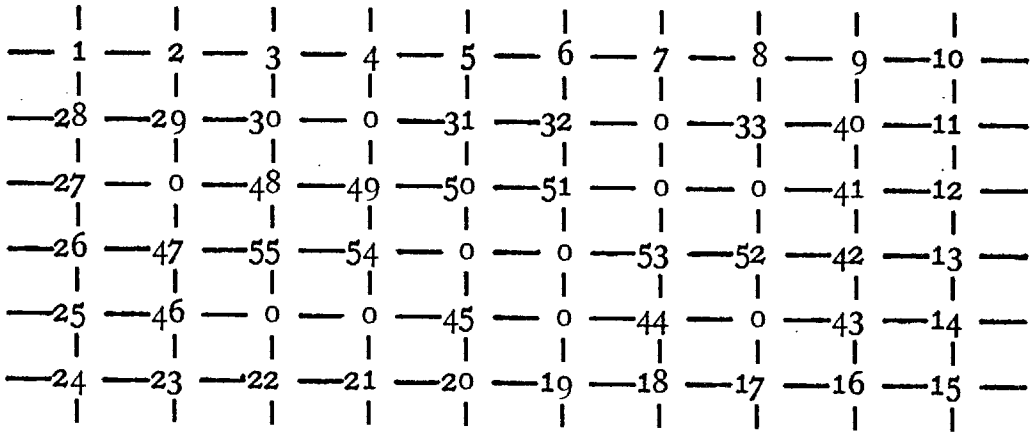


Figure 7.1:- Representation of data points on a planar topological grid.

a data structure point of view, the data points, the parameter values and the weights are stored as linear arrays, and the value of the function T, if non-zero, is the pointer to the data at the node under consideration.

7.2.3 The 'ordering' of the parameters, and the constraints

In the PCF algorithm, it was stipulated that the elements of the parameter vector, $\bar{u}$, be in ascending order, i.e.

$$u_i \leq u_{i+1} \quad , \quad 1 \leq i \leq n-1 \ ,$$

so as to preserve the order of the data points in the fitted functions. With two parameters, the concept of 'order' is not quite so self-evident, and must be defined.

Consider an off-boundary point, O, on a (k X 1) topological grid, at node (i,j) (figure 7.2), where

$$(i,j) = [\ i,j : 2 \leq i \leq k-1 \ , \ 2 \leq j \leq l-1\ ] \ .$$

Define the neighbours P, Q, R and S, of point O, as the nearest valid data entries along the grid lines that intersect at O, i.e.

P is at (i_1, j), $i_1 = \max [\ r : 1 \leq r \leq i-1 \ , \ T(r,j) > 0\ ]$

Q is at (i, j_1), $j_1 = \max [\ r : 1 \leq j \leq j-1 \ , \ T(i,r) > 0\ ]$

R is at (i_2, j), $i_2 = \min [\ r : k \geq r \geq i+1 \ , \ T(r,j) > 0\ ]$

S is at (i, j_2), $j_2 = \min [\ r : 1 \geq r \geq j+1 \ , \ T(i,r) > 0\ ]$

Let these points be mapped onto the uv-plane, via the function T, to produce the polygon shown in figure 7.3, where the primes indicate the mapping. The mapped co-ordinates are as follows:-

O' is at (u_o, v_o), $o = T(i,j)$

P' is at (u_p, v_p), $p = T(i_1, j)$

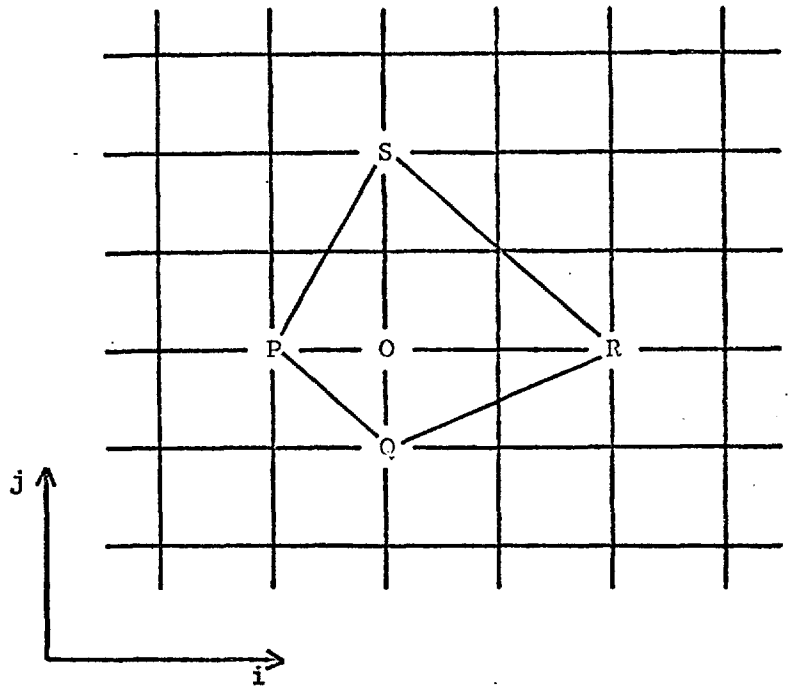


Figure 7.2 :- The topological polygon about an off-boundary data node.

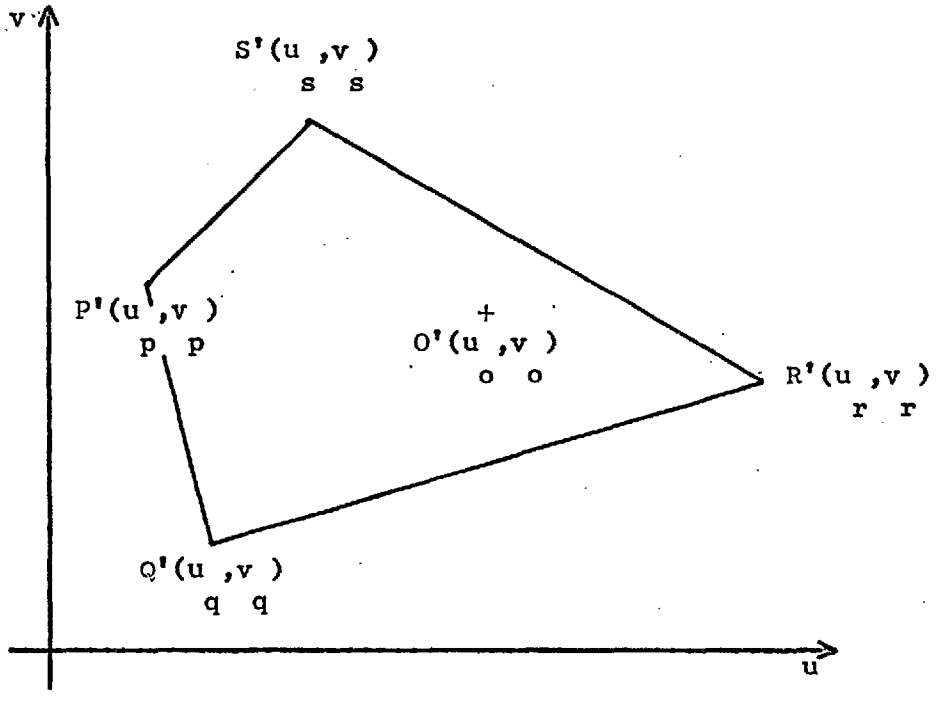


Figure 7.3 :- The mapping of Figure 7.2 onto the uv-plane.

$$Q' \text{ is at } (u_q, v_q), \quad q = T(i, j)_1$$

$$R' \text{ is at } (u_r, v_r), \quad r = T(i, j)_2$$

$$S' \text{ is at } (u_s, v_s), \quad s = T(i, j)_2$$

Having defined the neighbours of point O, there are several possible ways of defining the 'order' of the parameters. The one chosen was dictated by simplicity, and by the failures of several other definitions that had been tried.

Let the following 'order' be initially induced in parameter values:

a) $u_p \leq u_o \leq u_r$, i.e. along a horizontal grid line, that is, for j = constant, u varies as i. ..... (7.5a)

b) $v_q \leq v_o \leq v_s$, i.e. along a vertical grid line, that is, for i = constant, v varies as j. ..... (7.5b)

To complete the definition, the ranges of u and v must be specified. As for PCF, it can be shown that only the distribution of the parameter values within bounded intervals affects the results, not their absolute values. The final part of the definition is therefore:

c) $-2 \leq u_t \leq +2$,

and $-2 \leq v_t \leq +2$. $\qquad 1 \leq t \leq n$ ..... (7.5c)

The simplest geometrical constraints which preserve this order for all off-boundary points (on the boundaries, the problem reduces to a two-dimensional one, and is therefore a subset of the following) are the straight lines joining the points P'Q'R'S'P'. The region thus defined is termed the 'feasible domain', and any point within it is deemed 'feasible'. This is qualified algebraicly for a general point

(u,v), by specifying that the constraint vector, $\bar{d}$, be non-positive for a feasible point, i.e.

$$\bar{d}(u,v) \leq 0$$

where $\bar{d} = (d_1, d_2, d_3, d_4)$

and $$d_1(u,v) = (u_s - u_p)v + (v_p - v_s)u + v_s u_p - u_s v_p,$$

$$d_2(u,v) = (u_r - u_s)v + (v_s - v_r)u + v_s u_r - u_s v_r,$$

and so on, for the other two lines.

The case depicted in figure 7.3 is the simplest possible. Scrutiny of different mappings from the topological grid onto the uv-plane, e.g. figure 7.4, shows immediately that the four constraints are always sufficient to define a convex feasible domain for the point O'. However, on consideration of the effects of the movements of O' on the subsequent feasibility of its neighbouring points, it may be seen that more restrictions are necessary. For instance, let Q be the next node in the local minimisation procedure - figure 7.5. Then the latest position of O' is one of the apexes of the new constraint polygon; let F', G' and H' be the other apexes. It is clear that if O' is on the shaded side of either line F'Q'R'', or line H'Q'P'', point Q' is rendered non-feasible by the geometrical definition, although it may still obey the conditions of equations 7.5. Two (non-positive) constraints are added to prevent this, namely

$$d_5(u,v) = (u_f - u_q)v + (v_q - v_f)u + v_q u_f - u_q v_f$$

and $$d_6(u,v) = (u_q - u_h)v + (v_h - v_q)u + v_h u_q - u_h v_q$$

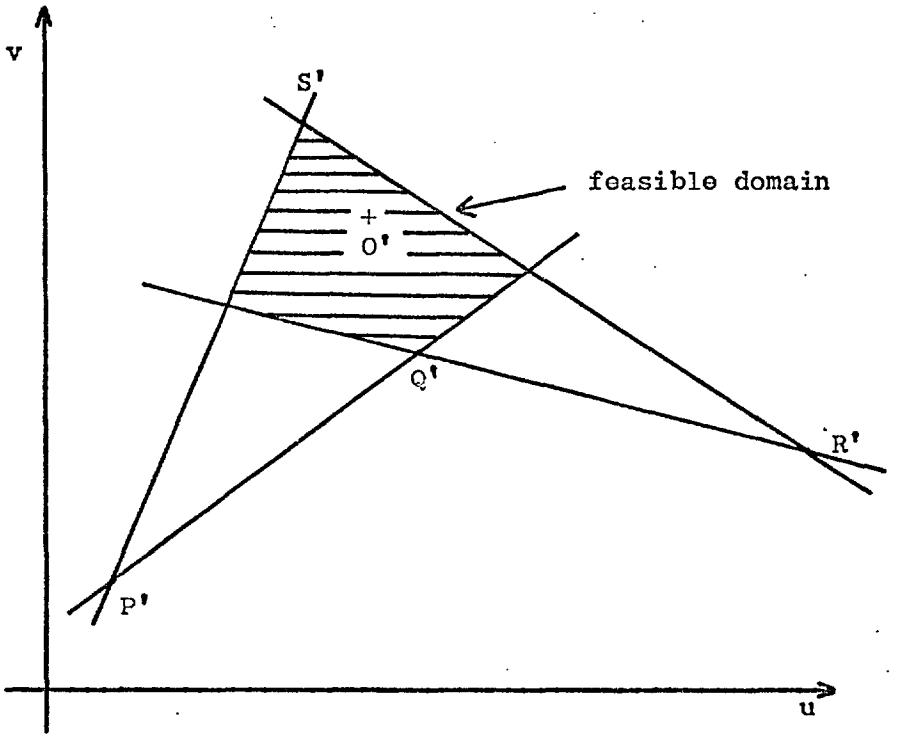


Figure 7.4 :- A more complex constraint polygon.

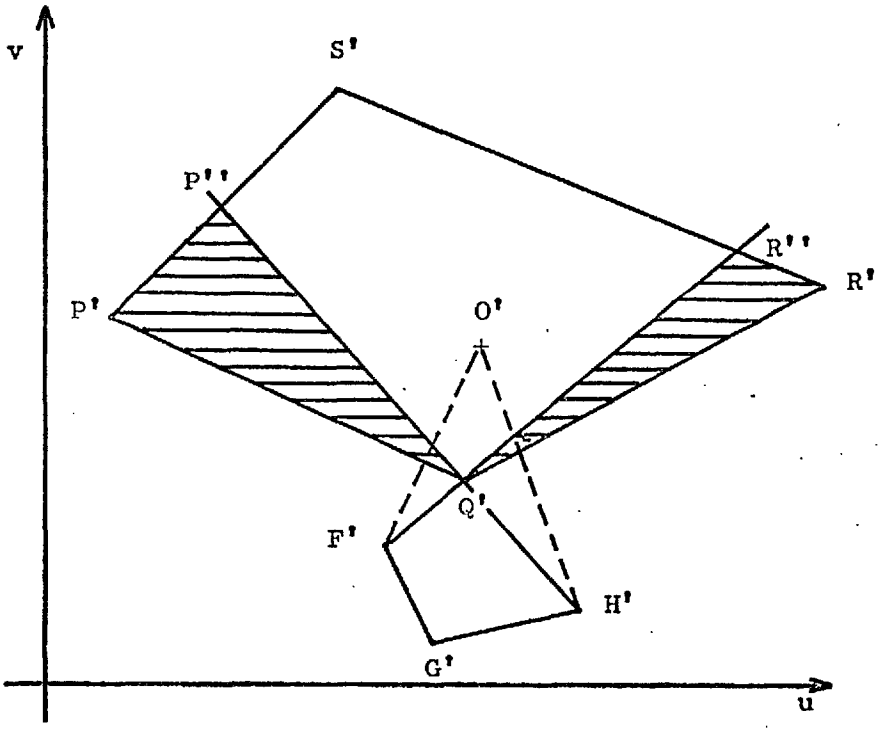


Figure 7.5 :- Effect of the current node on its neighbours.

Similar considerations apply to the points P', R' and S', leading to eight constraints that must be added to the basic set, making the total number of constraints twelve, at each node.

The initial definition of the 'order' of the parameters is now amended, or rather, replaced, by the stricter constraint definition:

$$\bar{d}(u,v) = (d_1, d_2, \ldots, d_{12}) \leq \bar{0} \qquad \ldots\ldots (7.6)$$

for each off-boundary point. On each boundary, equation 7.5c applies, i.e. one of the variables is fixed; a subset of $\bar{d}$ restricts the movements of the other variable. At each of the four corner points both variables are fixed (cf. similarity with the end-points in PCF).

