COMPUTER AIDED MODELLING
OF
SHEET METAL FORMING

By

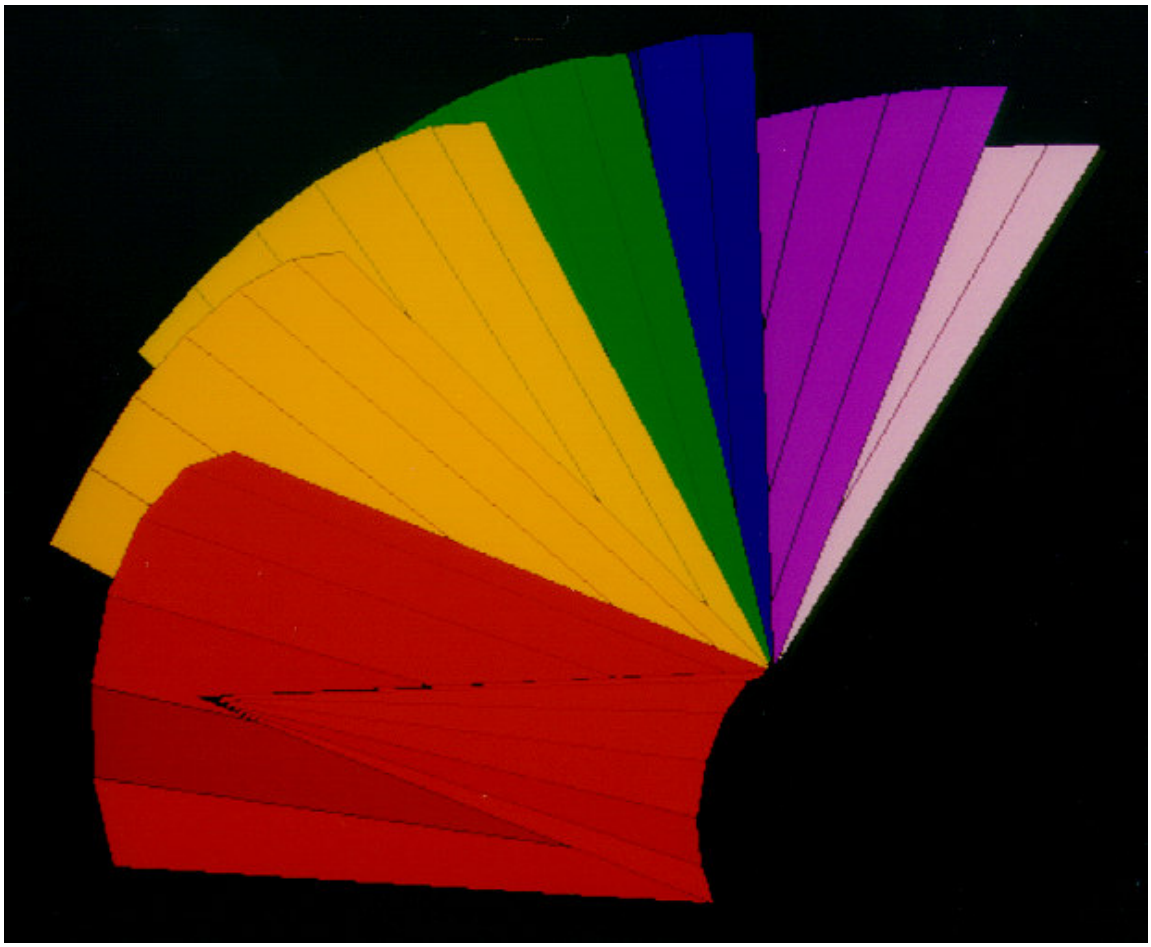
Richard Geoffrey Templer

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Department of Mechanical Engineering
The School of Engineering
University of Auckland

August 1994

TO LEEANNE



ABSTRACT

The work described by this thesis presents several new methods of the computer modelling of sheet metal forming. In particular it focuses on the design steps of sheet metal forming; blank shape prediction, part design, die design and part applications.

This work concentrates on developable surfaces, which are the tangent surfaces of space curves and include cones and cylinders. Developable surfaces are so called, because they can be rolled out (developed) onto a plane without stretching, tearing or creasing. A folded developable is formed when a developable surface is folded about a curve. Since folded developables are formed only by folding they are ideally suited to being constructed from sheet metal.

