University of Glasgow

Department of Naval Architecture and Ocean Engineering

Computer-Aided Development of Shell Plates

An M.Sc. Thesis by

(c)António José dos Santos Cruz Cacho, 1998

ProQuest Number: 13815580

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 13815580

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

GLASGOW UNIVERSITY
LIBRARY
M 290 (copy 1)

Abstract

Ship hulls and other curved shells, like gas tanks, aircraft bodies, and even clothes and shoes, offer a common difficulty in their manufacturing: it is necessary to produce them from a set of formerly plane elements. These plane elements, the raw materials like plates and fabric pieces, must be curved and assembled together to form the final product.

The reverse of the forming process of these curved elements, is the map of the curved surface onto the plane, which is improperly known as development. To develop a surface, in a proper sense, is to unfold it onto the plane without stretching or bulging. This is not possible with all kinds of shapes, such as spherical and saddle surfaces. Some common developable surfaces are the conical and cylindrical ones.

To form a non-developable shell requires much more work than to form an equivalent shell of developable shape. This increases the costs, the processing times and the defect content. Nevertheless, the fluid dynamists and the other designers are not always free to use developable shapes in their concepts; therefore, a pragmatic approach to the construction of curved shells has to cope with non-developable surfaces.

These subjects are chiefly of an advanced mathematic nature, and the required background is too widely spread in the bibliography. Therefore the necessary mathematical results are compiled and presented in CHAPTER 2 - THE MATHEMATICS OF DEVELOPABLE SURFACES, providing for a unified view of the concepts, the symbols and the nomenclature.

Since the advent of the digital computer, the increasing availability of computing power enabled new methods for surface development and for developable surface definition. By examining and comparing the methods reported in the literature, CHAPTER 3 - PLATE DEVELOPMENT AND

DEVELOPABLE SURFACES provides a broad view of the surface development issues, along with the developability conditions and the technologies for the definition of developable surfaces. Given the absence of developability conditions in some areas of the shell, a number of methodologies are reported which produce a plate map onto the plane.

In CHAPTER 4 - CONCEPT AND IMPLEMENTATION OF AN ALGORITHM, the concept and the implementation of a new development algorithm is described, analysed and applied to example cases. By geodesicaly mapping the surface onto the plane, this method avoids the implementation difficulties of both non-developable surfaces, and developable surfaces with ruling lines aligned in any direction. Therefore, the slightly non-developable plates, commonly found in actual ship hulls, are easily accommodated by this process, working as a map onto the plane.

In CHAPTER 5 - INDUSTRIAL APPLICATION OF THE IMPROPER GEODESIC MAP, the user interface of the method is presented. The method provides information about the surface developability and fairness, which assists the user in the decision to develop or otherwise to take corrective measures, like re-fairing or editing of seams and butts.

Results obtained from analytical plates, and comparisons with results from both a 1/10-scale electrostatic development jig, and a commercial software package, validate the method. Other results, obtained from actual ship plates, are also presented, further confirming the good accuracy of the method's developments and its good behaviour when processing non-developable plates. This method is in current use, as part of a shipyard system.

To

Sonia, Mimé and António Cacho

Acknowledgements

This author deeply appreciates the guidance received from Mr. Ian Winkle, at the Naval Architecture & Ocean Engineering Department of the University of Glasgow, and from Prof.Carlos Guedes Soares, from Secção Autónoma de Engenharia Naval of Instituto Superior Técnico, at Lisbon Technical University.

The computer programming required by this thesis was not an isolated piece of software. In fact it was part of a complex programmatic tool, co-developed by Eng.Jorge Vitória and Eng.Nuno Santos, which authored most of the plate interpolation functionality, and by Mr.Luís Almeida and Eng.Paulo Pires, which revised the code and assisted in the debugging and documenting. Moreover, Eng.Tiago Torrado provided for the tool integration within the CAD environment in which it was intended to work.

This work was also supported by Estaleiros Navais de Viana do Castelo, mostly by the invaluable collaboration received from Eng.Carlos Rodrigues and his excellent staff, working at the drafting and computing rooms, which is to numerous to mention here.

This work was sponsored by Junta Nacional de Investigação Científica.

Table of Contents

List of Tables

List of Figures

List of Diagrams

Chapter 1 - Introduction

The shell plate development problem has several different industrial domains, each with its own peculiarities. Aerospace, car, textile and shipbuilding industries are the most typical ones.

Cars, clothes and flying craft have small dimensions and are mass-produced, but ships are quite the opposite in both respects. These are fundamental differences since they lead to solutions very particular to the process of ship design and ship building.

Nevertheless, solutions for one industry can be adapted for others, as proved by the early adoption of ship lofting solutions in the aerospace industry, and later feed-back from Boeing and McDonnell Douglas into shipyard systems[1].

The need for good hydrodynamic performance, requires hulls with smooth but complex curved shapes which makes the production of plated shells a non-trivial task, requiring highly skilled workmanship, both for the lofting and for the workshop.

Hulls made of fibre reinforced plastics do not pose the same requirements on workmanship, since they are lofted at full-scale and frequently built from reusable moulds.

It is not unusual for a shipyard to spend 3 to 5 working days of a two-man team (6 to 10 man-days), just to give to a single plate the appropriate curvature. This does not account for the lofting, cutting, fitting and welding. Note that the amount of curved plates in a full bodied hull, is typically over 10% of all the steel plates required for the complete construction, and for other types of ship that proportion can be doubled[4].

The aerospace and car industries sell so many units of each model, that the curved plates used in its construction can be mass produced, justifying dies for any single shape. Those dies are very expensive and time consuming to produce.

Ships are frequently built on a single specimen basis, or at most in short series. In fact, even the symmetry of the ship along its longitudinal centre-plane does not make for such an economy, because the plate series are not long enough to justify dies or other mass production technologies.

However, the shipyard plate production process is required to be less expensive and quicker in its delivery of a unique plate shape. Thus, the process should remain highly flexible, but require much less human intervention.

Being an industry that creates one-of-a-kind products is one of the most compelling reasons why shipyards maintain such large workforces (along with the product size and the short delivery schedules).

To improve the curved plate production methods, shipyards have always been close to the state of the art in metallic sheet technologies, and have come a long way since the all template-based production of the last century, when workmanship was the cheap item in the process. Nowadays, it is common to have the flat plates (*blanks*) cut and marked with rolling curves, or even heating curves, as part of the computer-aided design and manufacturing system. Templates can also be automatically produced, but further automation has proved out of reach without a new generation of workshop tools, namely numerically controlled pressing/heating templates, able to handle at least a mid size plate. Smaller sizes require increased welding, and have therefore not been used noticeably.

Moreover, not all the curved surfaces can be developed, because many do not result from rolling a plane surface, but from heating, stretching, twisting, bulging, or a combination of these. To develop a plate is to unroll it to its isometric flat shape, the blank. Plate development procedures are neither easy nor simple for the newcomer. The reason is in part that it is a rather exotic mathematical topic, making it impossible to describe generic developable surfaces by Euclidean geometry. This can only be accomplished by resorting to differential geometry, a much more involved and advanced mathematical discipline, which can be quite stimulating. In Chapter 2, the reader will find a summary presentation of the relevant mathematical concepts and results applied to developable surfaces.

From the earliest metal ship construction, 150 years ago, full-scale lofting leading to complete wooden templates, was the basis for curved plate forming, in a context where the labour was cheap and abundant, and accuracy was not an issue, due to the overlap between adjacent plates required for riveting. The development of welding technologies during the Second World War, and the advent of 1/10 lofting in the late fifties, made way for much higher accuracy requirements.

Until the onset of digital computers, shipbuilders could only develop plates using complex manual drafting techniques, or analogue devices such as the electrostatic jig, which models the hull surface at a scale of 1/10. These development procedures were very time consuming, and required highly skilled people to be accurate, so they were expensive. With the availability of computers for technical applications in shipyards, plate development methods and developable surface design and fitting, were among the first applications to be programmed. Chapter 3 reviews the references about these topics in the last three decades.

The time required for plate development in a unique hull constitutes lead-time for the construction process, since before it the hull plates couldn't be cut and even the final production designs of the assembly blocks can't be settled. With computer-aided design, the time required for plate development was reduced dramatically and the accuracy has much improved. Recently, some shipyards start migrating towards robot-welding of complex assemblies, combined with new cutting and welding technologies, like laser. These technologies produce much improved cuts and welds, finer heat affected zones, and reduced cooling shrinkage. The resulting accuracy increase is rendering obsolete the traditional use of added material in the plate boundaries. Without this common allowance for process errors, including the development and forming ones, the quality requirements of these tasks are further stressed.

Unlike the fabrication processes themselves, lofting has benefited enormously from the integration of computer systems, in which cutting and marking machines can now act as mere peripherals to the computer, in the same way as printers do. The development process can benefit from an accurate and comprehensive design database, and even more from a much improved fairing process.

The first problem in preparing for the production of a curved plate, is the geometric definition of the blank to be cut. If a plate is developable, the mathematical properties of developable surfaces can be used to "unroll" that plate. Simply stated, a surface is developable if it can be unrolled into a plane without any resulting strain. Conical and cylindrical surfaces are developable, but those with at least one spherically shaped region, cannot be unrolled without strain, so they are not developable, as are those shaped like a saddle.

A plate is defined by the surface that follows its half thickness. This surface is called the plate's mid-surface. Therefore, a plate is developable if it is possible to unroll it into a plane without any mid-surface strain.

Mathematically cones and cylinders have zero second principal curvature at every surface point along the direction of their generators. Therefore, to form these surfaces one should roll along their generators.

Regions shaped as a sphere have two non-zero principal curvatures with the same sign everywhere, while saddle shaped regions have non-zero principal curvatures with opposite signs. Thus, a surface region shaped as a sphere or as a saddle has no strain-free rolling directions. (Pressing it into a plane would produce significant strains, as would happen with any other forming technology.) Since non-developability implies that there is no unique mapping between the curved plate and an eventual blank[5,6], the researcher must consider further constraints, reflecting the variety of workshop techniques, such as those proposed by Letcher[1].

Development methods seek that unique mapping between a developable plate and its blank. When applied to non-developable plates, such methods either provide no solution, or they lead to a solution that is not unique. To define an accurate map of a non-developable surface one should start to define what is the "correct" result. From the mathematical point of view, any result is equally valid, since the map can't be isometric. From the workshop point of view, the result should be feasible and economic. If not for the focus in the bottom-line, this last perspective is more interesting since it restrains the solution space. But still we have no unique solution, given that the economic and forming restraints would translate into quite different optimising conditions between different shipyards, due to differences in costs, technology and practices. Therefore, the application of a development method to non-developable plates is not acceptable unless as an approximation to a production optimum.

The research documented in this thesis was originally intended to support an order placed by Estaleiros Navais de Viana do Castelo. This shipyard required a basic development tool compatible to the existing CAD/CAM system[1], which should match the quality of the existing electrostatic development jig using the same data, and improve the labour efficiency. The discussions had with the shipyard for setting the specifications, made clear that the yard was not concerned with developability issues, just requiring that the tool behaved like the electrostatic development jig. This device was mostly intended to process accurately the developable plates. Nevertheless, the sheets used in modelling the plate material could endure some plastic deformation, but only to an extent known to be "small" and irrelevant for the project managers. Therefore, the software should be tolerant at least to marginally non-developable surfaces.

There are a danger in this to let the user process a non-developable plate, not knowing its true nature, which is more difficult and expensive to form. The issue here is to let the user evaluate the developability of each individual plate. The developability of the individual plates is both a matter of the shell shape, and of the straking methodology employed. These tasks are rather complex to be done manually in an optimal fashion. But the appropriate computer tools could offer the designer the information and the editing facilities to go further in the engineering for production, either re-fairing or using optimised straking methodologies.

---

[1] The shipyard launched an important software and hardware project to improve the design and production procedures. The author was only the developer of the plate development tool, almost as an isolated piece of software, and therefore was not involved in the design or implementation of the other elements of the CAD system.

Only properly faired surfaces should be subjected to the map into the plane, irrespective to the developability or the existing level of curvature. Otherwise the quality of results could be compromised by errors propagated from bumps in the surface. The focus of this work is the development however some of the software components necessary for the development process can be adapted to depict surface information, assisting the user in the evaluation of the surface fairness and straking optimality. Note that the curvature computations necessary for the geodesic tracing, required by the development algorithm, are closely related to the differencing used in fairness assessments.

When the plates are non-developable, the criteria for acceptance of any plate mapping onto a plane must be related to the extent of existing non-developable regions, to the magnitude of the 2nd principal curvature, and also to the particular workshop practices. This last issue is rather complex to quantify, so it was not an objective of this work to provide answers for it. However, one common strategy to deal with the non-developable regions of the hull, is to subdivide it in smaller and simpler plates, preferably increasing the area covered by developable plates.

It is not viable to quantify the accuracy of the method in general terms, since it is a map for which the objects do not form a known population (any conceivable surface can be admissible). However, one should study the actual performance of the method when dealing with typical plates, and compare it with other known and accepted development procedures. Moreover, since the basic step of the method is the tracing of geodesics, one can study the accuracy of this elemental procedure. For any surfaces whose geometry is defined analytically, the geodesic can be determined analytically and therefore accurately. For surfaces discretely defined (for instance, by a set of section lines), the accuracy assessment can only be made by comparison with the results of another approximate method. See Fig. 35 and Fig. 36 for comparisons of the two methods of geodesic tracing, without concern to the final development accuracy.

The influence of the principal curvatures of the plate on the accumulated geodesic curvature is studied empirically, for typical developable and non-developable surfaces (see Fig. 35 and Fig. 36). The influence of the developability on the computing efficiency of the tracing procedure is also interesting enough to be evaluated, since the method is intended for computer implementation (see Fig. 37).

The effects of thickness on the development process where not covered. The shipyard required that the software data should match the data for the electrostatic development jig, which worked reliably for decades without thickness considerations. This amounts to approximate the plate's mid-thickness surface with the surface passing through the offsets. Therefore, for most of this work this will be the referenced, instead of the mid-thickness one.

For the development tool to be compatible with the existing CAD/CAM system, it is required to accept the existing surface data, i.e. a grid of surface lines, with spacing typical of a paper-based drawing process, not proper CAD databases. Therefore, these discretisation practices are assessed by making comparisons with several frame spacings sampling a comprehensive range. See Fig. 34 for a depiction of the geodesic tracing accuracy related to the discretizing distance.

Testing with actual shipyard data is sensitive and expensive, but it is necessary for reasonably evaluating the algorithm, since it is intended for practical use. The outputs from the method are compared with results from a 1/10-scale electrostatic development jig, and also with results of commercial surface development software. See page 108 for results of comparisons made outside the work of this thesis[2]. The results in Table 7 and Table 8 ought to give confidence in the algorithm and provide for the validation of the software.

Conclusions can be found in Chapter 4, which summarises the findings of this study, and Chapter 6 briefly explores further research prospects.

The reader can find general guidance on definitions and terminology in Appendix A. See Appendix B for results obtained with surfaces designed analytically and translated to a discrete representation of sections lines. These cover the three surface categories, in regard to curvature: a plane surface (see page 126), seven developable surfaces (see pages 127 to 133), and six non-developable surfaces (see pages 134 to 139).

Chapter 2 - The Mathematics of Developable Surfaces

Classical books on mathematics are not known for the ease of use by the newcomer, since they are written for mathematicians, thus taking advantage of rather elegant but involved notations, and moreover, building the notation and naming conventions on top of traditional ones, which often are meaningless for non-mathematicians. Additionally, the most relevant definitions and results about surface development are widely spread in the bibliographic references, requiring a great deal of searching effort.

Therefore, this Chapter is not intended to substitute for the references, but it is rather a concise and structured compilation of the mathematical concepts, definitions and theorems, which support the rest of the thesis, and provide for general guidance in the consultation of the references.

2.1 Notation

In the text **boldface** stand for vectors and matrices, and *italics* stand for scalars and for terms to be defined.

Unless stated otherwise, every surface and curve is expected to be *regular,* which means it is clear of cusps, knuckles and self-intersections.

Any single proposition is numbered to allow for unambiguous reference, which would be otherwise impractical.

2.2 Preliminary Results

The *angle* between two vectors **a** and **b** is given by:

$$\alpha = \arccos\frac{\mathbf{a}.\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \arcsin\frac{\|\mathbf{a}\times\mathbf{b}\|}{\|\mathbf{a}\|\|\mathbf{b}\|} \tag{1}$$

The *projected length* of the vector **a** onto **b** is given by:

$$a_b = \frac{\mathbf{a}.\mathbf{b}}{\|\mathbf{b}\|} \tag{2}$$

The *component* of the vector **a** about **b** is given by:

$$\mathbf{a}_b = a_b \frac{\mathbf{b}}{\|\mathbf{b}\|} \tag{3}$$

For a point **P** to be on a plane which has a normal vector **n** and contains the point **Q**, it is necessary and sufficient that:

$$\mathbf{n}.\mathbf{P} = \mathbf{n}.\mathbf{Q} \Leftrightarrow n_x x + n_y y + n_z z = n_x Q_x + n_y Q_y + n_z Q_z \tag{4}$$

If the point Q is the one closest to the axis origin, among all plane points, then it is called the *central point* of the plane.

Its locating vector is normal to the plane, unless the plane contains the origin, in which case **Q**=0.

Therefore eqn.4 is equivalent to:

$$\mathbf{Q}.\mathbf{P}=\mathbf{Q}.\mathbf{Q} \Leftrightarrow Q_x x + Q_y y + Q_z z = Q_x^2 + Q_y^2 + Q_z^2 \tag{5}$$

Defining every plane by its central point establishes a one-to-one relationship between points and planes, and thus every plane can be defined simply by the three co-ordinates of the corresponding central point, in the given co-ordinate system.

2.3 Basic Concepts on Curves

Condition to be met by the *curve* **c**, in which u is the curve parameter:

$$\frac{\mathrm{d}\,\mathbf{c}}{\mathrm{d}\,u} \neq 0, \quad \forall u \tag{6}$$

The *natural parameter* is given by the arc length accumulated along the curve. If u_i is the initial parameter value on the curve, and u_f the parameter at some specified point, the natural parameter s for this point is computed by:

$$s = \int_{u_i}^{u_f} \left\| \frac{\mathrm{d}\,\mathbf{c}}{\mathrm{d}\,u} \right\| \mathrm{d}\,u \tag{7}$$

Using s instead of some non-natural parameter u, one benefits from this important simplification (see the following results, using s instead of u):

$$\left\| \frac{\mathrm{d}\,\mathbf{c}}{\mathrm{d}\,s} \right\| = 1 \tag{8}$$

The *unit tangent* to the curve **c** is given by:

$$\mathbf{t} = \mathbf{t}(u) = \frac{\dfrac{\mathrm{d}\,\mathbf{c}}{\mathrm{d}\,u}}{\left\| \dfrac{\mathrm{d}\,\mathbf{c}}{\mathrm{d}\,u} \right\|} \tag{9}$$

The *unit binormal* to the curve **c** is given by:

$$\mathbf{b} = \mathbf{b}(u) = \frac{\dfrac{\mathrm{d}\,\mathbf{c}}{\mathrm{d}\,u} \times \dfrac{\mathrm{d}^2\,\mathbf{c}}{\mathrm{d}\,u^2}}{\left\| \dfrac{\mathrm{d}\,\mathbf{c}}{\mathrm{d}\,u} \times \dfrac{\mathrm{d}^2\,\mathbf{c}}{\mathrm{d}\,u^2} \right\|} \tag{10}$$

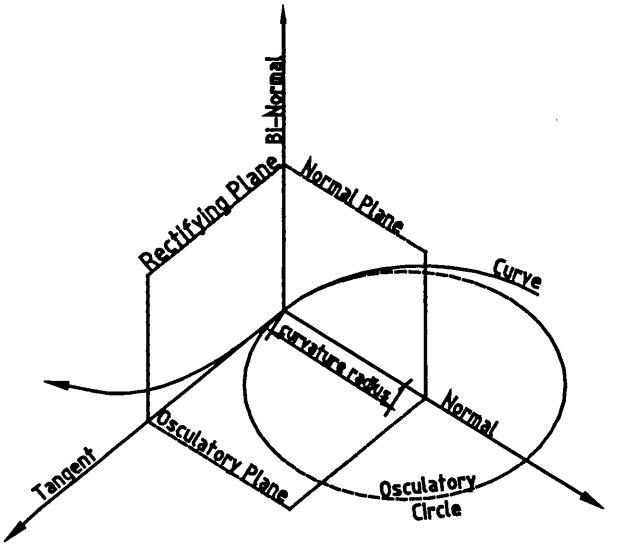


Fig. 1 - The moving trihedron, with the relevant planes.

The *unit main normal* to the curve **c** is given by:

$$\mathbf{n} = \mathbf{n}(u) = \mathbf{b} \times \mathbf{t} \tag{11}$$

The *moving trihedron* or *Frenet-Serret trihedron* is the system made at any curve point by the unit vectors **t**, **b** and **n** (see Fig. 1). A special circle traced from the curve centre of curvature with that same curvature is the *osculatory circle*. The plane containing it is the *osculatory plane*. The plane normal to the curve is the *normal plane*, and he plane defined by **n** and **t** is the *rectifying plane*.

The *curvature vector* of the curve **c** is given by:

$$\mathbf{k} = \mathbf{k}(u) = \frac{\mathrm{d}\mathbf{t}}{\mathrm{d}u} = k(u)\,\mathbf{n}(u) \tag{12}$$

The *curvature* of the curve **c** is given by:

$$k(u) = \|\mathbf{k}(u)\| = \left\|\frac{\mathrm{d}\mathbf{t}}{\mathrm{d}u}\right\| = \frac{\left\|\frac{\mathrm{d}\mathbf{c}}{\mathrm{d}u} \times \frac{\mathrm{d}^2\mathbf{c}}{\mathrm{d}u^2}\right\|}{\left\|\frac{\mathrm{d}\mathbf{c}}{\mathrm{d}u}\right\|^3} \tag{13}$$

The *curvature radius* of the curve **c** is given by:

$$\rho = \rho(u) = \frac{1}{k(u)} \tag{14}$$

Given the curve **c**, its *centres of curvature* lie in the curve **f**:

$$\mathbf{f} = \mathbf{f}(u) = \mathbf{c} + \rho\mathbf{n} \tag{15}$$

Torsion of the curve **c**:

$$\tau = \tau(u) = \mathbf{b}\frac{\mathrm{d}^3\mathbf{c}}{\mathrm{d}u^3} \tag{16}$$

The *Frenet-Serret* formulas relate:

$$\frac{\mathrm{d}}{\mathrm{d}u}\begin{Bmatrix}\mathbf{t}\\ \mathbf{n}\\ \mathbf{b}\end{Bmatrix} = \begin{bmatrix}0 & k & 0\\ -k & 0 & \tau\\ 0 & -\tau & 0\end{bmatrix}\begin{Bmatrix}\mathbf{t}\\ \mathbf{n}\\ \mathbf{b}\end{Bmatrix} \tag{17}$$

2.4 Basic Surface Concepts

The form of the *surface* σ, in which u_1 and u_2 are parameters and components of the two-dimension vector **u** is given by:

$$\sigma = \sigma(u_1, u_2) = \sigma(\mathbf{u}) \tag{18}$$

The *normal* to the surface σ, at the point with the parametric co-ordinates u_1 and u_2 is given by:

$$\mathbf{n}(\mathbf{u}) = \frac{\frac{\partial \sigma}{\partial u_1} \times \frac{\partial \sigma}{\partial u_2}}{\left\| \frac{\partial \sigma}{\partial u_1} \times \frac{\partial \sigma}{\partial u_2} \right\|} \tag{19}$$

The *normal curvature vector* of the curve **c**, lying in a surface σ, is the component of the curvature vector which is normal to σ (see Fig. 2):

$$\mathbf{k}_n = \mathbf{k}_n(u) = k(u)\mathbf{n}(u) \tag{20}$$

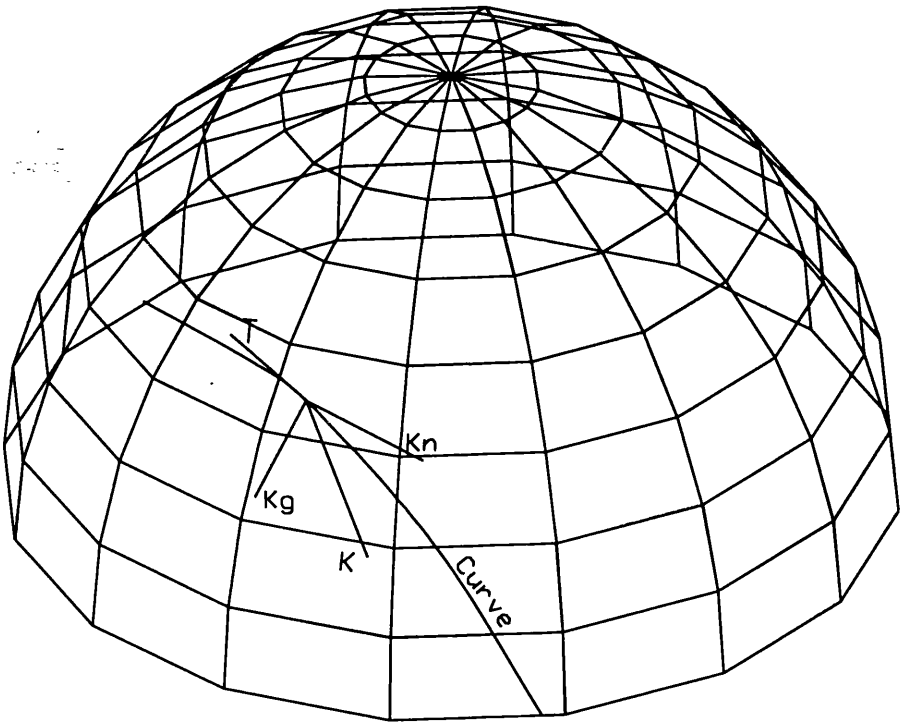


Fig. 2 - The curvature vector of a surface's line, and its components: the normal and the geodesic curvatures.

The normal curvature of a curve c, lying in a surface s, is the length of the normal curvature vector:

$$k_n = \left\| \mathbf{k}_n(u) \right\| \tag{21}$$

The *vector of the geodesic curvature* of the curve **c**, lying in the surface σ, is the component to the curvature vector which is tangential to σ:

$$\mathbf{k}_g = \mathbf{k}_g(u) = \mathbf{k}(u) - \mathbf{k}(u)\mathbf{n}(u) \tag{22}$$

The *geodesic curvature* of the curve **c**, lying in the surface σ, is the length of the geodesic curvature vector:

$$k_g = \left\| \mathbf{k}_g(u) \right\| \tag{23}$$

Let **c** be a curve lying in the surface σ, and **P** be a point in that curve. The direction of **c** at **P** is given by its tangent vector at that point, $\mathbf{t}_p$. When $\mathbf{t}_p$ and **c** are rotated around **P**, the value for the normal curvature changes between two extreme values, called the *principal curvatures*, k_1 and k_2 [7], which are the upper and the lower bounds.

The *Gaussian curvature* at a given point of the surface is given by[5]:

$$K = k_1 k_2 \tag{24}$$

Surfaces with *double curvature* are those with both principal curvatures simultaneously non-zero at some point.

Surfaces with *single curvature* are those with only one of the principal curvatures zero everywhere.

Non-curved or *plane* surfaces are those with both principal curvatures zero everywhere.

A surface is either of double curvature, single curvature or plane.

Definition of a developable surface: to be developable, a surface must have no more than one non-zero principal curvature, meaning it must be either singly-curved or already plane.

The *mean curvature* at a given point of the surface is given by[5]:

$$H = \frac{k_1 + k_2}{2} \tag{25}$$

The directions in which k_1 and k_2 are computed are the *principal directions*, h_1 and h_2, which are always orthogonal to each other[5,6,7], unless at umbilic points. These umbilic points are points in the surface where $h_1$and h_2 are not determined, since at these points the principal curvatures are equal.

A *line of curvature* is a surface curve aligned everywhere with one of the principal directions[5,6].

The principal directions are defined as scalars, not vectors, because they are taken as quotients on the parameter variations, measured along each principal direction. At each surface point, after measuring the variation of both surface parameters along the direction of k_i, the principal directions are given by the quocient:

$$h_i = \frac{\mathrm{d}\,u_2}{\mathrm{d}\,u_1} \quad i = 1, 2 \tag{26}$$

If $h_i = 0$ then the principal direction i is aligned with the co-ordinate curve of u_1. If $h_i = \infty$ then the principal direction i is aligned with the co-ordinate curve of u_2.

A *geodesic* is a surface curve with zero geodesic curvature everywhere[5,7], having the following properties[5,6,7]:

- the osculating plane of the geodesic is always normal to the surface;
- the normal plane of the geodesic is always normal to the surface;
- the rectifying plane of the geodesic is always tangent to the surface;
- any surface point is crossed by one and only one geodesic at any given direction.

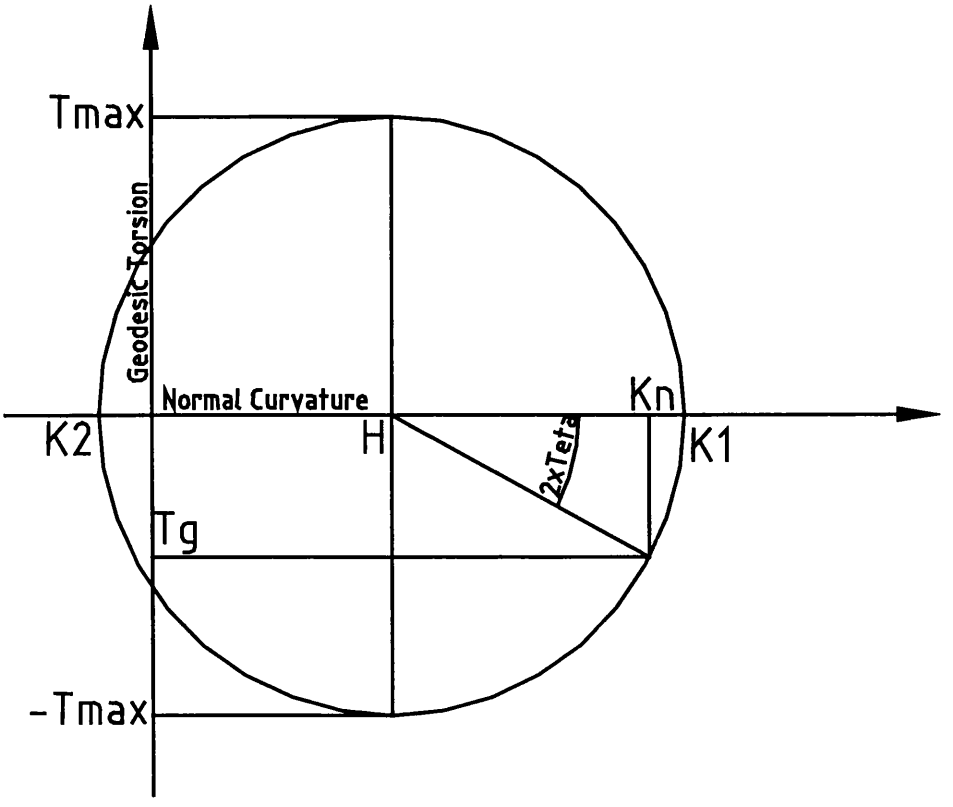


Fig. 3 - The curvature/torsion circle.

A geodesic is a line of curvature if and only if it is a plane curve[5]. Therefore, the lines of curvature aren't always geodesic.

The *geodesic torsion* of the surface σ at a given point **P**, in the direction *d*, is the torsion of the one and only geodesic crossing **P** with direction *d*. It is a scalar represented by τ_g. When *d* changes, τ_g changes accordingly with the surface shape in the vicinity. That's why τ_g is also called the *surface torsion*[7]. The surface torsion have two extreme values in every surface point (see Fig. 3). The directions at which these extreme values occurs are rotated by 45° about the principal directions[7]. At the principal directions the surface torsion is always zero[7], thus a line of curvature is a line of zero surface torsion.

An *asymptotic curve* is a curve lying in the surface which is always aligned with one of the surface directions for maximum torsion [5,6].

Along an asymptotic line the normal curvature is always equal to the mean curvature *H*.

For any surface point, the relationship between the normal curvature and the surface torsion can be plotted as a circle[7].

Any rotation on the considered direction by an angle α, is mapped in the circle to a 2α rotation, so the principal directions seem to be 180° apart in the circle.

Note that the geodesic torsion extreme values occur at the mean curvature directions (see Fig. 3).

The relation between the principal curvatures, and the normal curvature taken at an angle θ with the first principal direction is (see Fig. 3):

$$k_n(\theta) = \frac{k_1 + k_2}{2} + \frac{k_1 - k_2}{2}\cos(2\theta) \tag{27}$$

The relation between the principal curvatures, and the surface torsion taken at an angle θ with the first principal direction, is (see Fig. 3):

$$\tau_g(\theta) = -\frac{k_1 - k_2}{2}\sin(2\theta) \tag{28}$$

2.5 Ruled Surfaces

A *ruled surface* is the locus of a line, called a *generator*, whose direction is determined by successive values of a parameter, moving continuously along a

curve called the *directrix*, which intersects that curve at an angle other than zero[5] if the intersection point lies inside the surface boundaries.

A *generator* is a moving straight-line which sweeps out the entire surface[8]. A *ruling* is an instantaneous position of the generator[8].

Let the directrix be the curve $\mathbf{d}=\mathbf{d}(u_1)$, and $\mathbf{r}$ be the unit vector of the generator $\mathbf{g}=u_2.\mathbf{r}(u_1)$, in which u_1 and u_2 are the parameters for the directrix and the generator. Then the equation for the ruled surface $\sigma=\sigma(u_1, u_2)$ is:

$$\sigma(u_1, u_2) = \mathbf{d}(u_1)+ u_2\mathbf{r}(u_1) \tag{29}$$

The following expression is an alternative definition of a ruled surface σ, when it contains two curves $\mathbf{d_a}=\mathbf{d_a}(u_{1a})$ and $\mathbf{d_b}=\mathbf{d_b}(u_{1b})$, known respectively as the primary and the secondary directrices. The parameter values u_{1a} and u_{1b} define the intersection points of a ruling at both curves, and u_2 is the length measured along the ruling, from the primary directrix $\mathbf{d_a}$ to the considered surface point.

$$\sigma(u_{1a}, u_{1b}, u_2) = (1- u_2)\mathbf{d_a}(u_{1a})+ u_2\mathbf{d_b}(u_{1b}) \tag{30}$$

The *cylinder* is a ruled surface in which all rulings are parallel to each other. The common cylinder is the *circular cylinder*, because its admits an arc of a circle as directrix.

The *cone* is a ruled surface in which all rulings intersect at one and the same point, known as the apex. The cone generator is the *generatrix*. The common cone is the *circular cone*, because its admits an arc of a circle as directrix.

Definition of a developable surface: a *developable surface* is a ruled surface having the same tangential plane along one and the same generator[5].

A ruled surface is not developable if the rulings intersect each other[5].

Since a plane tangent to a ruled surface is tangent to the surface along some ruling, it must be tangent also to any curve lying in the surface where it crosses the ruling.

Consider two points, **R** and **S**, each one in a curve lying in a developable surface. For the existence of a ruling between them, it is necessary and sufficient that the curve tangents at those points, named $\mathbf{t_R}$ and $\mathbf{t_S}$, lie in the same plane (see Fig. 4). Therefore, the condition for the line **RS** to be a ruling, is:

$$(\mathbf{RS}\times\mathbf{t_R}).\mathbf{t_S} = 0 \tag{31}$$

Another way to put it, is to say that the surface normals at **R** and **S**, named $\mathbf{N_R}$ and $\mathbf{N_S}$, must be parallel:

$$\mathbf{N_R \times N_S} = (\mathbf{t_R \times RS}) \times (\mathbf{t_s \times RS}) = 0 \qquad (32)$$

The cylinder is a developable surface.

The cone is a developable surface, excluding the apex where all the rulings intersect each other.

The *tangent surface* (also called *convolute*[9,9]), is a ruled surface with the generator tangent everywhere to the directrix (see Fig. 5).

To be a ruled surface, the directrix cannot be inside the surface boundaries, as the angle from the directrix to the generator is always zero. But one can consider the tangent surface, as being the union of the directrix with the two surface sheets, each one produced by making $u_2>0$ and $u_2<0$ in Eqn.29.

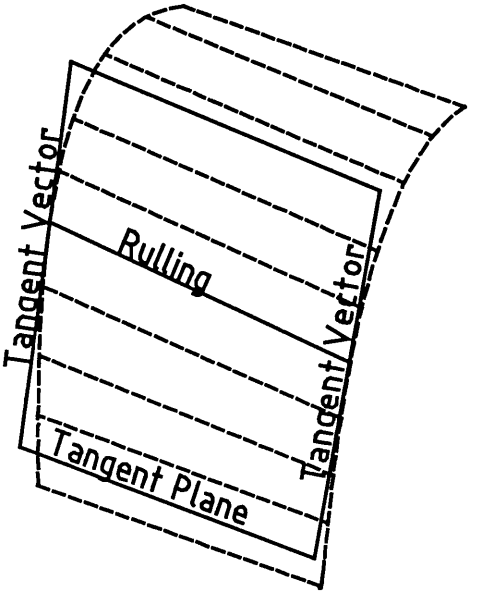


Fig. 4 - The tangent plane contains the ruling and both tangent vectors, if the surface is developable.

The directrix of a tangent surface is named *edge of regression*[5,6], *cuspoidal edge*[6,11], or *characteristic curve* (see Fig. 5). The edge of regression is also known as the *evolute*, by relation to some possible *involute* lying on the tangent surface[5,6]. The *involute* is a curve which is normal to every ruling.

The point of the edge of regression lying in some ruling of the tangent surface is called the *characteristic point* or the *apex* of that ruling (see Fig. 5).

Every tangent surface is developable. In a tangent surface the locus of all apexes is a curve called *the edge of regression.*

A plane tangent to a developable surface is an osculating plane at the cuspoidal edge (see Fig. 5).

Every rectifying plane of the cuspoidal edge is normal to the tangent surface, and intersects it along a ruling (see Fig. 5).

The edge of regression for a tangent surface contains the points of intersection of any two consecutive rulings.