If this argument seems self-contradictory, it must be remembered that the geometrical definition of feasibility is the important one and the algebraic definition has been constructed to fit it, and not vice-versa.

7.2.4 Non-linear programming

The constraints on the distribution of the parameter values identified the problem as a non-linear programming one, with a non-convex objective function and linear constraints. To calculate the size of the problem, consider n data points, on a $(k \times 1)$ grid, and parametric polynomials of degree m. Then, with the particular polynomials used, the dimensionality of the problem is

$$n_d = 2n - (k + 1 - 2) + 3(m + 1)^2$$

(where the middle term is due to the reduced number of independent variables on the boundaries, and the last term, giving the total number of polynomial coefficients, is added to completely generalise the definition of the problem), and the number of constraints is

$$n_c = 12(n - k - 1 + 2)$$

(boundary points have only four constraints each)

Typically, $n = 200$; $k = 10$; $l = 20$; $m = 3$, giving $n_d = 420$ and $n_c = 2064$. The solution of a problem of this size is very difficult with any established general method.

Unfortunately, this meant that there was no simple way in which the results of the technique described below could be verified, as was done for PCF. Therefore, more sophisticated convergence tests have had to be included in the implementation of the algorithm.

7.2.5 The algorithm

In priciple, the algorithm used for surface fitting is identical to that of curve fitting, although the computational procedures for the former are far more complex. The outline of the procedure is repeated here, for clarity:

a) Given an approximation to the parameter values, it is a simple matter to compute the best polynomial fit in the least-squares sense, to the given data.

b) Having computed the polynomials, it is possible to improve the approximation to the parameter values by minimising the weighted square of the deviation at each data point in turn. This is hence-forth referred to as the 'local optimisation problem'.

c) An overall convergence test is applied. If convergence has not been reached, the process is repeated from (a).

This method is still an Alternating Directions, or Direct Search, one (Box, 1965), although each of the n dimesnsions corresponding to the data points, spans E^2 (because the parameters are treated in u-v pairs). For general functions, this algorithm is very inefficient, as is usually the case for algorithms which predefine the search directions (Rosenbrock, 1960). Convergence to a minimum can only be

proved for unconstrained quadratic functions.

There is one aspect of non-linear optimisation, of which the direct search method used here is thought less likely to suffer from than other algorithms, and that is the collapse into a subspace of the original problem, of less than n_d dimensions. This is caused by convergence to a saddle point. The usual way of detecting and countering this is to perturb the solution vector, to let the algorithm converge again, and to compare the new solution with the old one. In the procedure described above, the least squares fit in each cycle provides this perturbation implicitly.

7.3 Computational Techniques

7.3.1 The parametric polynomials

7.3.1.1 The derivation

A least-squares fit with model (7.2) by orthogonal polynomials using a three-term recurrence formula, can be derived in exactly the same way as with model (7.1) (MacCallum, 1970 ; Kantorowitz, 1967). All the comments about weighting, stability precision and the imposition of boundary conditions, made about the latter in Chapter 3 (Section 3.3), apply equally to the former and are therefore not given here. It is the consideration of precision in the derivation, that governed the choice of the $[-2,+2]$ range for the parameters.

7.3.1.2 Three dependent variables

The parametric surface description has, of course, three dependent variables, rather than one, as is normally considered for standard surface fitting. The extension is straight forward. The orthonormal polynomials need only be calculated once (per fit), and the three sets of coefficients are calculated from them using the basic orthonormal vector relationship.

7.3.2 The local optimisation problem

7.3.2.1 Introduction

The local problem, as described above, is a NLP problem in its own right. It is the same as the global one - equation 7.3 - without the summation over the data points, i.e. generally a non-convex polynomial in two variables, with twelve linear constraints. It is a somewhat unusual constrained optimisation problem, inasmuch as the solution is expected to be an interior point most of the time. The algorithm chosen to solve it had therefore to be very efficient when the constraints are not effective, and at least reasonably efficient with constraints. The method finally chosen is a Conjugate Gradients method, in conjunction with Gradient Projection for handling the constraints. The former is presented in a completely general way; the latter, which is normally complex, reduces to a simple form, for three reasons, namely

a) The low dimensionality of the problem,

b) The bounded feasible region,

and c) The feasible initial approximation,

finally leading to an efficient procedure for solving the problem, including practical convergence tests and safeguards.

7.3.2.2 Minimisation by conjugate gradients

There are many conjugate gradients minimisation methods. The one used here stems from a particular way of solving a set of simultaneous linear equations, with a positive-definite coefficient matrix (Hestenes and Stiefl, 1952). It can further be shown to be a generalisation of the Gram-Schmidt orthonormalisation procedure (Beckman, 1960). A brief description only is given here, without proofs.

Let $\bar{x}$ be the column vector of the co-ordinates of the point

$(x_1, x_2, \ldots, x_n)$ in a n-dimensional vector space.

Consider a n-dimesional quadratic in $\bar{x}$, of form

$$f(\bar{x}) = a + \bar{b}\ \bar{x} + \frac{1}{2}\ \bar{x}'\ Q\ \bar{x}$$

where Q is the (constant) Hessian matrix of second partial derivatives of the function, taken to be positive-definite and the prime denotes the transpose operator.

The gradient of the quadratic is given by the column vector

$$\bar{g}(\bar{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots\ldots\ldots, \frac{\partial f}{\partial x_n}\right]$$

$$= \bar{b} + Q\,\bar{x}$$

The minimisation is effected by linear minimising searches in a sequence of directions, $\bar{p}_0, \bar{p}_1, \ldots, \bar{p}_n$, where the directions are related by

$$\bar{p}_{i+1} = -\bar{g}_{i+1} + \beta_{i+1}\ \bar{p}_i\ ,\quad 0 \leq i \leq n-1 \qquad \ldots\ldots (7.6)$$

where β is a scalar given by

$$\beta_{i+1} = \frac{\left\|\bar{g}_{i+1}\right\|^2}{\left\|\bar{g}_i\right\|^2} \qquad \ldots\ldots (7.7)$$

and the first search direction is that of steepest descent, i.e.

$$\bar{p}_0 = -\bar{g}_0 \qquad \ldots\ldots (7.8)$$

In each search, $\bar{x}$ is modified by

$$\bar{x}_{i+1} = \bar{x}_i + t_{i+1}\ \bar{p}_{i+1} \qquad \ldots\ldots (7.9)$$

Where t is a scaler, computed in the lineer search so as to minimise

$$h = f(\bar{x}_i + t\,\bar{p}_{i+1}) \qquad \ldots\ldots (7.10)$$

implying that

$$\bar{p}_i' \; \bar{g}_{i+1} = 0 \qquad \ldots\ldots (7.11)$$

Equations 7.6 - 7.10 generate a sequence of conjugate, or Q-orthogonal, vectors, that is $\bar{p}_i' Q \bar{p}_j = 0$ for $i \neq j$. The procedure can be proved to reach the minimum of the quadratic in n steps, or less, should any of the t's be zero, regardless of the starting point (Fletcher and Reeves, 1964). It is noted that all the equations, except equation 7.7, are quite general. The latter, while perhaps not necessarily producing the most efficient minimisation, is certainly one of the simplest to calculate, and probably the most economical in terms of computer storage space.

In applying this algorithm to a non-quadratic function, the process becomes iterative, rather than n-step. For non-convex functions, a local, rather than global, minimum is all that can be ensured, and even that subject to the following precautions:-

a) A descent angle is maintained, i.e. $\bar{p}_{i+1}' \; \bar{g}_{i+1} \leq 0$.

b) Every few iterations, the procedure is re-initialised, using the current point as the first approximation.

The reason for these precautions is that the algorithm approximates the general objective function by a quadratic, and uses the properties of the latter; for the general function, this approximation is valid only in a small region, near the sought minimum. In the non-quadratic region, the information accumulated by the algorithm during some iterations can cause the newly calculated search direction to be far from optimal. In extreme cases, an ascending direction may be chosen. If such directions were allowed, convergence becomes

difficult to guarantee, although it is still likely in practice.

A convenient criterion for (a) has been found to be:-

$$\text{Reset to steepest descent if } \frac{-\bar{p}'_{i+1}\ \bar{g}_{i+1}}{\left\|\bar{p}_{i+1}\right\|\ \left\|\bar{g}_{i+1}\right\|} < \Theta ,$$

i.e. when $\bar{p}_{i+1}$ differs from the zero-descent direction by less than $SIN^{-1}(\Theta)$, where Θ is some threshold value, which depends on the problem and on the computer used, of order 10^{-3} to 10^{-4}.

The number of iterations to be executed before a complete re-initialisation is less obvious. It seems advisable, and is necessary for quadratic convergence, that the entire n-space spanned by $\bar{x}$ should be explored, entailing at least n iterations. There is no equivalent for the maximum number of iterations. Experience has indicated that a reset should be forced every (n + 1) iterations (i.e. every 3 for this particular problem).

No sharp analytical results are available, that correlate the efficacy of the modified algorithm with the above criteria. The usual bench-marks (e.g. Rosenbrock's banana shaped valley function), and particularly, examples of the function under consideration, have been found insensitive to the exact values used.

7.3.2.3 The gradient projection method

Gradient projection techniques are lucidly treated in the literature (Rosen, 1960 ; Goldfarb, 1969). The general form is complex and therefore only the much simplified subset needed for this work is given here.

The constraints and the constraint basis

Consider the constraints as defined in Section 7.2.3. The feasible domain has arbitrarily been chosen to be in the negative

half-spaces of the lines defined; that is, for any feasible point (u_0, v_0),

$$a_{i1} u_0 + a_{i2} v_0 - b_i \leq 0, \quad 1 \leq i \leq n_c ,$$

where n_c is the number of constraint lines, usually 12, (a_{i1}, a_{i2}) are the coefficients of each constraint line, which have been normalised for convenience, i.e.

$$a_{i1}^2 + a_{i2}^2 = 1, \quad 1 \leq i \leq n_c ,$$

and b_i is a constant for each line.

It is convenient, for the purpose of the following discussion to denote the general point $(u_0 \quad v_0)$ by the two-dimensional column vector $\bar{x}$ (as in the conjugate gradients section), and to group all the normal vectors in a $(n_c \times 2)$ matrix A, so that the constraint vector $\bar{d}$, defined in equation 7.6, is given by

$$\bar{d} = A\bar{x} - \bar{b} \leq \bar{0}$$

If q linearly independent constraints intersect at $\bar{x}$, they and their unit normals are termed the 'constraint basis'.

The constraint basis must be updated after each linear search. In practice, the intersection can only be computed within the precision of the computer used. Instead of letting this be an arbitrary quantity, the constraint basis is defined as the intersection of all the lines that are within δ of $\bar{x}$ - figure 7.6 - where δ is a value on the limit of precision of the particular computer, usually of the order of 10^{-8} to 10^{-6}.

For this problem, the number of constraints that apparently intersect at (u_0, v_0) could be any between zero and twelve. Clearly, the basis cannot consist of more than two vectors and a way of eliminating the redundant ones has had to be developed.

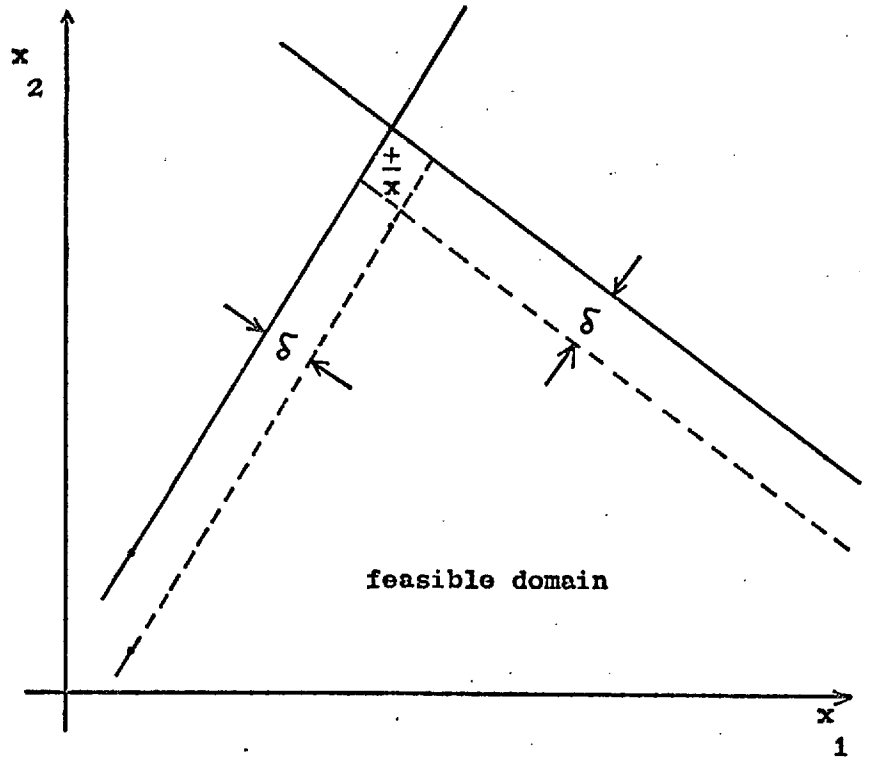


Figure 7.6 :- Determination of constraint basis, with two constraints.

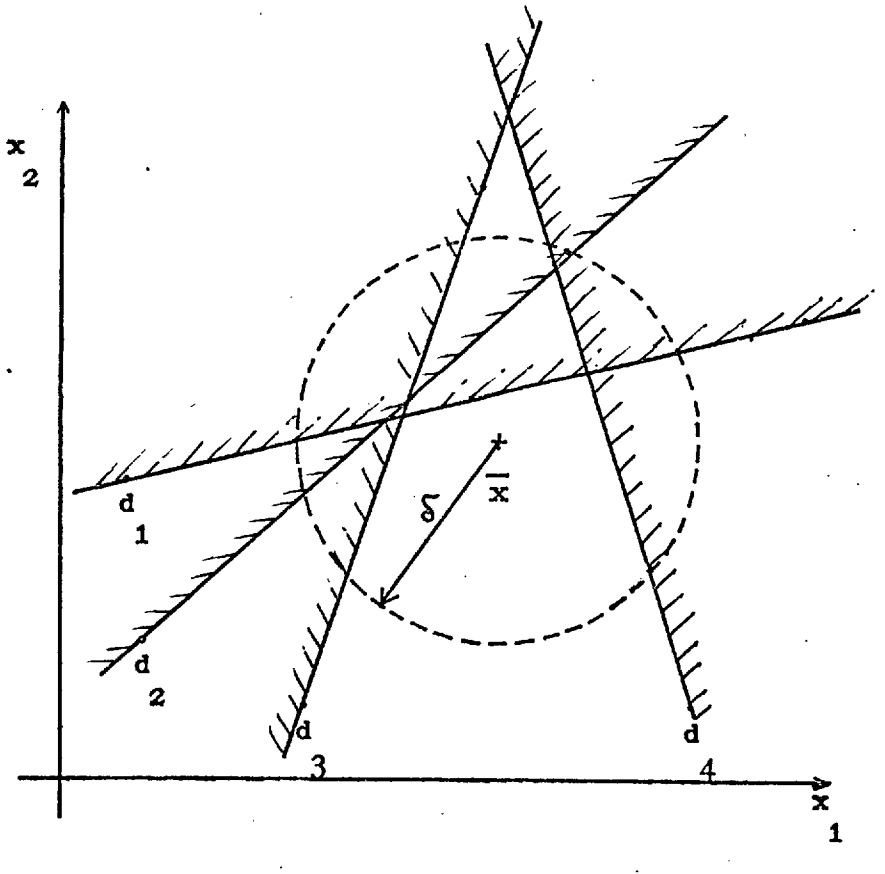


Figure 7.7 :- Elimination of redundant constraints.