New theories are presented that accurately predict the surface that will be formed if a general developable surface is folded about a general curve. The theories have been developed into a computer program, 3FD, that allows the rapid and accurate design of folded developables. Several different folded developables have been designed using the program and compared with physical results, with excellent geometric correlation.

An improved method of computer aided blank shape prediction has been developed. The method can be applied to both folded developables and to general sheet metal components. The method uses new boundary conditions to increase the accuracy of the predicted blank shape. The method also indicates possible areas of forming problems. A pressed automobile component is used to illustrate the increased accuracy of the new method.

The design program can also generate the geometry necessary to create the die set to form a folded developable. Such a die set has been created and a folded developable formed from it. The formed folded developable closely matches the computational model.

The design program has also been used to investigate the kinematics of folded developables. The mechanism of a simple folded developable has been determined and the implications and possible applications of this are discussed.

Frontispiece: Multiple images of conical folded developable. The coloured images show the surfaces formed by changing the initial surface radius of curvature from 0.1 (red) to 1 (violet). The folding curve geodesic radius of curvature remains constant at 0.5.

ACKNOWLEDGEMENTS

Prof. John Duncan, my supervisor and the first person, back in 1986, to show me that paper didn't have to fold in straight lines. Thank you for all the inspiration and advice.

Prof. Gordon Mallinson, my supervisor and the first person to explain to me the importance of the G.W. (Gee Whiz) factor. Thank you for the inspiration and assistance, particularly with the programming sections in my thesis.

My Advisory Committee especially Assoc. Prof. Debes Bhattacharyya, Assoc. Prof. Peter Hunter and George Moltschaniwskyj, for support, encouragement and advice.

Prof. K.L. Johnson and Prof. R. Sowerby for advice and encouragement.

The Mechanical Staff, particularly, George Blanchard, John Ward, Joe Deans and Prof. Peter Jackson for advice, encouragement and conversation.

Technicians Fred Kwee, Rex Halliwell, Graeme Moffat, Francis Tsang and Aubrey Mathias for help and friendship above and beyond the call of duty.

Dr David Kenwright, Stu Norris and Mike Johnston for their help in creating 3FD. Jonathan Roberts for assistance with the translation from computer model to die set. Tony Gray for the programs to translate data to MOVIEBYU. Dr Zhao-Tao Zhang for his assistance in digitising the auto part.

The Auckland University Research Committee, the Lotteries Board and IBM New Zealand for their contributions to the CAD technology used.

Fisher and Paykel, firstly as the providers of the Maurice Paykel Scholarship and secondly as my extremely understanding employers. Special thanks to Chris Hutchinson, Mark Benson, David Keeley, Graeme Murray, John Eccles, Mark Suckling and Lewis Gradon.

My family, Mum, Dad, Paul, Michelle, Jonathan and all my extended family for love, assistance and support over the years. My other family, the Olivers, for love and support.

The E-Team and the Skybird Crew, especially room mates Greg Lowe, Tony Davis and Niven Brown, for proving that there is more to engineering than hard work.

My friends, especially Andrew, Carl, Dave, Gowan, Jane, John, Marc, Matt, Megan, Melanie, Mike, Moo, Nicki, Paul, Soon, and Stu for boldly going where no-one has gone before.

Finally and most importantly my wife Leeanne. Without your love, support, encouragement, and help I couldn't have finished my thesis. Thank you!

TABLE OF CONTENTS

1. INTRODUCTION

This thesis demonstrates several new methods of computer modelling sheet metal forming. These methods provide design tools which apply to all stages of sheet metal component design. The methods are as follows:

1. An existing method of blank shape prediction is expanded to give greater accuracy and versatility.

2. New theories and computational models have been developed to produce a program that permits the rapid design of folded developables, a previously little used engineering structure.

3. A method of constructing a die set for the manufacture of folded developables has been developed and successfully trialled.

4. The kinematics of folded developables have also been investigated to show an application of folded developables.

These developments provide tools that will allow future sheet metal designers to design more creatively and with greater accuracy.

In the following sections of this chapter metal forming and sheet metal forming are introduced. The specific developments in computer modelling of sheet metal forming, described in this work, are then introduced, together with their potential applications.

1.1. METAL FORMING.

'Metal Forming' refers to a group of manufacturing processes by which the shape of a workpiece (a solid body of metal) is converted to another shape [Lange et al. 1985]. This change in shape occurs without change in the mass or composition of the metal of the workpiece.