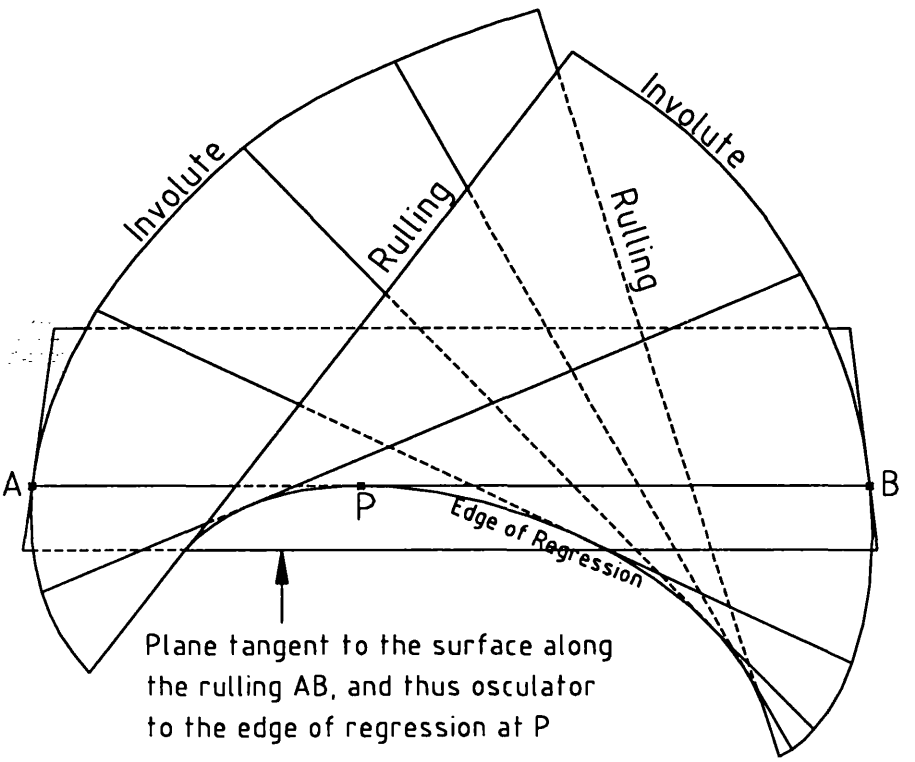


Fig. 5 - A tangent surface, bounded by two involutes and two rulings, showing a plane tangent to it.

Taking points at every ruling at the same parameter u_2 (recall Eqn.29), the distance between them vanish when u_2 changes in the direction of the apex, so the apex curve is described by:

$$\frac{\partial \mathbf{p}}{\partial u_1} = \frac{\partial \mathbf{d}}{\partial u_1} + u_2 \frac{\partial \mathbf{r}}{\partial u_1} = 0 \tag{33}$$

A curve is called a Bertrand offset of another if they have the same main normals. A surface is a Bertrand offset of another if they have the same unit normals[5,6,11].

Suppose a tangent surface offsets another developable surface by a distance D, and moreover, there exists a constant angle δ between their regression edges

tangents. Then its regression edge must have the curvature and torsion related by the expression[12]:

$$(1 - k\,\mathrm{D}) \sin\delta - \tau D \cos\delta = 0 \tag{34}$$

If a set of superimposed surface layers, like the ones in a composite material, are developable, then they satisfy Eqn. 34.

2.6 The Envelope of a One-Parameter Family of Planes

A one-parameter family of planes has the general equation:

$$n_x(u)x + n_y(u)y + n_z(u)z = q(u) \Leftrightarrow \mathbf{n}(u).\mathbf{p} = q(u) \tag{35}$$

Definition of a developable surface: every developable surface is the envelope of a one-parameter family of planes, and every envelope for a one-parameter family of planes is developable[6].

Any ruling in the envelope of the one-parameter family of planes is described by[11]:

$$\begin{cases} \mathbf{n}(u).\mathbf{p} = q(u) \\ \dfrac{\mathrm{d}\mathbf{n}(u)}{\mathrm{d}u}.\mathbf{p} = \dfrac{\mathrm{d}q(u)}{\mathrm{d}u} \end{cases} \tag{36}$$

The edge of regression of the envelope of the one-parameter family of planes is described by[11]:

$$\begin{cases} \mathbf{n}(u).\mathbf{p} = q(u) \\ \dfrac{\mathrm{d}\mathbf{n}(u)}{\mathrm{d}u}.\mathbf{p} = \dfrac{\mathrm{d}q(u)}{\mathrm{d}u} \\ \dfrac{\mathrm{d}^2\mathbf{n}(u)}{\mathrm{d}u^2}.\mathbf{p} = \dfrac{\mathrm{d}^2 q(u)}{\mathrm{d}u^2} \end{cases} \tag{37}$$

Every developable surface can be subdivided into regions so that any of these regions is a portion of either a plane, a generalised cylinder, a generalised cone, or a tangent surface[5]. However, a surface entirely made of developable

regions is not necessarily developable, due to possible non-developable boundaries between the developable regions.

2.7 Development Properties

Definition of a developable surface: a surface is developable, if and only if it has zero Gaussian curvature everywhere [5,6,7]. This is equivalent to the statement that a surface is developable if and only if it has the lower principal curvature zero-valued everywhere.

The development of a surface preserves the angles, the lengths, and the geodesic curvature of the curves lying in it[5,6], since it is an isometric mapping. The normal curvature is eliminated by the development, and the curvature of the developed line is the geodesic curvature existing before the development. For any given surface the development is unique, if it exists[5,6].

If a surface is not developable, to map it onto a plane one can always consider several flattening maps, so any mapping of a non-developable surface onto the plane is not unique[5,6].

For a line lying in a surface, the integration of the Frenet-Serret equations give its development. For this it is necessary to eliminate the torsion in the Frenet-Serret equations, which then becomes a system of only two equations, in which case it is enough to determine the two unknown functions on the plane co-ordinates x and y.

The resulting equations on the developed co-ordinates are[9]:

$$\begin{cases} \dfrac{\mathrm{d}^2 x}{\mathrm{d}s^2} + k_g(s)\dfrac{\mathrm{d}y}{\mathrm{d}s} = 0 \\ \dfrac{\mathrm{d}^2 y}{\mathrm{d}s^2} - k_g(s)\dfrac{\mathrm{d}x}{\mathrm{d}s} = 0 \end{cases} \tag{38}$$

2.8 The Spherical Map

Consider the map between any surface point and the unit sphere point where the locating vector is the surface unit normal. This map is known as the *Gauss map* or the *spherical map*, and is represented by **N**. The image of a surface or a

curve lying on it by the Gauss map, is called the *spherical indicatrix,* or simply the *indicatrix*[5,6,11].

The tangent plane at a surface point is parallel to the tangent plane at the image of that point, made by the spherical map [5,6]. If a surface is developable, then the surface normals are parallel along each ruling[5,6]. Therefore, the indicatrix of any ruling is just a point on the unit sphere, an the indicatrix of a developable surface is just a line[5,6]. If the indicatrix of a developable surface has a cusp, then the surface have an inflexion at the surface ruling which was mapped on that cusp[11].

The indicatrix of a circular cylinder is an arc of a maximum circle (see Fig. 6). The rulings are both tangent to the unit sphere and normal to the arc of circle.

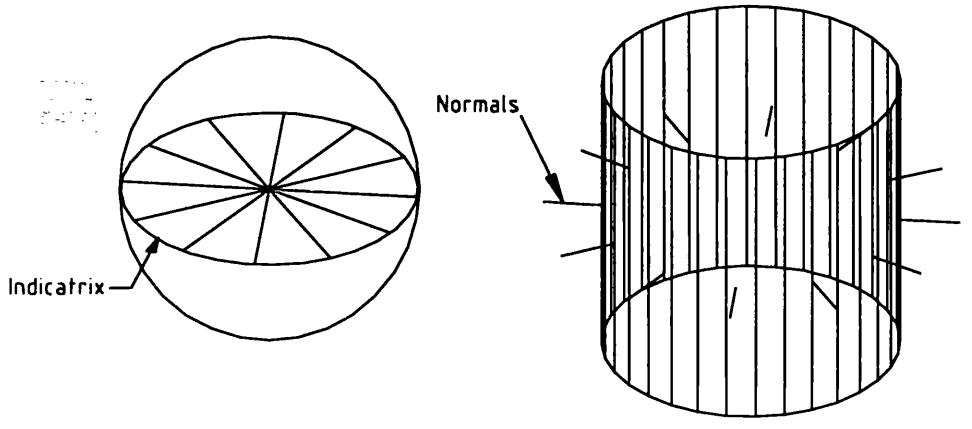


Fig. 6 - The spherical indicatrix of a circular cylinder.

The indicatrix of a circular cone is an arc of circle (see Fig. 7). The rulings are both tangent to the unit sphere and normal to the arc of circle.

The spherical indicatrix for a generalised cylinder is given by:

$$\mathbf{n}.\left(\frac{\mathrm{d}\mathbf{n}}{\mathrm{d}u}\times\frac{\mathrm{d}^2\mathbf{n}}{\mathrm{d}u^2}\right)\equiv 0 \tag{39}$$

While for the generalised cone it is:

$$\mathbf{n}.\left(\frac{\mathrm{d}\mathbf{n}}{\mathrm{d}u}\times\frac{\mathrm{d}^2\mathbf{n}}{\mathrm{d}u^2}\right)\equiv \text{constant}\neq 0 \tag{40}$$

And for a tangent surface is:

$$\mathbf{n}.\left(\frac{\mathrm{d}\mathbf{n}}{\mathrm{d}u}\times\frac{\mathrm{d}^2\mathbf{n}}{\mathrm{d}u^2}\right)=f(u)\neq 0 \tag{41}$$

The proportion between an element of area at the surface and its spherical map is the absolute value of the Gaussian curvature[5]:

$$\frac{\mathrm{d}\,A_{spherical}}{\mathrm{d}\,A} = |K| \tag{42}$$

Consider the differential of the Gauss map, d**N**(**u**), with **u**=(u_1, u_2) being the parameter vector of the original surface (not the unit sphere). The determinant of d**N** is the Gaussian curvature K[6]. The negative half trace[2] of d**N** is the mean curvature H[6].

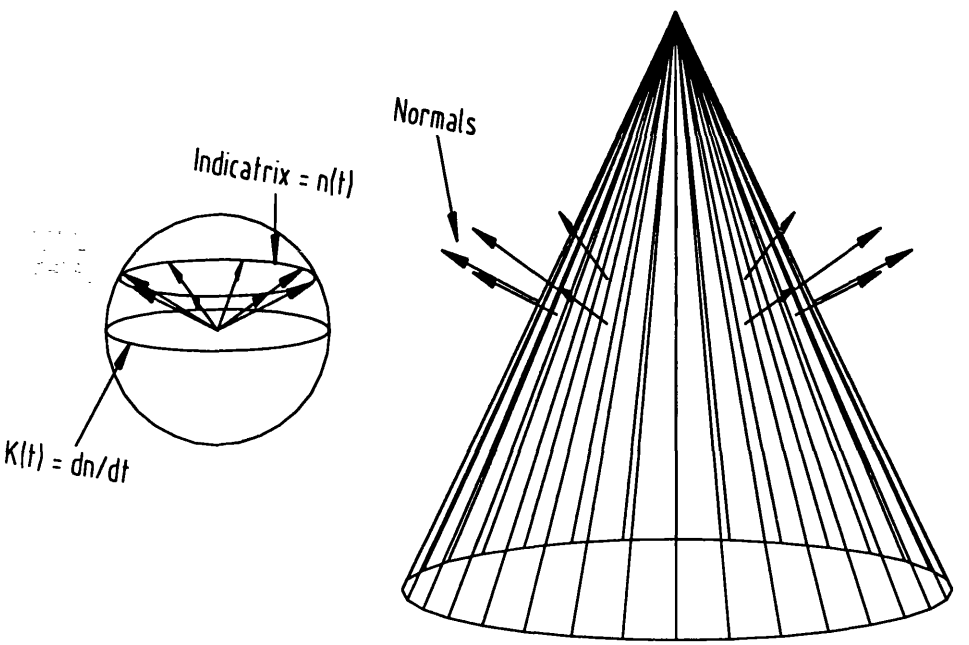


Fig. 7 - The spherical indicatrix n(t) of a circular cone, and its curvature K(t) = dn/dt.

2.9 The Metric and Curvature Tensors

The form of the *metric tensor*, at a given point of the surface σ is:

$$g_{ij} = \frac{\partial\sigma}{\partial u_i}\frac{\partial\sigma}{\partial u_j} \qquad i,j = 1,2 \tag{43}$$

[2] The trace of $\mathbf{a}_{ij}$ is $\Sigma\mathbf{a}_{ii}$.

The metric tensor is symmetric, implying that:

$$g_{ij} = g_{ji} \Rightarrow \frac{\partial^2 \sigma}{\partial u_i \partial u_j} = \frac{\partial^2 \sigma}{\partial u_j \partial u_i} \tag{44}$$

If a map between surfaces is isometric, then the metric tensors of both surfaces are equal between any mapped points[5]. Thus any mathematical entity depending exclusively on the metric tensor is preserved by an isometric mapping.

The components of the metric tensor are the coefficients of the first fundamental form *I*, which is a quadratic form. It gives the elementary displacement on the surface, due to elementary increments on the parameters, made along a direction **du**:

$$I(\mathbf{du}) = d\sigma^2 = g_{ij} du_i du_j = \mathbf{du}.\mathbf{g}.\mathbf{du}^T, \quad \mathbf{du} = \{ du_1 \; du_2 \} \tag{45}$$

The form of the *curvature tensor*, in a given point of the surface σ is:

$$b_{ij} = \mathbf{n} \frac{\partial^2 \sigma}{\partial u_i \partial u_j} = -\frac{\partial \mathbf{n}}{\partial u_i} \frac{\partial \sigma}{\partial u_j} \qquad i, j = 1,2 \tag{46}$$

The curvature tensor is symmetric, thus:

$$b_{ij} = b_{ji} \Rightarrow \frac{\partial \mathbf{n}}{\partial u_i} \frac{\partial \sigma}{\partial u_j} = \frac{\partial \mathbf{n}}{\partial u_j} \frac{\partial \sigma}{\partial u_i} \tag{47}$$

At the *umbilic points* the curvature tensor has determinant zero.

The coefficients of the second fundamental form *II*(**du**) are the components of the curvature tensor. This quadratic form doubles the elementary displacement *h* normal to the surface, due to elementary increments on the parameters.

Notice that the displacement normal to the surface is in fact half the second order differential, projected along the surface's normal:

$$II(\mathbf{du}) = \mathbf{n}\, d^2\sigma = b_{ij} du_i du_j = \mathbf{du}.\mathbf{b}.\mathbf{du}^T = 2h, \quad \mathbf{du} = \{du_1 \; du_2\} \tag{48}$$

The mean curvature, at a given point of the surface σ, is given by[5,7]:

$$H = \frac{g_{11} b_{22} - 2 g_{12} b_{12} + g_{22} b_{11}}{2(g_{11} g_{22} - g_{12})} \tag{49}$$

The Gaussian curvature, at a given point of the surface σ, is given by[5,7]:

$$K = \frac{b_{11}b_{22} - b_{12}^{\ 2}}{g_{11}g_{22} - g_{12}} \tag{50}$$

If the surface parameters are orthogonal to each other (meaning that there are at least one allowable orthogonal parameterisation), the Gaussian curvature can be expressed only upon the metric tensor[5]:

$$K = -\frac{1}{\sqrt{g_{11}g_{22}}}\left[\frac{\partial}{\partial u_1}\left(\frac{1}{\sqrt{g_{11}}}\frac{\partial\sqrt{g_{22}}}{\partial u_1}\right) + \frac{\partial}{\partial u_2}\left(\frac{1}{\sqrt{g_{22}}}\frac{\partial\sqrt{g_{11}}}{\partial u_2}\right)\right] \tag{51}$$

The principal curvatures, in a given point of the surface σ, are given by[5,7]:

$$k_{1,2} = H \pm \sqrt{H^2 - K} \tag{52}$$

The surface torsion, or geodesic torsion, at some given direction h, is given by[5,7]:

$$\tau_g = \frac{\left(g_{11}b_{12} - g_{12}b_{11}\right) + \left(g_{11}b_{22} - g_{22}b_{11}\right)h + \left(g_{12}b_{22} - g_{u_2u_2}b_{u_1u_2}\right)h^2}{\sqrt{g_{11}g_{22} - g_{12}}\left(g_{11} + 2g_{12}h + g_{22}h^2\right)} \tag{53}$$

The principal directions, in a given point of the surface σ, are given by[5,7]:

$$h_i = \frac{b_{11} - g_{11}k_i}{g_{12}k_i - b_{22}} = \frac{b_{12} - g_{12}k_i}{g_{22}k_i - b_{22}} \quad i = 1,2 \tag{54}$$

On a twice differentiable surface, the normal and the tangent vectors along each co-ordinate direction form a basis, on which one can define the surface's second derivatives by the *Gauss equations*[5,7]:

$$\begin{cases} \dfrac{\partial\sigma^2}{\partial^2 u_1} = \Gamma_{11}^1\dfrac{\partial\sigma}{\partial u_1} + \Gamma_{11}^2\dfrac{\partial\sigma}{\partial u_2} - g_{11}\mathbf{n} \\ \dfrac{\partial\sigma^2}{\partial u_1\partial u_2} = \Gamma_{12}^1\dfrac{\partial\sigma}{\partial u_1} + \Gamma_{12}^2\dfrac{\partial\sigma}{\partial u_2} - g_{12}\mathbf{n} \\ \dfrac{\partial\sigma^2}{\partial^2 u_2} = \Gamma_{22}^1\dfrac{\partial\sigma}{\partial u_1} + \Gamma_{22}^2\dfrac{\partial\sigma}{\partial u_2} - g_{22}\mathbf{n} \end{cases} \tag{55}$$

Whose coefficients Γ_{ij}^l are called the *Christoffel symbols of the first kind.* They only depend upon the metric tensor and its derivatives[5,7]:

$$\begin{cases} \Gamma_{11}^1 = \dfrac{g_{22}\dfrac{\partial g_{11}}{\partial u_1} - 2g_{12}\dfrac{\partial g_{12}}{\partial u_1} + g_{12}\dfrac{\partial g_{11}}{\partial u_2}}{2\left(g_{11}g_{22} - g_{12}^2\right)} \\ \Gamma_{12}^1 = \dfrac{g_{22}\dfrac{\partial g_{11}}{\partial u_2} - g_{12}\dfrac{\partial g_{22}}{\partial u_1}}{2\left(g_{11}g_{22} - g_{12}^2\right)} \\ \Gamma_{22}^1 = \dfrac{2g_{11}\dfrac{\partial g_{12}}{\partial u_2} - g_{22}\dfrac{\partial g_{22}}{\partial u_1} - g_{12}\dfrac{\partial g_{22}}{\partial u_2}}{2\left(g_{11}g_{22} - g_{12}^2\right)} \\ \Gamma_{11}^2 = \dfrac{2g_{11}\dfrac{\partial g_{12}}{\partial u_1} - g_{11}\dfrac{\partial g_{11}}{\partial u_2} + g_{12}\dfrac{\partial g_{11}}{\partial u_1}}{2\left(g_{11}g_{22} - g_{12}^2\right)} \\ \Gamma_{12}^2 = \dfrac{g_{11}\dfrac{\partial g_{22}}{\partial u_1} - g_{12}\dfrac{\partial g_{11}}{\partial u_2}}{2\left(g_{11}g_{22} - g_{12}^2\right)} \\ \Gamma_{22}^2 = \dfrac{g_{11}\dfrac{\partial g_{22}}{\partial u_2} - 2g_{12}\dfrac{\partial g_{12}}{\partial u_2} + g_{12}\dfrac{\partial g_{22}}{\partial u_1}}{2\left(g_{11}g_{22} - g_{12}^2\right)} \end{cases} \tag{56}$$

On a twice differentiable surface, the tangent vectors along each co-ordinate direction form a basis, on which one can define the first derivatives of the surface unit normal by the *Weingarten equations*[5,7]:

$$\begin{cases} \dfrac{\partial \mathbf{n}}{\partial u_1} = \beta_1^1 \dfrac{\partial \sigma}{\partial u_1} + \beta_1^2 \dfrac{\partial \sigma}{\partial u_2} \\ \dfrac{\partial \mathbf{n}}{\partial u_2} = \beta_2^1 \dfrac{\partial \sigma}{\partial u_1} + \beta_2^2 \dfrac{\partial \sigma}{\partial u_2} \end{cases} \tag{57}$$

Whose coefficients β_i^j depend not only upon the metric tensor, but also on the curvature tensor (unlike the Christoffel symbols)[5,7]:

$$\begin{cases} \beta_1^1 = \dfrac{b_{11}g_{22} - b_{12}g_{12}}{g_{11}g_{22} - g_{12}^2} \\ \beta_1^2 = \dfrac{b_{12}g_{11} - b_{11}g_{12}}{g_{11}g_{22} - g_{12}^2} \\ \beta_2^1 = \dfrac{b_{12}g_{22} - b_{22}g_{12}}{g_{11}g_{22} - g_{12}^2} \\ \beta_2^2 = \dfrac{b_{22}g_{11} - b_{12}g_{12}}{g_{11}g_{22} - g_{12}^2} \end{cases} \tag{58}$$

Note that the derivative of the unit surface normal must be normal to it, because the length of the normal is constant. Thus the unit normal derivative is always tangent to the surface.

A twice continuously differentiable curve $\mathbf{c}(s)=\mathbf{c}(u_1(s),u_2(s))$ is a geodesic if and only if there exists a natural parameterisation of it such that[5,7]:

$$\begin{cases} \dfrac{\partial^2 u_1}{\partial s^2}+\Gamma_{11}^1\left(\dfrac{\partial u_1}{\partial s}\right)^2+2\Gamma_{12}^1\dfrac{\partial u_1}{\partial s}\dfrac{\partial u_2}{\partial s}+\Gamma_{22}^1\left(\dfrac{\partial u_2}{\partial s}\right)^2=0 \\ \dfrac{\partial^2 u_2}{\partial s^2}+\Gamma_{11}^2\left(\dfrac{\partial u_1}{\partial s}\right)^2+2\Gamma_{12}^2\dfrac{\partial u_1}{\partial s}\dfrac{\partial u_2}{\partial s}+\Gamma_{22}^2\left(\dfrac{\partial u_2}{\partial s}\right)^2=0 \end{cases} \tag{59}$$

2.10 Surface and Line Continuity

A *cusp* is a sharp point where the tangent vectors are not continuous. Therefore, it cannot lie in a straight-line like a ruling, and any surface with a cusp is at least locally non-developable (besides being not regular).

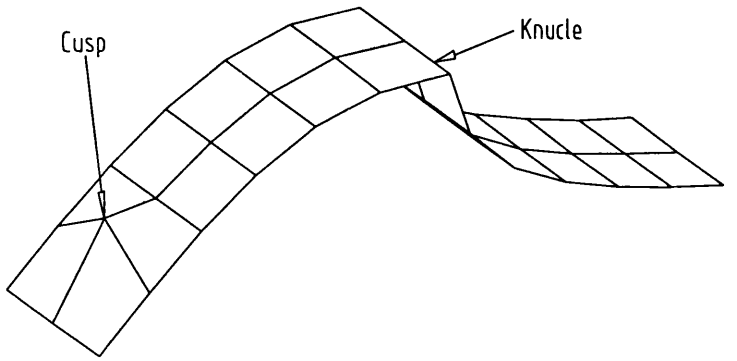


Fig. 8 - A cusp and a knuckle.

A *knuckle* is a ridge or a sharp curve lying in a surface, which therefore is not regular, so the results previously stated cannot be applied to it. For a surface containing a knuckle, to be developable it is necessary that the knuckle is a ruling and the tangency line for two unique planes, each tangent to each side of the surface along the knuckle.

G^n continuity means geometric continuity of n-th order, or equivalently, that the surface co-ordinates have continuous n-th derivatives on an orthogonal base of true length parameters. Some practical consequences of this are:

if n=0, then only point continuity are guaranteed;

if n=1, then the geometric surface tangents are continuous;

if n=2, then the geometric surface curvatures are continuous.

This is defined by opposition to C^n continuity, were the differentiation parameters could be any, thus not requiring geometric continuity, but only the mathematical continuity of the underlying surface expressions.

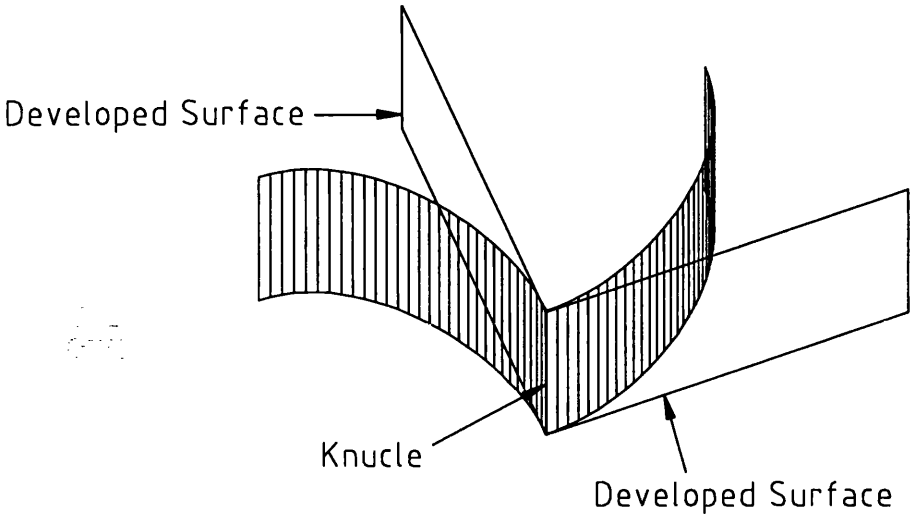


Fig. 9 - The development of a very particular case of a surface with a knuckle.

Chapter 3 - Plate Development and Developable Surfaces

At the outset of iron shipbuilding, in the first half of the 19th century, the traditional wooden planking procedures became obsolete, given the enormous differences in the aspect ratios, elastic modulus, stowage conditions and price of the materials.

For almost a century, until the First World War, the available workforce was continuously expanding in numbers, and its cost wasn't the issue it is today. Full-scale lofting was the only known way to build ships, with all its requirements for space, time and human resources.

Naturally, the technologies for sail vessel construction were not abruptly abandoned. The ancient iron riveting and copper planking technologies suffered gradual improvement and adaptation to the massive metal constructions required by ironclad construction, mostly in the British shipyards.

The First World War welding technologies led to gradual development between the wars, culminating in the dominance of welded fabrication by the early fifties. Before welding, it was common to use full-scale mock-ups and templates, despite the excess material in the overlapped boundaries of riveted plates. Welding further reinforced the requirements for full-scale mock-ups and templates, since it removed the need for plates to overlap, allowing economies in weight and material bill, but stressed the precision requirements of plate cutting to a level never experienced before.

After World War II, labour costs started to climb continuously, and this plate production process became increasingly more expensive, because of the enormous amount of work content in it.

The reaction came mostly from Germany, where in the fifties an enterprise started the marketing of a 1/10-scale electrostatic jig, for analogue plate development purposes, while other enterprises introduced lofting processes based on the optical projection of 1/10 drawings to its full-scale. In addition, automatic cutting machinery became available, which also used 1/10-scale drawings for optical reading. These 1/10-scale procedures did not discarded completely the need for full-scale mock-ups and templates, but required new skills to maintain accuracy.

Incidentally, the immense room for the full-scale lofting became welcome for other activities more vital for the shipyard operation: the increase in ship size and complexity stressed the need for space inside the yard, and the real estate evolution in most yard neighbourhoods restricted the possibilities of expansion outside the facilities perimeter.

In the fifties the computer was introduced to the shipyard environment. Initially the computer was mostly intended for the project design phase and administration, but gradually it was also perceived as the central tool for all design room procedures, including hull lofting. In the late sixties even the shop was using computer technology, used for the control of cutting machinery. The continuous stream of improvements brought to the yard by the computer technologies (which we are still witnessing in full vigour) was a true technological revolution. In fact, it enabled the return to complete full-scale lofting, by modelling the ship system in a broader and more accurate way, by improving operational control, and decreasing dramatically both the design and construction time. Computer-Aided Design (CAD), Computer-Aided Lofting (CAL), and Computer-Aided Manufacturing (CAM) are the technologies that have developed from this revolution, exploiting the opportunities of shared databases.

To accurately loft the hull plates, reliable plate development procedures (CAL), and practical methodologies for the design of developable surfaces (CAD), had to be created and introduced into the process. This Chapter reviews the bibliography on plate development and developable surface design methods that contributed to this evolution.

3.1 Graphical Plate Development

In addition to full-scale moulds, and the 1/10-scale electrostatic development jig, there were graphical methods for plate development, of which three were described by Branco[13]. Among these, the so called "French Method", is in fact a particular case of the "geodesic line method", so it will not be covered here, for reasons of conciseness.

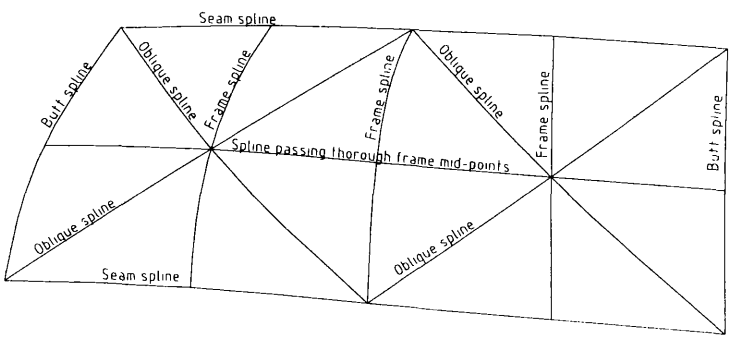


Fig. 10 - The splines fitting onto the frame body plan, as in the splines method.

The first method described by Branco, here called "the splines method", makes use of splines for the mapping of the hull curves onto the plane. It exploits the fact that a development is an isometric mapping[5,6], so any surface curve would have its arc length unchanged.

The method starts at the hull's frame body plan, where splines are used to measure the projected arc length of the following curves (see Fig. 10):

- the four plate boundaries (two seams and two butts)
- the centre line of the plate, which passes through the frames middle points
- the two diagonals of every inter-frame space, intersecting each other at the centre line of the plate.

For any of these curves, there exists a curved triangle, cylindrically shaped, of which the three sides are (see Fig. 11):

- the curve itself, of unknown arc length
- the projected length of the curve onto the vertical plane of the rear frame, is measured by the fitted spline
- the straight-line connecting the ends of the two former lines, whose length is the distance between frame planes, being orthogonal to the projection of the curve.

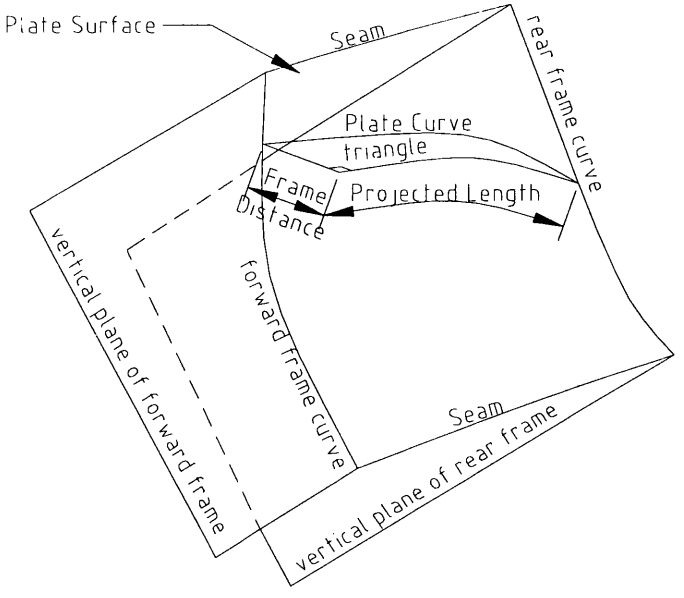


Fig. 11 - The triangle relating the hull curve with its projection onto the frame plane.

Since this cylindrical triangle is a right-angled one, Pythagoras' theorem is enough to obtain the arc length of the curve lying in the plate surface.

Knowing the arc lengths of all the plate curves, it is possible to compose the development of the previous spline frame. Each curve is represented by a spline of the same length, where all intersection points are marked. The spline assembly is done by bounding every two intersecting splines with nails. The resulting spline mesh is a scale model of the developed plate.

The corrections accounting for the spline thickness are made at the end, because it is not the spline edge that fits the true line path, but its medium fibre.

Another of Branco's geometric procedure is called "the geodesic line method", since it starts with the drawing of a geodesic curve located in the plate as a central line. It is drawn in the frame body plan.

The procedure to draw the geodesic is to discretise it frame by frame, from the mid-point of the plate's mid-frame, towards each butt. The initial direction of the geodesic is orthogonal to that frame. These initial conditions are intended to minimise the possibility of the geodesic intersecting a seam instead of a butt.

To be a geodesic, the line must have its osculating plane always normal to the surface, which requires that the line have a main normal parallel to the surface normal everywhere.

The intersection between the osculating plane and the plate surface, is discretised as a straight segment, which approximates the geodesic segment between two consecutive frames. The accuracy of this approximation decreases with the increase of both the frame distance and the normal curvature of the surface, in the geodesic direction.

As an initial condition, the osculating plane is to be orthogonal to the frame plane. Therefore its intersection with the next frame is just a matter of extending the initial tangent in the drawing, until it reaches the next frame, since the osculating plane is of vertical projection, as it is called in descriptive geometry.

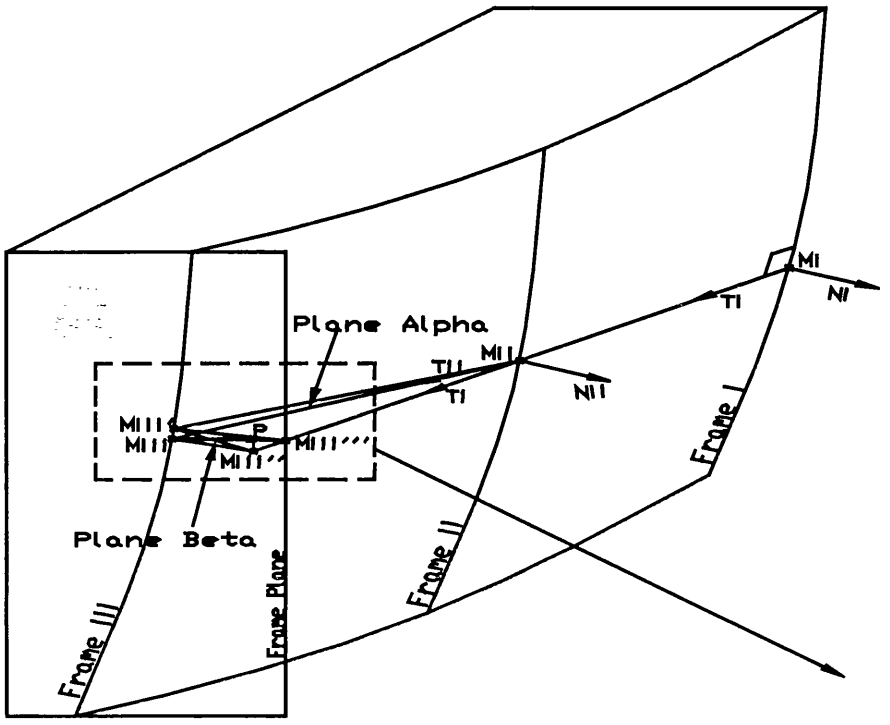
However, after this initial stage, the osculating plane can become an oblique one, due to the curvature of the surface, so the problem loses its initial simplicity.

For the intersection between the next frame and this oblique plane, Branco[13] uses a simplified geometric construction, designed to be as straightforward as possible for the loftsman.

The geodesic line and its tangent vector define a plane α, which is orthogonal to the frame plane. Fig. 12 depicts the initial geodesic segment $\mathbf{M}_i\mathbf{M}_{ii}$, between

frames *i* and *ii*. The plane β approximates the osculating plane to the geodesic segment at frame *ii*, and points $\mathbf{M}_{iii}'$ and $\mathbf{M}_{iii}$ are the intersection between frame *iii* and the planes α and β. The segment $\mathbf{M}_{iii}''\mathbf{M}_{ii}$ is the intersection between α and β.

If the surface is plane, $\mathbf{M}_{iii}$ would be the intersection between α and the frame *iii*. However, the curvature of the surface makes the geodesic rotate about $\mathbf{M}_{ii}$, thus the actual intersection is $\mathbf{M}_{iii}'$.



(Amplified in Fig. 14)

Fig. 12 - The projective plane α defined by the geodesic tangent, and the oblique osculating plane β.

To obtain $\mathbf{M}_{iii}''$ it is enough to draw from $\mathbf{M}_{iii}'$ the normal $\mathbf{n}_i$ until it reaches the extension of the segment that goes from $\mathbf{M}_i$ to $\mathbf{M}_{ii}$.

To this stage everything is accurate, the only approximations being hand drawing errors, say 1.5 millimetres[13], amplified by the scale factor to 15 millimetres (when drawing at 1/10-scale).

The triangle made by the distance between frames, and by the intersection line of the normal plane with the shell, is shown in its true length in Fig. 13.

Finally, to determine the real intersection between the osculating plane and frame *iii*, Branco assumes that the normal $\mathbf{n}_{iii}$ is approximately parallel to $\mathbf{n}_{ii}$. Therefore, drawing $\mathbf{n}_{ii}$ from $\mathbf{M}_{iii}''$ would intercept the frame *iii* at $\mathbf{M}_{iii}$. Fig.

14 depicts the three dimensional construction of $\mathbf{M}_{iii}$. The graphic (two dimensional) construction of $\mathbf{M}_{iii}$ is shown in Fig. 15.

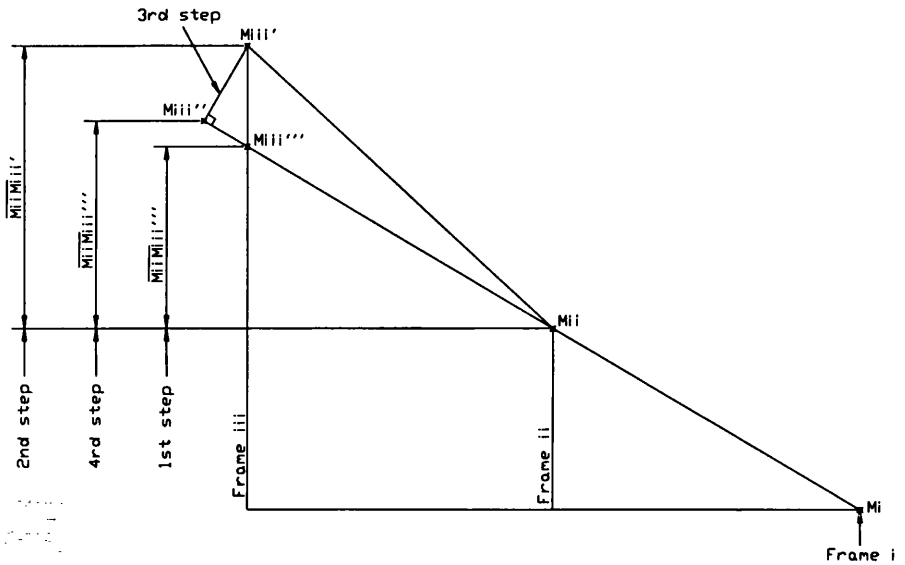


Fig. 13 - The projective plane defined by the geodesic tangent and the oblique osculating plane, in its true lengths. Note the sequence by which the geometric construction is drawn.

Note that Branco further simplifies the method, substituting $\mathbf{n}_{ii}$ by the normal to the frame *ii* at **P** (instead of at $\mathbf{M}_{ii}$), and therefore neglecting the frame curvature between **P** and $\mathbf{M}_{ii}$.