In two dimensions, this is simple. Referring to figure 7.7, it consists of starting from a feasible point, and eliminating all but the two lines which include the acutest angle about the point.

In the full gradient projection method, special techniques to update the constraint basis after each iteration are necessary, for efficiency. It would have been pointless to do so for a low order problem such as this, and therefore the basis is recomputed in each iteration.

Projecting the gradient

It can be shown that if the projection operator devised by Rosen is applied to two dimensional problems - illustrated in figure 7.8 - it simplifies to the the following set of rules (ignoring convergence tests for the moment) :-

a) One constraint in basis

If $\bar{a}_q \bar{g} \leq 0$, move along the constraint in the direction of descent; otherwise, treat as unconstrained.

b) Two constraints in basis

Resolve by Lagrange Multipliers (LM); if both LM's negative, then the current point is the constrained minimum; otherwise, select the constraint corresponding to the greater LM, and continue with rule (a).

7.3.2.4 The linear search

Having computed a search direction, $\bar{p}_i$, in the ith iteration, it is required to find the minimum of the function along $\bar{p}_i$; that is, to determine t_{min}, such that

$$\frac{dh}{dt}(t_{min}) = 0$$

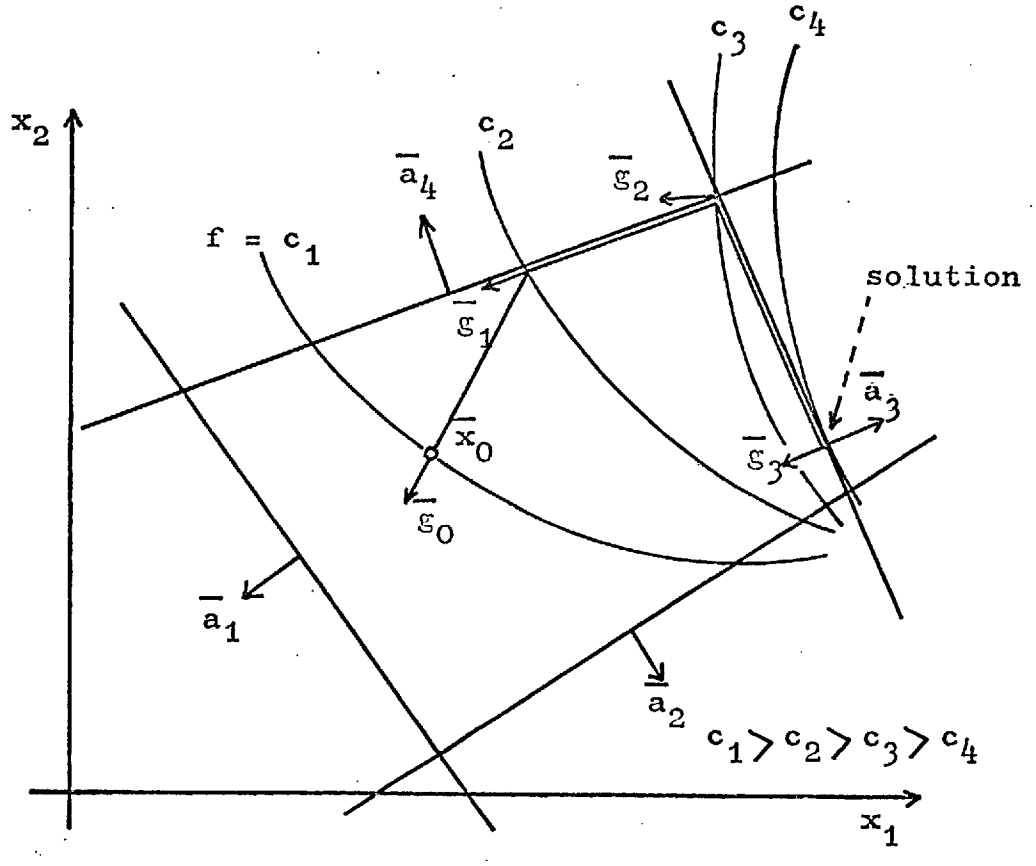


Figure 7.8 :- Gradient projection in two dimensions.

where $h(t) = f(\bar{x}_i + t\bar{p}_i)$

A cubic interpolation scheme proposed by Davidon (1959), and used by several other authors, has been found suitable. This scheme uses function values and gradients, and therefore requires more computation than schemes using function values only. On the other hand, the gradient at the starting point is known and that at the minimum is needed for the next iteration; also the procedure has been found very efficient in terms of the number of steps to the minimum.

The initial unconstrained estimate for t can be obtained by assuming $h(t)$ to be a quadratic, with a minimum value f_{min}, at

$$t_0 = 2(f_{min} - f_i) / \bar{p}_i' \bar{g}_i \qquad \ldots\ldots (7.12)$$

The function in question being of a constrained least squares type, f_{min} can safely be taken to be zero. A gross over-estimate by equation 7.12 is prevented by computing the upper bound of t, as follows:-

Given the constraint vector in the ith iteration

$$\bar{d}(\bar{x}_i) = A\bar{x}_i - \bar{b} \leq 0 \quad \text{for feasible } \bar{x}.$$

Find t_{max}, such that

$$t_{max} = \max [\, t : t \geq 0 , \bar{d}(\bar{x}_i + t\bar{p}_i) \leq -\delta/2 \,]$$

Consider the jth constraint. Two possibilities exist, namely

1) $\bar{a}_j \bar{p}_i \leq 0$, i.e. $\bar{p}_i$ does not intersect the jth constraint.

Then $d_j(\bar{x}_i + t\bar{p}_i) \leq \delta/2$ for all $t \geq 0$. This is excluded from further consideration.

2) $\bar{a}_j \bar{p}_i > 0$.

$$\text{Then} \qquad t_j = \frac{-[d_j(\bar{x}_i) - \delta/2]}{\bar{a}_j \bar{p}_i}$$

The upper bound of t is thus given by

$$t_{max} = \min_j [t_j], \quad 1 \leq j \leq n_c$$

In addition to imposing $t \leq t_{max}$, changes in the function value during the linear search must be kept 'reasonable' for accurate interpolation; also if the gradient vanishes, a check must be made that the point found is not a maximum.

The accuracy with which t_{min} needs be calculated has been a controversial subject for some time. Unnecessary precision is wasteful in terms of computing effort, but, on the other hand, the search directions are only mutually conjugate (for a quadratic objective function) if the true minimum is found in each linear search. For the function in question, the interpolation formula is exact in the quadratic neighbourhood of the minimum, and there is little to be gained from high precision away from the minimum. In practice, a parameter must be found for the accuracy of the linear search, which is a compromise between computing effort and the rapid convergence of the complete procedure.

7.3.2.5 Convergence tests

There are three standard ways for the minimisation procedure to terminate after i iterations, namely

a) Unconstrained

$\|\bar{g}_{i+1}\| < \gamma$, where γ is some small assigned value (formally zero).

b) One constraint in basis

$||(I - \bar{a}'_q \bar{a}_q)\,\bar{g}_{i+1}|| < \gamma$, where I is the 2X2 identity matrix, i.e. the gradient of the function is orthogonal to the qth constraint line.

c) Two constraints in basis

Let $\bar{g}_{i+1} = \lambda_j \bar{a}_j + \lambda_k \bar{a}_k$, where the λ's are the LM's. The current point is the constrained minimum if $[\lambda_j , \lambda_k] < 0$

These tests have been found adequate provided that the value of γ was selected judiciously, according to the problem and the computer used. Usually $\gamma = 0$ [10^{-6} to 10^{-3}].

7.3.2.6 The complete local optimisation algorithm

Step 1 Set 'reset to steepset descent' flag for the unconstrained conjugate gradients section.

Step 2 Given $\bar{x}_i$, compute f_i and $\bar{g}_i$.

If $||\bar{g}_i|| < \gamma$, exit; else

Step 3 Compute the constraint values, and find the ones that intersect at $\bar{x}_i$.

If more than two - step 9 ;

If two - step 8 ;

If one - step 7 ; else

Step 4 Unconstrained minimisation by modified conjugate gradients.

If 'reset to steepest descent' flag is set - step 6 ;

Compute β_i , $\bar{p}_i$

$$\text{If } \frac{-\bar{p}'_i \bar{g}_i}{||\bar{p}_i||\,||\bar{g}_i||} < \theta ,$$

i.e. nearly zero-descent - step 6 : else

Step 5 Linear search. Then step 2 .

Step 6 Reset to steepest descent, i.e. $\bar{p}_i = -\bar{g}_i$.

Clear 'reset to steepest descent' flag.

Then step 5 .

Step 7 Single constraint in basis

If $\bar{a}_q \bar{g}_i > 0$, constraint not effective - step 6 ;

Compute new search direction $\bar{p}_i = -(I - \bar{a}_q' \bar{a}_q) \bar{g}_i$.

If $||\bar{p}_i|| < \gamma$, constrained minimum reached - exit;

else step 5 .

Step 8 Two constraints in basis

Resolve $\bar{g}_i$ by Lagrange Multipliers.

If $(\lambda_j , \lambda_k) < 0$ - constrained minimum - exit ;

If $\lambda_j > \lambda_k$ $q := j$; else $q := k$.

Then step 7 .

Step 9 Redundant constraints

Remove the redundant constraints by finding the two lines which include the smallest angle about $\bar{x}_i$. Then step 8 .

In practice, an upper limit is set to the number of iterations allowed, both in the linear search and in the overall procedure.

7.3.3 The first approximation to the parameter values

It was shown in the PCF analysis that there is a direct, non-linear relationship between the distribution of the parameter values and that of the data points. This relationship was used to generate

the first approximation to the paremeter vector.

There is, unfortunately, no simple equivalent in the case of two parameters. The formal equivalent of 'linearised' arc-length in E^2 is 'linearised' surface area in E^3, but this yields only the values of the products of parameter increments from point to point, and is useless unless a relationship between the two parameters is known or assumed.

It is possible to treat the points on each topological grid line as a plane curve, and to use the arc-length approximation directly. This has been found to lead to non-feasible first approximations for all but very smooth data. Although a dummy iteration, that corrects the initial parameter matrix, can be included in the algorithm, the object of utilising the relationship between the arc-length and the parameter values would then be defeated.

The first approximation chosen ultimately was a topologically equally-spaced one, initially independent of the input data (except for the size of the grid); that is, at node (i,j) on a $(k \times 1)$ grid,

$$u'_{ij} = 2\,\frac{(2\,i - k - 1)}{(k - 1)} \qquad \ldots\ldots (7.13a)$$

$$v'_{ij} = 2\,\frac{(2\,j - 1 - 1)}{(1 - 1)} \qquad \ldots\ldots (7.13b)$$

where the primes denote dummy variables. The true parameter values are those mapped from equation 7.13 via the mapping function $T(i,j)$, - equation 7.3 - i.e. only at the nodes with data entries.

This first approximation has the advantages of simplicity and feasibility. In PCF, it was found that the equally-spaced approximation usually led to the same solution as the the arc-length one (which in all the test cases proved to be the global optimum),

although more iterations have had to be performed. It is thought that convergence to a global solution is also likely for parametric surface-fitting (PSF), although no way has been found to check this.

7.3.4 Overall convergence of algorithm

The convergence condition for the overall system is mathematically the same as that for the local problem. Computation may be terminated when the vector of derivatives of the global objective function S, with respect to all the variables (including the polynomial coefficients), or its projection onto the constraint subspace if any of the constraints are active, has all zero components.

In this case, the least squares fit in each major iteration ensures that at least $3(m+1)^2$ components of the gradient vector are exactly zero. Because of the way S is minimised here, the formal convergence test, as described above, would be both inconvenient and misapplied. Therefore, for reasons similar to those of PCF, the procedure is allowed to iterate, until either the decrease of S in two successive iterations is less than a prescribed threshold, or a specified running time or number of iterations have elapsed, whichever occurs earlier.

7.3.5 The simplification of data preparation

The data structure as defined in Section 7.2.2, permits great flexibility of input but it must be remembered that the topological ordering of the data affects the starting point of the iterative procedure, and thus probably the final result. The effect is highly data dependent and not easily calculable, and therefore care must be taken when preparing data.

In order to simplify the preparation of data, two facilities have been provided, as follows:

a) Given a grid of maximum size, the input data may be defined anywhere on it. That is to say, the boundaries of the data surface need not coincide with those of the grid, nor need they be explicitly defined. The rows and columns with the lowest and highest indices are taken to be the boundaries, thus implying the corner points, too.

b) The boundaries (i.e. the rows and columns of highest and lowest indices) may be input with missing points. An algorithm has been devised, that 'pulls' points outwards from inside the mesh, until the corners and boundaries are full. The operation is such that the original topology of the mesh is not altered, i.e. the topological plane is 'stretched' only, not 'folded', and no new intersections between grid lines are created. This is best illustrated by an example; figure 7.10 shows the data grid of figure 7.9 after the transformation. Although adequate, the algorithm is mechanical; a better arrangement of data can nearly always be produced manually.

7.3.6 Graphical output

Conventional graphical display of surfaces consists of either an isometric drawing of the surface, or sections parallel to the co-ordinate planes. Occasionally, a non-orthogonal view is also given, mainly for clarification. The former, which is only a guide to the shape of the surface, remains virtually unchanged when parametric functions are used; the lines plotted in the isometric drawing become space curves, with x, y and z generated on a regular u-v grid. The latter reduces to a contouring problem, because each parametric polynomial itself generates a surface. Say, for instance, that a $z - x$ section is wanted, for $y = y_a$. The values of the parameters that correspond to the contour $f_y(u,v) = y_a$ must be computed, and substituted in f_x and f_z, to calculate the required curve.