Metal forming is a subset of manufacturing processes. The manufacturing processes may be divided into six main groups:

Primary forming	Original creation of a shape e.g. ingot casting.
Deforming or Metal Forming	Change the shape of a solid body e.g. deep drawing.
Separating	Machining or removal of material e.g. turning in a lathe.
Joining	Uniting individual workpieces to form sub assemblies e.g. welding.
Coating	Application of thin layers to a workpiece e.g. galvanising.
Changing the Material Properties	Deliberately changing the properties of the workpiece in order to achieve desired characteristics e.g. tempering steel.

Metal Forming encompasses many different forming techniques. These can be classified by two methods:

- The means of deformation, and
- The initial shape.

1.1.1. The Means of Deformation.

Common means of deformation are;

Compressive Forming	• Rolling • Open-die forming • Closed-die forming • Indenting • Pushing through a die
Combined tensile and compressive forming	• Pulling through a die • Deep drawing • Flange Forming • Spinning • Upset bulging
Tensile Forming	• Stretching • Expanding • Recessing
Forming by Bending	• Bending with linear tool motion • Bending with rotary tool motion
Forming by shearing	• Joggling • Twisting.

1.1.2. The Initial Shape.

Metal forming may also be classified by considering the initial shape of the workpiece. The two major divisions are:

- bulk or massive forming, and
- sheet metal forming.

1.1.2.1. Bulk Forming

Bulk forming considers the deformation of bars and castings - workpieces in which all three linear dimensions are of a similar order of magnitude. By contrast, in sheet metal forming the initial workpiece will have one linear dimension orders of magnitude smaller than the other two e.g. 1000 mm by 1000 mm by 1.2 mm.

Bulk forming is characterised by three axis compressive forming whereas sheet metal forming is characterised by both compressive and tensile forming in the large axis of the sheet.

Historically metal forming was for a long time purely bulk forming. 6000 years ago in the Neolithic age primitive humans forged and hammered metal. The introduction of iron gave rise to hot forging best exemplified by historic blacksmiths who used the heat from a coal fire and the energy of their own muscles to turn pieces of bar and ingots into horse shoes, armour and swords.

Sheet metal did not make an appearance until the late 18th century. Initially the sheet was hammered flat but by the late 19th century, with the development of cast steels and the double action press, modern sheet metal forming began.

1.1.2.2. Sheet Metal Forming.

Sheet metal forming is one of the principal processes used in the manufacture of a wide variety of consumer and commercial products. Products such as automobiles, roofing, whiteware, kitchenware, medical equipment and kitchen sinks, to name a few, are formed wholly or partly from sheet metal.

The sheet metal industry involves primary metal suppliers, tool and die makers, metal stampers, lubricant suppliers and designers. Sheet metal forming is the culmination of all the above industries' inputs, to achieve a part with the desired shape, performance, finish, quality and cost [Karima 1989].

Historically this has been achieved by design/build/validate steps [Keeler and Stine 1989]. The part was designed by the part designer who then passed the design to the die designer, who modified the design and then passed it to the tool maker and so on.

Part and die design were undertaken by sheet metal forming craftspeople. These artisans based designs on past experience gained from trial and error with previous designs. Their success at designing a part or die was dependent on how closely it matched a part or die previously designed. Radical changes in shape or material made initial designs difficult, often inaccurate and the design process slow.

The development of analysis techniques such as Grid Circle Analysis and Forming Limit Diagrams, [Keeler and Backofen 1963], have increased the accuracy of sheet metal design.

In grid circle analysis, the flat sheet of metal about to be deformed, is covered in a regular grid of small circles. The deformation of the circles after forming provides information on the strain magnitude and direction in the deformed part.

A forming limit diagram is a plot of data obtained from grid circle analysis. The two principal strains ε_1 and ε_2 are plotted against each other for each circle. Tests performed on the same material provide an envelope within which the strains must lie for successful forming.

Grid circle analysis and the forming limit diagram allow the designer to quantify and analyse the strains in a deformed part, reducing the number of trials needed for a successful design.

1.1.3. Differences Between Folded Developables and Conventional Sheet Forming.

Sheet metal parts can be divided into two categories:

- stampings, and
- developables

[Duncan and Duncan 1979].

Stamping sheet metal parts are those in which permanent deformation, either thinning or shear distortion (drawing), must occur in the sheet in order to achieve the final shape. Sheet metal parts which can be formed by bending the sheet into place without stretching or drawing in the plane; are called 'developable' parts.

The forming techniques and material properties required are different for each category.