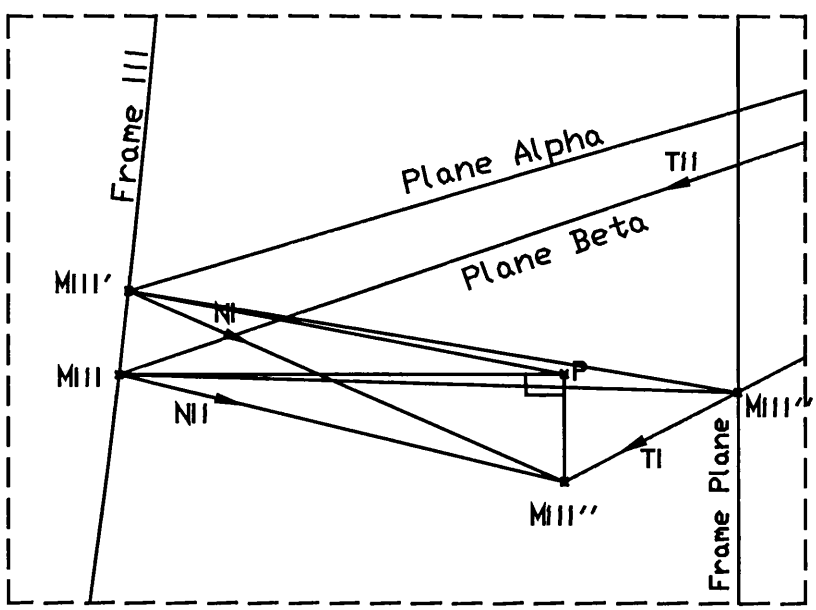


Fig. 14 - The point P, from which the normal $\mathbf{n}_{ii}$ is traced onto the frame *iii* (amplified from Fig.12).

After repeating this process in both directions, the geodesic is fully traced. To develop it, the true length of every geodesic segment is computed or graphically measured over its true length triangle, made by the vertical projection of the segment and the distance between frames. The accumulation of all these lengths gives the total length of the geodesic, which is developed as the straight-line obtained by the repeated application of Pythagoras' theorem, to each true length triangle along the line.

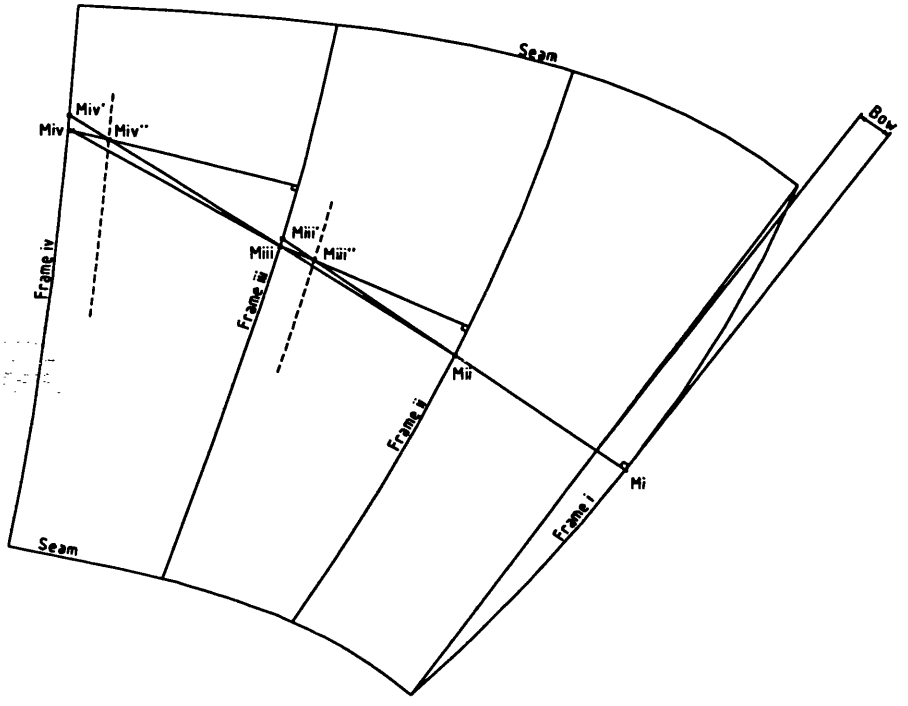


Fig. 15 - The drawing of the geodesic in the frame body plan, and the definition of bow_i.

The true lengths of the seams are also measured, segment by segment, and two splines are marked accordingly. Since the frame plane is projective, the lengths of the frames can be directly measured and marked on a spline, both from the geodesic up to the upper seam, and from the geodesic down to the lower seam. A single spline is enough for all the frames and both butts.

Relating to the frame bow, as defined in Fig. 16, the spline deflection at frame *i* (see Fig. 15) is computed as:

$$\text{spline deflection} = \frac{\text{bow}_i \times \overline{\mathbf{M}_{ii}\mathbf{M}_i}}{\sqrt{(\text{frame distance})^2 \times (\overline{\mathbf{M}_{ii}\mathbf{M}_i})^2}} \tag{60}$$

The final development procedure is the scale drawing of the developed plate. This is done by assembling both seam splines in a 1/10-scale drawing, following this steps:

1. drawing a straight-line as the development of the geodesic, with marks at the frame intersections

2. drawing a straight-line **q** orthogonal to the geodesic development, at a point distant from frame *i* by the spline deflexion, to the side that the frame bows

3. the central frame *i* is developed by placing the mid-point of the frame spline orthogonal to the geodesic rule, at its intersection with the frame *i*, and bending both sides of the spline until they reach the line **q** at the marked frame lengths

4. each spline with the seam marks is placed with its frame *i* at the **q** line, at the corresponding point, previously obtained

5. the frame spline is placed at the next frame, for one of the sides, and bent as in the case of frame *i*, but to meet the splines representing the seams, which must also bend until they meet at the proper markings

6. this last couple of steps is repeated for both sides, until the whole plate is developed.

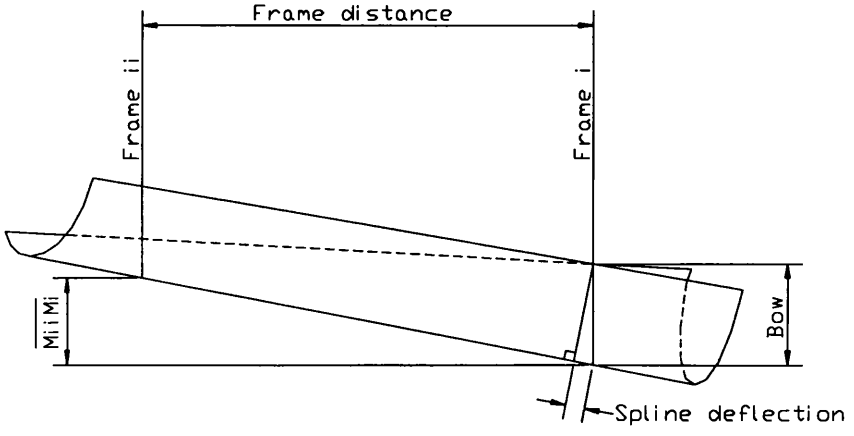


Fig. 16 - The spline deflection at the central frame, related to the bow.

The geodesic method is much more complex than the splines method, but Branco reported it as being more accurate. For a loftsman with a solid background in Descriptive Geometry, the geodesic tracing can be done by the Monge classical method for intercepting a line (the frame line) with a plane (the osculating plane)[14,15]. Doing that, one must use one of the two other projections of the shell, such as the lines plan. In this way, there are no methodical approximations, so it is more accurate. Nevertheless, it is not often that the loftsman has the education required by this procedure. Therefore, Branco's simplifications are in line with the actual workforce shortcomings.

Both methods are not easy to program, mainly because of the final spline assembly. This is mathematically modelled by a set of differential equations, expressing the boundary conditions of the curves and their invariant lengths.

3.2 The First Attempts at Developable Surface Design

In 1963 Barnaby[9] published at research work on developable surfaces and its applications on completely developable hulls. Developable surfaces were defined as the ones in which it is possible at any point to draw a straight-line in some direction, which extends to the boundaries of the shape. This is not correct if applied to general shapes, like the one in Fig. 17, which is not developable because of the constraint of the plane elements on the boundaries of the conical shape. For the generality of Barnaby's definition, it is necessary to state that those straight-lines should not cross each other, unless inside the contour of plane regions.

Barnaby's method for designing an entirely developable hull was based on an arbitrary curve called the directrix, chosen by the designer. At arbitrary points on this curve, the designer passes generators partly contained in the hull surface. Every two consecutive generators must intercept in a point, which is the apex of the cone element bounded by those generators and the directrix arc between them. In this way, a series of continuous cone elements defines the hull surface. To define cylinder elements, the designer must not intercept the consecutive generators, but make them parallel to each other.

Illustration is provided for a planing hull[9]. Planing hulls are quite convenient for the method, as the chine line can be taken as the directrix. Then, generators are passed through the chine, not only in the direction of the keel line, producing the conical bottom, but also in the direction of the deck line, producing the cylindrical side.

A drafting procedure is detailed, in which the initial data are the sheer line, the profile of the chine and its half breadth amidships. The slope of the cylindrical sides can be adjusted but the bottom is made of a single cone element, giving only the liberty to choose its apex location, which determines the keel line shape.

Barnaby points out the necessity to plastically shape all the plate, avoiding the residual stresses made in nearly developable plates, forced in position by elastic deformation. Obviously, the same can be said for any plate inaccurately shaped, developable or not. In fact, the advantage of an entirely developable hull is the economy of time, manpower and money obtained by the easily

shaped hull plates. Methods for developing the surfaces obtained are out of the scope of this short article.

Having the same the form coefficients, Barnaby expected a negligible penalty in hydrodynamic efficiency, relatively to classical hulls. He mentions a comparative study in an experimental tank that reported the same resistance for tugs. However, it should be noted that to get the same hydrodynamic efficiency, not only the form coefficients but also the hydrodynamic ones should be the same, not to mention other issues like particular hull differences in way of the propeller.

Hatch[16] researched mostly the applications of entirely developable hulls, called conic hulls. He pointed out several advantages of this type of design, such as the lack of skilled workmanship in many places where ships have to be built, savings in man-hours, earlier release of the expensive construction berths and reduced delivery time for the final product.

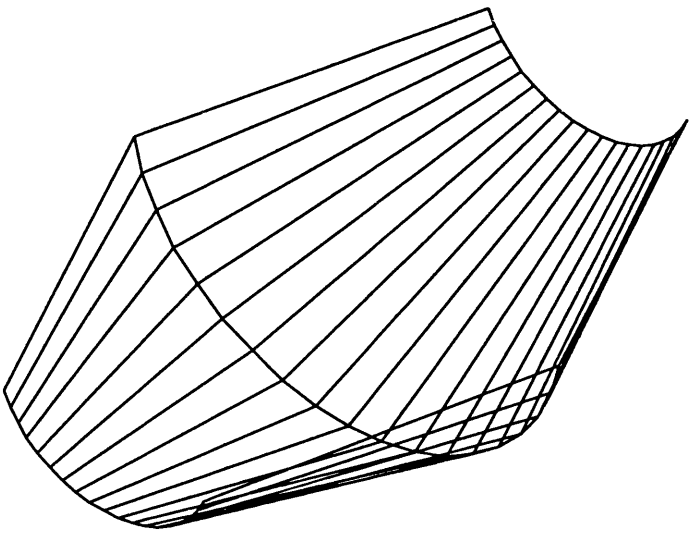


Fig. 17 - A non-developable surface made of developable elements.

Applications for this type of design, at the time (1963), were seen as "do-it-yourself" leisure boats, launches, tugs, barges, fishing vessels, passenger ferries, small coasters, and other mostly small craft. A great number of ships and small boats were already been built with this type of design, but none above 260 feet long, since increasing the ship size would decrease the economical savings relative to the total cost.

There was also a technical size constraint. Designers were using the conical surface design method presented by Barnaby, which requires the drawing of the apexes in the hull drawing. Since the apex tends to be far away for a small curvature cone, the size of the drawing boards becomes very constraining.

In principle, hull forms can be produced as close to conventionally round hulls as necessary, using multi-chines and several cone elements, possibly combined with cylindrical ones, to substitute for a soft bilge and a steep rise of floor. However, the hull complexity must be acceptable for the vessel being designed. Large tanker designs with conic hull weren't treated at the time, but Hatch pointed out that their typical hull shape was ripe for exploitation, because of the large amount of almost flat surfaces, and also the small bilge radius and small rise of floor.

Kilgore[17] studied applications on fishing boats of metallic developable hulls, in relation to the substitution for wooden hulls with traditional shapes.

If it exists, a developable surface can be found between two given hull lines, say line **a** and line **b**, by tracing the rulings between them. To estimate the ruling at a point **P** in **a**, the tangent vector to **a**, termed $\mathbf{t}_a$, is drawn at **P**, and then two straight-lines are drawn onto **b** (see Fig. 18). Since they intercept each other, they define a plane, named τ.

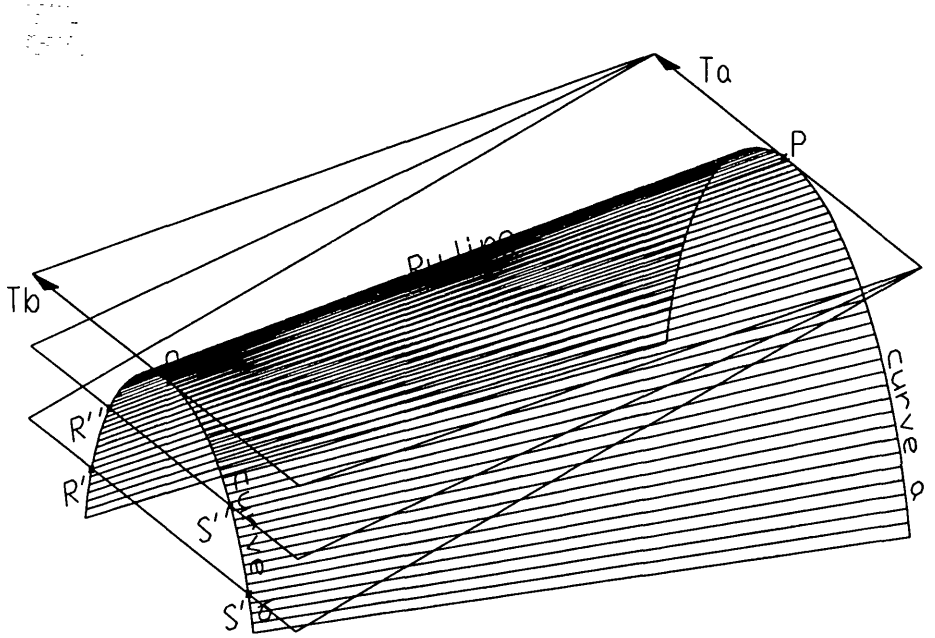


Fig. 18 - Plane α can be rotated around tangent $\mathbf{t}_a$ until it become tangent to the surface, and then **R** and **S** become the same point at **Q**.

R and **S** are the end points of those straight segments at **b**. This plane intersects the hull surface along the straight-lines from **P** to **R** and **S**.

Imagine that the plane α rotates around $\mathbf{t}_a$ until **R** and **S** become coincident at **Q**. Then α would be tangent to the surface, and the straight segment from **P** to **R≡S**, would be the tangency line, which is the ruling between lines **a** and **b** at **P**.

Kilgore takes the middle point of the arc from **R** to **S** as an approximation to **Q**, thus simplifying very much this procedure for drafting applications, so it can easily be repeated at points in **a** other than **P**, providing for as many rulings as necessary.

The paper makes no mention whatsoever to development procedures, and gives only simple illustrations on the developable surface fitting procedures. Guidance in the surface developability is also not provided, but it would be enough to state that rulings should not intersect each other for the surface to be developable [5]. In such a case, the designer should try to change the lines around the ends of the crossing rulings, or otherwise give up on the development of such surface regions.

Precise shape control of hull lines is required to produce a hull design that is hydrodynamically efficient and has hydrostatic properties close enough to the design figures. For conic hulls, developable surfaces are matched until hull lines are acceptable, so the shape of lines is only indirectly controlled (by the fitted surfaces), making the procedure error prone and a quite tedious one.

Applying Barnaby's method to a chine planing boat produces the profile centre-line, but Kilgore uses this line as initial data in the definition of the hull region between it and the chine. Kilgore's method for fitting developable surfaces largely surpasses the functionality of the conic methods, since it uses not only cone and cylinder segments, but also convolute surfaces, enabling the designer to cope with a much wider range of problems. This is achieved without having to care about the particular type of surface being used, or to try for the location of apexes, which is not the case with conic hulls. The simplicity and power of this method made it very popular[18].

3.3 The Outset of Computer Algorithms

Ferris[8] conceived a standard series of developable surfaces, covering not only surface fitting of the main hull lines, but also the development of these surfaces. In fact, the considerable amount of computing required by this methodology strongly suggests a computer implementation.

Regarding the fact that every developable surface is the envelope of a one-parameter family of planes, he chose a surface formulation with all coefficients of second degree in only one parameter, except for the dependent co-ordinate, for which the coefficient is of zero degree:

$$y = J_0\,[(J_1u^2+J_2u+J_3)x+(J_4u^2+J_5u+J_6)z+J_7u^2+J_8u+J_9] \qquad (61)$$

In Eqn.61 the half-breadth is the dependent co-ordinate, for which $-1/J_0$ is the coefficient usually found in the normal equation of the plane. This is the common expression for hull sides. For the bottom, the dependent co-ordinate is z. Ferris doesn't expressly state it, but for each hull region, he chooses as dependent co-ordinate the one with smaller variations, preserving the numerical conditioning of the expressions. Given some pair (x, z), the dependent co-ordinate y on the surface is the extreme value among the ones pertaining to the plane family.

Differentiating Eqn.61 and computing the root, the following result is obtained:

$$u = \frac{J_2 x + J_5 z + J_8}{2(J_1 x + J_4 z + J_7)} \tag{62}$$

Ferris didn't explicitly use this result to eliminate u in Eqn.61, but the generic equation he actually used for the developable surface was the result of this substitution.

Variations in the nine coefficients are illustrated, but besides J_0 and J_9, which are proportional to y, no simple relationship was shown between the coefficients and the shape, so it is not clear that working with these coefficients is practically viable. As the plane coefficients form a second-degree polynomial on the parameter u, the apex locus reduces to a point, since the second derivative of the y expression don't depend on the parameter. Therefore, this second-degree formulation always produces simple cones.

To fit any particular hull shape, Ferris proposed the use of fields of coefficients, which multiply the developable half-breadths. This ensures good fairness but deviates from the original developable condition. The reason for this was the admission that the are regions which cannot be made developable, as in the case of the concave bow flare. Although this region cannot be fitted with a single surface patch of second order, since it is not a cone, it can be fitted by subdividing into several developable elements, if care is taken of the fairness across element boundaries. Another way to deal with these particular hull shapes as well as cylinders could be to increase the complexity of the coefficients, but this was not explored in the paper.

The easiest way to derive hulls is to scale from a basic one, since scale factors do not affect the surface developability.

Ferris also proposed to establish a system to compute offsets of developable surfaces for use by naval architects. There appear to be no further reports on this initiative.

The resulting plate development procedure can be accomplished with relative ease, either computing or graphically, since all plates are simple conic

surfaces. Each of these have the apex at the intersection of any two rulings, and any plate point can be defined by the distance from the apex and the angle to some reference ruling, in a polar co-ordinate system.

The first report on a computer-aided method for fitting a developable surface is due to Nolan[30]. The method is based on the rulings computation, similar to the graphical method presented by Kilgore[17].

As the two curves bounding the developable surface must be fair to ensure fairness of the surface, he chose the Theilheimer spline to interpolate these curves, since its deformation follows a physical spline. These curves are usually the chines, knuckle lines, sheer line and the keel line, without further subdivision of the hull.

The method calculates the developable surface between the boundary curves, which is unique, if existing at all. The rulings are computed without restrictions on direction, so any developable surface can be fitted. They are computed by searching for parallel surface normals, one at each boundary curve, at both ends of a prospective ruling.

After computing the surface normal $\mathbf{N}_a$ at a particular point $\mathbf{P}_a$ in the boundary curve **a**, the other end of the ruling is searched in the boundary **b**, by taking in it some point $\mathbf{P}_b$, and computing there the normal $\mathbf{N}_b$. For the normals to be parallel it is enough that:

$$f(\mathbf{P}_b) = \| \mathbf{N}_a \times \mathbf{N}_b \| = 0 \tag{63}$$

When $\mathbf{P}_b$ is not the very root of Eqn.63, f is a positive non-zero quantity. Nolan studied the function f and found that its derivative about the $\mathbf{P}_b$ location is zero at the root and at other unpredictable points.

To calculate the roots of Eqn.63 it is necessary to use some numerical method. Nolan recommends the method of Newton for this purpose, arguing rightfully that the selected method should use all available information. But the Newton's method require the knowledge of the derivative of f at every interpolated point. This was possible by the analytic derivation of the Theilheimer polynomials. Instead, Nolan chose to calculate the derivatives numerically, incurring in efficiency and accuracy penalties. Moreover, the efficiency of the Newton's method is very much vulnerable to small derivatives, since any minor deviation on one tangent computation can throw the next iteration far away from the root.

Therefore, even if deriving the Theilheimer polynomials, it is best to use a method other than first degree, which better reproduces the function shape known to be far from linear.

The starting location for $\mathbf{P}_b$ is usually the last ruling intersection with the line **b**, but when the rulings tend to be parallel to **b**, the search method revealed itself unreliable. In these cases, Nolan opted for widening the search scope exponentially, until the acceptable result or an abort condition is achieved. He did not use all surface boundaries, only the seams, so the algorithm can fail with developable surfaces if the rulings tend to be parallel to the seams, and cross the surface butts.

Butts should be processed in a practical application, as the seams are. Another way is to extend the seams by extrapolating the Theilheimer polynomials, and then intercept the resulting rulings with the butts.

The extrapolation would lead to amplified errors on the computed points and vectors, and the computing is further complicated, so it is better to search for the rulings directly at the butts. If the method doesn't show convergence than it is assumed that the surface is non-developable (in fact it is non-ruled).

Nolan doesn't explore the possibility of the surface being ruled but non-developable, which should make the rulings cross somewhere, clearly exposing the problem and its location. If any problem arises, it is advised to change one of the curves until the procedure works.

At the time (1971) computing costs were remarkably negligible, at least for a test made on a small chinned hull.

The development of the resulting surface was not explored in Nolan's paper, and besides Ferris superficial description[8], no other report on development methods is known until 1972, with the Barkley's MSc thesis, about computer-aided surface development[19].

Barkley's geometric definition of the plate is based on surface patches of triangular shape, contoured by geodesics. Thus, any type of plate boundaries can be modelled, like the triangular plates, which was not the case with the quadrilateral patch and others.

The development of a geodesic is a straight-line; therefore, the surface triangular patch is developed onto a plane triangle. This triangle can be drawn, just by computing the length of each geodesic side.

The main issue in this method is to trace the geodesics between the given points, lying in the plate surface. To do that, Barkley started choosing a convenient co-ordinate system (u_1, u_2), where u_1 is the frame arc length, and u_2 is the waterline arc length, with origin at the initial point of the considered line. For any surface curve, the parameter is also its length s.

Therefore, for any curve lying in the surface, the following differential equation holds, where g is the metric tensor:

$$g_{11}(\frac{\mathrm{d}u_1}{\mathrm{d}s})^2 + 2g_{12}\frac{\mathrm{d}u_1}{\mathrm{d}s}\frac{\mathrm{d}u_2}{\mathrm{d}s} + g_{22}(\frac{\mathrm{d}u_2}{\mathrm{d}s})^2 = 1 \tag{64}$$

If the curve is a geodesic, then its arc length is minimal. From the minimisation of the arc length results:

$$g_{12}\frac{\mathrm{d}^2u_1}{\mathrm{d}s^2} + g_{22}\frac{\mathrm{d}^2u_2}{\mathrm{d}s^2} + \left(g_{12}{}^{u_1} - g_{11}{}^{0.5u_2} - \frac{g_{22}{}^{2u_1}}{2g_{22}{}^{u_2}}\right)(\frac{\mathrm{d}u_1}{\mathrm{d}s})^2 = 0 \tag{65}$$

To solve this equation for $u_1=u_1(s)$ and $u_2=u_2(s)$, the ends of the geodesic are given and finite differences can be computed, establishing the necessary boundary conditions. Once the differential equations are solved, inverting the relation $s \rightarrow (u_1, u_2)$ at the end gives the geodesic arc length by the final value of s, named sf.

Since the aim is to find the geodesic length sf, and not the geodesic curve equations $\{u_1(s), u_2(s)\}$, one can eliminate the redundant equation, and after several simplifications get:

$$s_f = \sqrt{\frac{g_{22}u_1{}^2}{g_{11}g_{22} - g_{12}{}^2} + \frac{(g_{12}u_1 + g_{22}u_2)^2}{g_{22}}}\Bigg|_{s=s_f} \tag{66}$$

Eqn.66 gives very good results when the geodesic is contained in a sphere, producing errors under 0.4% of the arc of the great circle between both ends.

Surfaces that are more realistic were not tested, but for developable surfaces, the method should give good results, including the final development.

Barkley supposed that it could work acceptably on surfaces with negligible second principal curvature (thus non-developable). However, reservations are to be taken about that, because the algorithm for aliasing the plane triangles during its assembly is simply non-existent, and because of that some unpredictability is to be expected.

The ICCAS 1973' conference[20,21,22,23] gathered most of the main software developers at the time, covering Computer-Aided Lofting and shipyard Computer-Aided Design and Manufacturing.

Hurst[20], representing the B.S.R.A., reported an initiative to gather information within the Member Firms, about the required production documents for the developed plates. This produced the specification for the development modules of the BRITSHIPS package, and included the representation of the building frames and other information not specified. Results should be

produced in paper drawings and control tapes for the numerically controlled machines.

At the time, the Member Firms were using development methods based either in triangulation or in the geodesic line[13]. The BRITSHIPS team opted for a development method that splits the plate into small segments assumed to be perfectly developable. These segments are either quadrilateral or triangular.

The formers are developed by the calculation of the arc lengths of the diagonal geodesics, and the angle between them.

For the triangular segments, only two geodesic sides and its angle are computed. Naturally, the development of the unfolded segments is done using straight-lines instead of the curved geodesics. The particulars of the inter-segment assembly process are not detailed.

Additional value is given to the user, by providing him with the language "2C,L", relatively easy to use, since it was purposely built for the shell lines specification, including the seams and butts.

Magnusson, presenting the VIKING package[21], reported on the traditional loftsman methods used for the computer-aided development of plates. The adapted manual technique, named the "angle line", was neither described nor referenced. The information for production includes both the template drawings, and in the drawing of the developed plate, the representation of the welding lines between the plate and the inner structural components.

Belda et al[22], argue that geodesic seams should reduce the scrap material resulting from the plate cutting, since the plate boundaries should be almost straight, and therefore it is easier to use the remains in smaller parts. Thus, the FORAN system uses geodesic seams for this purpose.

Instead of programming a traditional method, the FORAN team opted for fitting a developable surface onto the original frames. This surface spanned an entire panel. If the Gaussian curvature, integrated over the panel, exceeds some amount, typical of a reasonable plate, then the panel is to be subdivided until each plate is acceptable.

For the specification of the seams and butts, the FORAN system develops the entire panel, where the user then strakes the plates, specifying the seams as straight-lines. Besides the developed seams and butts, the FORAN system provides information about thickness, weights, quality, templates and a "developability index", which is not explained.

The Aster system, presented by Juranek et all[23], is only briefly described. It includes a development system based on geodesics and rectangular plate elements, but further information about it is not provided.

3.4 Computer-Aided Design Becomes Mainstream

The generalisation of computing devices like hand calculators, personal computers and powerful graphical workstations gave way for increased research and application development in many practical and theoretical fields, including ship hull design and plate development.

The author didn't succeed in finding published information on development methods or developable surfaces, since the 1973 world recession until 1980. In this year Grandpierre's published his book[24] on small boat design and construction, included in a series about leisure and fishing boat construction.

Despite the rather simple style of the book, it provides comprehensive and serious information on the covered topics. Grandpierre points out economies about 20-40% in the chinned hull construction, when compared with freely curved ones. He also acknowledges the absolute necessity of computer or electronic calculator assistance in the design process, given the lengthy and repetitive calculations required by this type of hull.

Conical, cylindrical and Kilgore's methods for fitting developable surfaces are presented clearly and thoroughly. The proposed development admits a plane approximating to the surface between every two consecutive rulings, if they are close enough. In this plane, the boundary chine lines form a quadrilateral figure, which diagonal is easy to compute. Therefore, the graphical development is just a mater of drawing sequentially every such plane element.

Being this book aimed at the unprepared reader, a systematic method for multi-chine design is detailed, and several applications on sailboats are thoroughly demonstrated. In these applications, the complex multi-chine hulls are derived from equivalent ones of traditional shape, with some admitted loss in the hydrodynamic efficiency.

Clements[25], maybe because of its background as a mathematician, published for the first time an algorithm for fitting truly developable surfaces, not just ruled surfaces. All the covered algorithms were designed for the computer.

The paper explored briefly the necessity of the proper fairing of the boundary chines, without which the surface could have non-developable spots. Weighted Theilheimer polynomials are fitted, minimising both the deviations from the surface data points and the curvature integral along the curve. The surface fitting is done by searching rulings with Barkley's algorithm[19], but controlling for the possible existence of rulings overlapping at each other, in which case the developability is locally compromised.

For those cases, a multi-conic procedure is locally executed, imposing the developability of the surface, at the cost of further deviations on the original

data points. This is done imposing mathematically a convenient vertex, located at the intersection of the projections of two consecutive rulings, onto the tangent plane to one of them. For the surface development, the intersections of rulings are computed and used for unfolding each cone segment.

This paper is symptomatic about a fundamental change in the knowledge areas involving the lofting, being that directly associated to the leading technologies. The descriptive geometry dominated the manual drafting techniques for fitting developable surfaces and for its development, until the computer methods started being explored and the differential geometry took the same role.

In this evolution, Munchmeyer[26] published a systematic study on the applications of the differential geometry to the ship design. Besides a synthesis of the necessary mathematics, it reports on several topics, like:

- Surface fitting methods for the regions of the hull.
- Computing of the surface area.
- Forming of developable plates by applying the roll onto the second principal direction.
- The basis for the optimal design of seams and butts.

Clements took again the subject of surface development[26], now with an entirely new algorithm. This algorithm can only deal with properly developable surfaces.

Up to a point it seems very much like the geodesic method described by Branco[13], because it also uses a geodesic fitted longitudinally, as a reference line for the development procedure. This geodesic is traced as the solution of the Eqn.67, which is a differential equation on the geodesic **g**, where **n** is the normal to the surface and u is the curve parameter.

$$\left| \frac{\mathrm{d}\mathbf{g}}{\mathrm{d}u} \times \frac{\mathrm{d}^2\mathbf{g}}{\mathrm{d}u^2} . \mathbf{n} \right| = 0 \qquad (67)$$

This equation is a formulation of the principle that at a geodesic, the curvature vector is always aligned with the surface normal.

The rulings intersecting the geodesic are computed and the angle between them and the geodesic is stored, along with the rulings lengths upward and downward to the seams.

After developing the geodesic as a straight-line, the rulings are developed by its stored lengths and angles with the reference geodesic. Then, the seams are

drawn connecting the ends of the rulings. This method is numerically tested with a chine hull, giving negligible errors.

Possible improvements suggested by the author include a second geodesic used as a check line for verification of the overall accuracy of the applications.

Nørskov-Lauritsen[4], working for Burmeister & Wein Shipdesign Aps, reported on the status of developable surface fitting, in the current production systems.

At the time (1985), big size ships were already built with hydroconic hulls (a Burmeister & Wein's series of 20 Panamax ships), providing evidence of the hydrodynamic efficiency of this design. This shipyard started the utilisation of a computer system, as early as in 1960, for the replacement of the vast manual lofting work.

The first generation of the software required the manual input of the rulings to provide the complete definition of the developable shell surface.

By the time of this report, the second generation were already very automated, ranging from the processing of the classification drawings to the final offsets used for steel cutting and construction work, and implementing fully the concepts of entirely developable hulls.

The hull surface is initially defined as a mesh of lines, bounding each region of particular geometry. Over this lines it is necessary to specify the boundary conditions, like continuity of tangents or curvature. For simple and intuitive manipulation, these lines are internally stored and processed as Bezier curves. The surface domains of flatness, of developability and of non-developability, are defined, dividing between them the entirety of the shell. Boundaries of the plane regions are often specified as straight-lines, for the sake of smoothness in the transitions to the adjacent regions.

Often the non-developable regions are the transitions between developable ones, and thus should be relegated for the last stages of the surface design, because they are mainly a consequence of the particularities of the surrounding regions.

The developable surfaces are generally defined by specifying a directrix segmented in relation to a corresponding set of apexes. In this way, generalised cylinders and generalised cones are fitted with absolute accuracy, and tangent surfaces are discretised as a continuous set of elements of generalised cones.

The boundary lines not used in the surface definition are projected onto it, for the trimming of the surface ends. The regions of the resulting knuckle lines can be faired by a variety of methods, and in some cases like the bilge, the blending between surfaces can be specified to be developable. The

developability of the blended transition depends on the parallelism of the rulings of the adjacent surfaces.

The developing method uses a two-dimensional co-ordinate system, local to each surface element. It is placed at the apex of the surface element, and the co-ordinates are the angles between rulings and the distance along the each ruling. Because the apexes are accurately known and the surfaces are exactly developable, the mapping onto the plane is of almost absolute precision, allowing for ship methods for construction requiring very little adjustments on the erection site.

The introduction of the developable surface fitting in the integrated production system yielded about 20% gains in man-hours for the curved panels. Depending on the fullness of the ship, up to 95% of the hull surface can be developable, without loss in the hydrodynamic performances.

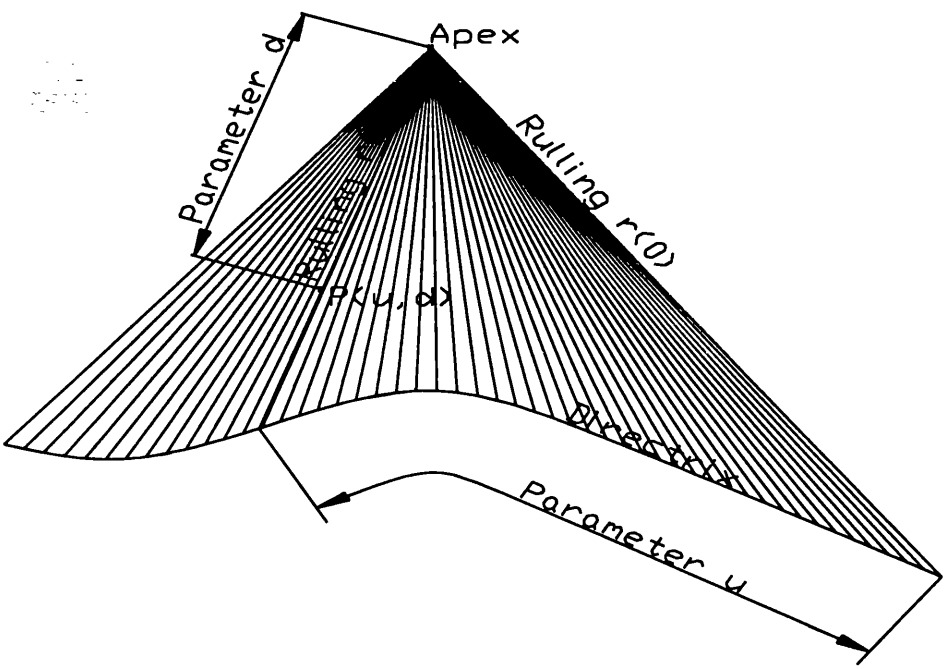


Fig. 19 - The polar co-ordinate system in the generalised cone.

Hansen[29], a mathematician, studied the numerical incidences of applications on generalised cones, chiefly in surface fitting, surface interpolation, surface development and the intersections between cones and cones and planes. This work was done at the Burmeister & Wein Shipdesign Aps.

The generalised cone is a ruled surface with directrix $\mathbf{c}=\mathbf{c}(u)$, were u is the curve parameter.

To know the position of a point **P** in the cone, it is enough to find the parameter u of the ruling that contains **P**, and the distance d from **P** to the apex **A** (see Fig.

19). Thus, the surface point **P** is defined by only two co-ordinates, implying the surface equation (which removes the third degree of freedom). However, this is clearly a polar co-ordinate system where u is a generalised angle, with origin at the ruling that passes at beginning of **c**.

The cone development is naturally defined in a polar co-ordinate system, so the development becomes just the translation of the three-dimensional polar co-ordinate system onto the conventional one. To do that it is enough to translate u to an angle α, in radians, which is simple, given the radian relation to the arc length:

$$\mathrm{d}\alpha = \frac{\mathrm{d}s}{d} = \frac{1}{d}\frac{\partial \mathbf{c}}{\partial u}\mathrm{d}u \tag{68}$$

Integrating $\mathrm{d}\alpha$, and noting that the distance from P to A is unchanged by the development, results the following relationship between the co-ordinates of P, before and after the development:

$$\begin{cases} d_{2D} = d_{3D} \\ \alpha = \int_{u_0}^{u_P} \frac{\sqrt{\left(\mathbf{c} \times \frac{\partial \mathbf{c}}{\partial u}\right)^2}}{\mathbf{c}^2}\mathrm{d}u \end{cases} \tag{69}$$

The expression for α is analysed for numerical implementation and an adaptive osculatory method is proposed, with differentiation done recursively. This rather evolved method tries to improve both the numerical accuracy and the computing efficiency.

Note that the expression for α is the basis not only for the development, but also for the interpolations and thus also for the surface fittings and intersections. Therefore, it is expected that the effort required by the design of this method pay off considerably.

Some pseudo-code is provided for programming the surface fitting, the surface interpolation and the surface intersection, based in elemental tasks, of which the evaluation of the α expression is the only one with significant complexity. It is desirable to have an overall accuracy of at least 7 decimal digits, or 1 millimetre per kilometre, because the apex is frequently located kilometres away of the surface.

Alternatively to the plate development, one can try to control the forming problem and, assuming a given flat plate, plastically bend it until the intended curved shape is obtained. In this line of research, Hardt et all[28] studied models

for plate rolling and punching, which can be matched with the geometric definition of the hull surface, and the plate thickness, to design the forming process.

These models can be used to simulate alternative production sequences and alternative production parameters (like the radius of the die or the rollers), leading to the optimisation of the forming procedures.

The bending model for die pressing starts with the machine geometry (die radius a), the punching penetration Y_p, and assuming the material as being of linear strain hardening, with elastic modulus E_e, plastic modulus E_p, and yield stress σ_y. The maximum deflection moment M_{max} is then matched to Y_p, by an iterative series of calculations. These integrate the curvature to find the deflection of the centre point.

Given M_{max} and the resulting linear moment distribution across the plate $M(s)$, the moment-curvature relationship can be applied to find the corresponding loaded curvature $K_l(s)$.

Finally, the unloaded curvature distribution, $K_u(s)$ is found by applying an elastic moment of equal and opposite magnitude to the original load, to account for the elastic spring-back, resulting:

$$K_u(s) = K_l(s) - \frac{M(s)}{EI(1-\nu^2)} \tag{70}$$

The model accounts for sequences of punches across the plate, and subtracts the machine penetration Y_p by the initial deflection produced by the previous punches. Residual stresses are computed and accounted for in the model, so the material behaviour is properly described. The results are the curvature, the deflection and the residual stress fields.

Sensitivity analysis of the curvature and of the plastic zone shape, show that the isotropy of the curvature field (the sphericality) can be improved by closely spacing shallow penetration punches, as expected.