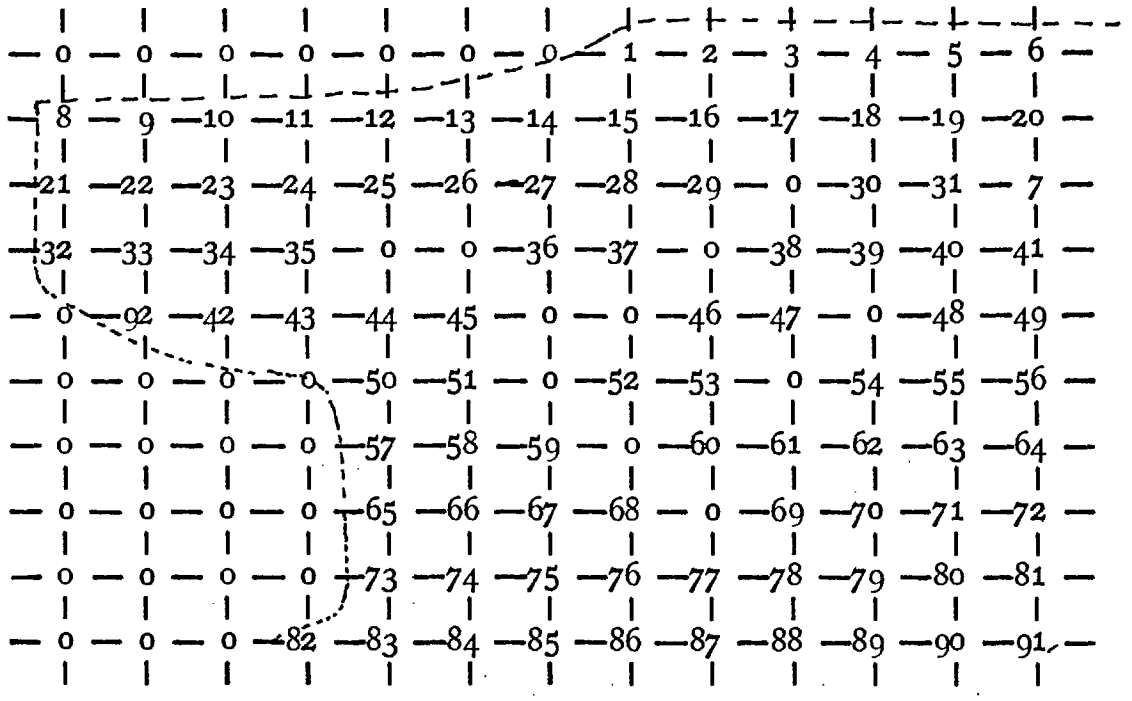


Figure 7.9 :- A typical stern topology.

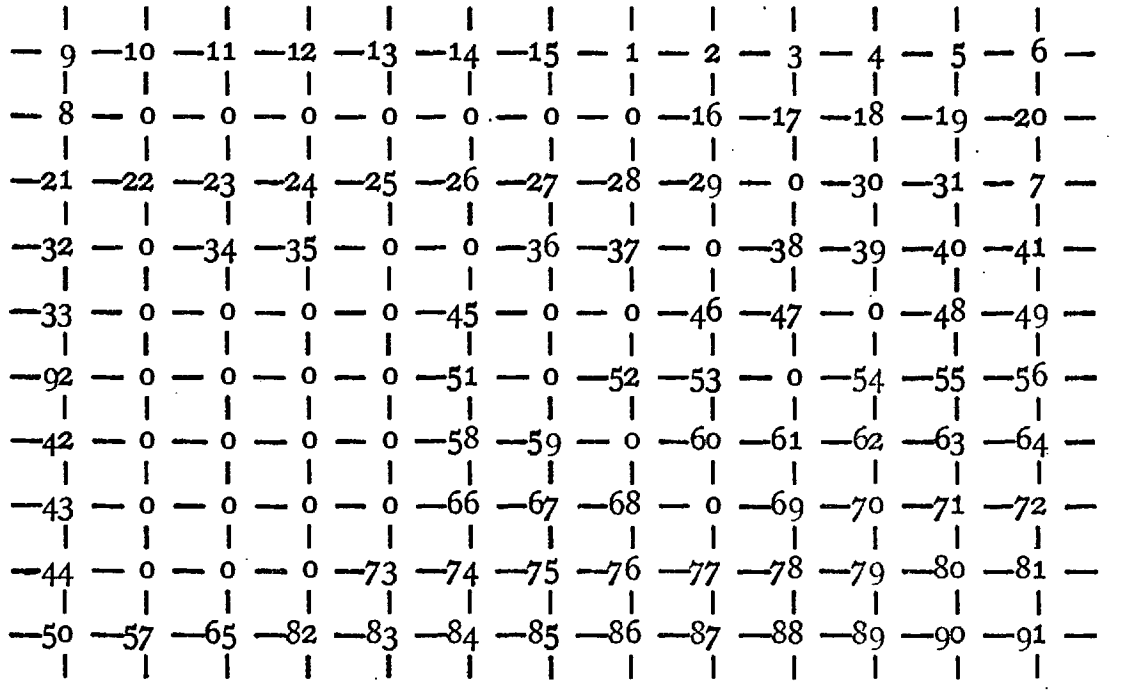


Figure 7.10 :- The stern topology of Figure 7.9, with filled boundaries.

Assume that no extrapolation is wanted. The problem can then be stated in its most general form as:

$$\text{Minimise} \quad h(u,v) = [\, f(u,v) - c \,]^2 \qquad \ldots\ldots (7.14)$$

$$\text{Subject to} \quad |u| \,,\, |v| \leq 2 \qquad \ldots\ldots (7.15)$$

where $f(u,v)$ is the parametric function that is being contoured, and c is the contour value. For the example quoted above,

$$h(u,v) = [\, f_y(u,v) - y_a \,]^2 .$$

Further, assume that the point (u_0, v_0) is known to lie on the contour. Let an exploratory step be taken in one of the parameters, e.g. let $u = u_0 + \delta u$. The approximate value of the corresponding increment in v for remaining on the contour can be found by equating the total differential of $f(u,v)$ to zero, i.e.

$$\frac{\partial f}{\partial u} \delta u + \frac{\partial f}{\partial v} \delta v = 0 ,$$

whence $$v = v_0 + \delta v ,$$

and the exact value of v can be found by univariate minimisation of $h(u,v)$ with respect to v, subject to the constraints (7.15).

The starting point of the contour can be found by univariate minimisation of $h(u,v)$ along each of the boundaries of the permissible parameter domain, until an exact solution is found (one of the parameters will have the value of ± 2).

This problem is clearly a subset of the local optimisation problem in the surface fitting procedure, and can be solved by the same method. The facility of graphical output on a digital plotter could therefore be included in the PSF system with little additional effort. The method used to find the starting point of each contour implies that the plotting is restricted to curves corresponding

to single segments of open contours. This is a little limited, but contour plotting was not the main aim of the work.

7.4 Results

7.4.1 Data used

The algorithm has been tried on a variety of data, ranging from simple surfaces that could have been fitted equally well using conventional methods, to data with extreme gradients and curvatures.

As a test of the capabilities of the PSF method, an attempt was made to fair the entire hull of Model 1 and the forebody of Model 2 (up to the knuckle), each with a single patch, using polynomials of degrees 3 , 4 and 5. For the former, it was difficult to find a suitable topological arrangement for the data; for the latter, a good fit to the relatively great length of the parallel middle body conflicted with a good fit to the bulbous bow. The 'fits' were good numerically, but impractical. They are of academic interest only, and the plots are not included here.

The method was finally applied to several patches, both from Model 1 and from Model 2. These patches have been derived from the hulls' offsets by dividing the hulls at convenient points, with simple boundaries. The results, which are not fully converged, are shown in figures 7.11 to 7.13.

7.4.2 Effects of the degree of the parametric polynomials

The method has been tried with polynomials of degrees between 3 and 5. Unlike the PCF method, dramatic changes in the quality of fit were usually experienced when the degree was changed. A probable reason is that surfaces require more degrees of freedom for adequate representation.

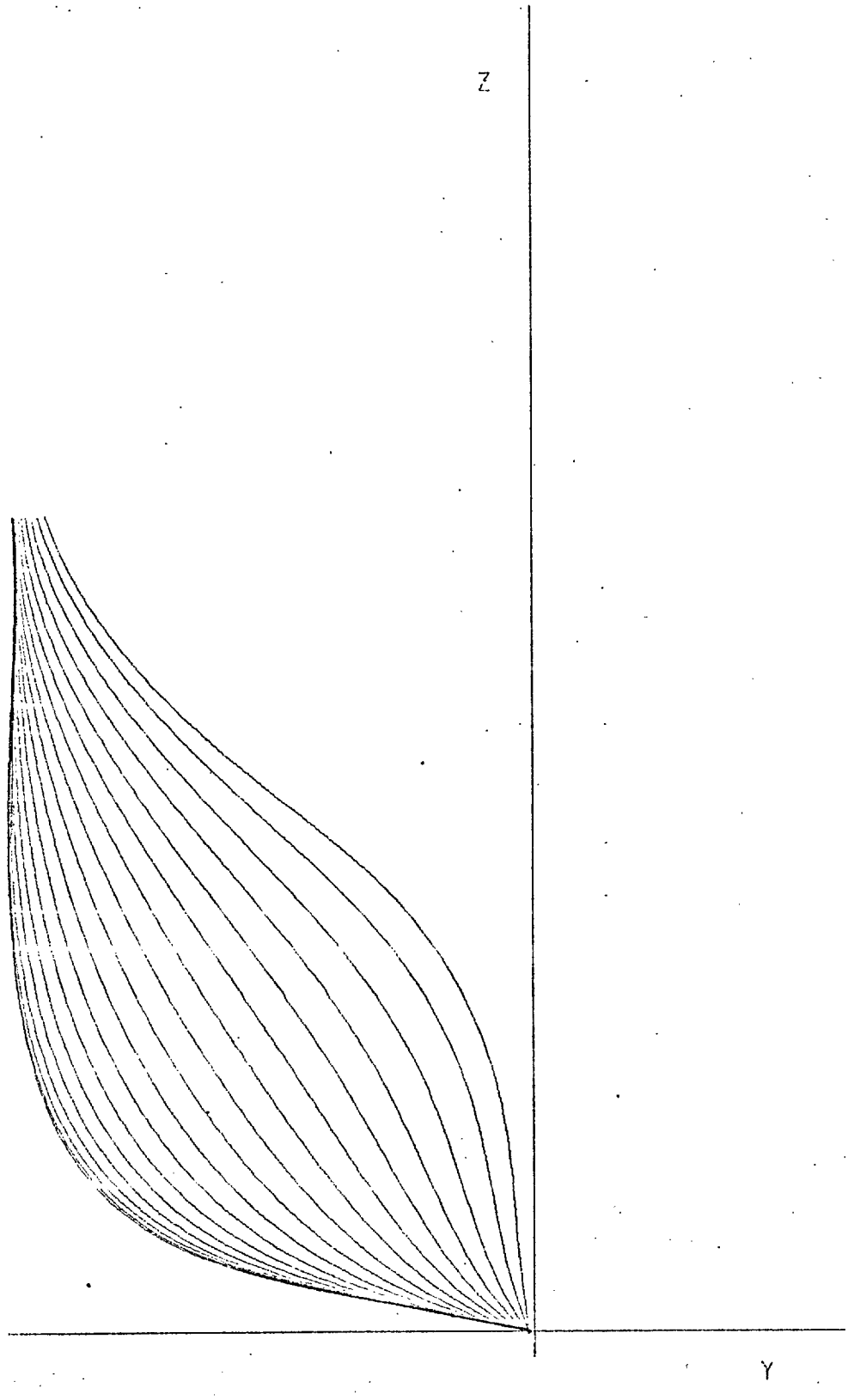


Figure 7.11:- Cross-sectional curves, at 1/4 station intervals, of a parametric surface fit to the Afterbody of Model 1. The surface is an unweighted quartic fit to the offsets from Station 1/2 to Station 5, from the Keel to the 160% waterline. The maximum deviation is 1.56% at Station 5, 100% waterline, after 36 iterations (incomplete convergence).

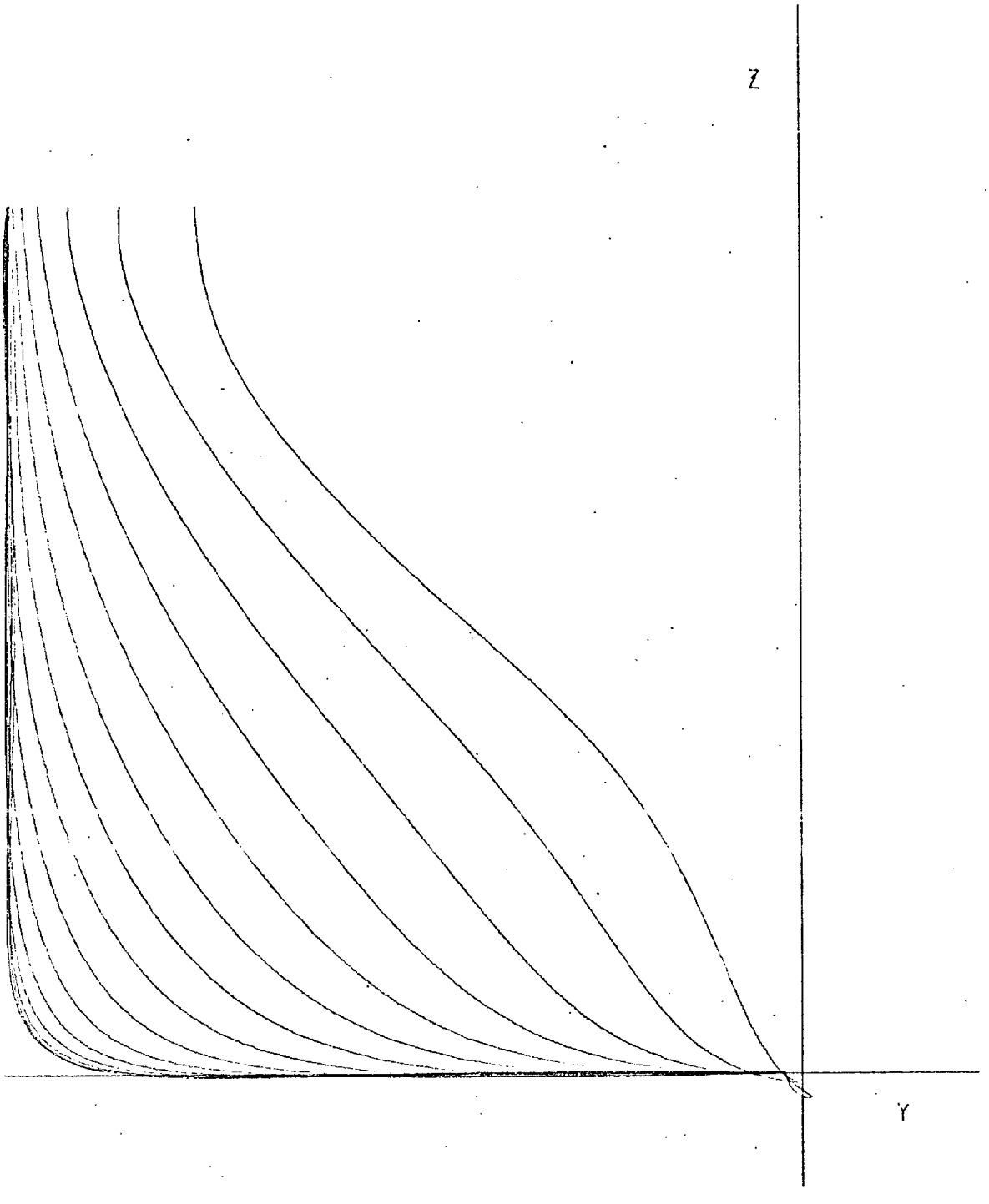


Figure 7.12:- Cross-sectional curves, at 1/4 station intervals, of a parametric surface fit to the Afterbody of Model 2. The surface is an unweighted quartic fit to the offsets from Station 1/4 to Station 4, from the Keel to Waterline 'K', with three points defining the Flat-of-Bottom at each section. The maximum deviation is 0.92°/o, at Station 1/4, Waterline 'F', after 80 iterations. The oscillations about the origin are unusually large, and are probably due to the incomplete convergence.