1.1.3.1. Stampings - Forming Techniques and Material Properties.

Stampings are most commonly produced in draw dies. In a draw die the material is retained at the edges and stretched over a punch. In draw die forming the material must have a large ultimate tensile stress to yield stress (UTS/Y) ratio. It may be shown that this ratio must exceed a lower limit given by,

$$\frac{UTS}{Y} = e^{\frac{\mu\pi}{2}}, \tag{1.1}$$

where μ is the average coefficient of friction between the sheet and the punch [Marcimak & Duncan 1992]. This friction coefficient is rarely less than 0.2, so the ultimate tensile/yield ratio must be greater than 1.4. Most materials used in die forming have an ultimate tensile/yield ratio of 1.5 to 2.0 to make die forming easy.

However the modern demand for lighter weight, higher strength products, especially in the auto industry, has caused problems. To reduce the weight of a sheet component such as a door panel, it must be made from thinner sheet. To retain function, the initial yield stress or 'strength' of the material must increase. Unfortunately the most convenient methods of increasing initial yield stress such as work hardening, precipitation hardening and grain refinement do not increase the ultimate tensile stress. Thus the UTS/Y ratio drops, making die forming difficult for high strength steels.

One solution to the problems associated with die forming of high strength sheet is to make the component a developable.

1.1.3.2. Developables - Forming Techniques and Material Properties.

Developables are most commonly formed by roll forming and bending. Roll forming involves passing the sheet through a series of rolls which bend the sheet to the desired section. Bending may be performed in a variety of ways such as press-brake bending, flanging, folding and pressing. Conventionally the bends are straight and have a constant bend angle. Thus the range of shapes that can be produced is small.

Material requirements are different from stampings. Ultimate tensile/yield ratio is not critical and materials with a *UTS/Y* ratio of one can be formed by bending. The critical material property is the maximum strain due to bending. In simple bending this is;

$$\varepsilon_{max} \approx 1/[2(r/t)] \tag{1.2}$$

where r is the radius of the bend and t is the thickness of the sheet. For a $5t$ bend, the outer strain is 10%. Provided this is less than the fracture strain which is approximately indicated by reduction in area of the material in a tensile test, satisfactory bending will occur.

1.2. COMPUTER AIDED MODELLING OF SHEET METAL FORMING.

As the name suggests, computer aided modelling of sheet metal forming uses computer programs to solve possible sheet metal design problems. Many different methods are used in the programs including Finite Element Methods, Geometric Rules and Metal Flow Analysis. Some programs run on Cray mainframes others run on simple PCs.

The problems this work attempts to answer are:

- What shape (blank shape) should the component be before forming?
- How can a folded developable component be designed?
- How can the above component be formed?
- What applications does the component have?

1.2.1. Blank Shape Prediction.

The blank shape of a part is its flat shape prior to deformation. Finite element analysis (described in section 2.2) has had limited success in predicting the blank shape of parts with complicated shape. Several alternative methods have been proposed including slip line field methods (described in Section 2.2) and the constant area transformation.

The constant area transformation assumes plane strain or zero thinning conditions during deformation. Developed by Sowerby, Duncan and Chu [1986] it is a simple geometric mapping process that conserves the area between the blank shape and the finished part. This is a predictive technique and is used as a means of reducing the number of costly press shop trials that are required with new tooling. The constant area transformation is not an exact solution to the actual metal forming process, but rather a technique that allows an insight into the forming process necessary to produce the desired shape.

In this work, the method developed by Sowerby, Duncan and Chu has been expanded and improved. The original model was restricted to straight boundaries, perpendicular to one another. Mathematical and computational methods for the modelling of any boundary have been developed. This increases the accuracy of the predicted blank shape and greatly increases the range of parts that may have their blank shape predicted.

1.2.2. Folded Developable Part Design.

Computer Aided Design (CAD) and Computer Aided Manufacture (CAM) have changed the way engineers design parts. Most manufacturing firms now use computer packages such as AUTOCAD™, CADAM™ and CATIA™ to assist in the design of their sheet metal parts. Designs are stored on computer disk rather than paper, and complex solid models of the part can be produced. However the parts are still designed from experience and previous parts are used as guides.

In this work a new method of part design is presented. The mathematical and computational method allows the designer to design folded developables interactively. A folded developable is a sheet surface that has been folded along an arbitrary curve to produce two separate developable surfaces joined along this fold curve.