In the rolling, instead of imposing a deflection as in the punching, the machine imposes a bending moment. However, the penetration of the centre roll, relatively to the supporting ones, is equivalent to the punch penetration Y_p. The other machine parameters relevant to the process are the roll spacing d, the centre roll radius a, and the plate displacement δ.

Rolling imposes a linear bending moment, which is zero at the supporting rolls (neglecting the plate weight), and is maximum at the centre roll, being that maximum proportional to the penetration and to the distance between rolls. When the plate is rolled the bending moment at each point changes in time, increasing from the first supporting roll until reaching the centre roll, and

then unloading linearly, without further plastic deformation. This produces a non-symmetric field of imposed curvatures $K_1(s)$.

For a desired plastic curvature it was demonstrated that the punch penetration should be almost linearly dependent to the plate thickness, so varying thickness across a plate would affect significantly the shape of the rolled plate.

The linking from the plate-forming model to the design of the surface and of the seams and buts are out of scope for this work. Nevertheless, it is a desirable feature, not only for the ship builder but also for the ship designer, since it allows improvements on the time and economic resources required by the overall project.

3.5 Computer-Aided Geometric Design of Developable Surfaces

Researching general Computer-Aided Geometric Design (CAGD), Gurunathan and Dande[9] described two development procedures, each one adapted to each formulation for ruled surfaces, as in Eqn.29 for a surface defined with a single directrix, and as in Eqn.30 for a surface defined with two directrices.

Both development procedures are based in the development of one directrix, by the integration of the Frenet-Serret equations (recall Eqn.17). The development preserves the arc length and geodesic curvature of every line lying in the surface. Therefore, knowing the accumulated arc length and the geodesic curvature along the directrix is to know the arc length and the curvature of its plane image.

The three Frenet-Serret differential equations express the relationship between the arc length, the curvature, and the torsion. Knowing only two of these functions, it is necessary to eliminate for the torsion, and reducing to a system of two equations, which is enough to determine the two unknown functions, the plane co-ordinates x and y. The resulting system of equations is (as in Eqn.38):

$$\begin{cases} \dfrac{\mathrm{d}^2 x}{\mathrm{d}s^2} + k_g(s)\dfrac{\mathrm{d}y}{\mathrm{d}s} = 0 \\ \dfrac{\mathrm{d}^2 y}{\mathrm{d}s^2} - k_g(s)\dfrac{\mathrm{d}x}{\mathrm{d}s} = 0 \end{cases} \tag{38}$$

The solution in $x=x(s)$ and $y=y(s)$ can be obtained numerically, defining the developed directrix.

In particular cases it is possible to get an analytical solution, depending on the complexity of the function of geodesic curvature, but this subject is not explored in the paper. One of such cases is when the directrix is a geodesic, becoming the solution a straight-line.

Applications to surface and curve polynomial formulations, which are common in computer graphics, could lead to interesting results. The development preserves the rulings lengths and angles with the directrix, so any point or line lying in the surface can be easily mapped onto the plane once the directrix is developed.

Exemplification is not provided in the paper, but applications in air conditioning ducts and other metal sheet structures were being done by the authors, at the time.

The development of thick plates was to be tried later, but reports are still not available.

The computer-aided design and development of hydroconic hulls is studied again by Akbarabad[1], which used a surface fitting process like Kilgore's[17]. The resulting surface is subdivided in conical patches to be developed one by one, and then fitted together to assemble the plate development.

This cone assembly can only give approximated results for tangent and cylindrical surfaces which is rather restrictive, compared with previous works like Clements'[26], which solved this problems already, without method approximations.

Weiss and Furtner[33] studied the problems of connecting two curves by a developable surface. The targeted application domain is the design of metallic sheet surfaces, like the transition between two tubes of different shape or alignment.

The surface fitting is done by a sequence of small four-sided plane elements, or by small triangles. Therefore, the development of the surface is simplified at the expense of the systematic error of underestimating the true lengths of the lines lying in the surface (see Fig. 20).

This technique requires very little work and technology to produce the fitted surface, and therefore it is desirable for the given application domain, which is not demanding about accuracy.

For higher accuracy, say 0.1% (about a centimetre for a line 10 meters long), and for the complex curvatures found in ship hulls, it is necessary to decrease the step of the discretisation, loosing computing efficiency.

It is conceivable to have an osculatory method for predicting the accurate development, based on this discretisation algorithm, but this is out of scope for this work.

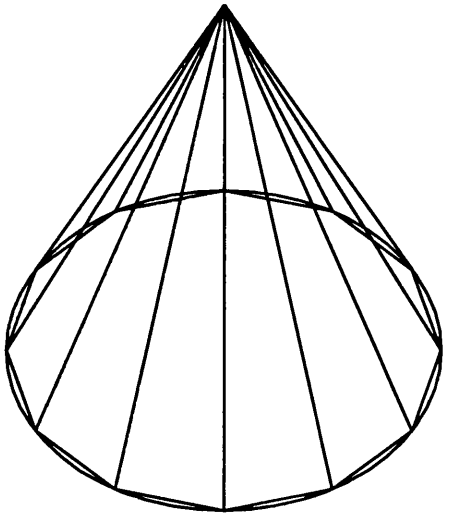


Fig. 20 - Cone discretisation as proposed by Weiss and Furtner.

The Fairline software system was developed by Letcher and Brown[34] for the fitting of spline surfaces. These surfaces can be constrained to be developable by using Kilgore's method. The development method is very similar to Gurunathan and Dande's.

In a paper presenting the Fairline system[34], the authors and the audience made several important remarks about the design of chine lines bounding a developable surface. The chines should be thoroughly faired. Than, plotting the abscissa for both chines in the same graphic, the points of maximum second derivative should be removed, because they could lead to non-developable spots. The intersecting rulings, showing a non-developable spot, are reported to occur more frequently at the ship ends. When this happen it is advised to rise the end of the upper chine.

They give no recommendations for non-developable spots lying in the middle of the surface, besides moving the nearest control points, one at a time, by trial and error. Accuracy and reliability considerations point to a minimum of 50-60 rulings defining each developable surface patch. Exemplification is provided only for a very simple chinned hull, but with an error under 0.1 millimetre.

Hamlin acknowledges the popularity of Kilgore's method for developable surface fitting in the Principles of Naval Architecture[18]. Only after 25 years of published works, is the subject of surface developability covered by this classical compendium on naval architecture, which says a lot about the lack of awareness for this issue among the community of ship designers. Nevertheless

it still doesn't give any help on plate development, maybe because it is more of a construction topic, rather then a design one.

Redont's research[11] is oriented towards generic computer aided design of developable surfaces, and is not a specialised hull surface application.

He explores the advantages of dualities between complex graphical entities and a simpler, more manageable, graphical ones, namely the duality between developable surfaces and its spherical indicatrix.

For the definition of the developable surface, he uses the spherical indicatrix of one geodesic line lying in it, and the rulings. The rulings are defined by the angles and the points at which they meet the geodesic line, plus the lengths measured from these points to both surface borders. The spherical indicatrix is discretised as a sequence of circular arcs on the unit sphere. Because of this, the surface is also discretised, but as a sequence of cones, as seen in the section 2.8 The Spherical Map.

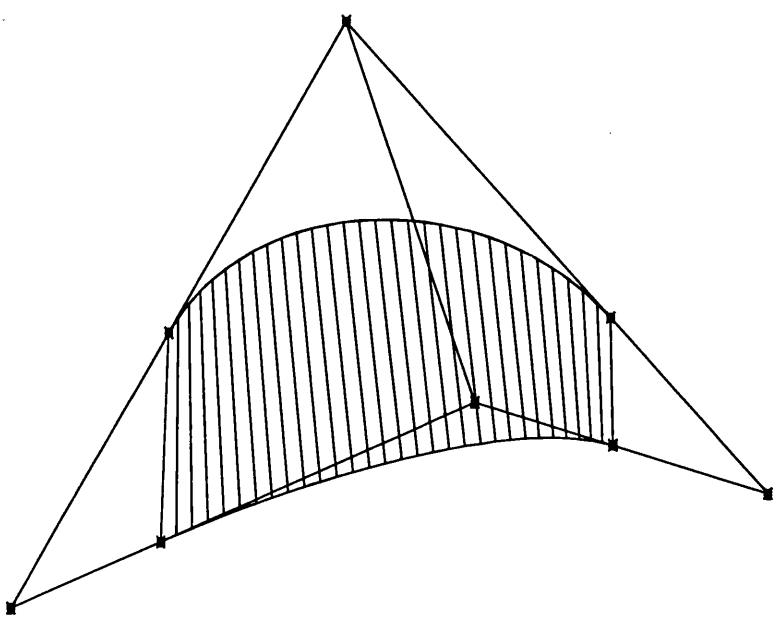


Fig. 21 - The control net for a developable rational Bézier surface, of degree [1×2]. Note that consecutive rows are coplanar.

This formulation allows the user to edit the resulting surface only by changing the indicatrix on the unit sphere, which being very powerful is not easily related with the final surface shape, for a common ship designer. For each indicatrix point there are a corresponding ruling in the surface, and thus a corresponding sheet of normals along that ruling, which are equipollent to the locating vector at the indicatrix point[5,6].

The designer can handle the surface orientation by changing the indicatrix path along the unit sphere, and thus changing the orientation of the surface normals. Imposing a cusp in the indicatrix, the designer produces a surface inflexion along the corresponding ruling. For the design of cylindrical shapes, it is enough to impose arcs of maximum circle along the spherical indicatrix.

The angles between the rulings and the geodesic, the lengths of the rulings and the geodesic arc lengths are known. Therefore, the development of the surface is just a mater of drawing the geodesic as an isometric straight-line, over which are marked the rulings intersections and angles. Consequently, the rulings can be mapped into the plane.

Aumann published on developable surface fitting [33]. He researched the criteria for establishing the developability of interpolating developable patches, when fitted onto two boundary curves, lying in planes parallel to each other. Both curves, used as the directrices of a ruled surface, are Bézier curves[36], being the primary one of the third degree, and the secondary of the fourth degree.

The criteria proposed is quite involved and not at all of intuitive use. They relate the location of the control vertices of both curves, the baricentre of the triangles made by them, and the slopes made by the lines connecting them. The continuity between adjacent patches, both of the tangent and of the curvature, also produces expressions almost impossible to use directly.

Ordinarily, Bézier surfaces are defined by a rectangular mesh of $[n \times m]$ control points, being $[(n-1)\times(m-1)]$ the degrees of the corresponding Bézier polynomials. Lang and Röschel[37] used this classical approach to model developable surfaces. Because developable surfaces should be ruled, they chose Bézier degrees $[1 \times n]$ so one of the directions is made of straight-lines.

Moreover, they investigated conditions for the developability of the rational $[1 \times n]$ Bézier surfaces, which being rational are not required to be linear in the direction of degree 1.

The conditions obtained for the developability of the rational $[1 \times n]$ Bézier surfaces are too complex for a handy utilisation, unless the degree n is two. In that case, the control net has consecutive coplanar rows in the direction of n, as shown in Fig. 21.

Because of its complex results, these papers don't give the ship designer the ability of the direct manipulation of Bézier developable surfaces. At least until there are software tools that hides from the user the intricacies of this task. For that purpose, other simpler and more intuitive techniques were devised by Bodduluri and Ravani[38,38].

After Redont's work[11], these authors tried other dualities besides the classical one existing between developable surfaces and indicatrix lines. They used two

methodologies to define developable Bézier surfaces, one called a point representation of the developable surface, and the other called a plane representation of the developable surface.

They explored the duality between points, defined in homogeneous co-ordinates, and planes existing in the projective space[36], also defined by four real co-ordinates in its linear formulation. All the Bézier formulations employed were of the third degree, but other degrees could be employed, including higher ones for increased freedom in the shaping of the surface[36].

In the point representation of the developable surface, its edge of regression is constructed as a Bézier curve, defined by its control points[36].

The developable surface is defined by spanning tangent lines at the edge of regression, used as rulings, and making a tangent surface.

The parameter of the edge of regression, being one-to-one associated with the rulings, is used as the longitudinal parameter of the surface. The surface's transverse parameter is the length measured along the rulings, from the edge of regression.

The boundaries of the surface are defined as third degree Bézier relationships between both parameters. The four three-dimensional points controlling the shape of the edge of regression are thus associated with the two sets of four two-dimensional control points of the boundary lines.

The most promising technique in this work was the plane representation of the developable surface, by a one-parameter family of planes defined with Bézier coefficients. These coefficients are elegantly defined by a set of control planes, so there exists a relationship between these control planes and the resulting developable surface which envelope the family of planes[5].

The four linear coefficients of a plane (recall Eqn. 35) of the enveloping family form a four-dimensional vector. This is defined by the four-dimensional vectors of the control planes (which are also four, because the degree of the Bézier is three[36]).

The interpolating equation for the coefficients of the plane family is:

$$\begin{Bmatrix} n_x(t) \\ n_y(t) \\ n_z(t) \\ q(t) \end{Bmatrix}^T = \begin{Bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{Bmatrix}^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} N_x^{(1)} & N_x^{(2)} & N_x^{(3)} & N_x^{(4)} \\ N_y^{(1)} & N_y^{(2)} & N_y^{(3)} & N_y^{(4)} \\ N_z^{(1)} & N_z^{(2)} & N_z^{(3)} & N_z^{(4)} \\ Q^{(1)} & Q^{(2)} & Q^{(3)} & Q^{(4)} \end{bmatrix} \quad (71)$$

Note that each $\mathbf{U}^{(i)}$ vector contains the linear coefficients of the i-th control plane, and that u varies between 0 and 1.

Using Eqs.36 and 37, the rulings and the edge of regression are derivable from Eqn. 35 and 71. Therefore, the equation of the surface would take the form of Eqn. 29, where the directrix is the edge of regression.

Alternatively, to Eqn. 71, the authors illustrate the application of the De Casteljau classical algorithm for Bézier interpolation.

The main properties resulting from the third degree Bézier formulation for the control planes are:

- The first and the last control planes are tangent to the developable surface;
- The intersection of the first two control planes and also of the last two control planes, are respectively the initial and the final rulings of the surface;
- The intersection of the first three control planes and of the last three control planes, are respectively the initial and the final characteristic points of the surface, which is known to be a tangent one.

At the patch ends the characteristic points must lie simultaneously in the 2nd and the 3rd control planes, thus they must lie in the line of intersection of these planes, implying that the characteristic curve is a straight-line. This is an important topic for practical applications, because the characteristic curve should have the proper degree of freedom, to adapt for the required constraints.

The Farin-Bohem construction is recommended for G^2 continuity across adjacent surface patches. The following relationships between the centre points $\mathbf{U}_i$ and $\mathbf{V}_i$ (i=1,2,3,4) of the control planes of the adjoining surface patches **U** and **V** (see Fig. 22) is a consequence of that:

$$\begin{cases} \mathbf{V}_1 = \mathbf{U}_4 \\ \mathbf{V}_2 = \mathbf{U}_4 + (\mathbf{U}_4 - \mathbf{U}_3) \\ \mathbf{V}_3 = \mathbf{U}_2 + 4(\mathbf{U}_4 - \mathbf{U}_3) \end{cases} \tag{72}$$

Note that it can be implemented in a simple way for the designer to use: every plane can be graphically handled by its central point, and the Farin-Bohem equations are programmed as an intrinsic relationship between those central points, so changing the location of a centre point (control plane) of one patch, moves the points/planes of the other patch accordingly. Obviously, this needs to be properly interfaced for an effective use by the designer.

For the development of the resulting surface it is suggested to discretise the surface as the set of tangent plane strips, each one using a ruling as a centre line. The angles between these strips are the angles between the surface

normals at the centre line rulings, and so they are defined by the following expression, where δu is the difference in the longitudinal parameter at consecutive rulings:

$$\alpha(u) = \arccos \frac{\mathbf{n}(u)\,\mathbf{n}(u+\delta u)}{|\mathbf{n}(u)|\,|\mathbf{n}(u+\delta u)|} \tag{73}$$

Flattening of the surface is done by a simple sequence of rotations. The dependency of the final accuracy on the curvature and on the strip width is not explored.

Conversely, the bending sequence required by the forming process is described by the rotation of $-\alpha(u)$ of the strips chinned at the intersections of the consecutive tangent planes.

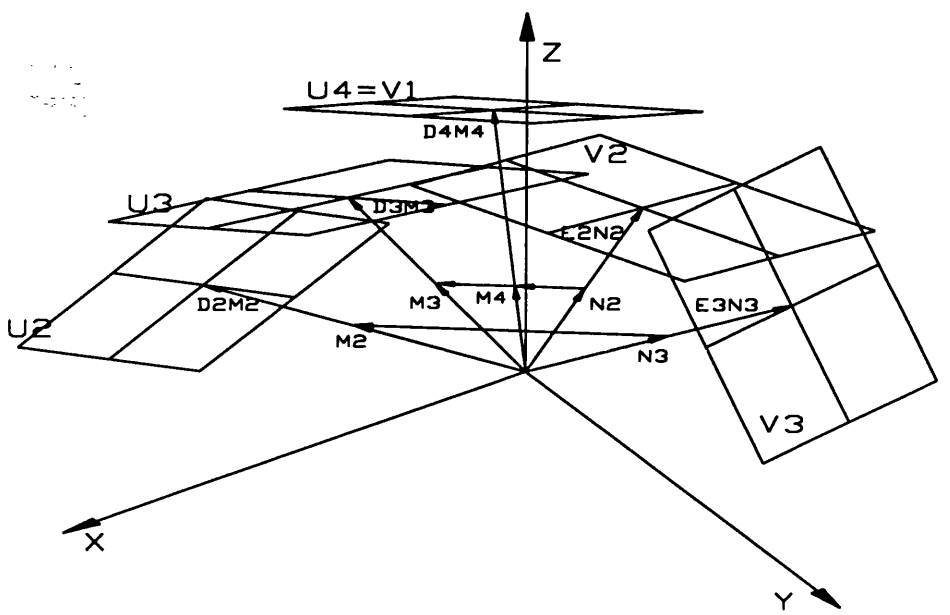


Fig. 22 - The Farin-Bohem construction for G^2 continuity.

Very simple illustrations are presented in both papers, but the intuitiveness and simplicity of the proposed method of defining developable surfaces using control planes are made clear.

Because of its elegant concept of duality between planes and surfaces and its true integration in all relevant aspects of surface design, surface development and user interface, this work can not be overlooked, and can be seen as a major proposal for a systematic methodology of developable surface design and development, since Nolan's[19], 20 years before.

Nevertheless, it still offers room for further research and improvement, not only in the user interface concepts and ergonomics, but also in the underlying formulations for the family of planes.

In fact, Bodduluri and Ravani tried periodic B-Spline, as an alternative to the Bézier formulation. However, the link from the control planes to the characteristic points and to the rulings becomes more complex and loose the intuitiveness of the Bézier formulation.

However, open B-Spline, not covered in this work, offers some of the useful characteristics of Bézier's, with the advantage of not having to consider compatibility conditions between the consecutive patches, like tangent or curvature continuity.

Moreover, also out of scope of Bodduluri's paper, NURBS could be another obvious choice, greatly improving both the degree of freedom and the user control on the surface shape.

3.6 State-of-the-art on Plate Development

In 1993, Lamb[1] made a report on the state-of-the-art on plate development methods, 25 years after Ferris published his leading work on this subject[8]. This work was carried out with the co-operation of several shipbuilders, to present the point of view of the user and of the main Computer-Aided Lofting software houses, invited to demonstrate their solutions for plate development.

Components	Quantity
Lofted Parts	35,000
Parts Cut by N/C Machines	16,000
Shell Plates	800
Non-developable Shell Plates	80
Plates Requiring a Bed to be Formed	40

Table 1 - Figures for a typical high speed container ship.

The shipyard expectations about the plate development and fit-up processes were still to be met, because of claimed software inaccuracies, forcing the workshop to use excess material around the plate borderline.

Cutting the plate neat to the developed outline is desired by the shipbuilders, in order to economise the labour intensive fitting, cutting and edge preparation at

the building berths. An overall accuracy in the millimetre figure is required to avoid expensive rework of almost finished parts, due to fitting mismatches.

Construction costs of curved plates are not reported in this work, but they are known to be way above the average costs for other ship structural parts.

Table 1 illustrates for a high-speed container ship built in USA, the amounts of curved plates relative to the entire set of the ship structural components. Table 2 reports on the relative use of each forming technology, for each ship and shipyard (Avondale/IHI Shipbuilding, also in USA). Note that the areas of the plates aren't reported, but only the number of plates done with each bending process.

The plates in the table's first row ("no roll") are plane ones, and given that they are the largest ones in the shell, there are not many. The table's second row counts the developable plates. Note that this kind of plates are simply rolled or pressed.

The other rows of Table 2, dealing with non-developable plates, account for more than 3/4 of the total number of shell plating, but not that much of shell area because complex surface shapes require further subdivision of the strakes. The traditionally extensive use of line heat in USA shipyards accounts in this case for almost 100% of the non-developable plates.

Bending Process	Plate Quantity	%
No roll	26	8.7
Roller or press only	45	15.1
Roller and line heating	196	65.8
Line heating only	20	6.7
Roller and forming jig	11	3.7

Table 2 - Curved shell plates for a tanker built at Avondale/IHI Shipbuilding (USA)

The inaccuracies found in the prediction of the non-developable plate boundaries led to errors as much as 3 inches in the plate corners.

Full-scale mocks are used to avoid this in the more problematic areas of the shell. Typically, these areas are:

- Clipper bows - soft nose stem;
- Cruiser sterns;
- Single screw apertures;
- Forebody and aft body shoulders;

- Blocks in the fore and aft bodies with vertical butts and horizontal seams;
- Bulbous bows;
- Domes (sonar's and other);
- Heavy flare in "fine" hulls.

All of these areas are typically non-developable. Therefore, the software developers aren't able to provide a consistent map of the curved plate onto the plane, because this is not uniquely defined, as discussed in section 2.7 Development Properties.

In fact, what shipbuilders ask for is not the plate development itself, but the convenience of accurately predict the plate shape to be cut in all cases. This is only achievable by the complete integration between the workshop forming technologies and the Computer-Aided Lofting software, which is still not available commercially.

The Computer-Aided Lofting software houses claim that applying development methods to non-developable surfaces give good approximations in most cases, and recommend criteria for acceptance of this approximation.

Albacore Research indeed programmed for this purpose its CAL package ShipCAM, where the spots expected to be unable to provide good development approximations have no surface mesh fitted over. Nevertheless, the criteria employed are not documented.

The straking of the shell is one of the most effective tools to improve on the developability of the plates. Some guidelines recommended when designing the shell straking are:

- To make the straking suit the shell shape, instead of the modular construction of the hull;
- To place the buts and the seams at the curvature inflexions;
- To decrease the plate size as the surface double curvature increases;
- To optimise the material utilisation, by trying to maintain orthogonality between butts and seams, and to adapt the plate dimensions to the available stocks of plating.

Thickness effects on the development accuracy are told not to be meaningful for thin plates. For the thicker plates, it is suggested to add its half thickness to the mould offsets, and develop the obtained plate shape instead of the one defined by the mould itself.

The experience of plate development from the aircraft industries is compared with the ship construction ones. Most significantly, the utilisation of triangulation algorithms for the developable aircraft parts still require stock material as an allowance for fitting up. Lamb didn't explain this, but it could be related to the bounding and riveting technologies that this industry uses extensively with the composite materials.

The development algorithms, being of commercial value, are described only to a point:

- The Senermar's FORAN package develops the surfaces used for the definition of the shell by a cylinder or a multi-cone approximation. However, because these defining surfaces are not unique across the shell, it seems to require a further approximation for the nesting of the individual development patches contained in the plate.

- The cylinder and conic fittings are validated by least squares criteria, otherwise it refines the discretisation.

- If the overall Gaussian curvature of the plate exceeds some values, the system itself imposes a re-strake on the panel using more plates.

- All the other five Computer-Aided Lofting development modules compared in this paper, use triangulation of many small panels subdividing the plate, each formed by four space points, to obtain the flat plate's shape by some nesting approximation.

- The Albacore Research's ShipCAM package uses a conical approximation for the development method of each plate partition, starting from the centre to the seams and then to the buts, and preserving the mesh arc lengths.

- The BMT ICoNS Limited's BRITSHIPS uses a single triangulation for each set of four mesh points.

- It also starts in the middle of the plate, developing first towards the seams, and then towards the buts.

- The lengths of the seams and buts tend to be maintained, to try to preserve the mating to the adjacent plates.

- The Coastdesign Inc's AutoPLEX uses an algorithm apparently inspired by Nolan's, intended for developable surface applications.

- Its other product, AutoPLATE, aims at non-developable plates, for which it uses a finite element method to expand the surface patches, each one defined by four space points[3].

- Kokums Computer Systems AB markets AUTOCON This package includes a development module working with a triangulation of Coon's patches, fitted onto the mesh defining the shell.

- The Cali & Associates Inc.'s SPADES nests the panel's developed triangulation from one end of the plate, whereas the others start in the middle to spread more evenly the algorithm errors.

- All development modules, except ShipCAM and AutoPlex/AutoPlate, account automatically for the plate's thickness relatively to the moulded shape.

- For all development modules, the information provided for production include N/C code, but only ShipCAM, AutoShip and AUTOCON give information about the strain fields required to form the plate.

- Several other limitations affect the usability of these applications. One of them is the difficulty in handling non-conventional plate boundaries, like triangular plates or buts not parallel to the frames.

- Also stock material is not automatically handled in some products, or is not considered at all, as in the case of the AutoShip line of products.

To do a practical comparison on the available commercial software tools, five plates said to be difficult to develop were used as test cases and the results compared. These plates are located in double curvature areas, therefore they where differently mapped into the plane by the software packages.

Differences as much as 50 millimetres in length and 25 in width were found at the corners, which is far too much when compared with common accuracy requirements for the fitting up. Fit up accuracy requirements is shipyard dependent. Nevertheless, to mount a steel block without stock material for in-place adjustment, the required accuracy should be well under the centimetre range. Otherwise, coherent errors in adjacent parts would render it impossible to fit.

---

[3] A representative from Coastdesign told the author that there was an association with Letcher[1] for the first releases, but they adopted independent development paths since some stage near the first publication of this paper (1993).

Lamb remarks that cutting all plates neat to the predicted plane boundaries is not expected to be possible in the foreseeable future, because of non-consistencies in workshop procedures, making it virtually impossible to model by the software.

Contrary to this idea, Letcher[1] took on the effort to develop a geometric formalism for the process of plate forming. He was already known by its work on America's Cup projects, the FairLine Computer-Aided Design system, and its collaboration with CoastDesign on the AutoShip software package.

The problem of non-developable plates can be addressed by modelling the fields of in-plane strain, classically known as membrane strain, imposed by each forming tool. These fields are either isotropic, like in the case of pining, die pressing and point heating, or orthotropic, as in the cases of rolling or line heating.

The metric tensor $g_{\alpha\beta}$ of the curved plate, is related to the metric tensor of the blank $G_{\alpha\beta}$, by the membrane strain tensor $e_{\alpha\beta}$:

$$e_{\alpha\beta} = G_{\alpha\beta} - g_{\alpha\beta} \tag{74}$$

This strain tensor models the output of the workshop forming process. Note that $g_{\alpha\beta}$ is known for the given hull, and that $G_{\alpha\beta}$ is arbitrary for the blank.

Therefore, instead of dealing with continuum mechanics, this model reduces the forming process to a pure geometric problem, which is in fact the very nature of forming procedures.

For any particular workshop, the forming procedures are more or less standardised in face of any typical plate shape.

The field $e_{\alpha\beta}$ models those procedures. Therefore, given the particular plate shape and its metric tensor, one should rewrite equation 74 in order to the unknown metric tensor in the plane blank:

$$G_{\alpha\beta} = e_{\alpha\beta} + g_{\alpha\beta} \tag{75}$$

For the isotropic strain procedures, the derivative vectors of the patch σ and the blank β, along any direction θ, are proportional to each other. Being so, there exist a field ε at the patch that satisfies the following equation:

$$\frac{\partial \beta}{\partial u_\theta} = (1+\varepsilon)\frac{\partial \sigma}{\partial u_\theta} \tag{76}$$

Thus, for the metric tensors, the following relationship holds:

$$G_{\alpha\beta} = (1+\varepsilon)^2\, g_{\alpha\beta} \tag{77}$$

From Eqn.74 one concludes that the strain field metric tensor is, for the isotropic cases:

$$e_{\alpha\beta} = \varepsilon(2 + \varepsilon)\, g_{\alpha\beta} \approx 2\varepsilon\, g_{\alpha\beta} \tag{78}$$

If ε is not negligible, one can easily reduce the above expression, just by choosing orthogonal parameters for σ, so $g_{12} = 0$, and using the Beltrami second differential parameter and the Gaussian curvature[5]:

$$\Delta\,[\ln(1 + \varepsilon)] = K \tag{79}$$

To make this problem one of determined solution, it is enough to specify ε all around the σ boundary.

For the orthotropic strain procedures, the elliptic dependency of ε on the direction θ can be stated in terms of a factor E:

$$e(\theta) = (\cos^2\theta + E \sin^2\theta)\, \varepsilon \tag{80}$$

Note that isotropic procedures can modelled by making $E = 1$, and unidirectional ones by making $E = 0$. In fact, E has a characteristic value for each forming procedure.

Because of Eqn.80, for orthogonal co-ordinates the metric tensors are related by:

$$\begin{aligned} G_{11} &= (1 + E\varepsilon)^2\, g_{11} \\ G_{12} &= g_{12} = 0 \\ G_{22} &= (1 + \varepsilon)^2\, g_{22} \end{aligned} \tag{81}$$

By omitting terms other than linear on ε, it results the following expression, whose form is closely related to Eqn.79, if expanded:

$$\frac{1}{\sqrt{g}} \left\{ \begin{aligned} &\frac{\partial}{\partial u_1} \frac{\sqrt{g_{22}}}{\sqrt{g_{11}}} \frac{\partial \varepsilon}{\partial u_1} + E \frac{\partial}{\partial u_2} \frac{\sqrt{g_{11}}}{\sqrt{g_{22}}} \frac{\partial \varepsilon}{\partial u_2} + \\ &+ (1 - E) \left[\frac{\partial}{\partial u_1} \frac{1}{\sqrt{g_{11}}} \frac{\partial \sqrt{g_{22}}}{\partial u_1} - \frac{\partial}{\partial u_2} \frac{1}{\sqrt{g_{22}}} \frac{\partial \sqrt{g_{11}}}{\partial u_2} \right] \varepsilon \end{aligned} \right\} = K \tag{82}$$

The patch boundaries are ruled by equations of the form:

$$u_i = u_i(t),\ i = 1,2 \tag{83}$$

Once the differential equation is solved, the map between the patch and the blank is known, and the u_i equations are established for the blank.

For the solution of Eqn.82 (or 79) it is necessary to use a numerical method. For this purpose, Letcher recommends to geodesicaly triangulate the patch.

The distances between any nodes i and j, are measured either in the blank (d_{plane}), and in the patch ($d_{geodesic}$), and related by the average strain between those nodes:

$$d_{plane} = [1 + .5\,(\cos^2\theta + E\sin^2\theta)\,(\varepsilon_i + \varepsilon_j)\,]^2\, d_{geodesic} \qquad (84)$$

If the triangulation has N nodes, connected by L links, than there exists $3N$ unknowns (on the node plane co-ordinates and strain), and L equations of the above form.

To determine this set of equations, it is necessary to consider $3N$-L additional conditions. These could be the strains at the boundary nodes, plus the specification of the co-ordinates of one of the plane nodes (eliminating the three rigid-body degrees of freedom).

The application of this numerical method seems to be quite satisfactory, at least for the very simple cases explored in the paper: a spherical cap and a Wigley patch.

Despite the lack of testing on true industrial plates, the essentials of this method show great promise for the efficiency and quality of forming technology, and can be the basis for full automation, when using N/C forming tools.

Martins and Aravena[31] published on developable surface design, improving on Kilgore/Nolan previous works. Acknowledging the indirect skills required for the designer when using the existing systems, they propose a system closer to the actual problem domain.

The developable surface is discretised with cone and plane elements. Instead of just defining it by two base lines, they offer the user the possibility to specify a directrix, plus either the direction of the ruling or the surface normal, along this curve.

Explicit concern is taken with the application's user interface, despite some rather primitive characteristics, like alphanumeric entries where other more interactive ways could be offered. Nevertheless, the potential for effective and intuitive design practices seem quite obvious, given the more direct control of the very nature of the developable surface.

The development procedure is not covered in this work, but is obvious, due to the underlying conical discretisation.

Chapter 4 - Concept and Implementation of an Algorithm

A new development method has been formulated, and the respective algorithm has been designed and implemented in an actual Computer-Aided Lofting System, ordered by a shipyard. The method is a map from any curved surface onto the plane, therefore it enables any plate to be mapped onto the plane, even if not developable, accommodating the processing of all kinds of ship plates.

In a mathematical sense, one could name the method as "Improper Geodesic Map", because the algorithm maps some in-plate geodesics onto straight-lines in the plane blank. However, it is not a proper Geodesic Map[5], since it only maps geodesics onto straight-lines if and only if the plate is developable. Alternatively, one can call it just "Geometric Map", since it is purely geometric, by opposition to analytic, but this is rather generic. The proposed name would be "Improper Geodesic Map", since it is more precise.

The next sections describe and discuss this method, starting with a full description of the geodesic tracing procedure, in which it is rooted. The immediate chapter follows with the analysis of the practical implementation of the method on an actual Computer-Aided Design system, currently in shipyard use.

4.1 Geodesic Tracing Discretisation

Because of compatibility requirements with an existing Computer-Aided Design system, the method had to address the limitations of defining surfaces not has a patch, but by a set of lines lying in it, like its transverse sections. Therefore, an analytical procedure being continuous along the surface has to be adapted to a finite approximation, as the surface itself is finitely approximated by those lines lying in it. The continuum/finite approximation called discretisation, is a source of errors and must be carefully evaluated.

Perhaps the most obvious analytical procedure for geodesic tracing, rooted in the very geodesic definition, is to subtract from the total curvature its geodesic component, along every infinitesimal curve step, and thus integrate the position vector from the corrected curvature. This would produce a geodesic path along the surface, since it will have zero geodesic curvature everywhere. However, this procedure doesn't endure the adaptation to a finite application, given the inappropriate fit between the system curves (for instance cubic splines) and the

very high degree of the geodesic curvature. Moreover, the finite surface approximation is clearly not isotropic, since:

- Longitudinally, between two consecutive surface sections, the geodesic curvature can only be approximated to the first degree, because there are only two geodesic curvature vectors to define it (one at each end).
- Transversely, being the system curves of degree *n*, their curvature is of degree *n-2*, and the surface normals along each curve are of degree *(n-1)(n-1)*. Therefore, the geodesic curvature at each end is of degree *(n-2)(n-1)(n-1)*.

If n is 1 or 2, there is no possibility to model the curve curvature itself, being it the second derivative. Therefore, using n= 3 (with cubic splines, for instance), the degrees of geodesic curvature approximation are 1 longitudinally and 4 transversely.

Therefore, the geodesic curvature profile of a surface curve, being of the 1st degree, is too stiff to fit the actual profile, which is expected to have roughly the same degree found in transverse variations (the 4th, for cubic splines). Note that the plate's definition system is isotropic.

Therefore, the prototyping of this geodesic tracing method produced almost meaningless results, with extraneous effects on all but the less curved segments, usually taking it clearly out of the surface, far from the geodesic path. However, when applied to patch-based Computer-Aided Design systems, this method should give reliable results, making it an attractive option, because of its simplicity and straightforward implementation.

This direction of research was naturally suspended, at least until a true surface definition system becomes available. Instead, a geometric method was devised for the solution required for the shipyard system. It was based in the same geodesic property that Branco exploited in his simple drafting method[13] for geodesic tracing. Furthermore, the computer implementation rended unnecessary several approximations required by hand lofting, which improves the procedure accuracy.

4.2 Geodesic Tracing by a Geometric Procedure

In the existing Computer-Aided Design system, a cubic spline is a sequence of cubic segments, each one traced between two consecutive lines defining the surface. The spline is geodesic if and only if all of its segments are geodesic. Hence, the core task in geodesic spline tracing is to trace a single cubic segment, between two consecutive surface lines. The successive trace of

each segment in a geodesic manner will render a final spline that will be geodesic.

As an iterative procedure, the geodesic spline tracing passes some information from step to step, along the surface path. This information is the boundary conditions of each segment, at the end line defining the surface, where the current segment meets the next one to be traced. Prior to anything else, one has to establish those boundary conditions.

The curvature of a surface curve can be decomposed in two components: the normal curvature and the geodesic curvature[5,6,7]. The former is the surface curvature in the direction taken by the curve. The latter is a component intrinsic to the curve, as already seen in the 2.4 Basic Surface Concepts section.

Different curves passing in one point of the surface with the same direction (the same tangent in that point) can have different geodesic curvatures in that point, but only one normal curvature (see Fig. 23).

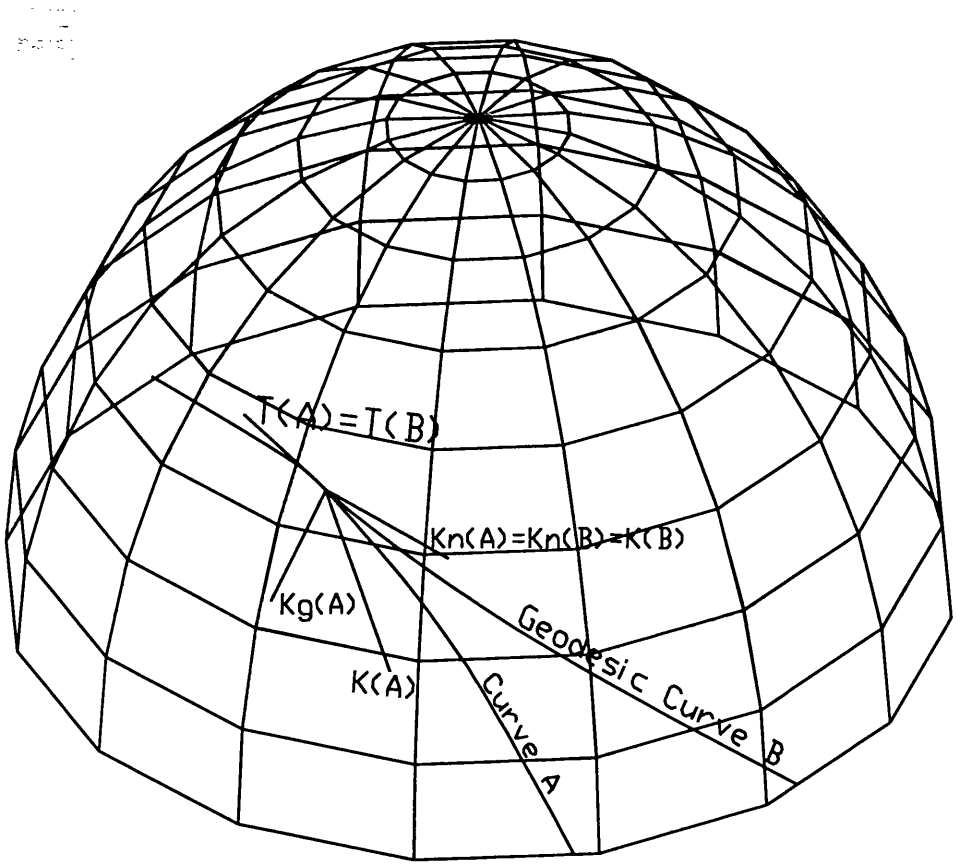


Fig. 23 - Two surface curves with the same tangent at a point.