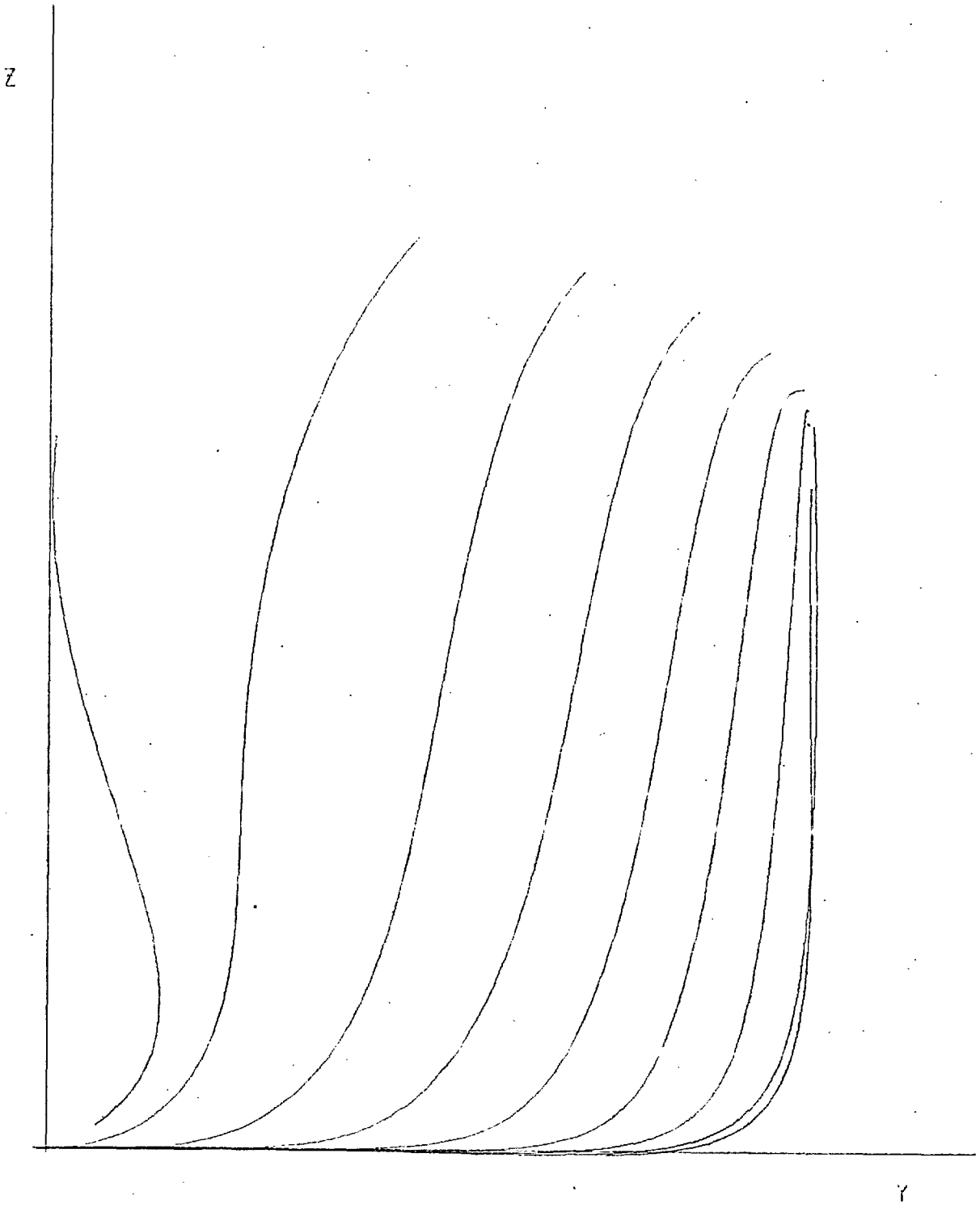


Figure 7.13:- Cross-sectional curves, at 1/4 station intervals, of a parametric surface fit to the Forebody of Model 2. The surface is an unweighted quartic fit to the offsets from Station 8 to Station 10, from the Keel to the knuckle, with three points defining the Flat-of-Bottom at each section. The maximum deviation is 0.58°/o, at Station 10, Waterline 'C', after 46 iterations. Despite a good fit numerically, the surface is practically unacceptable. It was, however, still improving substantially from one iteration to the next.

7.4.3 Effects of weighting

As expected, neither discrete weighting nor the alteration of the direction of the error vector affected the performance of the algorithm.

7.4.4 Effects of the arrangement of data on the topological grid

It has been found that when there was a close correspondence between the physical boundaries and the topological boundaries of a data set, e.g. x and y given on an approximately rectangular grid, reasonable changes in the arrangement on the grid of the 'internal' data points usually had little visible effect on the results, despite some numerical effect.

For data with missing boundary points, the 'boundary filling' procedure - Section 7.3.4 - has been found adequate. However, better results, in terms of visual quality of the fitted surface, could nearly always be obtained by filling the boundaries manually, using 'internal' points, in an intuitive manner; that is to say, in such a way that points lying on a grid line form as smooth a physical space curve as possible. For patches with truly awkward boundary shapes, several tries have usually been found necessary, for aesthetically pleasing results.

7.4.5 Number of iterations and processing time for convergence

As in the PCF method, the number of iterations required for convergence has been found unpredictable. In some cases, both initial and final fit were bad, resulting in extremely slow convergence. In other cases, practically complete convergence was attained within 3 to 4 iterations.

Further, as in the PCF method, the major consumer of computing time is the local minimisation procedure. The speed of the procedure

was little affected by the precise values of the parameters θ (the descent angle parameter) and δ (the 'thickness' of the constraints) but, on the other hand, was practically governed by the value of γ (the highest acceptable value of the gradient or its projection). The optimum value of γ is very machine and problem dependent. On the IBM-7094 Mk. II , the lowest value that can be used with impunity has been found to be 10^{-3}. With this value, the processing time has been found to be approximately 1/4 sec./point with bicubic polynomials, and to rise roughly as the square of the degree of polynomials used, as expected.

The effect of the number of points on the computing time cannot be so readily assessed. It is a combination of several factors, such as the size of the grid and the proportion and positions of the null entries. Very roughly, the effect is of an order between 1 and 2.

7.4.6 Oscillations between data points

With conventional least squares fitting methods, it is often the case that a numerically good fit to the data points produces unwanted oscillations between the the points, especially when relatively high degrees of polynomials are used. This imposes a severe limitation on the quality of fit attainable with such methods.

PSF appears far less prone to this problem. A probable reason is that relatively low order polynomials can be used to describe complex surfaces. When oscillations did occur, the data were frequently found to contain quite large errors, which the procedure attempted to fit too closely.

7.4.7 The formation of clusters

An important condition for the reliable convergence of this algorithm is independence between the parameter values that corres-

pond to different data points, which is manifested by the optimised parameter values being away from the constraints. As anticipated, this was so for most of the data tried. Occasionally, however, cases were found, where groups of neighbouring parameter values tended to coalesce. This problem had also been encountered with the PCF procedure, and a possible solution is discussed in Section 6.5. The solution is difficult enough with PCF, and considerably more so with PSF. It is therefore very encouraging to note that whenever this problem occured, it was later found to have been due to bad initial topological arrangement of the data.

7.5 A Possible Improvement

The operation of the algorithm, as described, is quite slow. It has been used simply because readily available NLP methods require the storage of a $(n \times m)$ matrix for a problem in n variables and m constraints, which, for this problem, tends to be larger than present computer storage sizes. In any case, the constraint descriptor matrix for this problem need only be approximately of size $(12\ n \times 3)$.

In recent years, there have been considerable advances in the field of non-linear optimisation, both constrained and unconstrained (a good survey may be found in Sargent and Sebastian, 1971, which is concerned primarily with unconstrained methods; no such collective treatment of constrained minimisation methods is available, but the above paper includes some useful references). Therefore, it should now be possible to construct a much more efficient algorithm for PSF, possessing formal convergence properties and faster convergence, and yet making use of the reduced storage requirements of this problem.

7.6 Discussion and Conclusions

PSF is a flexible and versatile surface-fairing method. Although

somewhat lacking in theoretical proofs of converegence, results obtained for a wide range of data indicate that the method is both valid and practical.

There is still no doubt that the hull-surface must be divided into several patches, which must be faired individually. However, by being able to describe compound surfaces successfully, the parametric surface definition eliminates many of the restrictions imposed on the patch-boundaries by other methods. In other words, the naval architect can define the boundaries at points which appear natural to him, or which are imposed by practical considerations, rather than ones which are convenient mathematically.

Further, PSF can also be used to convert a previously designed hull-form to the approximately equivalent Coons-patch representation (approximately, because some deviation may ensue from the fitting procedure. The resulting surface can then be moulded into a new, but related, shape (MacCallum, 1970), in several patches if necessary.

CHAPTER 8 :-

CONCLUSIONS

8.1 Summary

In this thesis, experiments with some unconventional lines' fairing methods have been described. It has been shown that many decisions relevant to the mathematical definition of hull forms can be quantified, and hence made by computer. However, it has been found that the most critical decision, namely the division of the hull into patches, is best made by the designer, in an intuitive manner.

Further, it has been shown that instead of attempting to replace the designer's intuition by uncertain methods, it is possible, and practical, to use definitions of curves and surfaces that are insensitive to variations in the patch boundaries. Three such methods have been treated, two of which have proved very successful - the parametric curve and surface definitions. It is not suggested that the fairing of a practical hull form can be made into a 'single shot' affair, even with the parametric definitions, but the designer is allowed considerable freedom to experiment, whilst still obtaining 'fair' results. In this way, this work provides the designer with a new and powerful tool.

8.2 Future Work and Recommendations

8.2.1 A practical system

Despite the strong practical slant of most of this work, even the successful methods, i.e. the parametric curve and surface definition, are of limited usefulness as they stand, particularly the latter. For their use in an integrated ship design system, a closer examination of multi-patch fairing is needed; also, possibly, more efficient methods of computing the parametric definitions.

8.2.2 Interactive Graphics

It has been shown in this thesis that it is very difficult, if not

impossible, to devise a completely automatic lines' fairing system. Moreover, for many aspects which can be automated, the results do not justify the effort involved, i.e. it is often far simpler to let the designer make a decision. Considering also the delays and frustration caused by computer batch-processing, the conclusion is inevitable - some convenient form of man-machine interaction is needed for practical computer-aided ship-design. While Teletype terminals are very useful, there is nothing like a Visual Display Unit (VDU) for conveying certain information simply, quickly, accurately and quietly, both ways.

It is beyond the scope of this work to discuss Interactive Graphics in any detail, but its application to the preliminary design of ships has been treated by MacCallum (1970). At present, it seems unlikely that the VDU will replace plotters for final assessment of hull-forms, because of its limited size and precision. On the other hand, little imagination is needed to visualise the power of a ship-design system which employs efficient lines' fairing programs, and which can be 'driven' via a VDU with a light-pen. Consider, for instance, the simplicity of:

a) Removing coarse errors from input data, by observing the display of offsets and waterlines,

b) Correcting less obvious errors by checking the display of the first and second finite-differences of the data,

c) Splitting the hull into patches, by direct indication on the screen (assuming implicitly that the feature of selective scaling of portions of the display is available),

d) Choosing a particular fairing method from a displayed selection, and so forth. The background computation must, of course, be performed fairly rapidly, although reasonable delays are tolerable. Hence the emphasis on the efficiency of the fairing programs.

At the present state of computers and computing-science, there seems

to be vast scope for an integrated hull-design system, using a VDU linked to a host computer, almost certainly a time-sharing one, and possibly an on-line plotter. The entire design procedure can then be a smooth, closed-loop, operation, from the preliminary stages up to the final design.

8.2.3 A different 'goodness-of-fit' criterion

The goodness-of-fit criterion used in this work is that of least-squares, although in an unusual form. A better criterion, or norm, would be the 'minimax' norm, i.e. the function is computed, which minimises the maximum deviation ('deviation' is here and henceforth used in the commonly accepted sense, i.e. the difference in one of the co-ordinate directions, deemed the 'dependent') between the given data and the corresponding mathematical definition. The minimax norm is the most powerful and critical norm, by its definition, and the knowledge that the magnitude of the maximum deviation is below a given bound is very satisfactory. However, the numerical application of this criterion is difficult, because its mathematical expression in terms of any regular hull descriptor function is not differentiable with respect to the parameters of the function, even in the vicinity of the solution.

The minimax norm has been applied to lines' fairing by Berger (Berger et al., 1966). In that interesting experiment, the hull was described by a piecewise spline, enabling the problem to be formulated as one of Linear Programming. Its solution was usually costly in computer time, and difficulties were encountered with end-conditions. However, much has been discovered in the field of minimax approximation since Berger's attempt, and a renewed investigation should prove fruitful.

The only reason, apart from analytical simplicity, for using the least-squares criterion is that it penalises large deviations. It has

been found in this work that the magnitude of the deviations can be quite misleading as to the quality of the fit, and that excellent results could be obtained by redefining the concept of 'deviation' to mean the perpendicular distance from each data point to the mathematical definition. It would therefore seem worthwhile to investigate the application of the minimax norm to this perpendicular deviation, using well established functions, e.g. polynomials, for the basic hull description. This does not appear to have been considered either analytically or numerically.

8.2.4 Different hull description functions

Polynomials are very simple to calculate and to manipulate. Consequently, they have been predominant in the field of mathematical ship lofting. It is noted that practically whenever another function is used, e.g. a conic, to fit a stern ending, it is fitted exactly, i.e. without fairing. While the parametric hull definitions, which also employ polynomials, seem quite satisfactory for most applications, many, possibly better, hull definition functions remain unexplored.

In recent years, much research has been carried out in the field of rational approximations, using the minimax norm. A rational approximation can, by its nature, describe a singularity, and could be useful for describing curves with almost vertical, linear regions. At present, the treatment of rational approximations seems to have been restricted to curves. The extension to surfaces and its application to hull forms should prove interesting.

APPENDICES

APPENDIX A:- Body Plans of Two Typical Simple Ship Forms

The particular ship forms given in this appendix were selected because they have some of the properties that make lines' fairing difficult mathematically, e.g. a bulbous bow, and yet they are simple forms. The names Model 1 and Model 2 are arbitrary.

The data given here is incomplete. Only the details relevant to this work are given.

Table A1:-

Basic Dimensions of Model 1.