A geodesic is a curve with zero geodesic curvature in all its extents. Being so, the geodesic's main normal is always aligned with the surface normal, and the

geodesic's osculatory plane is normal to the surface itself[5,6,7]. This is the essential geodesic property exploited by the present method.

The end of the geodesic spline segment is then approximated as the intersection between the osculatory plane at the segment's initial point (see Fig. 24), and the other mesh line. Therefore, it is admitted that the osculatory plane is constant along the segment span, or at least it doesn't change significantly.

Of course, this implies that the frame lines defining the surface should be spaced accordingly to the surface torsion, or otherwise close enough to deal with the worst expected case. Addressing this problem, some guidance can be found in the validation tests, in the final sections of this chapter.

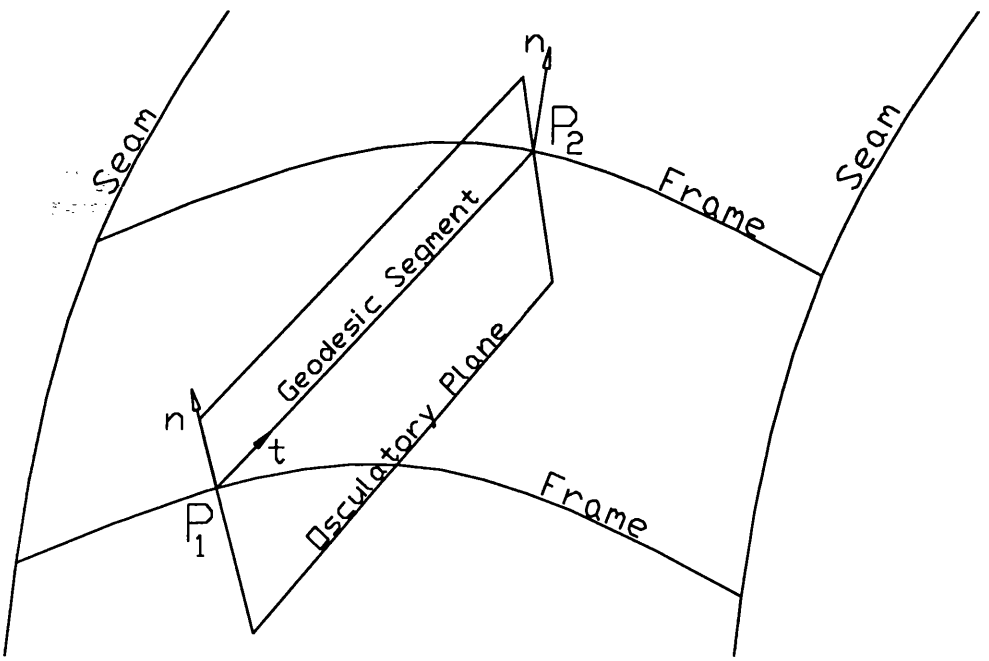


Fig. 24 - Tracing of a geodesic spline segment, geometrically.

The trace of the osculatory plane on the surface is fitted by a 2nd degree curve, for computing its end tangent vector. The component of this vector that is normal to the surface is discarded, making it a true surface tangent (see Fig. 25). The use of a curve of the 2nd degree is a consequence of having just 3 boundary conditions, namely the initial tangent vector and both end-points.

The tangent is computed from a 2nd degree segment, which is stiffer than a cubic segment. Therefore, the surface errors eventually existing along the path, like small bumps, would not affect the direction as much as if using cubic splines. Therefore, the 2nd degree interpolation should improve a little the tracing stability, hence the end tangent is expected to match more closely the true geodesic's direction when the surface definition has some error content, which is often the case.

Knowing the end point and the end tangent vector, the geodesic cubic spline is trivially computed, and the method proceeds to the next frame line, taking as initial conditions the end point and end tangent just computed (see Fig. 25). This process of tracing a geodesic spline geometrically is depicted in Diagram 1.

For validating this geometric method, it is advisable to compute the actual geodesic curvature along the curve, and compare the resulting paths with known geodesics, to have some measure of the method's accuracy. The Chapter final sections deal with validation issues, and the next section presents another procedure, which traces splines of geodesic curvature as low as prescribed, appropriate for comparison with the geodesics traced geometrically.

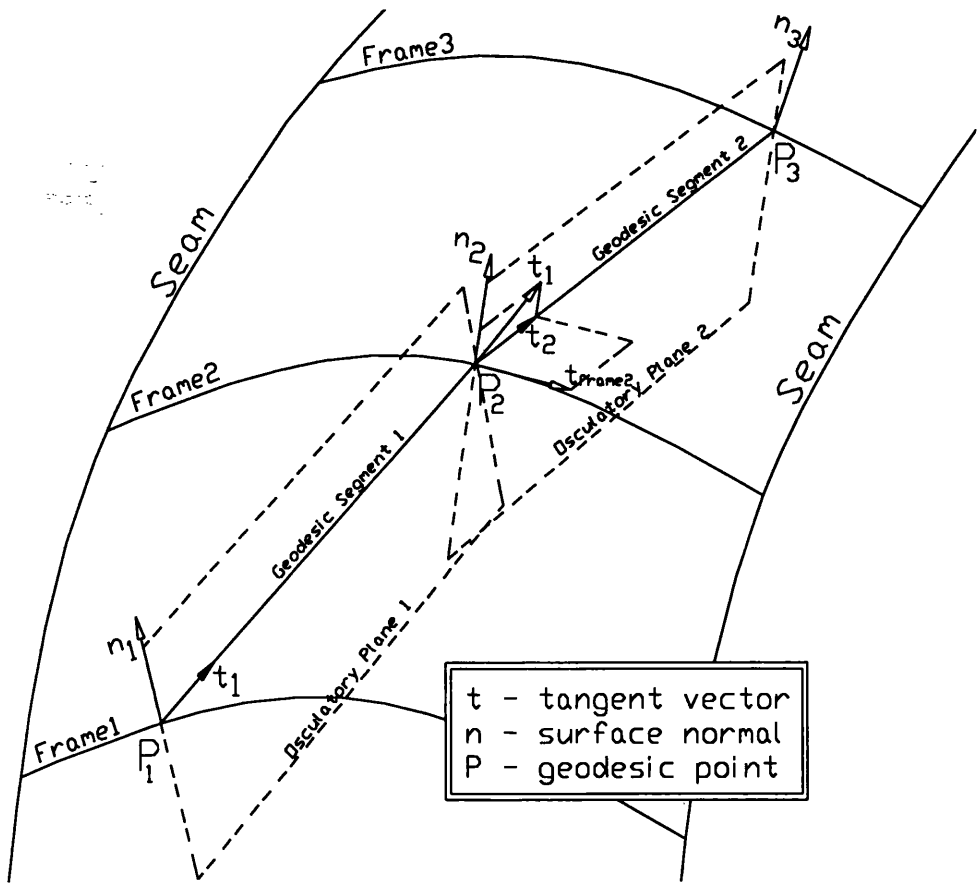


Fig. 25 - Tracing a geodesic segment which follows a previous one.

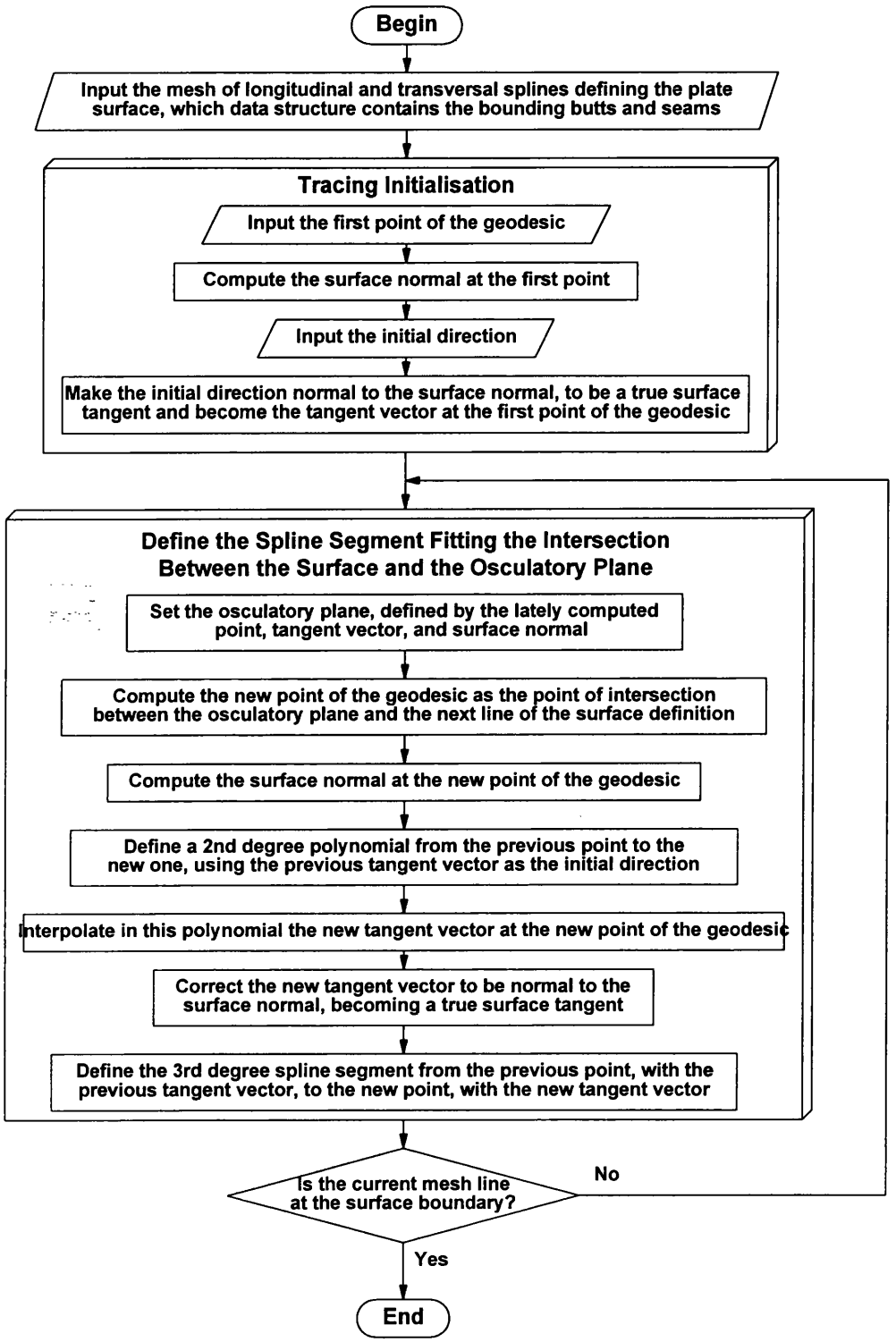


Diagram 1 - The process of tracing a geodesic spline by the geometric method.

4.3 Geodesic Tracing with an Optimised Procedure

Based on the previous geometric method, a new one is presented which computes the geodesic curvature of each spline segment, and changes its path according to a minimisation procedure controlled by the geodesic curvature. Notice that any single geodesic segment must have zero absolute geodesic curvature accumulated along its span, so the integral of the norm of Eqn. 22's vector should vanish when the end conditions are properly defined.

The absolute geodesic curvature accumulated along the segment span, will increase when the end conditions deviate from the geodesic ones (see Fig. 26). Therefore, a minimisation procedure can be used to find the geodesic by using the end and the end tangent vector as the optimising parameters.

From a given initial mesh point, on a specified initial direction (the initial tangent vector), the geodesic spline is traced through the mesh, one segment at a time. The geodesic segment tracing procedure addresses the problem of finding the end and the end tangent vector of a spline geodesic segment, given its start point and its start tangent vector.

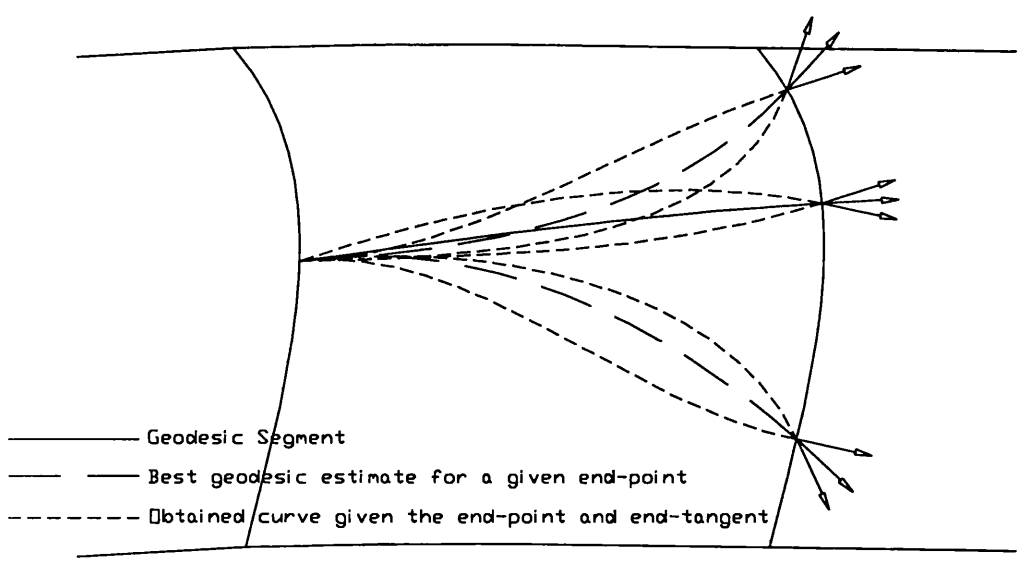


Fig. 26 - Varying the end conditions of the spline segment affects significantly its geodesic curvature.

The coefficients of the cubic spline segment are stated both in terms of the fixed initial conditions (the initial point and the initial tangent) and of the final unknown conditions.

For each spline segment there are only two known geodesic curvatures, one at each end, therefore the geodesic curvature profile is modelled as a first-degree polynomial, becoming its integration very simple and efficient.

The accumulated absolute geodesic curvature of the spline segment is non-linear on the two end conditions. Therefore, it is natural to use a second-degree minimisation method to compute those end conditions.

The analytical derivatives of the geodesic curvature in order to both end conditions are not known. Therefore, a method not depending on derivative calculation is preferable, to avoid the computing penalty of derivative estimate, which would require two function evaluations for computing each derivative.

Given the availability of good public domain routines for numerical analysis[40], the choice has to be among them. In regard to the previous considerations, and to avoid the slow progression of the simplex method, the Brent's method was selected, applied in two nested steps. The outer step varies the location of the end of the segment, and the inner varies the tangent vector direction (see Fig. 26). Separating the optimisation in two nested steps it is easier to control the iteration, and it allows for efficiency improvements. (See Diagram 2 for a depiction of the general structure of this algorithm.) The relevant issue is that tangent vector is not entirely independent on the end-point, as would be assumed by multidimensional algorithms. By nesting the tangent vector iteration inside the end-point iteration, it is possible to have the initial bracketing triplet closer to the expected result.

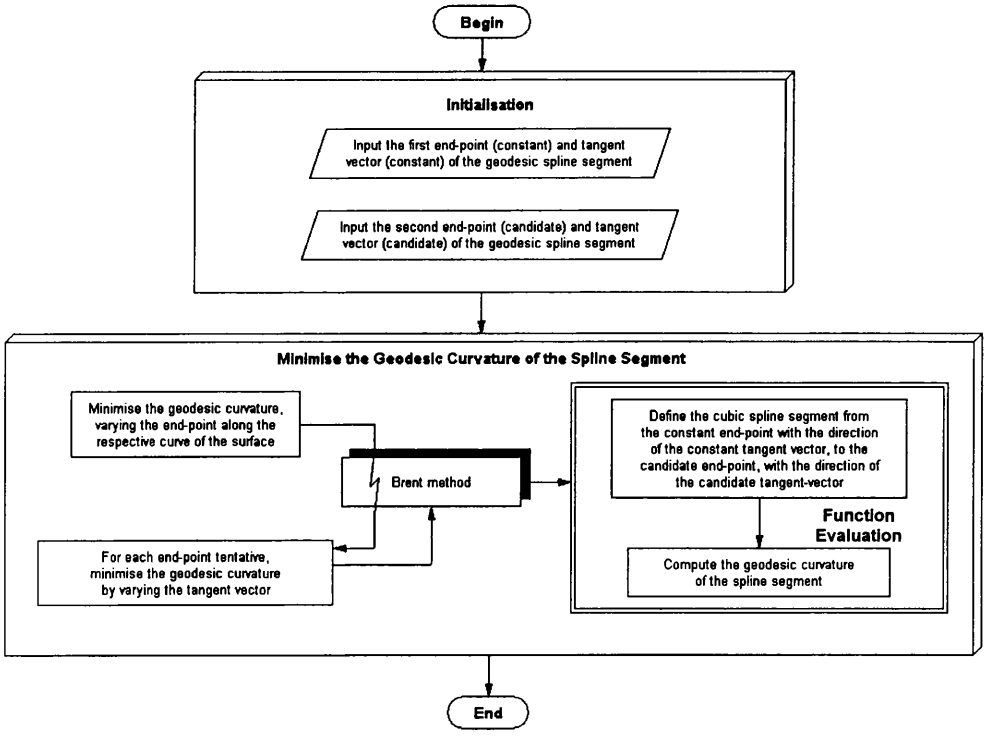


Diagram 2 - The process for minimising the geodesic curvature of a spline segment.

The bracketing triplets initialising both levels of the iteration have as mid-point the most "probable" candidate, and use as outer values the most extreme ones, for reasons of reliability and simplicity.

The initial bracketing triplet of the end-point are the following:

- The mid-point is computed by the geometric tracing method.
- The two extreme points are the ends of the section curve.

For bracketing each tentative end tangent vector the following three unit vectors are used:

- The mid tangent vector is computed as the end tangent vector of a second degree arc, defined by the constant conditions of the spline segment, and by the candidate end-point; this vector is not forcedly parallel to the surface, therefore it is projected on it, and only then normalised.
- The other two vectors are heuristically obtained by rotating the previous one 30° in each direction.

Note that the inner optimisation variable is in fact not the tangent vector, but the angle it makes to the centre vector of the triplet. The iteration control parameter is the mean absolute value of the geodesic curvature, computed along the segment's span.

The parameters whose values control the stop criteria should be the ones relevant to the computation. The iteration stops when is either one of the three stop criteria listed in Table 3 is accomplished. The first criteria, which limits the total of inner iterations, exists for practical reasons, since the process can't go forever. The value of 100 is just an heuristic result which seems to be reasonable.

Maximum steps in each iteration:	100
Reduction of *Kg* per iteration under	2.5%
Target geodesic curvature per unit length under	$0.0001 m^{-2}$
(Allows a deviation rate of the geodesic curve under	0.5 mm/m)

Table 3 - The three stop criteria for the geodesic curvature minimisation.

The iteration is successful as soon as becomes negligible the mean geodesic curvature along the span of the spline. Note that a typical spline segment, say 2

metres long, can have its end-point at most 1 millimetre apart from its exact location.

The second criteria is necessary to abort eventual iterations with negligible gains, therefore too inefficient to be really improving. It is just a control to avoid blocking situations, like variations in the geodesic curvature of the order of the round-off error.

The adjustment of this value was made with the validation tests (see Appendix B. - Validation Data). The method was to let the iteration go until success, and then increase this value and try again. When problems where found, some fiddling with the values of the other criteria was done, but without clear results, so the value was settled.

The process of geodesic tracing with optimisation is synthesised in Diagram 3. The use of the methods previously described, both for the tracing initialisation and for the tracing iteration, provide for the problem decomposition and reduced the overall process complexity. The re-use of the code already made for geometric tracing and the code available for the Brent's method[40], simplified the phases of implementation and testing.

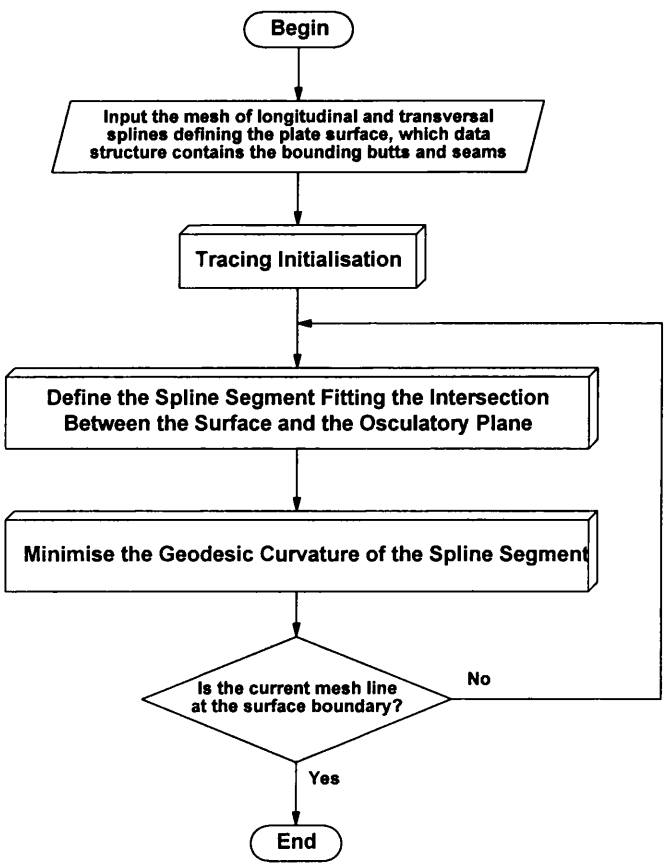


Diagram 3 - The process of tracing a geodesic spline by minimising its geodesic curvature.

The actual tracing implementation in the map has to adapt for the particularities of the Computer-Aided Lofting System, already in use at the shipyard. Since the existing surface definition system was exclusively based on lines, the method has to collect additional information about the mesh lines intercepted by the geodesic, for efficiency reasons, as it will be seen in the next Sections. The process used in testing was the one depicted in Diagram 4, which is more flexible, since it implements both tracing procedures. Notice that the conditional block of code that contains the optimisation was removed from the final production code, as discussed in the final Sections of this Chapter.

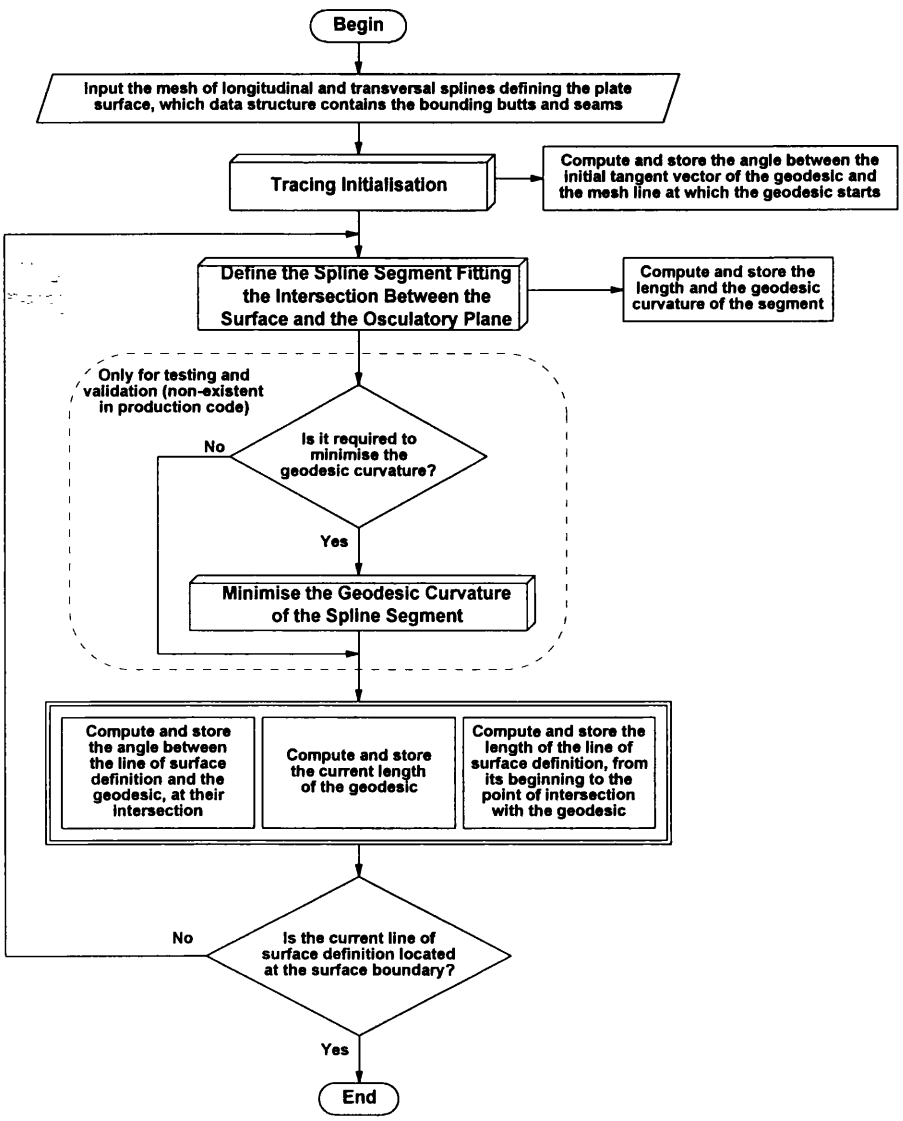


Diagram 4 - The process of tracing a geodesic spline, as implemented.

4.4 The Map Requirements

The only plates theoretically developable, are the ones with zero Gaussian curvature everywhere, but that condition is not generally met on a typical ship hull. Therefore, this development method was designed to accommodate the processing of non-developable plates. In such cases it provides less meaningful results, to be taken as what they are: approximations to the plate outline which are as good as the second principal curvature is less significant.

In fact, this is not a serious limitation for the plate manufacturing process, because in those circumstances, the particular shipyard and operator practices, though being mandatory, are not at all uniform[1,1]. Therefore, the map of non-developable plates is already seen by the practitioners as an approximation onto which they have to consider added material allowances, at least in one of the tops and one of the seams.

Notice that a geodesic on a developable surface is developed as a straight-line[5,6]. However, if the surface is not developable, there is no unique map between the shell plate and the plane blank[1], and the same geodesic can be mapped onto the plane by several different transformations, onto several different plane curves. Therefore, some constraints must be set, so the non-developable surfaces could be uniquely mapped onto the plane. Moreover, to be effective, the map constraints should model the workshop procedures, as closely as possible.

Given the trial-and-error interactive nature of the forming work, the forming procedures of non-developable plates are seldom repeatable, therefore the map constraints have little to be modelled of. The increasing use of numerically controlled tools, and the acceptance of Total Quality Management, is expected to improve the predictability of the forming procedures.

There was no data available to define a set of map constraints, to preserve the uniqueness of the map results, as Letcher did[1]. Therefore, a non-unique map was devised, which preserves the curved plate's metrics to a reasonable extent. For that purpose, the simple geodesic map of developable surfaces was adapted for the processing of non-developable surfaces.

Trying to assure the map relevancy both for technical applications, and as a research product, a small set of requirements where to be met:

1. It has to be isometric when applied to developable surfaces.
2. It has to be reliable even when applied to non-developable plates, so a properly developable plate and one slightly disturbed by small non-developable spots must have almost identical maps.
3. It has to be simple to implement.

4. It has to be reasonably efficient in computer resources.

The fulfilment of the first requirement is a consequence of the map being a geodesic one, when applied to developable surfaces[5].

The second requirement is met by the unlimited existence of geodesics all over the plate, from which it is always possible to use one to map any single plate point. However, it remains to be shown that the map results only suffer smooth variations when the surfaces are deviated from the developable condition. Being this clearly subjective, the map behaviour must be assessed for some representative set of plate shapes, as seen in the final sections of this Chapter.

The third requirement was met by re-using the procedure for geodesic tracing, already available at the time the development method was ordered. Moreover, further simplifying the concept and implementation, the geodesics are always mapped into straight-lines, even when the plate is non-developable.

Finally, the efficiency requirements are chiefly of an algorithmic nature, to be dealt with in the next section.

4.5 The Map Concept

A simple geometric map would be to trace a longitudinal geodesic along the plate, serving as a curved abscissa axis, and a transverse geodesic acting as an ordinate axis. Being orthogonal, the two curved axes, say *uu* and *vv*, would be mapped on the plane onto two orthogonal straight axis, say *xx* and *yy*.

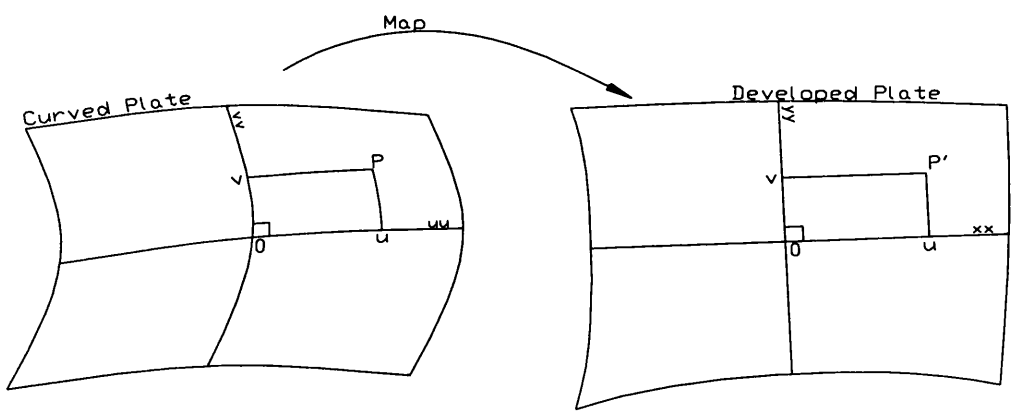


Fig. 27 - A simple geometric map, based in a frame of geodesic co-ordinate axis.

Every single plate point would have co-ordinates on those axes. Each co-ordinate would be computed by tracing a geodesic from the point onto one of

the axis, such that at the intersection the geodesic and the axis would be at a right angle (see Fig. 27). The axis span from its origin onto the intersection point measure the co-ordinate, which is a map invariant for developable surfaces.

Conversely, one can understand this co-ordinate system as a net of orthogonal geodesics all over the plate (see Fig. 28), each one being the locus of the plate points with a particular co-ordinate value.

Paradoxically, this rather uncomplicated concept produces a complex schema to be implemented, which is quite intensive in computing resources. The reason for this is the requirement of orthogonality between co-ordinate geodesics, which requires iterating through several geodesic tracings, for each co-ordinate, meaning twice for **every** mapped point.

One can simplify this map procedure, by tracing either the *uu* geodesics or the *vv*, but not both. This way one co-ordinate would be the distance from the point to the intercepted axis, and the other would be measured from this intersection point to the axis origin (see Fig. 29).

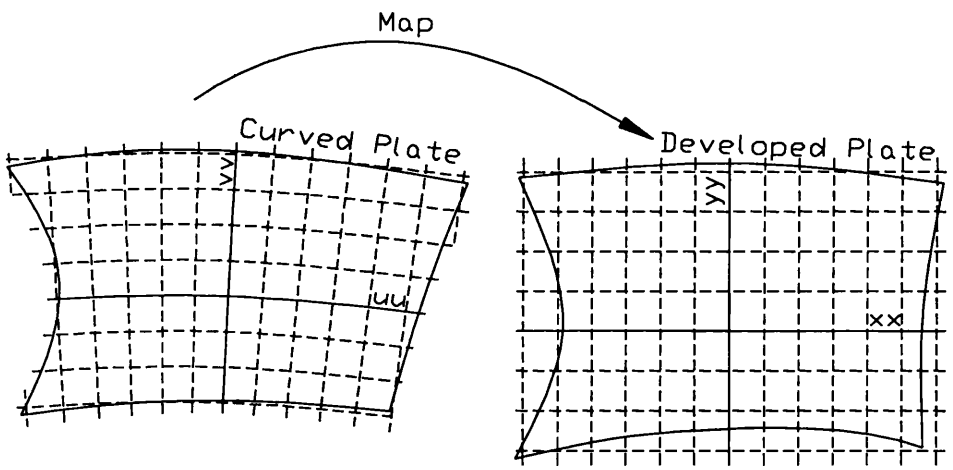


Fig. 28 - Geodesic co-ordinate grid.

The amount of orthogonally traced geodesics is cut by half. Nevertheless, the computing toll is still high, due to the complexity of tracing a single geodesic, subjected to these boundary conditions. Notice that the lines-based surface definition system doesn't allow for an analytical and direct orthogonal tracing procedure, as the geodesic equations formulated from an analytical expression of the surface. This is an interesting option for Computer-Aided Design packages built on top of analytical geometry engines.

Other problem, common to both maps, is tracing the geodesic out of the plate boundaries, risking passing across possible rough spots. These spots should only affect their containing plates, to localise eventual numeric effects, and to

assure the procedure robustness. Otherwise, the procedure behaviour would be unpredictable, in the sense that perfectly developable plates sometimes would be improperly mapped.

A third map can be derived from this last one, keeping the only geodesic axis and the one only locating geodesic per mapped point, but relieving the orthogonality condition at the axis intersection. Instead of locating the mapped points with one or two co-ordinate geodesics, a single geodesic is traced from each point onto the abscissa axis (see Fig. 30). Then, the angle *beta* is measured at the intersection with the axis, to be used to compute the two orthogonal components of the mapping geodesic. Notice that this procedure no longer requires the computing toll of tracing geodesics until a reasonable orthogonal approximation is reached. Therefore, it is by far more efficient.

Furthermore, the mapping geodesics should always be directed to the plate' centre region, avoiding completely the possibility of tracing out of plate' boundaries, which improves the map reliability.

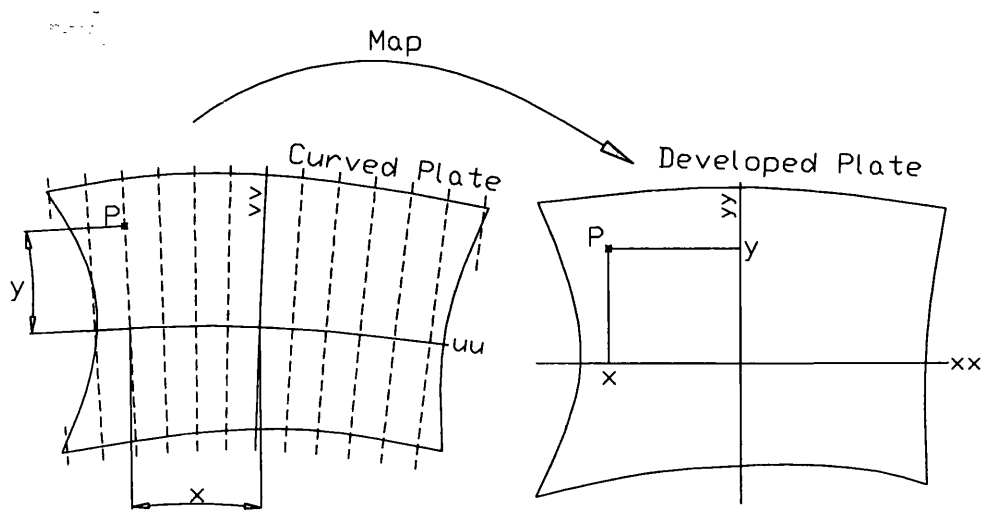


Fig. 29 - The simplified map, using only the transverse geodesics.

4.6 Map Elaboration

The previously stated map concept is the basis of this thesis work, and is hereafter scrutinised, developed, and tested in a particular implementation.

As already seen, the *uu* geodesic, spanning the plate's length, is the development directrix, and is mapped onto the straight-line *xx*. On the other hand, the mapping geodesic, particular to the point P, is labelled *vv(P)*. It is developed onto the straight segment *yy(P)*.

When the plate is developable the angles are invariant, due to the isometric nature of the development map[5,6]. This allows making a quite efficient line map, as can be seen by the map of the frame point P, in Fig. 30. Notice that the mapped geodesic *yy(P)* not only reveals the frame point P', but also the frame angle *alfa*. Moreover, the angle *gama*, that is another development invariant, is easily computed on the curved plate using the tangent vectors of the frame and of the directrix at their intersection point Q. Therefore, the frame cubic spline is completely defined between P and Q, tracing only one map geodesic.

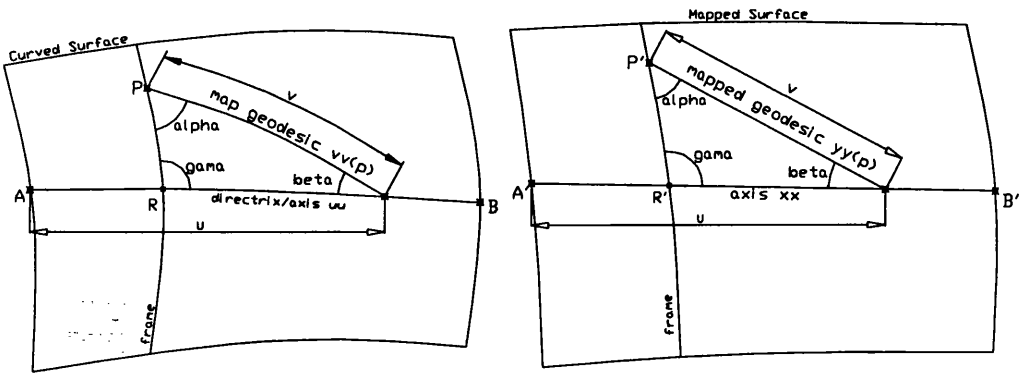


Fig. 30 - The map of a frame point P onto P', in the plane.

At the intersection points between frames and the seams, the map is even more efficient, due to the knowledge of the angles with both lines (see the point T, in Fig. 31).

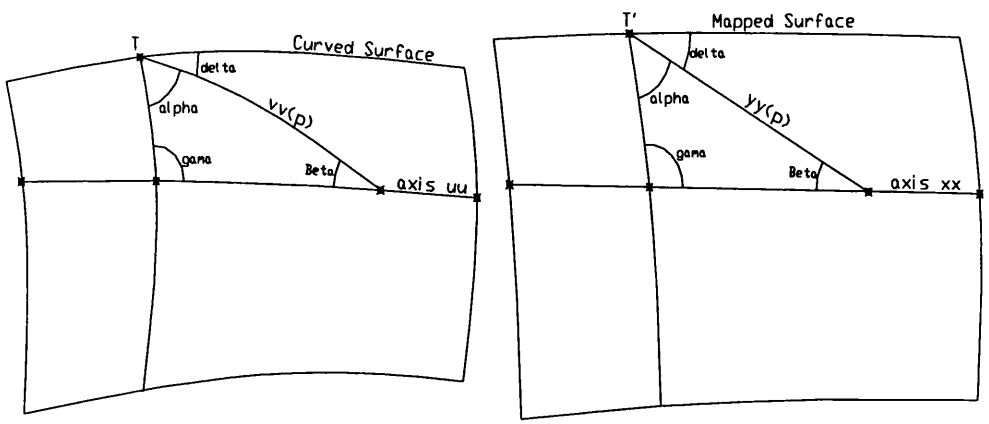


Fig. 31 - The map of the point T, located at the intersection of the frame and the seam.

Theoretically, if there are non-developable spots, the same point can be mapped differently, if using various map geodesics *vv*, as seen in Fig. 32. Moreover, due to the inescapable numeric errors, even for the developable surfaces, two different mapping geodesics would usually produce two mapped points numerically different, even if they weren't conspicuously apart. The only

coincidences to be expected to occur are for the trivial cases, for instance the plane plates.