Type of hull:	Optimised Trawler Form
Length (overall)	216.46 ft
Length Between Perpendiculars	200.00 ft
Breadth	38.4667 ft
Draught	17.4889 ft
Rise-of-Floor	20 percent draught
Type of Bow:	straight

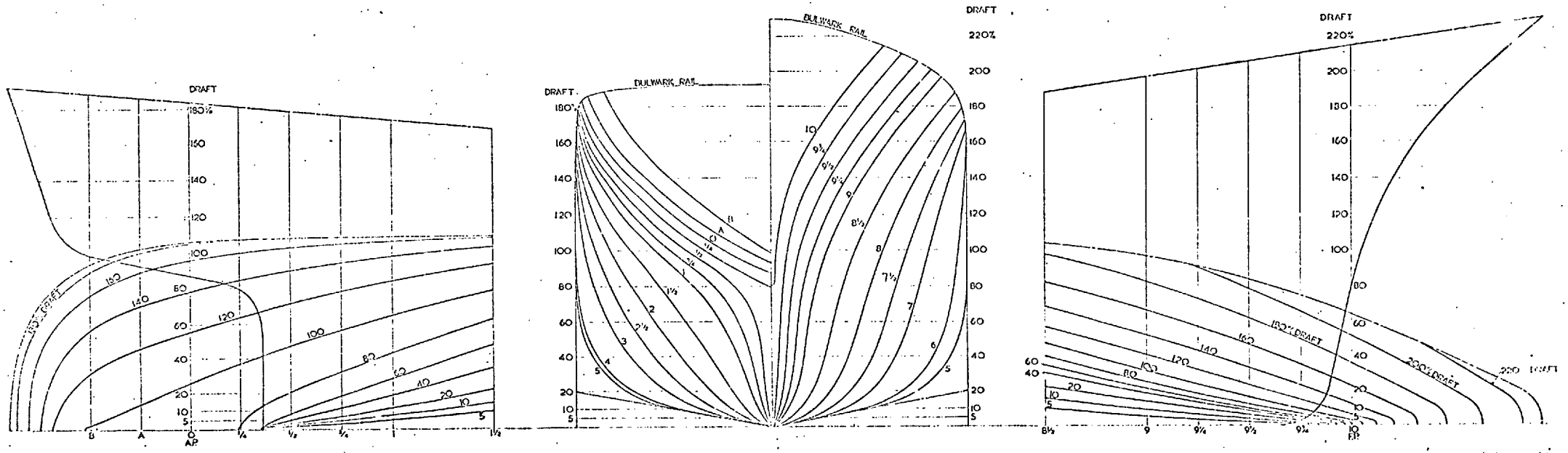


Figure A.1 :- Body-plan of Model 1.

Table A2:-

Offsets of Model 1

The offsets are expressed as percentage of maximum half-breadth

Station	Waterline heights (expressed as percentage of Load Draught)												
	0 (keel)	5	10	20	40	60	80	100	120	140	160	180	200
B	—	—	—	—	—	—	—	2.5	29.2	51.9	69.7	81.7	—
A	—	—	—	—	—	—	—	13.2	42.4	64.4	81.3	91.3	—
AP 0	—	—	—	—	—	—	—	24.3	50.7	71.4	87.3	95.8	—
1/4	—	—	—	—	—	—	1.3	34.3	58.4	77.5	90.7	98.1	—
1/2	0.7	1.4	2.0	3.3	4.9	8.2	19.1	43.4	65.5	82.1	93.6	—	—
3/4	0.7	2.6	4.2	6.6	10.8	17.4	28.6	51.7	71.3	85.9	95.0	—	—
1	0.7	4.3	7.0	10.8	17.5	26.4	39.8	59.6	76.9	89.3	96.8	—	—
1 1/2	0.7	8.9	13.7	20.4	31.8	43.8	57.5	72.0	85.2	93.7	98.3	—	—
2	0.7	12.9	20.9	31.3	46.4	59.8	71.8	82.6	90.7	96.7	99.2	—	—
2 1/2	0.7	18.0	29.0	42.7	60.5	72.8	82.6	90.0	95.5	98.7	100.0	—	—
3	0.7	21.8	36.7	53.9	72.4	83.6	90.9	95.5	98.3	99.6	100.0	—	—
4	0.7	25.1	47.1	70.7	89.0	96.4	99.3	100.0	100.0	100.0	100.0	—	—
5	0.7	25.1	50.4	74.5	91.4	97.9	100.0	100.0	100.0	100.0	100.0	—	—
6	0.7	26.1	46.6	66.5	82.2	89.3	92.9	95.2	97.1	98.5	99.5	—	—
7	0.7	23.8	36.6	50.3	63.2	70.1	75.4	80.4	85.4	91.0	97.7	—	—
7 1/2	0.7	19.8	29.1	40.0	51.4	57.8	63.5	69.4	76.1	83.9	92.9	—	—
8	0.7	14.5	21.5	29.6	39.1	44.7	50.3	56.7	64.7	74.0	85.2	—	—
8 1/2	0.7	9.2	14.2	20.3	27.0	31.4	36.2	42.8	51.1	61.5	74.3	88.3	—
9	0.7	4.8	8.1	12.4	16.3	18.6	22.2	28.3	36.6	47.1	59.6	74.1	—
9 1/4	0.7	3.4	5.5	8.5	11.7	13.4	15.9	20.9	28.5	38.5	51.1	65.4	81.0
9 1/2	0.7	2.1	3.4	5.4	7.3	8.6	10.6	14.4	20.4	29.8	41.8	55.6	70.5
9 3/4	—	0.9	1.3	2.0	3.3	4.2	5.7	8.0	12.5	20.5	31.5	44.4	58.9
FP 10	—	—	—	—	—	—	—	2.4	5.0	10.8	19.9	32.1	46.1

Table A3:-

Bulwark Curve for Model 1

Station	Height (as percentage of Load Draught)	Offset (as percentage of half-breadth)
(Aft tip) -0.894	193.10	0
B (- 1/2)	189.10	85.36
A (- 1/4)	186.60	93.36
AP 0	183.90	96.81
1/4	181.30	98.18
1/2	178.60	98.63
3/4	178.70	98.72
1	172.60	99.18
1 1/2	168.30	99.27
2	167.20	99.45
2 1/2	166.10	100.00
3	166.10	100.00
4	166.10	100.00
5	166.10	100.00
6	166.10	100.00
7	166.10	99.27
7 1/2	172.00	98.63
8	179.10	97.18
8 1/2	188.00	93.81
9	197.10	87.00
9 1/4	201.60	81.72
9 1/2	206.20	75.00
9 3/4	210.70	66.72
FP 10	215.10	57.18
(Foreward 10.919 tip)	231.40	0

Table A4:-

Stem and Stern Profiles of Model 1

Waterline height (as percentage of Load Draught)	Stern Profile (as percentage of LBP relative to AP)	Stem Profile (as percentage of LBP relative to FP)
0 (keel)	3.57	- 3.56
5	3.57	- 2.07
10	3.57	- 1.60
20	3.57	- 1.15
40	3.57	- 0.77
60	3.49	- 0.39
80	2.27	0
100	- 5.24	0.45
120	- 6.89	1.12
140	- 7.46	2.00
160	- 8.02	3.11
180	- 8.57	4.49
200	-----	6.25
220	-----	8.13

Table A5:-

Flat-of-Bottom Boundary Curve of Model 1

The offsets are given as percentage of maximum half-breadth.

Station	Offset
1/2	0.7
3/4	0.7
1	0.7
1 1/2	0.7
2	0.7
2 1/2	10.1
3	16.8
4	40.5
5	54.7
6	40.6
7	21.3
7 1/2	14.2
8	6.5
8 1/2	0.7
9	0.7
9 1/4	0.7
9 1/2	0.7

Table A6:-

Additional data for Model 1

This table contains the lowest points of those cross-sectional curves which do not end on standard waterlines.

Station	Height (as percentage of Load Draught)
B	98.6
A	93.3
0	87.6
1/4	79.4
9 3/4	2.5

Table A7:-

Basic Dimensions of Model 2.

Type of hull:	bulk carrier
Length (overall)	434.0 ft
Length Between Perpendiculars	400.0 ft
Breadth	55.0 ft
Draught	22.0 ft
Rise-of-Floor	nil
Type of Bow:	bulbous

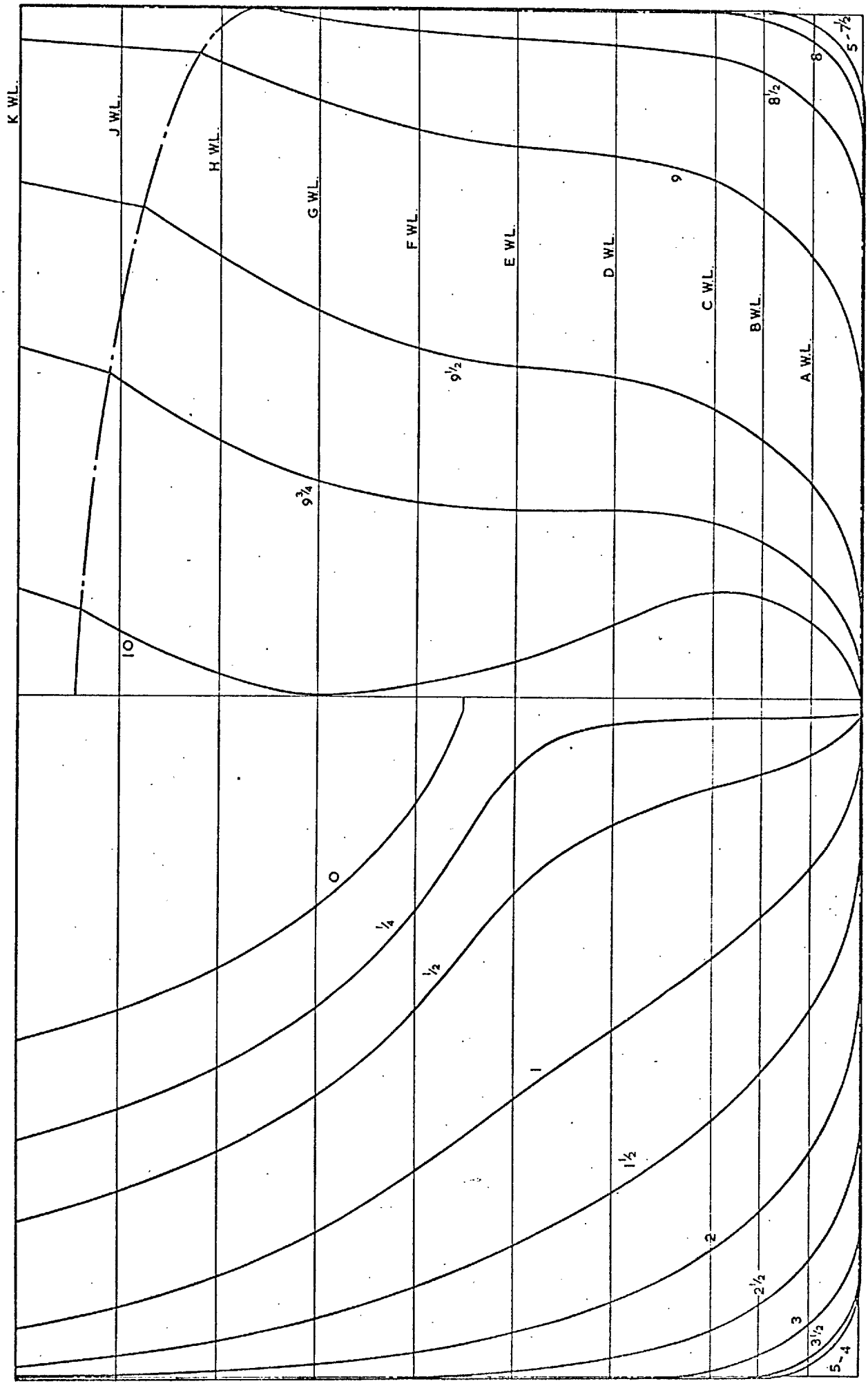


Figure A.2 :- Body-plan of Model 2

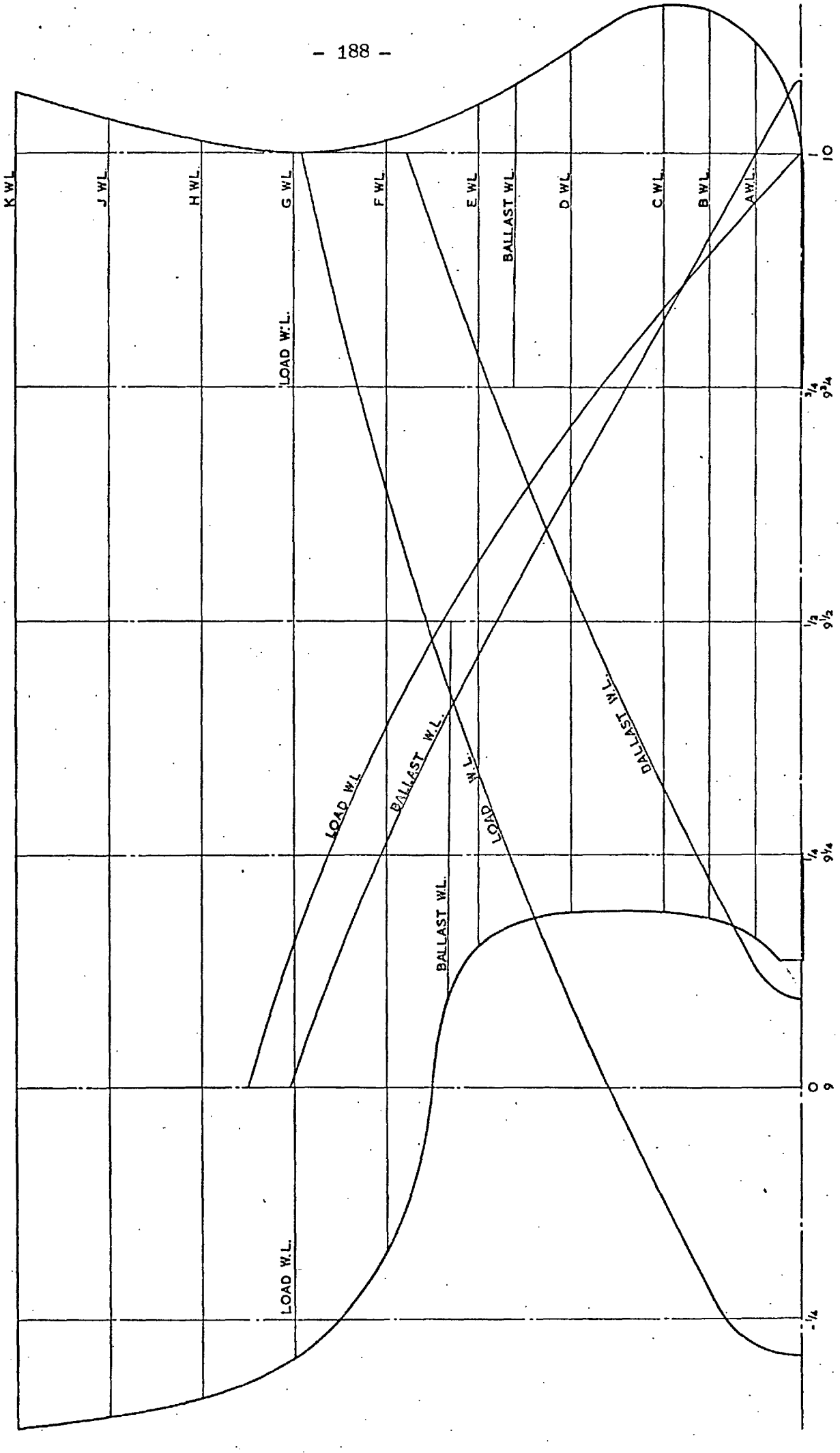


Figure A.3 :- Stem and Stern Profiles and Waterline Endings of Model 2.

Table A8 :-

Offsets of Model 2.

Waterline	Keel	A	B	C	D	E	F	G	H	J	K
Percentage of Load draught	0	9.09	18.18	27.27	45.45	63.64	81.82	100.00	118.18	136.36	154.55
Station	Waterline offsets expressed as a percentage of maximum half-breadth										
AP 0	——	——	——	——	——	——	15.7	30.9	40.1	46.2	50.5
1/4	2.2	2.7	2.8	3.0	4.1	10.7	31.4	45.9	54.7	60.8	65.2
1/2	2.2	8.3	11.0	13.2	19.0	28.9	46.2	58.7	66.9	72.7	77.1
1	0	24.8	32.0	38.3	48.8	59.2	69.7	78.6	85.4	89.7	92.6
1 1/2	0	45.6	55.0	62.1	73.0	80.7	86.7	91.3	94.7	96.8	98.3
2	0	66.6	75.6	81.3	88.9	93.7	96.6	98.0	99.0	99.5	99.8
2 1/2	0	82.6	89.4	93.6	97.3	99.2	99.8	100.0	100.0	100.0	100.0
3	0	92.1	96.8	99.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
3 1/2	0	96.7	99.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
4	0	98.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
5	0	98.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
6	0	98.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
6 1/2	0	98.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
7	0	98.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
7 1/2	0	98.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
8	0	95.5	98.8	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0
8 1/2	0	86.1	91.2	93.4	94.8	95.7	96.8	98.6	100.0	100.0	100.0
9	0	64.2	71.4	75.4	78.8	80.4	82.5	86.8	92.3	94.4	95.3
9 1/2	0	31.1	37.8	42.3	46.8	48.2	50.7	56.3	64.2	71.9	74.6
9 3/4	0	17.7	23.0	25.6	27.2	27.3	28.5	31.5	37.2	46.2	50.7
FP 10	0	11.3	14.6	15.2	10.5	4.9	1.6	0.3	3.2	9.9	15.7

Table A9:-

Knuckle Curve for Model 2

Station	Height (as percentage of Load Draught)	Offset (as percentage of half-breadth)
8 1/2	111.5	100.0
9	122.0	93.5
9 1/2	132.4	71.4
9 3/4	138.4	47.0
FP 10	143.4	12.4
(Stem) 10.919	144.3	0

Table A10:-

Stem and Stern Profiles of Model 2

Waterline (height as percentage of Load Draught		Stern Profile (as percentage of LBP relative to AP)	Stem Profile (as percentage of LBP relative to FP)
Keel	0	0.97 (extrapolated)	0
A	9.09	1.61	1.18
B	18.18	1.82	1.51
C	27.27	1.89	1.57
D	45.45	1.89	1.08
E	63.64	1.52	0.51
F	81.82	- 1.76	0.13
G	100.00	- 2.91	0
H	118.18	- 3.34	0.13
J	136.36	- 3.55	0.36
Knuckle (stem)	144.30	----	0.48
K	154.55	- 3.66	0.65

Table A11:-

Flat-of-Bottom Boundary Curve of Model 2

The offsets are given as percentage of maximum half-breadth.

Station	Offset
1/4	2.2
1/2	2.8
1	8.2
1 1/2	23.1
2	42.4
2 1/2	63.3
3	78.9
3 1/2 up to 7 1/2	86.0
8	82.5
8 1/2	71.4
9	44.3
9 1/2	12.7
9 3/4	4.2
FP 10	0

APPENDIX B:- The Calculation of the Distance from a Point in the xy-plane to a Curve $y = f(x)$

Given A point $P(x_p, y_p)$ in the xy-plane
A plane curve described by $y = f(x)$.

Find The shortest distance, δ, from P to the curve, i.e. the length of the normal dropped from P unto the curve, at the point $T(x_t, y_t)$, say.

There are two equivalent approaches to the solution. The first is to calculate the length of the normal, and the second is to find the general expression for the length PT and to minimise that. The latter is given, because it illlustrates the problem better.

Define the function d(x) by

$$d(x) = [x - x_p]^2 + [y - y_p]^2$$

$$= [x - x_p]^2 + [f(x) - y_p]^2 \qquad \ldots\ldots(B.1)$$

i.e. the square of the distance from P to an arbitrary point on the curve. The value of δ corresponds to a minimum of d(x). For a stationary point of d(x),

$$[x - x_p] + f'(x)\,[f(x) - y_p] = 0 \qquad \ldots\ldots(B.2)$$

where $f'(x)$ is the derivative of $f(x)$ with respect to x.

There is no general solution to equation B.2; it can only be solved analytically for particular functions. The function of interest here is a polynomial, generally of degree greater than 1, and the equation must be solved numerically.

Define the function g(x) by

$$g(x) = [x - x_p] + f'(x)\,[f(x) - y_p] \qquad \ldots\ldots(B.3)$$

where $g(x_t) = 0$.

Let the value of the ith approximation to x_t be given by x_i. Applying the Newton-Raphson method, the improved value of x, x_{i+1}, is related to the current approximation by

$$x_{i+1} = x_i + h_i$$

where $h_i = - g(x_i) / g'(x_i)$.

By differentiating equation B.3, expanding and simplifying, it is found that

$$h_i = \frac{[x_i - x_p] + f'(x_i)[f(x_i) - y_p]}{1 + f'^2(x_i) + f''(x_i)[f(x_i) - y_p]} \qquad \ldots\ldots(B.4)$$

The value of x is thus improved iteratively, until the computed correction, h, is less tham a prescribed value. It is noted that a maximum of d(x) is an equally valid solution, so that a check must be made that $d''(x_t) = g'(x_t) > 0$, for a minimum.

This method requires strict conditions for convergence, but if they exist, the convergence is rapid. Loosely speaking, the approximation to the root should be closer to the solution than any turning point and/or inflextion of g(x) (if it possesses any). The following procedure overcomes the obstacles:

a) Take the first approximation as $x_0 = x_p$, and compute an upper bound for d(x).

b) Compute a range, [a,b], which brackets the root, from several values of d(x) and g(x).

c) Iterate for the root using equation B.4, checking in each itoration that d(x) is decreasing, and that x remains inside [a,b]. If either condition is violated, the improved root is instead estimated by

$$x = a + \frac{(b - a)\,|g(a)|}{|g(a)| + |g(b)|}$$

d) If possible, a new, reduced interval, [c,d], is found, where $|d - c| \leq |b - a|$, and the process repeated from step (c).

Once the point T has been found, the value of δ can be calculated from equation B.1.

Despite the limitations outlined above, this method has been found to perform well. An example is included, for illustration.

Example

Let $f(x) = x^2 + x + 1$

$\therefore f'(x) = 2x + 1$

and $f''(x) = 2$

Take the point P(2,1) - uncommonly far from the curve. The exact solution is $x = 0.5$. The table below illustrates the rapid convergence.

ITERATION NO.	X	H	DELTA-
0	2.000000	-0.789474	6.000000
1	1.210526	-0.463292	2.789930
2	0.747234	-0.203792	1.809419
3	0.543443	-0.041786	1.680803
4	0.501657	-0.001654	1.677056
5	0.500003	-0.000003	1.677051
6	0.500000	0.000000	1.677051

APPENDIX C:- Approximation to a Subset of the Statistical F-Tables

Given tabulated values of $F_{1,\nu}$ against ν (e.g. in Pearson and Hartley, 1958), for the degrees of confidence 95.0, 97.5, 99.0 and 99.9 percent (in fact, the inverse distribution is normally tabulated, i.e. the 5.0, 2.5, 1.0 and 0.1 percent point, but this does not affect the following).

It is required to approximate each table, as closely as possible, by a simple function. By observing the trend of the tables, it seems reasonable to fit each by a conic function, of the form:

$$F_{1,\nu}^2 + a\nu^2 + b\nu + cF_{1,\nu} + d\nu F_{1,\nu} + e = 0$$

It is noted that although this observation is completely empirical and has no statistical justification, it is simple and convenient.

The coefficients of the conic can be calculated by least squares, as follows:

Let n be the number of entries in an F table, i.e. the entries are (ν_i, F_i), $1 \leq i \leq n$

Let the sum of squares of residuals be given by

$$R = \sum_{i=1}^{n} (F_i^2 + a\nu_i^2 + b\nu_i + cF_i + d\nu_i F_i + e)^2$$

By differentiating R with respect to each of the coefficients a, b, c, d and e in turn, and equating the derivatives to zero for a minimum of R, the following set of equations is obtained:

$$a:- \quad h_{40}a + h_{30}b + h_{21}c + h_{31}d + h_{20}e = -h_{22}$$

$$b:- \quad h_{30}a + h_{20}b + h_{11}c + h_{21}d + h_{10}e = -h_{12}$$

$$c:- \quad h_{21}a + h_{11}b + h_{02}c + h_{12}d + h_{01}e = -h_{03}$$

$$d:- \quad h_{31}a + h_{21}b + h_{12}c + h_{22}d + h_{11}e = -h_{13}$$

$$e:- \quad h_{20}a + h_{10}b + h_{01}c + h_{11}d + h_{00}e = -h_{02}$$

where $$h_{kl} = \sum_{i=1}^{n} \nu_i^k F_i^l$$

Unlike the normal equations for standard polynomial approximation, these equations are well-conditioned, and may be solved in the usual way.

Having found the values of the coefficients of the function, the value of $F_{1,\nu}$ can be found for any ν, by substituting ν in the conic equation, and solving the resultant quadratic. The wanted root is real and positive, and by solving for one known point, it is deduced which of the two roots of the quadratic is the wanted one.

Table C.1 contains the coefficients of the approximations to the standard four F-tables, and the maximum deviations of the approximations from the tabulated values.

The approximations have been found sufficiently accurate only for $2 \leq \nu \leq 500$, which is the normal range of degrees of freedom in practice. The extra precaution has been taken for the special cases of $\nu = 1$ and $\nu > 500$, by using the tabulated values for the former, and the asymptotic value of $F_{1,\infty}$ for the latter.

Table C.1 :- Approximations to F-tables

confidence point (percent)	n	a	b	c	d	e	maximum deviation (percent)
5.0	37	$1.9975450*10^{-3}$	-149.35007	-75.330845	38.818797	-82.082405	1.75
2.5	37	$6.2106713*10^{-3}$	-397.49358	-165.28960	79.043665	-401.52732	2.21
1.0	36	$7.8340514*10^{-3}$	-373.57679	-157.00656	56.183688	-422.62041	1.35
0.1	35	$2.5939629*10^{-2}$	-946.06202	-342.97817	87.202490	-2061.5735	1.98

APPENDIX D:- The Derivation of Spline Functions

Spline functions originate from the flexible wooden battens used by draughtsmen to draw smooth curves between data points. When used for such purpose, the beam is bent to the desired shape, and weights are placed on it at the data points. The usefulness of the mathematical equivalent of the beam, for curve fitting, came to light in recent years. Spline functions are treated thoroughly elsewhere (Asker, 1962; Theilheimer and Starkweather, 1961; Handscomb , 1965; Powell, 1967) and, therefore, only a brief description and derivation, necessary for this work, are given here.

It is possible to derive the functions without reference to the beam, but it is interesting to consider two elementary physical properties of thin beams. Assuming that a thin beam of uniform flexual rigidity, R, is subjected to a bending moment M, and lies along the plane curve $y = \Phi(x)$, the following expressions ensue from the theory of stress and strain (Timoshenko, 1951) :-

a) The curvature of the beam is given by

$$\frac{\Phi''(x)}{[1 + \Phi'(x)^2]^{3/2}} = \frac{M}{R} \qquad \ldots\ldots (D.1)$$

b) The potential energy of the beam is proportional to the integral along its length of the square of the curvature, i.e. proportional to

$$\int \Phi''(x)^2 [1 + \Phi'(x)^2]^{-5/2} dx \qquad \ldots\ldots (D.2)$$

and the shape it takes up is such as to minimise this energy under whatever constraints are in force, namely the weights at the data points.

In deriving spline functions as an interpolation tool, it is

assumed that the slope, $\Phi'(x)$, is small enough to be omitted from expression (D.1), The bending moment applied to the beam is transmitted at the data points; with a uniformly flexible spline, equation (D.1) implies second derivative continuity at the points.

Given data points (x_i, y_i), $1 \leq i \leq n$. Consider one segment of the spline, lying between (x_i, y_i) and (x_{i+1}, y_{i+1}).

Let $d_i = x_{i+1} - x_i$

and $\Phi''_i = \Phi''(x_i)$; $\Phi''_{i+1} = \Phi''(x_{i+1})$.

i.e. the second derivative of the spline, not of the function that is being interpolated. Assume, further, a constant third derivative throughout the segment; this assumption leads to the lowest order practical spline functions. Then, for $x \in [x_i, x_{i+1}]$,

$$\Phi''(x) = \frac{\Phi''_i(x - x_i) + \Phi''_{i+1}(x_{i+1} - x)}{d_i} \qquad \ldots\ldots (D.3)$$

Integrating twice with respect to x and rearranging,

$$\Phi(x) = \frac{\Phi''_i(x - x_i)^3 + \Phi''_{i+1}(x_{i+1} - x)^3}{d_i} + A(x - x_i) + B(x_{i+1} - x) \qquad \ldots\ldots (D.4)$$

where A and B are constants of integration.

By substituting the known values y_i and y_{i+1} for $\Phi(x)$ in equation (D.4), A and B are found to be

$$A = \frac{y_{i+1}}{d_i} - \frac{1}{6} d_i \Phi''_i \qquad \ldots\ldots (D.5)$$

and $$B = \frac{y_i}{d_i} - \frac{1}{6} d_i \Phi''_{i+1} \qquad \ldots\ldots (D.6)$$

It remains to compute the values of Φ''_i, $1 \leq i \leq n$. The derivative of the cubic in the segment is given by

$$\Phi'(x) = \frac{\Phi''_i (x_{i+1} - x)^2 - \Phi''_{i+1} (x - x_i)^2 + 2 (y_{i+1} - y_i)}{2 d_i}$$

$$+ \frac{1}{6} d_i (\Phi''_{i+1} - \Phi''_i) \qquad \ldots\ldots (D.7)$$

Substituting $x = x_i$ in equation (D.7) and simplifying,

$$\Phi'(x_i) = \frac{y_{i+1} - y_i}{d_i} - \frac{1}{6} d_i (\Phi''_{i+1} + 2 \Phi''_i) \qquad \ldots\ldots (D.8)$$

For continuity of first derivative at the data points, this value must equal the one obtained from the $(i - 1)$th segment, i.e.

$$\Phi'(x_i) = \frac{y_i - y_{i-1}}{d_{i-1}} + \frac{1}{6} d_{i-1} (\Phi''_i + 2 \Phi''_{i-1}) \qquad \ldots\ldots (D.9)$$

Equating the expressions (D.8) and (D.9) and simplifying,

$$d_{i-1} \Phi''_{i-1} + 2 (d_{i-1} + d_i) \Phi''_i + d_i \Phi''_{i+1} =$$

$$6 \left[\frac{y_{i+1} - y_i}{d_i} - \frac{y_i - y_{i-1}}{d_{i-1}} \right] \qquad \ldots\ldots (D.10)$$

Equation (D.10) holds for $i = 2, 3, \ldots, n-1$. Hence $(n - 2)$ equations can be generated, that relate the values of Φ''. The remaining two degrees of freedom can be resolved either by imposing two more constraints on the fit, or by using the thin beam analogy. Take the latter - assume that for $x \notin [x_1, x_n]$, the beam is linear, i.e. $\Phi''_1 = \Phi''_n = 0$.