In fact, because of computing several products *distance* **x** *angle*, the development method is exposed to inaccuracies in the evaluation of trigonometric functions, and to the error content in those angles. *Alpha* is the only angle directly set by the algorithm. Hence, it should use one of the values less prone to error propagation. The angle *beta* and the longitudinal *y* co-ordinate are computed using, the direct and inverse cosine functions, for which it is best to use:

$$\alpha_{cos} = n\,\pi, \quad (n \in Z) \tag{85}$$

The sine function is also required for the co-ordinate *y*, but α_{cos} defines the worst angles for that purpose, in a numerical sense, since the sine differential is maximum. Therefore a generally more reliable value for *alpha* would be (among others admissible):

$$alpha = \pi/6 \tag{86}$$

The map, as just defined, will hereafter be called "Improper Geodesic Map", as it was reasoned in the beginning of this Chapter. The following Diagrams depict the processing of the map. The core process of the map is the computing of the in-plate distances and angles existing between the relevant points and curves (see Diagram 7).

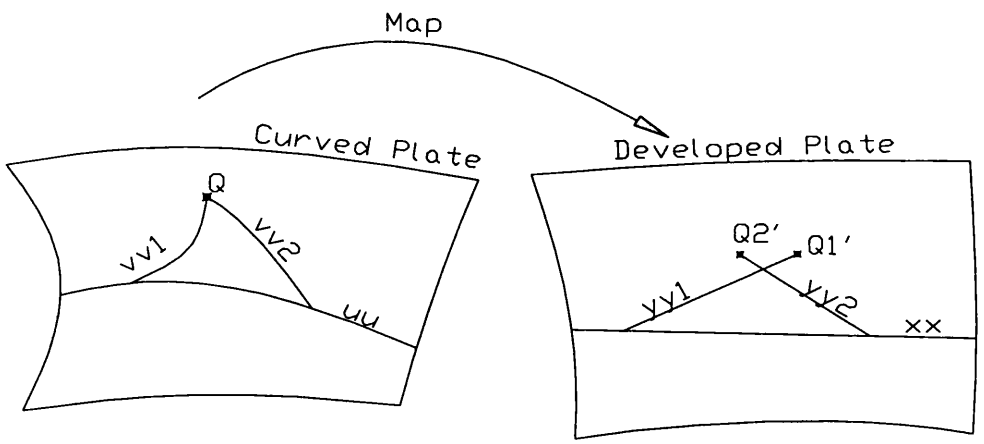


Fig. 32 - When the plate is non-developable, the map is non-unique.

The input data-structure that defines the plate surface is made of two sets of splines: the frames and the buts, plus the splines computed transversely to the frames, and the seams. These two spline sub-sets compose a mesh of lines almost orthogonal to each other, like a grid.

All this lines are defined as cubic splines, so if in the data structure the frames are transposed with the longitudinal splines and the butts are transposed with the seams, the procedures accepting the former data-structure will still work. Therefore, the tracing procedure of the longitudinal geodesic can be reused to trace transversal geodesics. Furthermore, given that the geodesic tracing procedure accepts as input the initial tracing direction, then it is able to trace oblique geodesics.

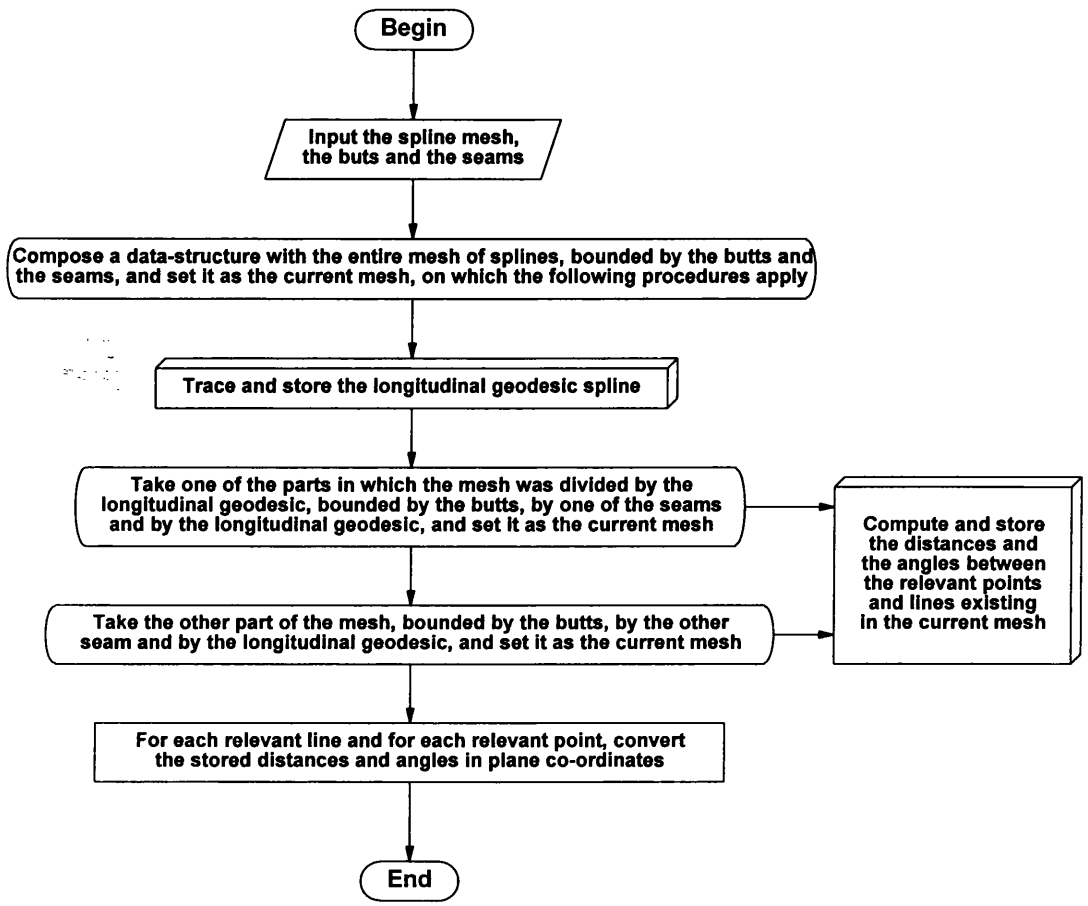


Diagram 5 - The overall computing process of the Improper Geodesic Map.

Notice that the tracing of oblique geodesics in the spline mesh requires some care in the specification of the initial direction. Because the trace process works by intersecting only the mesh lines in the same sub-set as the one that contains the starting point. Therefore, it should not go directly to one of the border splines, that is contained in the other sub-set of the grid splines.

This generalised tracing procedure is used not only to trace the longitudinal spline, but also to trace the locating splines. Each of these maps not only the respective point, but also the tangent vector of any intersected curve. This aggregate map of several vectors along with each point, improves the re-use

inside the overall procedure (see Diagram 5), and consequently improves its compactness, reliability and easiness of coding.

The map procedure starts inputting the required data to assemble the data-structure of the spline mesh that represents the plate surface (see Diagram 5). Using this data-structure the longitudinal spline is traced and its data stored (see Diagram 6). Following, from each relevant point or curve lying in the surface, the locating geodesics are traced, and the resulting data is stored. Finally, this data is translated to the plane co-ordinates of the map, as already explained above.

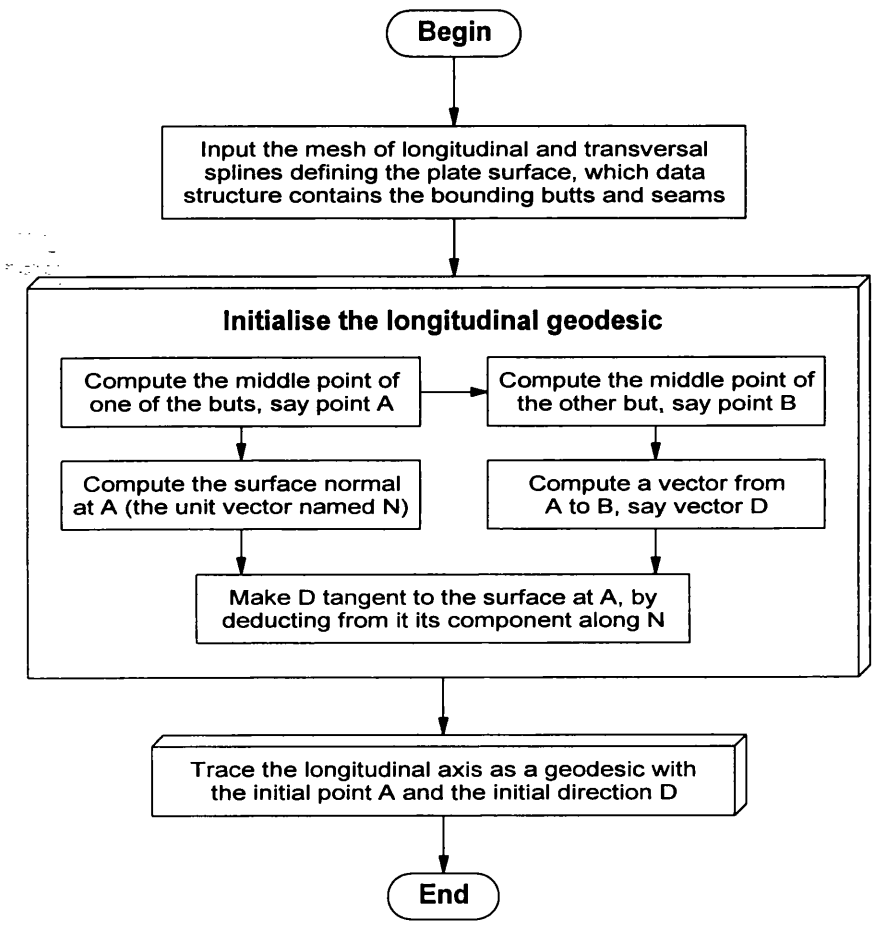


Diagram 6 - The process of tracing the longitudinal axis along the plate surface.

The tracing of all the locating geodesics is organised in two similar sub-processes, one for the upper part of the plate, the other for the lower, as the plate itself is divided by the longitudinal geodesic axis. Each sub-process uses a different, simpler surface model (see Diagram 7), without a longitudinal geodesic across the middle plate, making it possible to be re-used elsewhere. Notice that despite being simpler, the direct use of this plate model is prevented by the longitudinal geodesic, which must be as central to the plate domain as possible, to minimise the errors propagated by the distance.

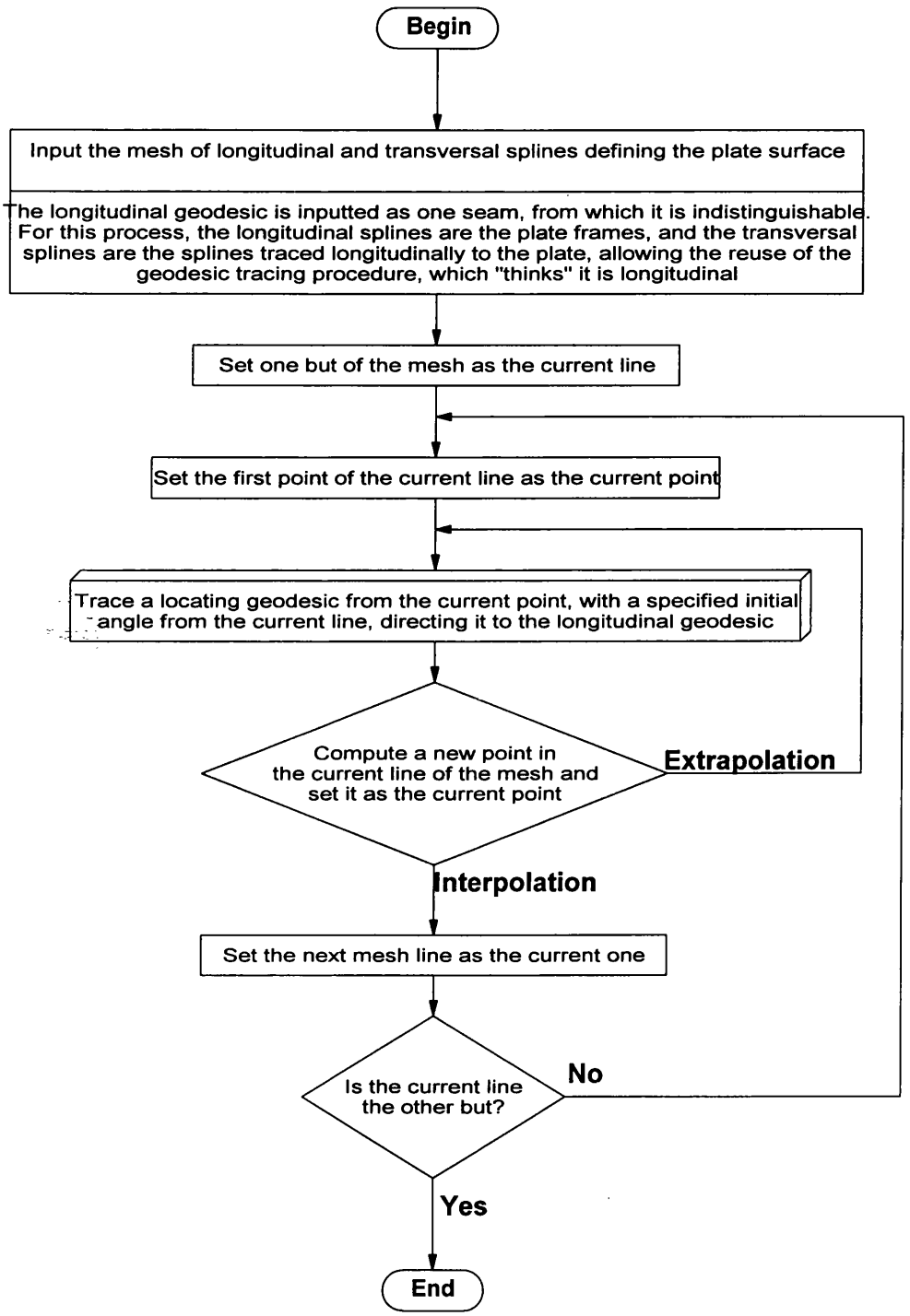


Diagram 7 - The process of computing the in-plate distances and angles between the relevant points and curves.

Notice how the overall process is so heavily dependent on the handling of spline lines, which define the plate surface. This topic is further discussed in the next Chapter.

4.7 Uniqueness Issues

Maps are relations from initial objects onto their images. The usefulness and workability of a map benefits from two reverse properties: each object has only one image, and each image has only one object.

If a plate is non-developable, then two different *alpha* values can produce two different points in the plane, since there would be two diverse traces across non-developable plate regions. Therefore, the Improper Geodesic Map can produce more than one image of the same point

Moreover, even for developable plates, the irretrievable numeric errors would render different maps for the same initial point, if using different values for *alpha*.

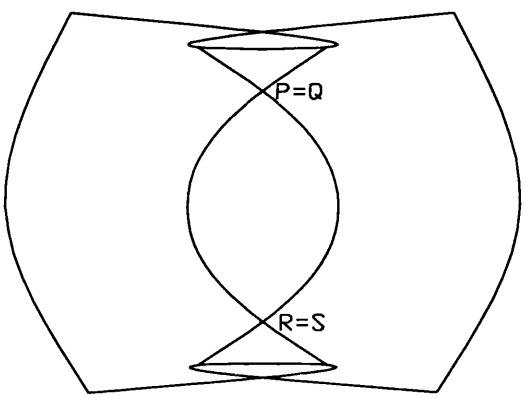


Fig. 33 - The map of an extremely non-developable surface, showing the plane surface folding along the seams, and two parallel frames intersecting each other. Note the extraneous coincidence of frame points P-Q and R-S.

Therefore, to preserve the one-to-one requirement of any useful map, one must do only one tracing from each mapped point. However, the reverse problem still persists, even if only for the extremely non-developable plates: two different points can be mapped onto the same plane location, producing folded plate outlines, as seen in Fig. 33.

Fortunately, these errors do not happen but for plates of abnormally high double curvatures, or for rather long ones, if sparsely discretised, which is also uncommon.

4.8 Accuracy Requirements for the Traced Geodesics

The traced geodesics are expected to deviate from its exact path due to several error sources, including:

1. The data errors, like the ones committed to data during the digitising of frame lines from traditional tables of offsets;

2. The rounding errors due to the limited number representation in the computing devices;

3. The error propagation in the algebraic operations;

4. The errors of the method like the discretisation of the geodesic path.

The first error source does not concern this study, since it is independent from the processing method. Fortunately, this error source can be disregarded just by making the comparisons relative to data already in digital format, and not with the data in any analogue format, like the one registered in paper.

The second error source is intrinsic to the common computing devices and programming languages, therefore it is not possible to simply remove it. Yet, rounding errors are not particularly interesting for this study, since it is not intrinsic to the method itself, so to minimise its effects, this study is to use data with excess of accurate digits. Because the data is not digitised, but designed purposely for the tests, the data sets where easily represented by real values in single precision (about seven digits). The third error source is concerned directly to the procedure and the fourth to the method. Therefore, they should be the focus of this study.

The proposed measure for the geodesic accuracy, is its deviation from the accurate path per unit length, due to the last three error sources. It is given by the following relation, where K_g^{ave} is the average geodesic curvature per unit length:

$$\delta = [1-\cos(K_g^{ave})] / K_g^{ave} \tag{87}$$

However, since the geodesic curvature is small for a tentative geodesic, the Taylor approximation to the cosine would give:

$$\delta \approx K_g^{ave} / 2 \tag{88}$$

The non-developable plate processing is intrinsically inaccurate, so it doesn't constrains the geodesic tracing accuracy. Regarding the developable plates, it is useful to relate their dimensions and the length of the greatest generators in the definition of the allowable δ.

In the ship, most developable plates are generated either diagonally or longitudinally. The seldom plates generated transversely can be seen as a simplification of the former case.

The plates with diagonal generators are usually either of short length or low curvature, because the common workshop practices so dictates. Therefore, their curved geodesics do not accumulate as much curvature as the ones existing in plates with longitudinal generators. Among these, the round bilge of small ships constitute the worst cases for the Improper Geodesic Map, given their high curvature, which makes them the reference for the definition of an accuracy criteria.

The workshop requires that the plate development accuracy be at the centimetre level. The bilge plates with shorter radius occur in the smaller vessels, for which plate beams of 2m are quite common. Based on this reasoning, the worst developable geodesic should be about a meter long, which makes the allowed relative error close to the 1% level. This translates to the following criterion to be met by the geodesic curvature:

$$K_g^{ave} < 0.020 \text{ m}^{-2} \quad \text{(for developable plates)} \tag{89}$$

Notice that K_g^{ave} is the geodesic curvature accumulated along the traced curve, divided by the geodesic length.

This simplistic accuracy target was made for the round bilge plates of small ships, the worst design case. More realistically, one should define a set of K_g^{ave} figures relating to each problem class. For instance, the non-developable plates should have more relaxed accuracy constraints, since the forming practices vary widely, and the plate outlines resulting from any map are inevitably given added material, rendering worthless the effort on an increased accuracy. For the time being no such set of K_g^{ave} are in research, so this topic remains to be handled.

The proposed accuracy criteria for the geodesic tracing is much more relaxed than the 0.0001m^{-2} stop criteria for the geodesic tracing (see Table 3). Consequently, all traced geodesics would meet the accuracy criteria, but the ones too inefficient to be economically traced. Moreover, these last ones are expected to occur in plates either acutely non-developable or roughly discretised, therefore with other more significant accuracy problems.

4.9 Plate Set for the Preliminary Validation

A small set of test plates was made for the preliminary validation of the plate development procedure, using both geodesic tracing procedures.

This *validation* is the process of making evidence that the *map procedure* consistently provides acceptable results for the target plates. What is to be validated is the map procedure, not the map itself, since what is actually put to test is the particular map implementation. Notice that any procedure degrades the numeric behaviour of the implemented method, due to the inevitable trade-offs between speed, size and complexity.

The test set must be simple and small enough to be feasible, yet representative of the actual ship plates on which the procedure is to be used. To be *representative*, means to capture all the meaningful features of the actual shipyard data. Therefore, a clear perception of this data must be established before designing the test set.

Given the limitations of the existing plate forming technologies, the ship plates are designed as small surface patches of a uniform geometry. The most complex hull regions are assembled from smaller plates, and the most uniform hull regions use the longer plates. Consequently, the hull plates are relatively even in respect to the intrinsic geometry.

To properly model such surface patches, one has to consider both the most typical examples like planes, cylindrical and conical plates, but also the most extreme cases of non-developable shapes, like saddles and paraboloids. Furthermore, at least the validation set has to explore the influence of different frame spacings and different plate dimensions, given the method's dependence on discretisation issues.

One can argue that actual ship plates would constitute a better test set, but that would result in a biased one. Unless it is big and varied enough to support statistic validation, rendering it impractical for this stage of work. Moreover, an eventually extreme case of a hull plate would be too complex to expose any single map feature, defeating the efforts to assess the map behaviour. This behaviour is the very target of a preliminary assessment like this, and to be fully appreciated it should be exposed in views, as independent from each other as possible.

The process data are easier to model if classified by prevalent features, representing each class an independent data type. For instance, the primitive surface shapes, like cones, planes, saddles, etc., being the fundamental intrinsic geometry types, represent the purest kinds of data. To have a generic representation of the data, one should start with these basic data types. Any particular composition of the pure types would inevitably favour the prevalent ones, unless one knows how to balance them. This balance is of a statistic nature, requiring the map of many and varied ship hulls, which is still to be accomplished.

Before that, the map has to be preliminary validated, eliminating any outstanding method misconceptions or procedure bugs. If the procedure fails to properly process the test set, then it must be further analysed and reworked, or otherwise discarded. If it succeeds, its ultimate value remains to be established, for which it is to be fully implemented in a production release, for shipyard evaluation and current use. This last and telling stage is still in the way. Yet, some findings where already collected and are presented in Chapter 5 - Industrial Application of the Improper Geodesic Map.

Description	Radius or Equation	Developable ?
Plane		Yes
High Curvature Cylinder	R= 2	Yes
Medium Curvature Cylinder	R= 3	Yes
Low Curvature Cylinder	R= 4	Yes
Rough Frame-Spacing Cone	R= 5 r= 0.5	Yes
Medium Frame-Spacing Cone	R= 5 r= 0.5	Yes
Close Frame-Spacing Cone	R= 5 r= 0.5	Yes
Extreme Spacing Cone	R= 5 r= 0.5	Yes
Slight Curvature Paraboloid	z= (x^2 y^2)/ 200.0	No
Low Curvature Paraboloid	z= (x^2 y^2)/ 0.126	No
Medium Curvature Paraboloid	z= (x^2 y^2)/ 0.251	No
High Curvature Paraboloid	z= (x^2 y^2)/ 0.375	No
Extreme Curvature Paraboloid	z= (x^2 y^2)/ 0.500	No
Saddle	z= (x^2 -y^2)/ 2.000	No

Table 4 - The 14 validation plates.

The validation set is briefly described in Table 9. Notice it is made of the fundamental surface shapes, namely: planes, cylinders, cones, paraboloids, and saddles, which cover both the developable and the non-developable fundamental cases. Moreover, the discretisation issues are considered by varying the frame spacings, the number of points defining each frame, and the principal curvatures (see also Table 5).

	Length	Beam	Number of ...	
Description	(m)	(m)	Frames	Frame Points
Plane	8	2	5	5
High Curvature Cylinder	8	0	5	5
Medium Curvature Cylinder	8	0	5	5
Low Curvature Cylinder	8	0	5	5
Rough Frame-Spacing Cone	8	5 to .5	3	3
Medium Frame-Spacing Cone	8	5 to .5	5	5
Close Frame-Spacing Cone	8	5 to .5	8	8
Extreme Spacing Cone	8	5 to .5	15	15
Slight Curvature Paraboloid	2	2	15	15
Low Curvature Paraboloid	2	2	15	15
Medium Curvature Paraboloid	2	2	15	15
High Curvature Paraboloid	2	2	15	15
Extreme Curvature Paraboloid	2	2	15	15
Saddle	2	2	15	15

Table 5 - Discretisation of the validation plates.

4.10 Comparative Validation

The plates are developed two times each, with the Improper Geodesic Map using both geodesic tracing procedures. The average geodesic curvature per unit length is a measure of the accuracy of the geodesic tracing itself, and therefore is to be zero for a perfectly developed plate. The arc lengths of the developed frames and seams are compared with the exact ones existing in the curved plates, and are to be the same if the development is exact.

Both developments are compared by measuring the distances of the developed plate corners, which should be coincident if the methods are the same. The correction for rigid body motion is not included. Therefore, the given results are more pessimistic than the reality.

4.10.1 Assessing the error lower bound

The plane plate is included not only to validate the simplest case, but also to assess the most basic numeric errors, due to the digital numeric representation in the computer microprocessor (see Table 10).

For this plate, both tracing procedures give the same errors. This suggests that the measured error occurs inside the plate interpolation procedure and the geometric inception of the geodesic segment, offering less than the required 2.5% gain in geodesic curvature, to be optimised. The magnitude of these errors defines the lower bound for the errors to be expected.

Nevertheless, the figures obtained are extremely small, and thus acceptable for a single precision computation as this one (no more than 7 significant digits), and for industry application.

4.10.2 Discretisation effects

The cylindrical and the conical plates validate both tracing methods, being the errors again quite negligible. The 4 conical plates represent the very same surface, only with different frame spacing, to better illustrate the discretisation effects.

As expected, the non-optimised tracing procedure presents higher values of mean geodesic curvature (see Fig. 34). If the optimisation process is allowed to go further the differences found should increase, since the optimisation procedure will be allowed diminishing improvements, which are computationally less efficient.

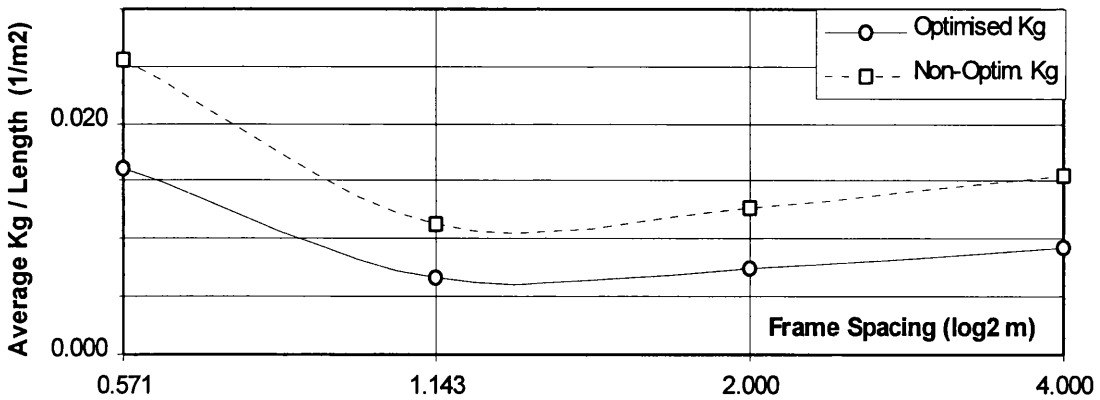


Fig. 34 - Developing the same cone with different discretisation spacings.

The respective developments results indicate that the 2 lower resolution cones had somewhat rough spacings, which could be improved as in the cone with 1143mm frame spacing. Possibly less evident, is the numeric error propagation in the cone with the finer spacing, which being the worst case among all, is nevertheless a quite good development. Too closely spaced frames aggravate the inaccuracies contained in the frames co-ordinates, in the same way as do the denominators approaching zero.

Clearly, there seems to be an optimal spacing around levels usual in the ship design and construction, which being mostly empirical, are expected to capture the best numeric practices. Away from this optimal spacing, increasing it will slowly degrade the accuracy, due to the surface worsening fit, but decreasing it, at least under the unity, will numerically inflate any fairing imperfection.

4.10.3 Effects of the First principal curvature

The geometric geodesic tracing procedure assumes a low rate of change for the osculatory plane along the geodesic, which is normal to the surface. Therefore, when the surface twists along the geodesic path, the procedure behaviour is expected to degrade accordingly. One such common case is when the geodesic direction is not aligned with any principal direction. In this case, even if the second principal curvature is zero everywhere, the rate of change of the osculatory plane increases with the First principal curvature.

The transverse geodesics on a cone or a cylinder are to suffer of such effects. Therefore, the 3 cylinders of the validation set show a clear progression in the average geodesic curvature when K_1 decreases (see Fig. 35). That suggests that the geometric algorithm could improve if it initially aligns the geodesic along one principal direction, or at least with one of a minor

geodesic torsion, instead of assuming a specific initial angle, as is presently implemented.

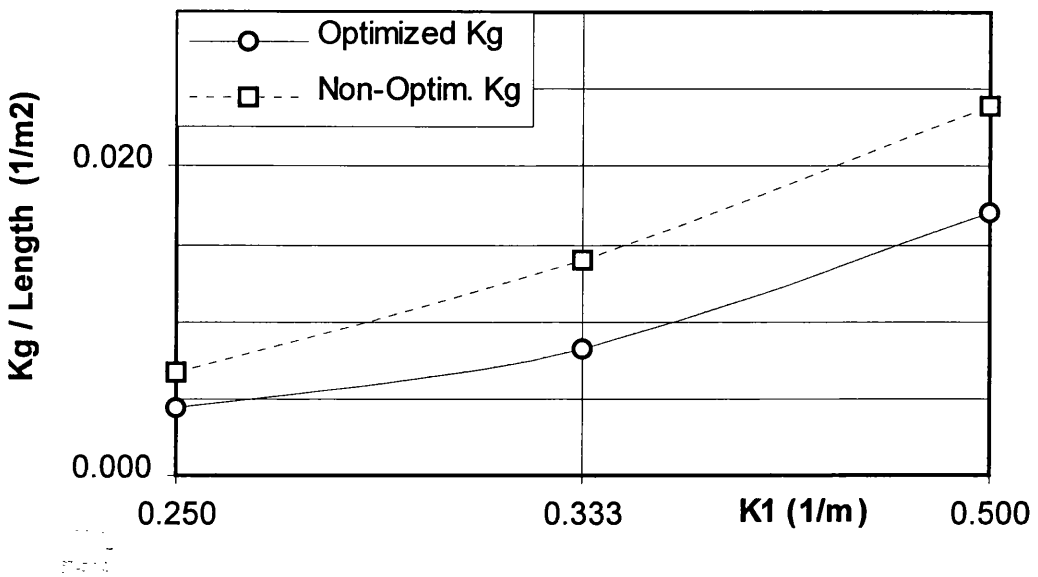


Fig. 35 - Variation of the average geodesic curvature with the First principal curvature, for the 3 cylinders.

Note that even the optimised procedure, due to the presently relaxed optimisation parameters, tends to be too inaccurate for the higher curvatures, which are to be found in the round bilge of small ships.

The non-optimised tracing procedure shows higher mean geodesic curvatures, as expected.

4.10.4 Effects of the second principal curvature

The influence of the second principal curvature on the algorithms behaviour was checked by processing 5 paraboloids, which results were compared (see Fig. 36). For these surfaces the two principal curvatures are equal everywhere, so the first principal curvature have no distinct effects by itself.

As can be seen in the tables of results of these plates' developments (in "Appendix B. - Validation Data"), when the principal curvatures rise the map deviates strongly from an isometric one. However, for the 3 less curved plates, the results still seem quite acceptable.

Note that with the flatter plate, the procedure strikes almost exact geodesics at first chance. This reflects in Fig. 36' curves, where the almost linear increase in geodesic curvature should have a root halfway the two first data points. This

point represents the K_2 maximum value for which it is to expect an almost exact initial geodesic.

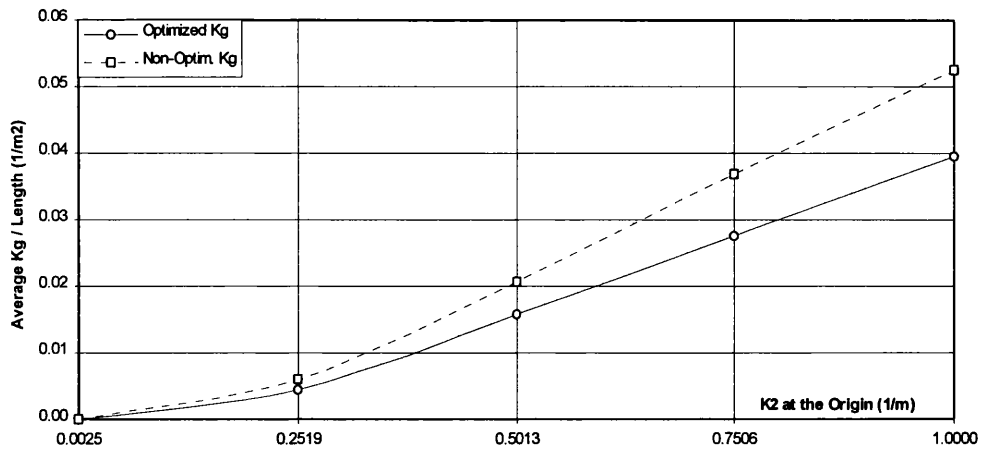


Fig. 36 - Dependency of the geodesic curvature on the principal curvatures, for the non-developable paraboloids.

4.10.5 Computing efficiency

As expected, the optimised procedure expends more computing time than the purely geometric one (see Fig. 37). As seen before, the flattest paraboloid plate behaves like a plane one, showing no iterations at all, and making the two tracing procedures almost indiscernible. For this data point (k_2=.0025m^{-1}), the negligible performance difference between both procedures, is due to the slight overhead of the optimisation process.

Besides this particularity, both procedures perform each almost independently with the principal curvature, which being expected for the geometric one, given its straightforward nature, is a little surprising with the iterative optimisation. The fact is that the steps to minimise the geodesic curvature do not show any particular dependence on the principal curvatures, as shown in Table 6, only denoting a small curvature dependency in the cylinder set.

More relevant to the computing time is the number of plate frames, as shown by the cone set. Thus, larger plates are also expected to require longer processing, since a proper fit would demand more frames and more points in each frame, lengthening the data structures and the time to assess and query them.

With respect to the executable size, the optimised tracing procedure carried to the program an extra 25KB of code and data. This is irrelevant in the present 32bit personal computing environments. However, it was not so for the 16bit computing persisting in the industry for long during this project. Moreover,

when running the development tool inside AutoCAD 11 mixed protected mode and real mode environment).

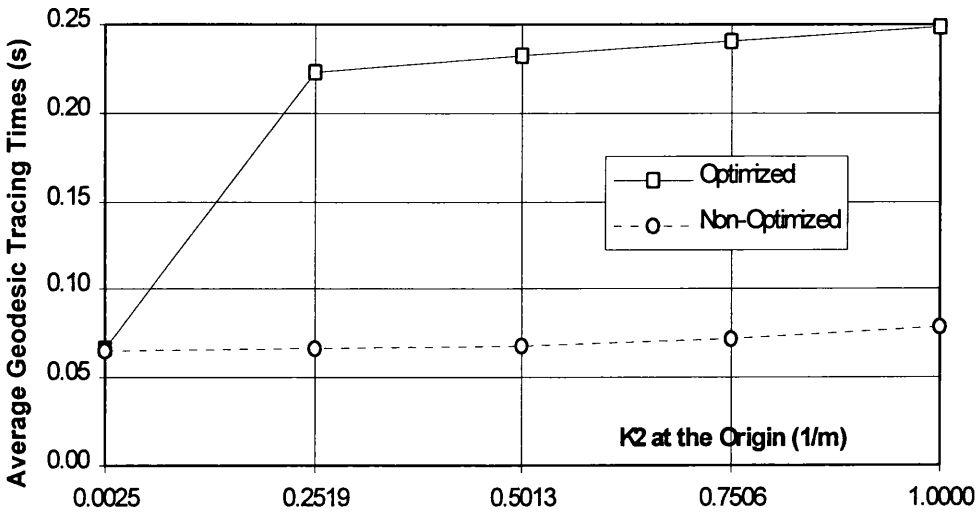


Fig. 37 - Variation of the geodesic tracing speed with the principal curvature, for the non-developable paraboloids.

Test Plate	Maximum K1	Maximum K2	Geodesic Segments	Average Iterations for Optimization of the ... ...Parameter	...Tangent	...Segment	Total Optimisation Iteration Steps
Plane	0.000	0.000	84	1	1	1	84
High Curvature Cylinder	0.500	0.000	154	5	10	50	7700
Medium Curvature Cylinder	0.333	0.000	304	6	11	66	20064
Low Curvature Cylinder	0.250	0.000	484	6	11	66	31944
Rough Spacing Cone	2.000	0.000	434	5	12	60	26040
Medium Spacing Cone	2.000	0.000	724	5	10	50	36200
Close Spacing Cone	2.000	0.000	1159	6	10	60	69540
Extreme Spacing Cone	2.000	0.000	2174	6	10	60	130440
Slightly Curved Paraboloid	0.003	0.003	254	1	1	1	254
Low Curvature Paraboloid	0.252	0.252	254	5	13	65	16510
Medium Curvature Paraboloid	0.501	0.501	254	5	13	65	16510
High Curvature Paraboloid	0.751	0.751	254	5	13	65	16510
Extreme Curvature Paraboloid	1.000	1.000	254	5	13	65	16510
Saddle	1.000	1.000	254	5	12	60	15240

Table 6 - Optimisation iterations for the validation plates.

Chapter 5 - Industrial Application of the Improper Geodesic Map

A software tool resulting from the implementation of the Improper Geodesic Map, was integrated in an actual Computer-Aided Design System in current shipyard use. The shipyard wished to emulate its electrostatic development jig (see Fig. 38). This device is a set of flexible frames at 1/10-scale (see Fig. 39). It is still a very reliable tool, despite expensive in labour. Therefore, its developments were used as the validation standard for accepting the software tool.

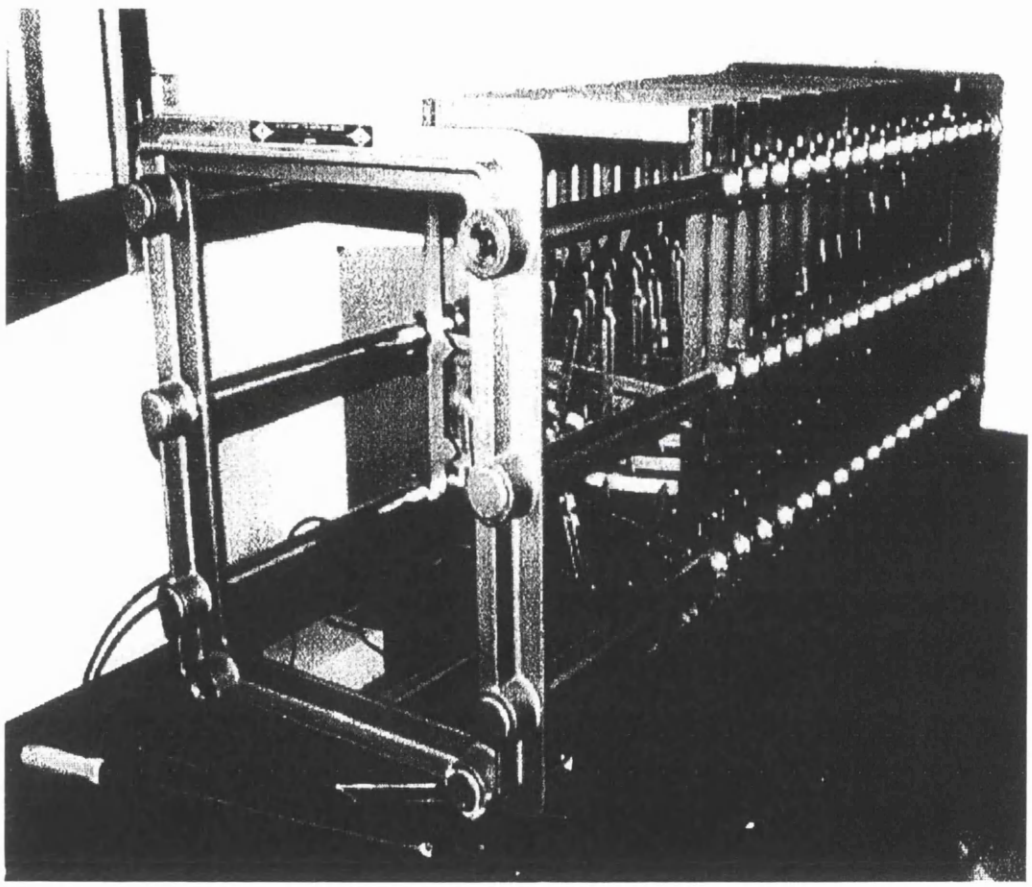


Fig. 38 - An electrostatic development jig, with the electrical stick over the table, in the left. Notice the 1/10-scale frame templates already in place.

The shipyard already experienced problems when using development software on non-developable plates, which otherwise were acceptably processed by the electrostatic development jig, for most cases. Consequently, the Improper Geodesic Map was the implemented solution, making it possible to map onto

the plane even non-developable plates. Hence, the user should be assisted in the judgement of the development validity, since the non-developable plates should not be cut neat to its mapped boundaries, and could even require partitioning in several patches, due to the extensive in-plane strains necessary to their shaping.

The relationship between the intrinsic plate geometry, and the stock material added around its boundaries, can only be established with extensive studies on the actual workshop practices, which are out of scope of this thesis. Nevertheless, this topic is briefly explored, in an attempt to shed some light on it, and to propose some directions for further work.

For introduction purposes, the shipyard Computer-Aided Design system is briefly presented. The shell plate geometry database is discussed, related to the pre-processing of the initial data for plate development.

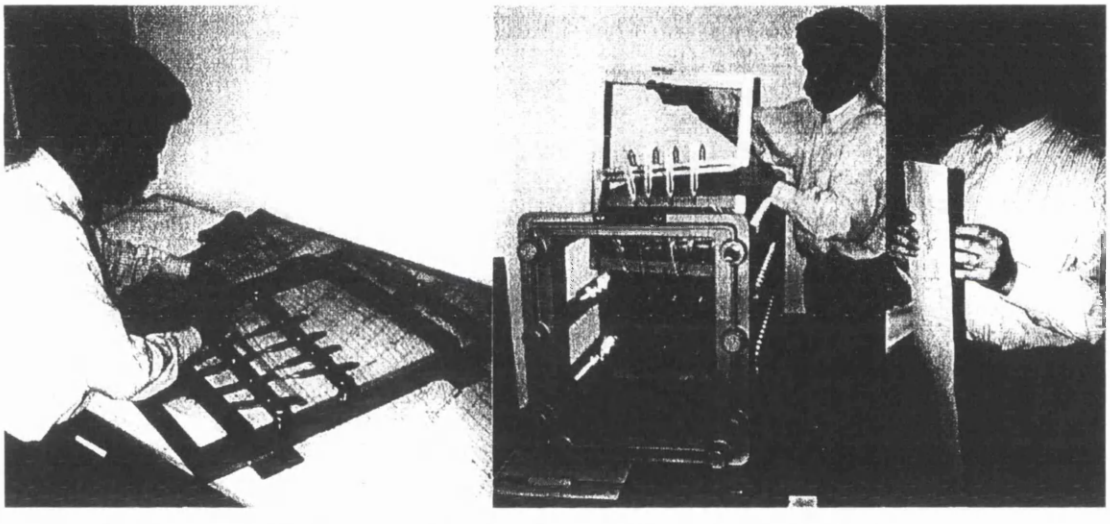


Fig. 39 - Three phases of the manual development procedure, from left to right:

- Shaping the two templates for a frame, over an 1/10 drawing (note the needles in each template, which later will electrically puncture the paper along the plate's relevant curves).
- Placing the templates at 1/10 frame spacing.
- Preparing a metalled paper sheet (which is the plate 1/10-scale model), to be fitted between the templates, and then marked by the sparks made by touching each needle with an electrical hand stick (seen over the table in the middle photo).

Post-processing and user interface issues are described and discussed, being fundamental for the effective use of a development package. Other outputs are the Computer-Aided Manufacturing information and the workshop documentation, both described and analysed here.

After this first industry implementation, a research on software for hull repair support was conducted for another yard. The testing conduced then included

several comparisons with other software and hardware development tools, and is reported and analysed here, in the way of the precedent chapters.

5.1 The Shipyard's Computer-Aided Design System

The development software was integrated in the Computer-Aided Design System running in the shipyard[41]. This system was initially developed by the shipyard personnel, and is running since the beginning of the eighties. Conceptually, it evolved from the traditional loftsman manual techniques, with the continuous work of the in-house project and programming staff.

The shipyard originally used Data General mini-computers with text and graphical terminals. The programming language adopted was the powerful DG Basic.

Now, instead of the minis, they are running local area networks of PCs, running AutoCAD and common desktop applications, abandoning the graphical terminals completely, and most of the text ones.

The Basic was entirely replaced by FORTRAN, AutoLISP, Cliper and C languages.

Aiming at greater system portability, and both independence from proprietary computer makers and restricted system integrators, the designing and production process was to be entirely re-implemented based in *the facto* standards, like:

- IBM PC compatible architecture of personal computers;
- DOS computer operating system;
- Netware local networks of computers;
- AutoCAD drafting software package;
- FORTRAN programming language.

The old system provided less then appropriate depicting techniques in its graphical terminals, making imperative that the visual analysis of hull lines was conducted entirely on paper. That made the big and slow pen plotters become the bottleneck of the system, with reported plotting queues of more then a week.

Presently, with the aid of new methods for depicting, analysing and editing the curve shape, common PCs are being used for the same purpose, much faster than before, and with far better quality[41].

The system of hull geometry definition was originally, and still is, entirely single precision (4 byte real numbers), defining the shell surface by the net of its transverse sections, waterlines, buttocks and knuckle lines.

The system still is not able to assure its coherence, because its not based on the intersecting points between those lines, but in the individual lines themselves. If some editing is made on one of those lines, the others aren't changed accordingly by the system, unless by user interaction prone to error.

This, and mostly the lack of a consistent service for interpolation of points and curves over the hull surface, made it extremely difficult to stabilise the requirements for the development module, and accounted for the great majority of the software development time. This interpolation service, instead of unique and centralised, was in fact replicated in several parts of the system, making it vulnerable to version discrepancies and configuration errors.

All the curves have the same information structure: they are cubic parametric splines, defined as a sequence of points with unit tangents, where a knuckle point is represented as a repeated point with different tangents.

Naturally, the internal representation of curves in the development tool follows this same structure. It was designed to operate in a batch either way, as a DOS executable, or interactively, inside the AutoCAD graphical environment, as an ADS application. In both cases, the memory size of code data were to kept roughly under the 500KB figure, because no only of the 640KB limit of PC's 16bit real mode, but also because under AutoCAD11, running in 4MB RAM 386 PC's, there were scarce resources to play with.

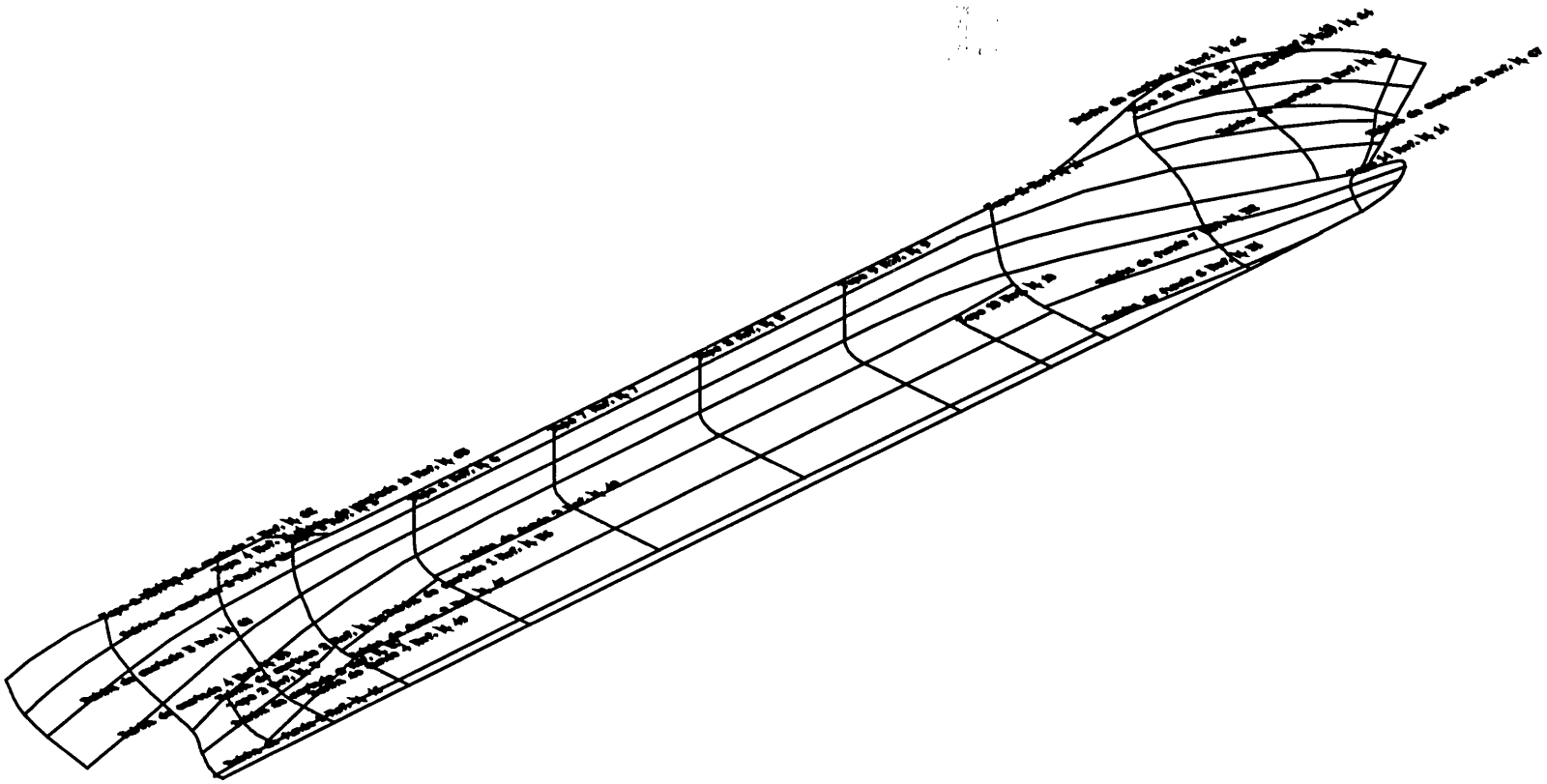


Fig. 40 – A typical mesh of seams and butts (with the labels not readable given the extreme scale required by A4 format).

5.2 Shell Plate Definition System

Surfaces are defined in full-scale through its curves in this Computer-Aided Design system, which uses an interactive smoothing and analysis method for individual curves (see Fig. 41). For each section curve all sudden variations of curvature are to be scrutinised to decide if it is a feature of the local hull shape or if it is an unfair spot. In the end the global behaviour of the curvature profile could be appreciated and compared with the other sections in the vicinity.

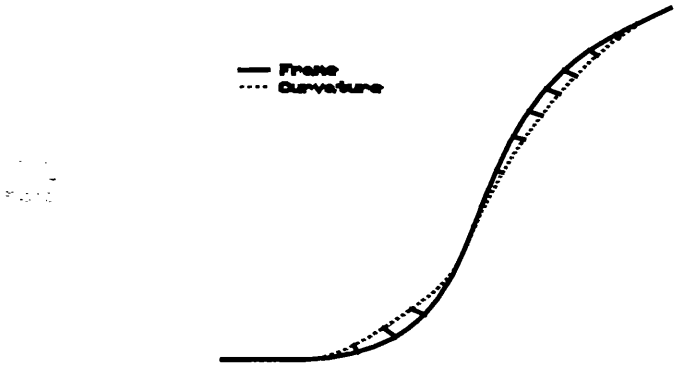


Fig. 41 - Curvature distribution along a hull line being faired.

This procedure is not enforced by the system, it is only expected that the user follows these guidelines. The surface fairing procedure also relies much on the draftsman's experience, because it is necessary to judge the shape of the surface from these curves. Note that a region of unfair shape, maybe with just a single wrinkle, can be bounded by four perfectly fair lines (see Fig. 42). Note that the problem in this plate is that the user wasn't aware of the existence of a knuckle within the designed plate boundaries. This can happen if the system depends to much on user intervention.

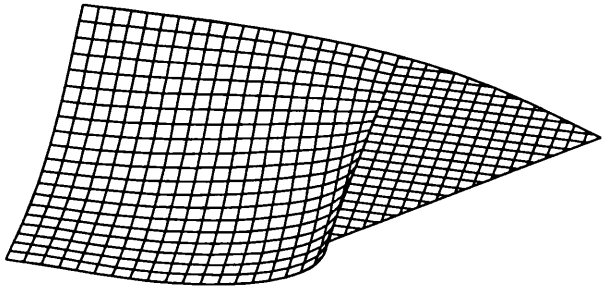


Fig. 42 - A non-smooth patch can be bounded by smooth lines.

Moreover, a collection of hull curves, usually smoothed to accuracy within the centimetre range (some millimetres), cannot guarantee that smoothness on the surface itself, because of error propagation in the interpolation process.

Therefore, the smoothing process, being entirely dependent on user intervention, and also being made one curve at a time, is not able to assure a consistent level of fairing quality. In fact, during the removal of big shape errors, it can introduce small localised ones, out of the user's perception, due to the absence of tools to perceive the entire patch as a proper surface.

In other hand, the design process of seams and butts, bounding every plate, is usually conducted in this shipyard with little concern to accuracy, because of the manual lofting experience of the users, which still expect late adjustments, were required. Adding to this the common use of excess material around the plate boundaries, providing for in-place adjustment, it is understandable why the three-dimensional seams are frequently found some centimetres out the shell surface.

5.3 Plate Surface Model and Data Pre-processing

The plate boundaries are the seams and the butts. In this system, the butts are even in the abscissa, so they are interpolated very precisely, as common transverse sections. However, the seams are not so accurately traced, because they are only roughly defined, and can be found way out of the shell surface, sometimes over the centimetre range.

Instead of relying on those ill-behaved seams, the development program uses projections of them onto the interpolated transverse sections, using a proximity algorithm. For each section this algorithm computes its closest point to the misbehaved seam, and then takes it as the seam projection onto the section. The algorithm is as a second-degree iterative optimisation method, with a penalty function when outside any of the lines spans. It is quick, reliable and accurate, but a little lengthy in the programming, if not using available code libraries, as the author did.

Underlying this system, the lines-based surface representation provides no direct access to the surface's intrinsic geometry. That can only be done by building a surface's model on top of the lines system, through which one can get not only the surface points, but also tangents, normals, curvatures, areas, etc.. This model must be as accurate and efficient as possible, therefore it has to stay formally close to the original information structure to avoid the respective overhead.

The plate surface is defined originally as a set of interpolated transverse sections, each one starting and ending at the bounding seams. Then, those curves serve as a basis for the interpolation of longitudinal curves, making for a set of lines, which includes the seams, and is non-collinear to the transverse sections. These longitudinal lines are traced in a way, that at along any frame, the longitudinal lines and the seams are evenly spaced, as shown in Fig. 43.

Note that the longitudinal set of surface lines is obtained by a two-fold interpolation on the original data, which is not the best practice, at least for the point of view of numerical accuracy and performance. Nevertheless, it is the only practical option, given the nature of the data provided by the system.

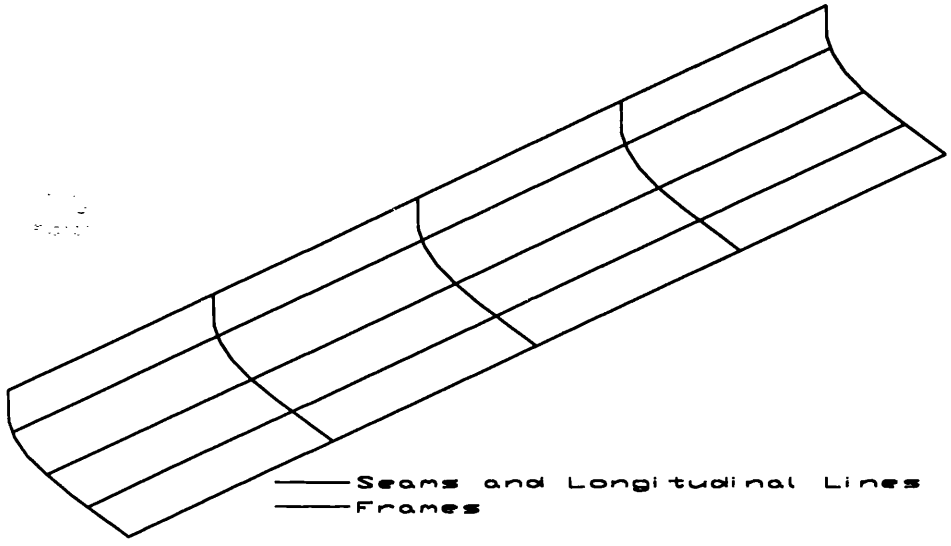


Fig. 43 - The defining mesh of the plate surface.

The mesh formed by this set of curves and by the transverse sections makes the data framework of the surface model. For the sake of the model's integrity, further operations inside the plate are always done on the mesh data by the mesh procedures, and no more system data is used, so the model is in fact a self-contained subsystem.

To conclude the surface formal model, it was necessary to implement a set of procedures for evaluation of points, tangents, normals, curvatures, elemental areas, curves lying on the surface, etc..

Along any mesh line, it is simple to interpolate a surface tangent just by differentiating the line at the given point. Between two parallel mesh lines, a surface tangent can be interpolated from the tangents on those lines.

Let a surface tangent $\mathbf{t}_\theta$ make an angle θ with one of the mesh lines, say line **a.** Composing the tangent from the unit tangents $\mathbf{t}_a$ and $\mathbf{t}_b$, interpolated on the closest mesh lines **a** and **b** one gets:

$$t_\theta = \cos\theta.t_a + \cos(\arccos(t_a.t_b) - \theta).t_b \qquad (90)$$

To compute a normal vector it is enough to evaluate the external product of two of such non-collinear unit tangent vectors.

The procedures for evaluating the curvature, the elemental areas, etc., uses the common formulae, already presented in Chapter 2.

The tracing procedure for a discretised line, lying in the plate, had to be accurate but simple. The surface's curve is discretised as a cubic spline, like the ones of the existing Computer-Aided Design system. Each spline segment starts and ends in points interpolated over the mesh lines, thus there aren't point interpolations outside the mesh lines, in all the processing after the construction of the plate's mesh, keeping the point interpolation simple and accurate.

The spline tangents are defined as the surface tangents aligned with the curve, at the starting and ending points of each curve segment. This was quite simple to implement, due to the already available surface model infrastructure.

Obviously, the accuracy of the method depends on the distance between consecutive mesh frames, however it is not possible to increase the accuracy effectively beyond the centimetre error range, because the data errors are already in that level.

From the Numerical Analysis, we know that in a division the error propagation grows as the reciprocal square of the denominator. The author experienced that when mesh curves are interpolated too close, say under the decimetre figure, the frame interpolation errors become quite amplified, building up bumps in the surface, which where otherwise inconspicuously small.

Keeping the step between mesh curves in the range of 200 to 500 millimetres, usually smoothes those bumps, since the same absolute errors of the co-ordinate evaluation are divided by larger distances in the slope computations.

5.4 Implementation of the Development Software Tool

When the software requirements were established, the shipyard's drafting rooms were just starting to use AutoCAD10. Moreover, the target machines were 2MB RAM 16 MHz 386Sx and 20MHz 386Dx. The development tool was to run inside the AutoCAD environment, which translated to critical constraints on the tools' size and speed.

The speed, lower memory occupation, and reliable behaviour made the non-optimised geodesic tracing procedure the solution to adopt. Presently, the

previous constraints are partly overcome, but the little accuracy gain provided by the optimised tracing procedure are still offset by its slowest operation.

Besides, the designers find quite satisfying the present development procedure. Indeed, they seem to prefer improvements in the post-processing rather then in the development procedure itself. Therefore, the optimised procedure is still discarded for industry application, being still quite useful in research work, as an accurate reference.

5.5 Plate Data Post-Processing

Any development method for industry applications is to help the users not only in the cutting, but also in the shaping operations. The forming process, despite being a somewhat dirty and heavy work, do require skilful people, due to the complex behaviour of materials, like spring-back, heat distortions, etc.. Along with the proper training, this people should be given adequate documentation, not restricting to template drawings and other full-scale mock-ups.

Today it is common that the development software provides strain maps, rolling lines and even heat lines[1]. Given the requirements posed by the yard hull workshop, the only strain measures provided where along the seams and but. Other forming information was not desired in this first phase, to allow the shop people to adapt, and maybe to rethink the specification the new documents should obey.

In the drafting room, there were also particular needs to be addressed. The designers need to be prescient of the handling difficulty presented to the shop people by each plate, before they settle the final seams and buts design. These concerns are rooted in the difficult forming of the non-developable hull regions.

The usual technique dealing with spots of pronounced second principal curvature, is to design finer plates locally, so any single one have no exaggerated amount of accumulated K_2. To quantify that "exaggerated amount" is clearly out of scope for this thesis. The fact is that the designers, which are rather experienced, feel confident to rely exclusively on their intuition, and if things eventually go wrong, possibly in the very hull shop, they decide to further partition the plates, and make the whole plate as an assembly of smaller ones.

In a first attempt to support the designer decision process, R_2 contour plots were included on the drawings of the mapped plates (see Fig. 44). The decision to plot $R_2 = 1/K_2$ instead of K_2, was concerned with the background of the people in the design rooms, most of which had no appropriate analytical

education. Therefore, since a radius is a simpler concept than a curvature, the option was made for R_2.

The plots with the higher curvature figures are both the most likely to trouble both the map, at the design room, and the forming procedure, at the workshop. Inside the problem-areas the R_2 field display either extreme values or contour curves too close to each other (denoting intense gradients). Isolated maxima are also plotted, independently of its the actual values, because they are helpful to spot the unfair bumps.

The plots proved difficult to understand by the yard designers, since they were not trained at all, and neither had the desirable analytical education. Nevertheless, along with strain maps it is for the time being the only help they get, besides directly inspecting the hull geometry.

The depicted strain values refer to each plate curve. In the case of extreme deformations the designer can easily perceive the problem. Naturally, he should be aware of the allowable figures of the intended construction materials.

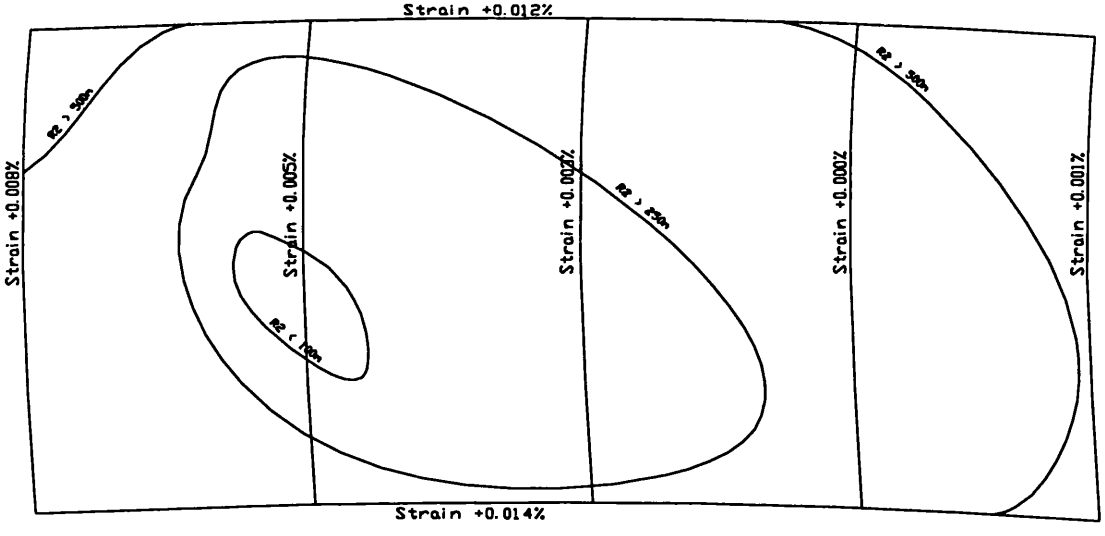


Fig. 44 - The development of a plate, showing strain figures and R_2 contour plots.

Interestingly, these plots offer two opposite comments:

- For the first, they were difficult to interpret because they appear quite confusing, showing many small spots and irregularities in the curvature.
- For the other, they present the designer with the fairness reality of the hull, making this a powerful tool to assess the fairing quality, and to eventually select areas for re-design.

Presently, this is one of the areas where some progress is to be made, either in the yard's Computer-Aided Design rooms, with a better enlightenment of the tool usage, and by the research and development people, as the yard's feed-back is incorporated in the tool's requirements, for posterior improvements.

5.6 Discussion of Results

A research on software for hull repair support[2], was conducted after this first industry implementation of the Improper Geodesic Map. In the eighth Chapter of the resulting report[2], the non-optimised variant of the method was used in development comparisons with both the electrostatic development jig, and a commercial software package, the KCS' STEERBEAR (presently known as TRIBON). The data collected by the research will be analysed here.

It was not possible to conduct both comparisons over the same data set, since the collaborating shipyards could only provide data about the current works of their own. Consequently, two different sets of plates were used, one developed by the STEERBEAR package and the Improper Geodesic Map, and the other developed by the electrostatic development jig and the Improper Geodesic Map.

Each plate set contains only three specimens. These were chosen to be challenging, but not so far from developable that the map results could become unpredictable. Therefore, none of the test plates were either developable or of pronounced second principal curvature. The test was focused on the curved plates from the big full hulls, which are the most common in the line of business of the involved yards.

(All measures in mm)

Plate	Development Procedure	Corner A		Corner B		Corner C		Corner D		Diagonals Length	
		X	Y	X	Y	X	Y	X	Y	AD	BC
3E	Non-Optimised Geodesic Map	0	0	-564	2763	7820	7	7491	2201	7807	8826
	Electroestactic Development Jig	0	0	-562	2761	7800	12	7465	2194	7787	8799
	Corner Distances:	0		3		21		27		Differences:	21 27
	Average Distance = 13										
6E	Non-Optimised Geodesic Map	0	0	31	2946	9952	-32	9843	2816	10238	10358
	Electroestactic Development Jig	0	0	35	2948	9941	-13	9837	2820	10233	10341
	Distances:	0		4		23		7		Differences:	5 17
	Average Distance =										
4F	Non-Optimised Geodesic Map	0	0	-62	2719	7979	-37	7697	2741	8171	8500
	Electroestactic Development Jig	0	0	-61	2719	7978	-53	7697	2748	8179	8494
	Distances:	0		1		16		7		Differences:	-9 6
	Average Distance = 6										

Table 7 - Results for the 3 ship plates, mapped into the plane by the electrostatic development jig, and the Improper Geodesic Map, as implemented[2].

The study compared the lengths of the diagonals of the developed plates, and the corner co-ordinates (see Table 7 and Table 8). The developed co-ordinates

were not corrected for differences in rigid body motions. Hence, de found differences exaggerate the actual figures, so they are conservative.

The electrostatic development jig is accurate only to the centimetre, making the millimetre digit irrelevant. This is so because it is operated at 1/10-scale, and the measuring is done down to the millimetre. For the other side, the STEERBEAR package, and the implementation of the Improper Geodesic Map, have an expected accuracy around the millimetre.

In Table 7 the average distances between the developed corners of each plate, and also the differences in the developed diagonals, are quite reasonable, because the inaccuracy of the electrostatic development jig envelopes reasonably this deviations.

In the other hand, comparing the results from STEARBEAR and the Improper Geodesic Map, the Table 8 reports a close match between both programs, naturally due to the higher precision of computer processing.

(All measures in mm)

Plate	Development Procedure	Corner A		Corner B		Corner C		Corner D		Diagonals Length	
		X	Y	X	Y	X	Y	X	Y	AD	BC
602-3-2	Non-Optimised Geodesic Map	0	0	15	2701	8980	15	8157	2748	7898	9359
	STEERBEAR's	0	0	15	2704	8981	15	8155	2750	7896	9360
	Distances	0		3		1		3		Differences: 2	-2
	Average Distance = 2										
603-3-1	Non-Optimised Geodesic Map	0	0	3	2206	7032	0	7110	2209	7445	7368
	STEERBEAR's	0	0	1	2206	7031	-2	7110	2208	7445	7369
	Distances	0		2		2		0		Differences: 0	-1
	Average Distance = 1										
603-3-3	Non-Optimised Geodesic Map	0	0	14	2320	7120	16	7204	2324	7550	7471
	STEERBEAR's	0	0	15	2320	7125	15	7207	2323	7553	7474
	Distances	0		1		5		3		Differences: -2	-3
	Average Distance = 2										

Table 8 - Results for the 3 ship plates, mapped into the plane by STEERBEAR and the Improper Geodesic Map, as implemented[2].

Evidently, this research confirmed the preliminary testing of the Improper Geodesic Map, using the theoretical surface forms. Moreover, at the time this report was being published, the contracting shipyard had already validated the development software tool, by conducting internal tests from which the data was not made available. This testing was reported to the author to be entirely satisfactory, showing fully acceptable agreement of the plates developed by the system and the ones developed by the electrostatic development jig. The found deviations were told to be less than 10 millimetres, after discounting the differences in rigid body motion. Since then, the software as been in daily use, and presently, there is already sailing ships, which production data was entirely processed by the development software.

Chapter 6 - Conclusions

A survey of the published materials on surface development and developable surface definition has been undertaken, covering both the mathematical background and the other published works since 1963.

The research relating to surface development has been segregated predominantly into two fields:

- mathematics on one side, relying on the rather involved notations of Differential Geometry; and
- practical engineering on the other, by approaching the problem in more empirical ways.

The application of digital computers in shipyards made it possible for engineers to implement more complex development methods, closer to the theoretical base of Differential Geometry. At present, after the publication of several successful methods, such as Nolan's[30], research is shifting towards the processing of non-developable plates, which has still to show results[34].

From the mathematical definition of geodesic lines, a new map has been conceived in this study, to allow for the processing of any plate surface, even if not developable, as is frequently the case in ship hulls. This map has been implemented as the development tool of an actual software package, and has been in shipyard use since 1995.

6.1 The Improper Geodesic Map

This new map (presented in CHAPTER 4 - CONCEPT AND IMPLEMENTATION OF AN ALGORITHM) uses geodesic lines to map points and vectors onto a plane. Since there is always one (and only one) geodesic passing through every point on a surface in any particular direction, this the map will always produce the plane image of any conceivable point or vector. However, in cases of extreme second principal curvature (K_2), this mapping procedure can produce the same image for different points (see Fig. 33). Since it is applicable to non-developable surfaces, it is not always a one-to-one mapping of points in the curved surface into the plane, i.e. such mapping is not unique. In the mathematical sense, this means it is not isometric and therefore it is not a geodesic map, consequently it has been termed an "Improper Geodesic Map".

In industrial use, this mapping procedure can only be disrupted by these extreme K_2 values, as the curved plate will be impossible to form because

different elemental areas will map into the same spot in the plane, resulting in a folded outline (see Fig. 33). To correct this, one can use the traditional technique of dividing the plate into smaller ones, reducing the accumulation of K_2 to be processed by the map in each run. Nevertheless, as it takes abnormal values of K_2 to produce such cases, this should not happen in practice, and in fact the shipyard never requested software support related to this.

The possibility of a fairing error in the plate surface, results in a localised extreme value of K_2. The software exposes such cases both by folding the plate outline onto itself, which is a conspicuous error, and by letting the user appreciate the R_2 field across the plate (see Fig. 44).

The decision to plot $R_2 = 1/K_2$ instead of K_2 was concerned with the background of the people in the design rooms. Most of the designers had no appropriate analytical education, therefore the curvature radius, which is more intuitive than the curvature itself, was the preferred measure of shape variation.

Inside the problem-areas the R_2 field will display either extreme values or contour curves too close to each other (denoting intense variations). Isolated maxima are also plotted, independently of its the actual values, because they are helpful to spot unfair bumps.

If the plate is correctly faired and mapped, the plot of the R_2 field still offer the user an alternative way to perceive the surface and possible forming difficulties. One such difficulty is due to improper straking leading to disparate curvatures in the same plate, like a plate shaped partly as a saddle and partly as cylinder. (In this particular situation the user will find the R_2 field restricted to saddle region.) This sort of problems is handled by partitioning the plate into regions where the curvature has a more uniform variation.

It is important to note that if the designer doesn't have the proper background to understand the contour plots, this tool is of no use, and could even be detrimental. In such cases, it is probably better to freeze the CAD layer in which these plots appear, so that the user is not confused by data unintelligible to him.

The only plates in which the map has been found to fold have been theoretical ones with extreme K_2 values. These plates were purposely conceived to stretch the mapping procedure to the limit. So far, there has been no attempt to establish criteria to predict the folding of a plate. These criteria would be rather complex and of little practical value for the contracting yard, since the software already copes with the problem by first exposing it and then supporting the iterative redesign of seams and butts. Note that the most obvious candidates as parameters for such criteria are the accumulated curvatures in the plate surface, which indicates the complexity and computational load that might be involved.

Comparing the software solution developed with the alternative of the electrostatic development jig, it is not only more economic in labour costs, but also provides the user with information about the actual plate curvature and strain, assisting him in the detection and evaluation of the problem areas of the hull. This allows for further economic gains, by improving the design for production of the plate straking definition.

As discussed in CHAPTER 4 - CONCEPT AND IMPLEMENTATION OF AN ALGORITHM, each plate point is mapped on the developed plane by rectifying (straightening) the geodesic traced through it within the curved plate. Therefore, the central procedure of the mapping process is the tracing of geodesics, and the efficiency and accuracy of the resulting map is largely controlled by it. Branco's discrete algorithm for geodesic tracing[13] was reused, since having been found to be both accurate and efficient enough for hand drawing, it should be even more so for computer tracing. In fact, this is confirmed in Fig. 34 and Fig. 37. However, as Branco's procedure has no accuracy control parameters, another tracing method has been derived from it, just by coupling an optimisation procedure, to minimise the geodesic curvature. As a result, this derived tracing procedure is computationally heavier, so it is less suited for field use; but since it is has controllable accuracy, it serves well for research purposes.

The map is made in two sequential steps:

1. the tracing in the plate
2. the rectifying in the plane.

These are the only error sources inside the map, so knowing the error of one, the error of the other can be deduced from the observed total error. Moreover, the difference between the final errors of both mapping procedures is a measure of the error content of Branco's tracing procedure, even if a little underestimated.

The optimised tracing procedure requires an accuracy measure for the traced geodesics. The proposed measure is the ratio of the geodesic length to the accumulated geodesic curvature. As this represents an average geodesic curvature, it is termed "K_g^{ave}". A reasonable figure for this, related to the typical plate dimensions, expected maximum curvature and workshop accuracy (which is about one centimetre) is (Eqn.89):

$$K_g^{ave} < 0.020 \text{ m}^{-2} \tag{89}$$

This value is less significant in the case of non-developable plates, since their map requires significant strains of an arbitrary nature, possibly above this figure.

To test the tracing procedures and the maps, it was decided to use basic geometric shapes as data, instead of practical plate cases, which would be too complex for initial assessment, and could bias the results towards the particular examples assessed. The set of basic shapes considered included both developable and non-developable surfaces (see Table 4). The former where represented by cones and cylinders, and the latter where represented by paraboloids. A saddle and a plane complete the set, being representative of the universe of simple surfaces.

To assess the discretisation effects, the developable surfaces contained different discretisation distances (see Table 5). This study confirmed that the spacings used by the shipyard in the hand drawing system are close to the optimum for accuracy (see Fig. 34). Since the CAD system was specified to optimise of the previous hand-drawing system, maintaining many of the old techniques, the coincidence of this optimum in both systems is understandable. Note that both systems represent the surface by section lines, instead of surface patches, and therefore both rely on similar discretisations. Possibly, another surface representation scheme would have different optimum spacing.

To assess curvature effects, the test surfaces where made with different principal curvatures. This study suggests that for small curvatures, the method behaves extremely well, as if there is no curvature at all and the surface is plane. At around the $0.25m^{-1}$ curvature level, the curvature starts to affect the geodesic tracing procedure, and from that point onwards the tracing error increases regularly. Nevertheless, this increasing error is a negligible fraction of the overall curvature of the geodesic line, as can be seen by comparing the relative scales of both axes in Fig. 35 and Fig. 36. Therefore, the numeric contribution to the overall tracing error can be relatively important.

The results compiled in Appendix B. - Validation Data, demonstrate the good fit of both optimised and non-optimised maps to their true developments, for the developable surfaces, as shown by the strain figures. Note that the differences between both maps are of the same order as the difference between the optimised maps and the true developments. Therefore, the errors made in the geometric tracing are about the same order as the errors in the rectification procedure.

Fig. 35 and Fig. 36 show some expected results, like the lower curvature developable surfaces having more accurate developments than those with higher curvature, the same also applying to non-developable surfaces. However, not so evident are the lower distortions and geodesic curvatures of the very low curvature paraboloid, in comparison with developable surfaces which, despite the absence of second principal curvature (k_2), do have much higher first principal curvature (k_1). Once more, mapping accuracy is clearly dependent on first principal curvature.

The computing efficiency of these methods is measured against the principal curvatures, as demonstrated in Fig. 37. This is more relevant in the interactive environment of the CAD package in which the development tool is used. As expected, the optimised procedure expends more computing time than the purely geometric one. The flattest paraboloid behaves like a plane, showing no iterations at all, and making the two tracing procedures almost indiscernible. For this data point (k_2=.0025m^{-1}), the negligible performance difference between both procedures, is due to the slight overhead of the optimisation process. Besides this peculiarity, both procedures perform almost independently of the principal curvature, which might be expected for the geometric one, given its straightforward nature, but is a little surprising in the case of iterative optimisation. The fact is that the steps required to minimise the geodesic curvature do not show any dependence on the scale of principal curvatures, as shown in Table 6. Therefore, perfectly developable data sets are mapped with negligible errors, and the development software tool is suitable for the present accuracy standards. This means two things, namely that the tool is reliable in its:

- functioning - meaning that it accepts the current accuracy levels of the data used by the contracting shipyard, including the common deviations from the developable condition, which can disrupt other software procedures.

- and its results, meaning that it produces accurate results, regarding the data quality, including the cases of plates which are "almost" developable. Note that for extremely non-developable plates, the development results are intrinsically meaningless.