Then the values of $\Phi''_2, \ldots, \Phi''_{n-1}$ may be computed by solving the linear equations

$$S\,\Phi'' = \bar{z},$$

where S is the well conditioned tri-diagonal symmetric matrix

$$S = \begin{vmatrix} 2(d_1 + d_2) & d_2 & & & \\ d_2 & 2(d_2 + d_3) & d_3 & & \\ & \cdots & \cdots & \cdots & \\ & & \cdots & \cdots & \\ & & & & 2(d_{n-2} + d_{n-1}) \end{vmatrix}$$

and

$$z_i = 6\left[\frac{y_{i+1} - y_i}{d_i} - \frac{y_i - y_{i-1}}{d_{i-1}}\right]$$

Having found the values of Φ''_i, the coefficient matrix of the spline can easily be calculated from equations (D.4), (D.5) and (D.6).

APPENDIX E:- Properties of the 'Flatness' Function of Polar Transformation.

Given :- Data points (y_i, z_i), $1 \leq i \leq n$,
A point $P(y_o, z_o)$ in the yz-plane.

Let $r_i = \sqrt{(y_i - y_o)^2 + (z_i - z_o)^2}$

$r_{min} = \min [r_i]$

$r_{max} = \max [r_i] \qquad 1 \leq i \leq n$

Let Q and R be the extreme points of the data, i.e. the two points between which the distance is greater than between any other pair (Fig. 5.8). Assume, to start with, that there is only one such pair.

Define the Flatness function Φ as

$$\Phi = r_{max} - r_{min}$$

The following properties of Φ may be deduced by elementary considerations:-

a) It is clearly non-analytic.

b) It is non-negative and finite everywhere.

c) It is continuous with, generally, discontinuous first and higher derivatives.

d) As $|y_o|$ and/or $|z_o| \longrightarrow \infty$, $\Phi \longrightarrow 0$.

e) Φ has a maximal ridge, of value 'QR', along the line generated by QR, but only outside the range QR. When there are more than one pair like QR, there is a ridge corresponding to each pair. When the ridges form a closed figure, Φ has a minimum inside the figure.

f) The most interesting property of Φ is the existence of a minimum for finite values of y_o and z_o. The minimum lies in the same half-

space of the curve ('curve' being defined as the figure created by joining all the data points, in their given order, by straight line segments, and taking the end segments to infinity) as the centroid of the data points, G. It has been found empirically that the minimum lies approximately on the line MGN, where GN is perpendicular to QR.

g) Beyond the minimum, in the same direction and moving away from the curve, Φ has a maximum. Further still it decreases, to finally blend in with its behaviour in the other half-space of the curve, which is a monotonic decrease.

APPENDIX F:- Univariate Minimisation by Quadratic Interpolation, without Gradients.

Given: Three values of a function, f_a, f_b and f_c, at points a, b and c. For convenience, and without loss of generality, let

$a < b < c$.

It is required to compute the point d, at which the parabolic approximation to the function (not necessarily the function itself) has a minimum.

Let the function be approximated by the three point Lagrangian formula

$$f(x) = \frac{(x-b)(x-c)}{(a-b)(a-c)} f_a + \frac{(x-a)(x-c)}{(b-a)(b-c)} f_b + \frac{(x-a)(x-b)}{(c-a)(c-b)} f_c$$

By differentiating with respect to x and equating to zero, it can be shown that the turning point of the parabola is at

$$d = \frac{1}{2} \frac{(b^2 - c^2) f_a + (c^2 - a^2) f_b + (a^2 - b^2) f_c}{(b - c) f_a + (c - a) f_b + (a - b) f_c}$$

It is a minimum if the second derivative of the approximation is negative at d, i.e. if

$$\frac{(b-c) f_a + (c-a) f_b + (a-b) f_c}{(a-b)(b-c)(c-a)} < 0$$

There are three basic possible outcomes, as follows

a) A minimum predicted at $d \in [a,c]$, and $f(d) \leq \min [f_a, f_b, f_c]$. This is the ideal situation. The position of the minimum can be refined by discarding the point furthest from d, substituting d for it and repeating the computation, until d is within some prescribed tolerance of one of the points.

b) A minimum predicted at $d \notin [a,c]$, and $f(d) \leq \min [f_a, f_b, f_c]$. The precise action in this case depends on the configuration of the initial three points. With reasonably well behaved functions, the point furthest from d may be discarded, and the computation repeated until the minimum is bracketed, as in (a). The last is necessary, because only in that situation can a true minimum of the function itself be guaranteed.

c) A maximum predicted, or a minimum outside some prescribed limits on the value of d. As in (b), the precise action depends on the initial points, but, generally, the search step is reduced, and a local search in [a,c] is carried out.

REFERENCES

APT Part Programming	IIT Research Institute Chicago McGraw-Hill 1967.
Asker, B.	"The Spline Curve, a Smooth Interpolating Function used in the Numerical Design of Ship Lines", Nordisk Tidsskrift for Informationsbehandling (BIT), Vol. 2, pp.76-82, 1962.
Beckman, F.S.	"The Solution of Linear Equations by the Conjugate Gradients Method", in "Mathematical Methods for Digital Computers", Ralston, A. and Wilf, H.S. (Eds.), Wiley, 1960.
Benbow, J. and Whitehead, P.	"A Method of Feasible Directions in Non-linear Programming", Ph.D. Thesis (in preparation) at the University of London (Imperial College), (1968).
Berger, S.A. Webster, W.C., Tapia, R.A. and Atkins, D.A.	"Mathematical Ship Lofting", Journal of Ship Research, 1966, December.
Berztiss, A.T.	"Least-squares Fitting of Polynomials to Irregularly Spaced Data", SIAM Review, Vol. 6, pp. 203-227, 1964.
Box, M.J.	"A Comparison of Several Current Optimisation Methods, and the Use of Transformations in Constrained Problems", Computer Journal, Vol. 8, 1965.
Cadwell, J.H. and Williams, D.E.	"Some Orthogonal Methods of Curve and Surface Fitting", Computer Journal, Vol. 4, pp. 260-264, 1961.
Cheney, E.W.	"Introduction to Approximation Theory", McGraw-Hill Book Company (New York), 1966.
Clenshaw, C.W.	"Curve-Fitting with a Digital Computer", Computer Journal, Vol. 2, pp. 170-172, 1960.
Clenshaw, C.W. and Hayes, J.G.	"Curve and Surface Fitting", J. of Inst. of Maths. Applics., Vol. 1, 1965.

Coons, S.A. "Surfaces for Computer Aided Design of Space Figures", M.I.T. MAC-TR-41 1967, June.

Davidon, W.C. "Variable Metric Method for Minimisation", A.E.C. Research and Development Report ANL-5990 (Rev.), 1959.

Eisenhart, L.P. "A Treatise on the Differential Geometry of Curves and Surfaces", Dover Publications, 1960.

Fletcher, R. and Reeves, C.M. "Function Minimisation by Conjugate Gradients Computer Journal, Vol. 7, pp. 149-153, 1964.

FORAN "FORAN System of Computer-aided Ship Design", Proc. of FORAN Seminar, The British Oxygen Company, 1969, October.

Forsythe, G.E. "Generation and Use of Orthogonal Polynomials for Data Fitting with a Digital Computer, Journal of SIAM, Vol. 5, pp. 74-88, 1957, June.

Goldfarb, D. "Extension of Davidon's Variable Metric Method to Minimisation under Linear Inequality and Equality Constraints", SIAM Journal of Appl. Maths., Vol. 17, pp. 739-763, 1969, July.

Grossman, M. "An Experimental Lines' Fairing Package" Computer Technology Group Report 67/10, Department of Computing and Control, Imperial College of Science and Technology 1967, October.

Handscomb, D.C. (Ed.) "Methods of Numerical Approximation", Symposium Publications Division, Pergammon Press, 1966.

Hayes, D.G. "A Method for Storing the Orthogonal Polynomials used for Curve and Surface Fitting", Computer Journal, Vol. 12, pp. 148-150, 1969, May.

Hayes, J.G. "The Mathematical Fairing of Ships' Lines", NPL Seminar on 'Computers in Ship Research and Design', 1962.

Hestenes, M.R. and Stiefel, E. "Methods of Conjugate Gradients for Solving Linear Systems", J. Res. N.B.S., Vol. 49, p. 409, 1952.

Horsham, W. "Ship Hull Form Design using Remote Time-Sharing Computers", M.Sc. Thesis at the University of Strathclyde, 1970, July.

Householder, A.S. "Principles of Numerical Analysis", McGraw-Hill (New-York), 1953.

Hudson, D.J. "Statistics Lecture Notes", (two vols.) CERN, Geneva, Report Nos. 63-29 and 64-18, 1963-1964.

Kantorowitz, E. "Mathematical Definition of Ship Surfaces" Danish Ship Research Institute, Report No. DSF-14, 1967, April.

Kerwin, J.E. "Polynomial Surface Representation of Arbitrary Ship Forms", Journal of Ship Research, Vol. 4, No. 1, 1960, March.

Kuo, C. "Computer Methods for Ship Surface Design" Longmans, 1971.

MacCallum, K.J. "The Application of Computer Graphics to the Preliminary Design of Ship Hulls", Ph.D. Thesis, Faculty of Engineering, University of London, 1970, March.

Miller, N.S. and Kuo, C. "The Mathematical Fairing of Ships' Lines" European Shipbuilding, Vol. 12, pp 72-85, 1963.

Pearson, E. and Hartley, H. "Biometrica Tables for Statisticians", Cambridge University Press, 1958.

Pien, P.C. "Mathematical Ship Form", David Taylor Model Basin, Report No. 1398, 1960, January.

Pitidis-Poutous, T. "Mathematical Design of Ship Lines", National Engineering Council, 1964.

Powell, M.J.D. "An efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives" Computer Journal, Vol. 7, pp. 155-162, 1965.

Powell, M.J.D. "Curve Fitting by Cubic Splines", A.E.R.E. Report No. TP-307, 1967, September.

O'Neill, M., Sinclair, I.G. and Smith, F.J. "Polynomial Curve Fitting when Abscissæ and Ordinates are both Subject to Error", Computer Journal, Vol. 12, pp. 52-56, 1969.

Rosen, J.B. "The Gradient Projection Method for Non-linear Programming. Part 1 - Linear Constraints", J. SIAM, Vol. 8, pp. 181-217, 1960, March.

Rosenbrock, H.H. "An Automatic Method for Finding the Greatest or Least Value of a Function", Computer Journal, Vol. 3, pp. 175-184, 1960.

Sargent, R.W.H. and Sebastian, G. "Numerical Experience with Algorithms for Unconstrained Minimisation", Symposium on 'Numerical Methods for Non-Linear Optimisation', University of Dundee, 1971, June.

Saunders, H.F. "Hydrodynamics in Ship Design", S.N.A.M.E., New York, 1957.

Theilheimer, F. and Starkweather, W. "The Fairing of Ship Lines on a High-Speed Computer", Mathematics of Computation, 1961, October.

Timoshenko, S.P. "Theory of Elasticity", McGraw-Hill (New-York), 1951.

Watson, D.G.M. "Estimating Preliminary Dimensions in Ship Design", Trans. I.E.S.S., Vol. 105, 1962.

Weisfeld, M. "Orthogonal Polynomials in Several Variables", Numerische Mathematik, Vol. 1, pp. 38-40, 1959.

BIBLIOGRAPHY

Atkins, D.A. and Tapia, R. — "Mathematical Ship Lofting", California Section of S.N.A.M.E., 1962, October.

Bakker, A.R. — "Application of a Computer to Some Shipbuilding Problems", Institute of Shipbuilding Progress, Vol. 12, pp. 245-264, 1965, June.

Benson, F.W. — "Mathematical Ship Lines", I.N.A., Vol. 12, pp. 129-150, 1940.

Berghuis, J. — "Experience with Mathematical Shipform", N.P.L. Seminar on Ship Research and Design, Paper 42, 1962.

Chapman, F.H. — "A Treatise on Ship-Building", Cambridge ans London, 1760 (English translation 1820)

Di Bella, A. — "Equations of Ships' Hulls", Genoua University Report, 1948.

Engvall, L.O. and Jacobsson, A. — "Mathematisk Formbestamning" (Swedish) Institution for Skeppshydromekanik, (C.T.H.) Report No. 33, 1965.

Gerritsma, J., Kerwin, J.E. and Newman, J.N. — "Polynomial Representation and Damping of Series 60 Hull Forms" International Shipbuilding Progress, Vol. 95, 1962, July.

Gohin, F., Perrot, P. and others — "Numerical Fairing and Representation of Ship Hulls", Ass. Tech. Mar., 1966.

Gospodnetic, D. — "Numerical Definition of Ships' Hulls by Means of Elastic Interpolation", National Research Council of Canada, Report No. MB-258, 1965, January.

Guida, A. — "Application of a Computer to Find the Hull Equation of a Ship", (Italian) Proc. Associazione Di Technica Navale, Vol. 15, pp. 25-47, 1961-1962. English Abstract in: Journal of the B.S.R.A., Vol. 19, No. 20.923.

Hamilton, M.A. and Weiss, A.P. "An Approach to Computer Aided Preliminary Design", M.I.T. M.Sc. Thesis No. 21/8-64, 1964.

Hammil, P.A. "A Simple Approach to the Mathematical Ship Form", National Research Laboratory of Canada, Mechanical Engineering Division, Ship Laboratory, 1960.

Harloff, A.F. "A Method for Describing Ship Lines Mathematically", N.P.L. Seminar on Computers in Ship Research and Design, 1962.

Jacobsson, A. "Die Verwendung von Rechenautomaten fuer Schiffstechnische Probleme in Schweden", Hansa, Vol. 102, pp. 1201, 1965.

Jinnika "On Ship Forms and Wave Making Resistance" (Japanese) Journal of Zosen Kiokai, Vol. 14, Part 1, 1961, February.

Lidbro, N. "Analytische Formbestimmung von Schiffen", Schiffstechnik, Vol. 8, pp. 91-96, 1961.

Rosingh, W.H.C.C.E. and Berghuis, J. "Mathematical Design of Hull Lines", International Shipbuilding Progress, Vol. 6, 1959, January.

Taggert, R. "Mathematical Fairing of Ship Lines for Mould Loft Layout", Journal of A.S.N.E., Vol. 67, pp. 337-357 1955, May.

Taggert, R. and "Mathematical Derivation of Destroyer Type Hulls", Naval Engineers (A.S.N.E.), Vol. 79, pp. 19-28, 1967, February.

Taylor, D.W. "On Ship Forms Derived by Formulae", S.N.A.M.E., Vol. 11, pp. 243-267, 1903.

Taylor, F. "Computer Application to Shipbuilding", R.I.N.A., 1962, April.

Thieme, H. "Systematische Entwicklung von Schiffslinien", Schiff und Hafen, Vol. 4, pp. 241-245, 1952.

Thieme, H. "Ueber Grundlagen fuer den Mathematischen Linienriss Eines Frachtschiffs", Schiffstechnik, Vol. 3, pp. 288-299, 1956, October.

Warrington, N.M. "A System of Mathematical Lines for Ships", S.N.A.M.E., Vol. 17, pp. 441-452, 1909.

Wattanab, K. "Mathematical Expression of Ship Forms", Journal of Zosen Kiokai, Vol 77, 1945, July.

Weinblum, G.B. "Systematische Entwicklung von Schiffsformen", Jahrbuch der Schiffsbautechnische Gesellschaft, Vol.47, pp. 186-215, 1953.

Williams, A. "Mathematical Representation of Ordinary Ship Forms", Schiff und Hafen, Vol. 16, pp. 414-433, 1964.