6.2 Industrial Implementation of the Map

The mapping procedure has been implemented in a software tool, used as part of a shipyard CAD package, and as such, has been in use since 1995. From the first applications in the yard, it proved as an effective and efficient tool, and was enthusiastically accepted by the personnel. Nevertheless, since it is included in a complex CAD package, all of it relying on surfaces defined by section lines, one should guard some care against the possible use of it with inaccurate data. Such case can easily happen just by using too few nodes in the discretisation of the splines that represent the section lines, the seams and the butts. A seam can be apparently well defined by placing two consecutive nodes 10 metres apart, but a particular seam point can be 20 centimetres apart from the hull surface. Note that the hull surface is defined by a set of splines unrelated to the seams in the present Computer-Aided Design specification. The integrity/coherence of this geometric database can only be assured by filtering every user attempt to define new lines or new points on the

hull surface, running special software. For the moment, this software does not formally exist in the system, despite being part of the development tool.

During the development of the software and the training of the shipyard personnel, several informal tests where made, but it was not possible to retrieve the reference data developed by other tools. Naturally, the shipyard conducted internal approval tests after delivery of the software, but the results where not formally disclosed, only the outcome of acceptance and its introduction into the routine operations. This testing was reported to the author as entirely satisfactory, showing acceptable agreement of the plates developed by software and the ones developed by the electrostatic development jig. This agreement means less than 10 millimetres deviation, after discounting the differences in rigid body motion. Since then, the software has been in daily use, and presently, there are already ships sailing, for which production data was entirely processed by the development software.

Further research on software for hull repair support[2] has been conducted after this first industry implementation of the Improper Geodesic Map. In the 8th Chapter of the resulting report[2], the non-optimised variant of the method was used in development comparisons with both the electrostatic development jig, and a commercial software package, the KCS STEERBEAR system (presently known as TRIBON).

It was not possible to conduct both comparisons over the same data set, since the collaborating shipyards could only provide data about their current projects. Consequently, two different sets of plates were used, one developed by the STEERBEAR package and the Improper Geodesic Map, and the other developed by the electrostatic development jig and the Improper Geodesic Map.

Each plate set contains only three specimens. These were chosen to be challenging, but not so far from developable that the map results could become unpredictable. Therefore, none of the test plates were either developable or of a pronounced second principal curvature. The test was focused on the curved plates from big full hulls, which are the most common in the line of production of the yards involved.

The study compared the lengths of the diagonals of the developed plates, and the corner co-ordinates (see Table 7 and Table 8). The developed co-ordinates were not corrected for differences in rigid body motions. Hence, those found differences are exaggerated.

The electrostatic development jig is accurate only to the centimetre, the millimetre digit being irrelevant. Note that as it operates at 1/10-scale, the actual measurement is to the millimetre. The implementation of the map compared favourably with results from the electrostatic development jig[2], which was shown to be less accurate. The electrostatic development jig, due to

its working scale, is in fact an error amplifier, which is not the case for the software, which works at full-scale.

Table 7 presents the average distances between the corners of each developed plate, and also the differences in the developed diagonals. These are quite acceptable, because the inaccuracy of the electrostatic development jig, which is still the standard for development accuracy, reasonably envelops these deviations. On the other hand, comparing results by STEARBEAR and the Improper Geodesic Map, Table 8 demonstrates a close match between both programs, given the higher precision of computer processing.

This evidence confirms the testing of the Improper Geodesic Map using theoretical basic surfaces.

6.3 Limitations and Improvements of the Map

Traditionally, fairing is mostly understood as a hydrodynamic necessity, but in fact, the complete fairing of the hull surface has the utmost impact on the development and effectiveness of the workshop. Even the most inconspicuous bumps affect the accuracy of the map and would be blindly reproduced by the workshop, with a penalty in labour costs. Hence, the development software should be complemented with tools to assist the identification of problem areas in the hull surface. In the case of this particular software, the depiction method adopted was a plot of the second principal curvature maxima and contour lines onto the mapped plate. It was not intended to replace the proper fairing tools, but only to offer the user a last chance to realise possible errors still existing in the design.

Besides proper non-developable features of the hull shape, the troubled areas will appear with high values or intense gradient (closer contours) of the second principal curvature. Therefore, the map can benefit from improved hull smoothing:

- by removing the bumps, the plates can have lower curvature levels, which as seen before (see Fig. 35 and Fig. 36), improves the behaviour of the development method.

- once having removed even the less conspicuous frame deviations, the accuracy gains allow for an increase in the frame spacing, reducing the amount of data, and thus the processing time and computer storage requirements.

Instead of assuming an initial angle along the bigger plate dimension, as is presently implemented, the geometric algorithm can also be improved if it

initially aligns each geodesic with one principal direction, or at least with one of minor geodesic torsion. With this strategy, the mapping geodesics would have smoother paths; and their discretisation errors will decrease.

Note that even the optimised procedure, due to the presently relaxed optimisation parameters, tends to be inaccurate for the higher curvatures which are to be found in the round bilges of small ships. It is also possible to make the stop criteria parameters depend on the plate's intrinsic geometry, which will allow the development procedure to adapt to wildly different cases. Alternatively, this accuracy improvement will allow for more relaxed spacings in the discretisation of the geodesics, improving the processing time and the storage requirements.

For the current implementation, the program size in the computer memory is not an issue, since its size is around 500 Kbytes, with PCs with 4 to 16 Mbytes RAM as the targeted computing platforms.

It should be noted that the shipyard is presently using (or at least testing) commercial software for hull fairing and straking. This is an acknowledgement of the inadequacy of the existing one, to much inspired in old practices of hand-work to exploit appropriately the powerful tool which is a computer.

Chapter 7 - Prospects for Further Work

After much experience with almost entirely developable hulls[4], and studies proving substantial cost savings when developable surfaces are fitted instead of compound curvature ones[5], most ship persist to be projected with extensive non-developable areas in the hull, so the plane mapping of non-developable plates continues to be an interesting subject for research.

Continuous advances in the capabilities of plate forming machinery could lead to a situation where the plate's developability is not an issue as it is today. Note however, that the workshop equipment still missing is now an imminent reality, because all the basic technologies are already available: fast digital computers, accurate modelling of the material deformation due to forces and heat, instrumentation for accurate monitoring, sophisticated control devices like robots, and an important base of experience with numerically controlled machines of disparate types.

The deformation of non-developable plates can be improved, by minimising the in-plane strain energy. The material behaviour could also be modelled more accurately. Thus, the purely geometrical methods originally intended for developable plates, as the one presented here, could become less relevant in Computer-Aided Design and Manufacturing packages.

As the survey of this thesis suggests, the methods for surface design are still in evolution, mostly the ones intended for the design of developable

---

[4] Since the `60's, with rowing dinghies, sailing craft, cargo launches, trawlers, tug boats, and the 20 *panamax* ships built by the Burmeister & Wain shipyards, between 1980 and 1985[1].

[5] In the `60's, a US Navy study on the plating production costs for two big carrier sponsons, found differences of about $100,000 between compound curvature and alternative developable designs[8].

surfaces. Designing surfaces is a task heavily dependent on the quality of the user interface: both the control and the rendering (the input and the output). Virtual reality technologies are available, and are extremely effective in rendering complex shapes. However, the necessary control methodologies to full exploit virtual reality environments seems to be still in the infancy.

References

1. Letcher, John S. Jr., "Lofting and Fabrication of Compound-Curved Plates", Journal of Ship Research Vol. 37, No. 2, June 1993, pp.166-175.

2. Secção Autónoma de Engenharia Naval/Instituto Superior Técnico, Lisnave, Estaleiros Navais de Lisboa, S.A., "Sistema de Computação Gráfica para a Reparação Naval", Projecto STRIDE STRD/TIT/120, Relatório Final, April 1996.

3. Lamb, Thomas, "Shell Development Computer Aided Lofting - Is There a Problem or Not?", The National Shipbuilding Research Program, 1993 Ship Production Symposium Sponsored by the Hampton Roads Section SNAME, pp.9.1-9.17.

4. Nørskov-Lauritsen, Ole, "Practical Application of Single Curved Hull, Definition, Background, Application, Software and Experience", 1985 ICCAS, Elsevier Science Publishers B. V. (North-Holland).

5. Kreyzig, E., "Differential Geometry", Mathematical Exposition No. 11, University of Toronto Press, 1959.

6. Do Carmo, Manfredo P., "Differential Geometry of Curves and Surfaces", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

7. Nutbourne, Anthony W., and Martin, Ralph R., "Differential Geometry Applied to Curve and Surface Design", Ellis Horwood Limited, 1988.

8. Ferris, L. W., "A Standard Series of Developable Surfaces", Marine Technology, Vol. 5, Jan. 1968, pp.52-62.

9. Barnaby, K. C., "Developable Hull Forms", Ship and Boat Builder International, Jan. 1963.

10. Gurunathan, B., Dhande, S. G., "Algorithms for Development of Certain Classes of Ruled Surfaces", Computer & Graphics Vol. 11, No. 2, 1987, pp.105-112.

11. Redont, P., "Representation and Deformation of Developable Surfaces", Computer-Aided Design Vol. 21, No. 1, Jan./Feb. 1989, pp.13-20.

12. Ravani B., Ku T. S., "Bertrand Offsets of Ruled and Developable Surfaces", Computer-Aided Design, Volume 23 Number 2, March 1991.

13. Branco, José Nogueira Rodrigues, "Traçagem, supporting notes on the Shipyard Technology lectures", Instituto Superior Técnico, Lisbon

Technical University, 1981 (in Portuguese). For a copy contact ajcacho@ist.utl.pt or I.S.T. Secção Autónoma de Engenharia Naval, Av. Rovisco Pais, Lisboa, Portugal.

14. Antomari, X., "Cours de Géométrie Descriptive", 4th Ed., Paris, Vuilbert et Nony.

15. Coxeter, H.S.M., "Projective Geometry", 2nd Ed., Springer Verlag, 1987.

16. Hatch, G. N., "Conic Boats (I)", Ship and Boat Builder International, Mar. 1964.

17. Kilgore, Ullmann, "Developable Hull Surfaces", Fishing Boats of the World, Vol.3, 1967, pp.425-431.

18. Hamlin, Norman A., Lewis, Edward V., Editor, "Principles of Naval Architecture", (Volume 1 Stability and Strength, Chapter 1 Ship Geometry), The Society of Naval Architects and Marine Engineers, 1988.

19. Barkley, K. A., "Shell Plate Development by Digital Computer", MSc Thesis, Univ. Heriot Watt, 1972.

20. Hurst, R., "BRITSHIPS - An Integrated Design and Production System", Computer Applications in the Automation of the Shipyard Operation and Ship Design, Computer Applications 5n Shipping and Shipbuilding, Vol. 2, Ed. Yuzuru Fujita, Kjell Lind, Theodore J. Williams, North-Holland Publishing Company, American Elsevier Publishing Company Inc., 1973, pp.6-7.

21. Magnusson, Lennart, "VIKING System for Ship Design and Production", Computer Applications in the Automation of the Shipyard Operation and Ship Design, Computer Applications 5n Shipping and Shipbuilding, Vol. 2, Ed. Yuzuru Fujita, Kjell Lind, Theodore J. Williams, North-Holland Publishing Company, American Elsevier Publishing Company Inc., 1973, pp.21-23.

22. Belda, António, Abarca, Eduardo Martinez-, and Torroja, Jaime, "The FORAN System, Computer Applications in the Automation of the Shipyard Operation and Ship Design", Computer Applications 5n Shipping and Shipbuilding, Vol. 2, Ed. Yuzuru Fujita, Kjell Lind, Theodore J. Williams, North-Holland Publishing Company, American Elsevier Publishing Company Inc., 1973, pp.35-37.

23. Juranek, J., Majewski, W., "The Development and Use of the ASTER Numerical Control System", Computer Applications in the Automation of the Shipyard Operation and Ship Design, Computer Applications 5n Shipping and Shipbuilding, Vol. 2, Ed. Yuzuru Fujita, Kjell Lind, Theodore

J. Williams, North-Holland Publishing Company, American Elsevier Publishing Company Inc., 1973, pp.39.

24. Grandpierre, Cyril, "connaissance du TRACÉ DES CARÈNES", voiliers-bateaux moteurs, Architecture et Construction Navales, Loisirs Nautiques, n° 8 hors série, Oct. 1980, pp.197-275.

25. Clements, John C., "A Computer System to Derive Developable Hull Surfaces and Tables of Offsets", Marine Technology, Vol. 18, No. 3, July 1981, pp.227-223.

26. Munchmeyer, Frederic and Haw, Robert, "Applications of Differential Geometry to Ship Design", Computer Applications in the Automation of Shipyard Operation and Ship Design IV, D. F. Rogers, B. C. Nehrling, C. Kuo (eds.), North-Holland Publishing Company, 1982, pp.183-188.

27. Clements, John C., "Developed Plate Expansion Using Geodesics", Marine Technology, Vol. 21, No. 4, Oct. 1984, pp.384-388.

28. Hardt, David, Wright, Andrew, and Constantine, Edward, "A Design-Oriented Model of Plate Forming for Shipbuilding", Journal of Ship Production, Vol. 6, No. 4, Nov. 1990, pp.212-218.

29. Hansen, Per Skafte, "The Numerical Treatment of General Cones in The Design of Developable Ship Hulls", 1985' ICCAS, Elsevier Science Publishers B. V. (North-Holland).

30. Nolan, T. J., Computer-Aided Design of Developable Hull Surfaces, Marine Technology, Apr. 1971, pp.233-242.

31. P.DP.D. Martins, J.A. Aravena R., "Designing Developable Hulls for Ships Using Quadric Surfaces", PRADS'95 The Sixth International Symposium on Practical Design of Ships and Mobile Units, Sept. 1995, pp.2.1358-2.1368.

32. Akbarabad, M. A. N., "Computer Aided Design of Hydroconic Hard Chine Hull Form", MSc Thesis, Univ. Strathclide, 1988.

33. Weiss, Gunter and Furtner, Peter, "Computer-Aided Treatment of Developable Surfaces", Computer & Graphics Vol. 12, No. 1, 1988, pp.39-51.

34. Letcher, J. S. Jr., Brown, J. M., and Stanley, E. S., "Developable Surfaces in the Fairline System", Proceedings Royal Institute of Naval Architects International Conference on Computer Aided Design of Small Craft, 1988.

35. Aumann, Günter, "Interpolation With Developable Bézier Patches", Computer Aided Geometric Design 8 (1991), pp.409-420.

36. Rogers, David F., Adams, J. Alan, "Mathematical Elements for Computer Graphics", 2th Ed., McGraw Hill Publishing Company, 1990.

37. Lang, Johann and Röschel, Otto, "Developable (1,n) - Bézier surfaces", Computer Aided Geometric Design 9, 1992, pp.291-298.

38. Bodduluri, R. M. C., Ravani, B., "Geometric Design and Fabrication of Developable Bézier Surface", Proceedings ASME Design and Automation Conference, Scotsdale, AZ USA, 1992.

39. Bodduluri, R. M. C., and Ravani, B., "Design of Developable Surfaces Using Duality Between Plane and Point Geometries", Computer-Aided Design Vol. 25, No. 10, Oct. 1993, pp.621-632.

40. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., "Numerical Recipes The Art of Scientific Computing", Cambridge University Press, 1986.

41. Ventura, A. Cacho, T. Torrado, C. Guedes Soares, "Computer-Aided Generation of Ship Hull Strutures on Small Computers", Proceedings of CADAP'95 Computer Aided Design & Production for Small Craft International Conference, The Royal Institution of Naval Architects, 1995.

Appendix A. - Glossary

ADS: AutoCAD Development System.

AutoCAD: is the commercially dominant drafting software package.

Blank: is the plane plate before being cut or subjected to other mechanical work.

CAD: Computer-Aided Design.

CAGD: Computer-Aided Geometric Design.

CAL: Computer-Aided Lofting.

CAM: Computer-Aided Manufacturing.

DOS: is the Disk Operating System, ordinarily used on personal computers.

FORTRAN: the first high level programming language, originally designed for scientific and technical applications.

IBM PC: the industry the facto standard for the architecture of personal computers.

LAN: local area networks of interconnected computers.

N/C: Numerically Controlled machines.

NURBS: Non-Uniform Rational B-Spline[36].

Patch: is the curved plate.

Stock or *green material*: the excess material considered around the plate boundaries, as an allowance for process errors.

Appendix B. - Validation Data

This test plates were processed in a 90MHz Pentium PC, running Windows NT3.51 in 32 MB RAM.

The code was compiled as a console application in Microsoft Powerstation 4.0 Fortran 90 compiler, configured with the default optimised options.

Other hardware and compiler configurations should produce quantitatively different results, but the qualitative assertions made on this results should still hold.

The distances between the developed plate corners of both development methods do not discount rigid body motion differences, thus providing for somewhat conservative figures.

No drawings are presented because they exibit no discernible differences at the fit scale to the A4 page format, as the numeric results demonstrate.

			Length	Beam	Number of...	
Description	Radius or Equation	Developable ?	(m)	(m)	Frames	Frame Points
Plane		Yes	8	2	5	5
High Curvature Cylinder	R= 2	Yes	8	2	5	5
Medium Curvature Cylinder	R= 3	Yes	8	3	5	5
Low Curvature Cylinder	R= 4	Yes	8	4	5	5
Rough Frame-Spacing Cone	R= 5 r= 0.5	Yes	8	5 to .5	3	3
Medium Frame-Spacing Cone	R= 5 r= 0.5	Yes	8	5 to .5	5	5
Close Frame-Spacing Cone	R= 5 r= 0.5	Yes	8	5 to .5	8	8
Extreme Spacing Cone	R= 5 r= 0.5	Yes	8	5 to .5	15	15
Slight Curvature Paraboloid	z= (x^2 y^2)/ 200.0	No	2	2	15	15
Low Curvature Paraboloid	z= (x^2 y^2)/ 0.126	No	2	2	15	15
Medium Curvature Paraboloid	z= (x^2 y^2)/ 0.251	No	2	2	15	15
High Curvature Paraboloid	z= (x^2 y^2)/ 0.375	No	2	2	15	15
Extreme Curvature Paraboloid	z= (x^2 y^2)/ 0.500	No	2	2	15	15

Table 9 - The plates of the test set

Plane

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.000007 1/m2
Non-Optimised: 0.000007 1/m2

Geodesics: 21

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.075300 s	0.072500 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	0.000000	0.000000	0.000008	0.000008	0.000068	0.000068	0.000000	0.000000
frame 2	0.000000	0.000000	0.000068	0.000068	0.000046	0.000046	0.000000	0.000000
frame 3	0.000000	0.000000	0.000046	0.000046	-0.000266	-0.000266	0.000000	0.000000
frame 4	-0.000048	-0.000048	0.000144	0.000144	-0.000266	-0.000266	0.000000	0.000000
frame 5	-0.000048	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
superior seam	0.000024	0.000024					0.000000	0.000000
inferior seam	0.000000	0.000000					0.000000	0.000000

Table 10 - Accuracy results for the development of a plane plate.

High Curvature Cylinder

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.017100 1/m2
Non-Optimised: 0.023800 1/m2

Geodesics: 31

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.290000 s	0.096300 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	0.040332	0.075542	-0.137120	-0.232348	-0.208215	-0.347110	0.000099	0.002442
frame 2	0.040431	0.075542	-0.137803	-0.232811	0.208852	0.347125	0.000096	0.002429
frame 3	0.040461	0.075580	0.137279	0.232621	0.208397	0.346769	0.000095	0.002429
frame 4	0.040530	0.075511	0.136012	0.232545	0.208397	0.346731	0.000094	0.002411
frame 5	0.040590	0.075542	0.000000	0.000000	0.000000	0.000000	0.000093	0.002402
superior seam	-0.076926	-0.137985					0.000000	0.048394
inferior seam	-0.076461	-0.138038					0.000000	0.049204

Table 11 - Accuracy results for the development of an high curvature cylindrical plate.

Medium Curvature Cylinder

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.008160 1/m2
Non-Optimised: 0.013900 1/m2

Geodesics: 51

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.342000 s	0.131000 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	0.031665	0.058401	-0.118557	-0.199092	-0.153216	-0.254500	0.000073	0.003174
frame 2	0.031837	0.058401	-0.118504	-0.199047	0.153224	0.254591	0.000071	0.003129
frame 3	0.031817	0.058381	0.119384	0.198880	0.151987	0.255032	0.000074	0.003138
frame 4	0.031705	0.058452	0.119012	0.199411	0.152427	0.254865	0.000078	0.003177
frame 5	0.031796	0.058401	0.000000	0.000000	0.000000	0.000000	0.000078	0.003143
superior seam	-0.088763	-0.156337					0.000001	0.058881
inferior seam	-0.089049	-0.156122					0.000000	0.057995

Table 12 - Accuracy results for the development of a medium curvature cylindrical plate.

Low Curvature Cylinder

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght

Optimised: 0.004380 1/m2

Non-Optimised: 0.006750 1/m2

Geodesics: 61

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.530000 s	0.152000 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	0.023908	0.043725	-0.099098	-0.166733	-0.120356	-0.200724	0.000024	0.003095
frame 2	0.023847	0.043755	-0.099766	-0.166376	0.120826	0.201248	0.000027	0.003124
frame 3	0.024007	0.043809	0.099463	0.166641	0.120196	0.200565	0.000024	0.003084
frame 4	0.024067	0.043680	0.099023	0.166550	0.119536	0.200390	0.000023	0.003032
frame 5	0.023733	0.043710	0.000000	0.000000	0.000000	0.000000	0.000024	0.003145
superior seam	-0.096387	-0.167000					0.000000	0.064094
inferior seam	-0.095749	-0.167274					0.000001	0.065735

Table 13 - Accuracy results for the development of a low curvature cylindrical plate.

Rough Spacing Cone

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.009290 1/m2
Non-Optimised: 0.015600 1/m2

Geodesics: 49

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.421000 s	0.113000 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	-0.019510	-0.018106	-0.136261	-0.136410	-0.152732	-0.152133	0.000014	0.000024
frame 2	0.041151	0.075706	-0.135295	-0.134790	-0.152792	-0.152139	0.000011	0.004447
frame 3	0.039501	0.074005	0.000000	0.000000	0.000000	0.000000	0.000000	0.000147
superior seam	-0.002265	-0.003657					0.005093	0.000039
inferior seam	-0.002629	-0.003211					0.004836	0.000008

Table 14 - Accuracy results for the development of a cone roughly discretised.

Medium Spacing Cone

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.007380 1/m2
Non-Optimised: 0.012600 1/m2

Geodesics: 81

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.465000 s	0.106000 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	-0.019594	-0.018167	-0.136800	-0.136883	-0.153722	-0.153771	0.000013	0.000026
frame 2	-0.019320	-0.017973	-0.136890	-0.137042	-0.152773	-0.152166	0.000007	0.000013
frame 3	0.040693	0.075104	-0.135899	-0.135387	-0.152755	-0.152165	0.000012	0.004404
frame 4	0.041214	0.075104	-0.135897	-0.135394	-0.152789	-0.152228	0.000003	0.001493
frame 5	0.040059	0.073697	0.000000	0.000008	0.000000	0.000008	0.000000	0.000140
superior seam	-0.002660	-0.004270					0.000005	0.000048
inferior seam	-0.002535	-0.004208					0.000003	0.000054

Table 15 - Accuracy results for the development of a cone with medium frame space.

Close Spacing Cone

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.006570 1/m2
Non-Optimised: 0.011300 1/m2

Geodesics: 129

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.501000 s	0.107000 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	-0.019811	-0.018360	-0.137005	-0.137090	-0.153730	-0.153775	0.000018	0.000026
frame 2	-0.019522	-0.018268	-0.137000	-0.137088	-0.153733	-0.153772	0.000010	0.000015
frame 3	-0.019403	-0.018062	-0.137008	-0.137090	-0.153736	-0.153777	0.000007	0.000013
frame 4	-0.019362	-0.017968	-0.137096	-0.137249	-0.152817	-0.152206	0.000004	0.000009
frame 5	0.041393	0.075836	-0.136136	-0.135604	-0.152818	-0.152193	0.000006	0.003449
frame 6	0.041207	0.075581	-0.136140	-0.135611	-0.152805	-0.152179	0.000004	0.001855
frame 7	0.041052	0.075257	-0.136163	-0.135589	-0.152717	-0.152156	0.000001	0.000751
frame 8	0.041778	0.074229	0.000000	0.000000	0.000000	0.000000	0.000000	0.000130
superior seam	-0.002566	-0.004603					0.000001	0.000077
inferior seam	-0.002400	-0.003761					0.000000	0.000035

Table 16 - Accuracy results for the development of a cone with the frames closely spaced.

Extreme Spacing Cone

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.016100 1/m2
Non-Optimised: 0.025500 1/m2

Geodesics: 241

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.584000 s	0.123000 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	-0.019556	-0.018299	-0.137008	-0.137103	-0.154214	-0.153662	0.000006	0.000020
frame 2	-0.018674	-0.018265	-0.137059	-0.137090	-0.154634	-0.153544	0.000121	0.000002
frame 3	-0.017432	-0.018219	-0.137092	-0.137072	-0.155084	-0.153397	0.000572	0.000006
frame 4	-0.015412	-0.017940	-0.137114	-0.137053	-0.155576	-0.153186	0.001370	0.000058
frame 5	-0.012276	-0.017555	-0.137266	-0.137017	-0.154354	-0.152843	0.002583	0.000223
frame 6	-0.010692	-0.016570	-0.137273	-0.136958	-0.154666	-0.152157	0.000052	0.000207
frame 7	-0.006839	-0.013985	-0.137341	-0.137030	-0.154199	-0.148687	0.000039	0.000250
frame 8	0.038337	0.144408	-0.136040	-0.132570	-0.154210	-0.150082	0.001573	0.041918
frame 9	0.041580	0.118360	-0.135751	-0.134886	-0.154687	-0.150827	0.001491	0.017121
frame 10	0.048398	0.079336	-0.135918	-0.133894	-0.153889	-0.150517	0.000337	0.002110
frame 11	0.089911	0.059465	-0.137163	-0.134169	-0.155669	-0.146796	0.000124	0.001401
frame 12	0.112042	0.166088	-0.138545	-0.130675	-0.158329	-0.143051	0.000092	0.003215
frame 13	0.137976	0.257683	-0.141412	-0.126197	-0.161344	-0.131458	0.000057	0.009741
frame 14	0.020381	0.428038	-0.143140	-0.117391	-0.154251	-0.146879	0.011424	0.054100
frame 15	0.035942	0.074616	0.000000	0.000008	0.000000	0.000008	0.000023	0.000182
superior seam	-0.015273	-0.052282					0.025497	0.021313
inferior seam	0.011928	0.043742					0.024530	0.018337

Table 17 - Accuracy results for the development of a cone with frames tightly spaced.

Slightly Curved Paraboloid

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.000004 1/m2
Non-Optimised: 0.000004 1/m2

Geodesics: 61

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.066300 s	0.065000 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	0.000101	0.000101	0.000068	0.000068	0.000273	0.000273	0.000000	0.000000
frame 2	0.000054	0.000054	0.000121	0.000121	0.000243	0.000243	0.000000	0.000000
frame 3	0.000101	0.000101	0.000129	0.000129	0.000205	0.000190	0.000000	0.000000
frame 4	0.000101	0.000101	0.000099	0.000099	0.000083	0.000083	0.000000	0.000000
frame 5	0.000101	0.000101	0.000114	0.000114	-0.000053	-0.000053	0.000000	0.000000
frame 6	0.000149	0.000101	0.000000	0.000000	-0.000144	-0.000144	0.000000	0.000000
frame 7	0.000149	0.000149	-0.000076	-0.000076	0.000129	0.000137	0.000000	0.000000
frame 8	0.000149	0.000101	-0.000008	-0.000008	0.000061	0.000061	0.000000	0.000000
frame 9	0.000101	0.000101	-0.000030	-0.000030	-0.000008	-0.000008	0.000000	0.000000
frame 10	0.000101	0.000101	0.000023	0.000023	0.000053	0.000053	0.000000	0.000000
frame 11	0.000101	0.000101	-0.000015	-0.000015	-0.000061	-0.000068	0.000000	0.000000
frame 12	0.000101	0.000101	-0.000030	-0.000076	-0.000083	-0.000083	0.000000	0.000000
frame 13	0.000101	0.000101	-0.000144	-0.000144	-0.000190	-0.000220	0.000000	0.000000
frame 14	0.000101	0.000054	-0.000068	-0.000068	-0.000046	-0.000046	0.000000	0.000000
frame 15	0.000101	0.000101	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
superior seam	0.000304	0.000304					0.000000	0.000000
inferior seam	0.000256	0.000256					0.000000	0.000000

Table 18 - The accuracy results for the development of a slightly curved paraboloid.

Low Curvature Paraboloid

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.004470 1/m2
Non-Optimised: 0.005970 1/m2

Geodesics: 61

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.223000 s	0.066400 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	0.049887	0.045922	0.174729	0.180264	2.958013	2.961166	0.000001	0.000012
frame 2	0.053155	0.049981	0.128765	0.134726	2.452660	2.454571	0.000000	0.000008
frame 3	0.057061	0.054441	0.078859	0.082729	1.938970	1.939487	0.000000	0.000006
frame 4	0.059385	0.057450	0.028435	0.031725	1.429232	1.430724	0.000000	0.000003
frame 5	0.057851	0.055727	-0.016474	-0.014606	0.887762	0.885754	0.000000	0.000004
frame 6	0.059043	0.057804	-0.049241	-0.045535	0.451142	0.464969	0.000000	0.000001
frame 7	0.037757	0.032412	-0.191931	-0.216916	0.400418	0.385096	0.000013	0.000024
frame 8	0.032955	0.027020	0.077272	0.065987	-0.135845	-0.149870	0.000001	0.000029
frame 9	0.037710	0.032412	0.187737	0.212787	-0.586439	-0.584440	0.000013	0.000023
frame 10	0.059137	0.057804	0.046353	0.042547	-1.157450	-1.158931	0.000000	0.000001
frame 11	0.057804	0.055727	0.014171	0.012073	-1.707844	-1.708276	0.000000	0.000004
frame 12	0.059291	0.057450	-0.028876	-0.032582	-2.274379	-2.276559	0.000000	0.000003
frame 13	0.057013	0.054382	-0.077188	-0.080638	-2.845066	-2.848046	0.000000	0.000006
frame 14	0.052907	0.049934	-0.123287	-0.128778	-3.406810	-3.412166	0.000000	0.000007
frame 15	0.049781	0.045969	0.001441	0.001448	0.000000	0.000000	0.000001	0.000012
superior seam	2.551708	2.566811					0.000002	0.000186
inferior seam	2.548640	2.562799					0.000000	0.000163

Table 19 - The accuracy results for the development a low curvature paraboloid.

Medium Curvature Paraboloid

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.015700 1/m2
Non-Optimised: 0.020700 1/m2

Geodesics: 61

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.232000 s	0.067300 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	0.251275	0.240790	1.013422	1.021233	9.176403	9.140724	0.000003	0.000096
frame 2	0.305924	0.311252	0.735089	0.740744	8.050438	8.078069	0.000012	0.000026
frame 3	0.180905	0.152074	0.395539	0.366911	6.604974	6.638161	0.000105	0.000725
frame 4	0.180423	0.153506	0.577741	0.637279	4.956666	4.997463	0.000011	0.000646
frame 5	0.173124	0.142746	0.290752	0.329644	3.197736	3.234627	0.000040	0.000804
frame 6	0.233869	0.218158	0.004429	0.041447	1.343686	1.374746	0.000001	0.000219
frame 7	0.190760	0.172253	-0.180587	-0.141924	1.829708	1.818068	0.000008	0.000298
frame 8	0.205232	0.196821	0.574563	0.570192	-0.215895	-0.247329	0.000004	0.000059
frame 9	0.190038	0.172207	0.173014	0.136307	-2.270496	-2.310894	0.000008	0.000275
frame 10	0.233823	0.218158	-0.003458	-0.038093	-4.386691	-4.433216	0.000001	0.000217
frame 11	0.173262	0.142700	-0.257544	-0.292237	-6.529565	-6.572259	0.000043	0.000814
frame 12	0.180767	0.153609	-0.493287	-0.544252	-8.581405	-8.614299	0.000015	0.000656
frame 13	0.181008	0.152498	-0.325672	-0.302696	-10.354390	-10.309330	0.000103	0.000708
frame 14	0.305775	0.311160	-0.586179	-0.590175	-12.571450	-12.565170	0.000012	0.000026
frame 15	0.251894	0.241615	0.061620	0.061781	0.000000	0.000000	0.000003	0.000092
superior seam	8.903421	8.933800					0.000363	0.000821
inferior seam	8.902265	8.933800					0.000392	0.000884

Table 20 - The accuracy results for the development of a medium curvature paraboloid.

High Curvature Paraboloid

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.027600 1/m2
Non-Optimised: 0.036800 1/m2

Geodesics: 61

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.240000 s	0.071600 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	0.544248	0.506251	2.887244	2.878906	15.754470	15.764490	0.000102	0.001338
frame 2	0.586609	0.537581	2.585093	2.635806	13.688660	13.726960	0.000079	0.002337
frame 3	0.620132	0.563745	1.916487	1.951356	11.297030	11.354250	0.000157	0.003036
frame 4	0.625802	0.565182	1.322360	1.373791	8.666308	8.755534	0.000169	0.003517
frame 5	0.598156	0.534587	0.709031	0.772994	5.563835	5.811082	0.000186	0.003875
frame 6	0.571290	0.422592	0.098698	0.173234	2.299885	2.381636	0.001794	0.020557
frame 7	0.488640	0.446739	-0.510873	-0.432399	3.526814	3.442978	0.000044	0.001671
frame 8	0.410541	0.379727	0.982052	0.909091	-0.129986	-0.217557	0.000017	0.000884
frame 9	0.488870	0.446048	0.470021	0.398297	-4.090977	-4.374444	0.000047	0.001737
frame 10	0.571476	0.422450	-0.083675	-0.147597	-8.450572	-8.559939	0.001793	0.020646
frame 11	0.598430	0.535278	-0.557366	-0.610698	-12.764800	-12.839940	0.000178	0.003824
frame 12	0.625988	0.565182	-0.973339	-1.011186	-17.218620	-17.271070	0.000168	0.003539
frame 13	0.620187	0.564535	-1.322276	-1.346641	-21.579100	-21.593670	0.000143	0.002948
frame 14	0.586971	0.537438	-1.685087	-1.719752	-25.913250	-25.918260	0.000076	0.002389
frame 15	0.547099	0.508642	0.409567	0.408952	0.000000	0.000000	0.000097	0.001374
superior seam	17.210540	17.318080					0.000087	0.010306
inferior seam	17.210040	17.318630					0.000088	0.010497

Table 21 - The accuracy results for the development a high curvature paraboloid.

Extreme Curvature Paraboloid

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.039600 1/m2
Non-Optimised: 0.052400 1/m2

Geodesics: 61

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.249000 s	0.079100 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	1.089044	1.013624	5.777053	5.753353	20.978230	21.027680	0.000430	0.005726
frame 2	1.237319	1.125206	5.034035	5.033293	18.365750	18.448710	0.000966	0.012791
frame 3	1.332654	1.199941	4.045599	4.084856	15.246270	15.355000	0.001193	0.018192
frame 4	1.338981	1.201168	2.897807	2.974710	11.657390	11.804930	0.001053	0.019797
frame 5	1.291349	1.156714	1.639475	1.763410	7.523676	7.680057	0.000837	0.019157
frame 6	1.159157	1.050275	0.370466	0.518150	2.885393	3.036053	0.000399	0.012520
frame 7	0.965390	0.891840	-0.811857	-0.656724	5.735283	5.588191	0.000096	0.005794
frame 8	0.751055	0.699726	1.801053	1.666202	0.398587	0.232705	0.000032	0.002714
frame 9	0.966865	0.889793	0.710520	0.574471	-5.719120	-5.908930	0.000117	0.006281
frame 10	1.158076	1.050795	-0.291460	-0.405999	-12.508580	-12.706540	0.000401	0.012311
frame 11	1.291744	1.156974	-1.139038	-1.226173	-19.749150	-19.908650	0.000822	0.019184
frame 12	1.338940	1.201116	-1.815760	-1.865489	-27.283860	-27.418230	0.001106	0.019932
frame 13	1.333360	1.201333	-2.319978	-2.343003	-34.732620	-34.820410	0.001256	0.018072
frame 14	1.236228	1.121425	-2.676686	-2.676965	-41.712220	-41.718540	0.000973	0.013406
frame 15	1.097978	1.012315	1.286477	1.283267	0.000000	0.000000	0.000560	0.007433
superior seam	25.444700	25.623090					0.001332	0.031365
inferior seam	25.444660	25.623090					0.001346	0.031396

Table 22 - The accuracy results for the development an extreme curvature paraboloid.

Saddle

(At least the digits after the 6th, could be numerically irrelevant)

Average Kg/lenght
Optimised: 0.042600 1/m2
Non-Optimised: 0.045300 1/m2

Geodesics: 61

Average Geodesic Tracing Time	
Optimized	Non-Optimised
0.225000 s	0.070900 s

	Strains relative to the curved plate (%)		Distorsion of the angle between the frame and the ... ... superior seam (%)		... inferior seam (%)		End-point distances between both mapped lines (mm)	
Plate Curve	Optimized	Non-Optimised	Optimized	Non-Optimised	Optimized	Non-Optimised		
frame 1	-0.161395	-0.151682	-3.308566	-3.291024	-46.349000	-46.110070	0.000017	0.000119
frame 2	-0.148804	-0.189195	-3.091610	-3.060344	-39.222730	-39.054440	0.000155	0.001394
frame 3	-0.169197	-0.203354	-2.880583	-2.893865	-31.170350	-31.112350	0.000170	0.001125
frame 4	-0.238821	-0.257250	-2.450893	-2.505862	-22.393980	-22.519870	0.000104	0.000386
frame 5	-0.344482	-0.327705	-1.784126	-1.909815	-13.353610	-13.708050	0.000003	0.000204
frame 6	-0.464655	-0.389671	-0.842491	-1.067203	-4.589323	-5.134785	0.000146	0.005419
frame 7	-0.588933	-0.440565	0.412148	0.079617	-6.903934	-6.254198	0.000930	0.023092
frame 8	-0.650277	-0.432150	-1.683467	-1.252323	0.359746	0.828176	0.002286	0.052513
frame 9	-0.589411	-0.439557	-0.473099	-0.092708	6.921731	7.185680	0.000947	0.023489
frame 10	-0.463876	-0.390191	1.075582	1.366555	12.270960	12.353480	0.000139	0.005217
frame 11	-0.345001	-0.328006	2.561022	2.742789	16.502610	16.467220	0.000003	0.000209
frame 12	-0.240088	-0.255774	3.896539	3.984328	19.748460	19.656870	0.000097	0.000286
frame 13	-0.169155	-0.204705	5.005052	5.028390	22.208820	22.084580	0.000171	0.001213
frame 14	-0.148108	-0.193381	5.795606	5.738380	24.087060	24.054780	0.000173	0.001782
frame 15	-0.160699	-0.169716	-2.791760	-2.792534	0.000000	0.000000	0.000071	0.000076
superior seam	-29.452280	-29.392590					0.011038	0.000625
inferior seam	-29.452200	-29.392590					0.011030	0.000623

Table 23 - The accuracy results of the development of a saddle.

GLASGOW UNIVERSITY LIBRARY