Because a developable surface (or developable) is folded and not drawn or stamped, material properties which govern drawing deformation may not be critical. Because the *UTS/Y* ratio is no longer critical, developables provide a means of forming high strength materials.

The method of modelling curved line folding of such developables, described in this thesis, greatly increases the range of products that can be produced by 'developable' methods. This gives the part designer a new option when designing parts. Computer modelling gives engineers and designers the opportunity to add folded developables to their list of choices when designing a new product.

1.2.3. Die Design for Folded Developables.

When a part has been designed, and its blank shape predicted, the tools for forming it must be designed. For parts that are to be pressed or folded the most common forming method is die forming. Two metal plates are machined to form male and female dies, as shown in Fig. 1.1. Pressing the two dies together, with the sheet blank between forms the part.

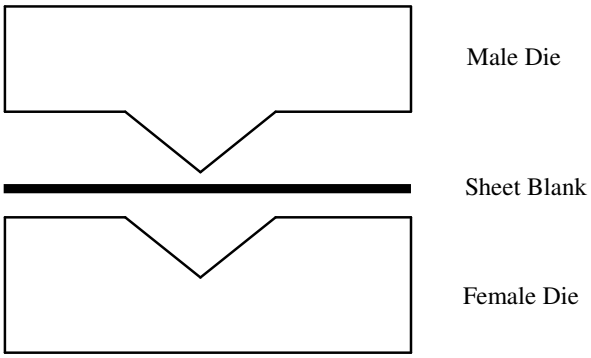


Fig. 1.1 Schematic of Die Forming.

In this thesis a computational method for the design of dies for folded developables is presented. This allows folded developables to be readily created, as the part can be designed, the blank shape predicted and the dies designed all in one integrated computer program. The theory and practice behind the die design, with an example is in Chapter 4.

1.2.4. Kinematics: An Application of Folded Developables.

An important skill in design is consideration of possible applications for the part to be designed. Can the designed part meet the design specifications? An intelligent approach to design involves going from the design idea to the finished part in the smallest possible number of prototypes, thus reducing the extra time and cost of multiple prototypes. In light of this, modern design methods allow the testing and evaluation of some parameters before the first prototype is even constructed, e.g. some solid modelling programs allow checking for interference between parts in an assembly.

The kinematics of folded developables is discussed in Chapter 5. The kinematics of folded developables results in linear displacements in response to changes in curvature of the initial surface.

Using this result folded developables can be used as simple mechanisms. The mechanism is described so the performance of the folded developable mechanism can be evaluated prior to manufacture.

2. COMPUTER AIDED BLANK SHAPE PREDICTION.

Computer Aided Blank Shape Prediction is the process of determining the blank shape of a desired part using a computer. The blank shape of a sheet metal part is the original shape of the flat sheet, before the sheet metal is deformed. To reduce waste of metal the blank shape should be such that trimming is minimised after forming. The relation investigated here is an assumption of constant area between the finished part and the blank shape. A computer program has been used to reduce the time taken and increase the accuracy of predicting the blank shape. Hence the name Computer Aided Blank Shape Prediction.

2.1. BLANK SHAPES AND METAL FORMING.

Metal forming of a simple shape can be defined as the transformation of a piece of metal into a useful part by means of plastic deformation [Berry 1988]. In sheet metal deformation the initial shape is either a flat sheet or tube and has a much smaller thickness than length or width dimension. Typically the thickness may vary from 1/50th to 1/5000th of the length or width.

Sheet metal forming involves the deformation of a basic shape (the blank shape) usually cut from a flat sheet, which is then plastically deformed into its final shape. The deformation process is characterised by large displacements but low strains. Studies at Toyota show that strains over large areas of auto body panels rarely exceed 4-5% [Duncan and Sowerby 1981].

Sheet forming includes brake forming, deep drawing, stretch forming, rubber pad forming and folding. All these forming methods use an initial blank shape. Accurately predicting this blank shape and better understanding the processes involved has a large scale industrial application.

2.2. REVIEW OF COMPUTER AIDED BLANK SHAPE PREDICTION.

2.2.1. Methods of Determining Blank Shape.

The determination of the most efficient blank shape for a deformed part, by either a physical or computer model, is the engineer's problem. He or she must determine the set of displacements that link the final and initial states of the part.

2.2.1.1. Non Computational Methods.

Historically blank shapes were predicted by experience, by trial and error and by rule methods as illustrated in Fig. 2.1. These methods give reasonable results for simple shapes but are unable to cope with the complex parts formed today.

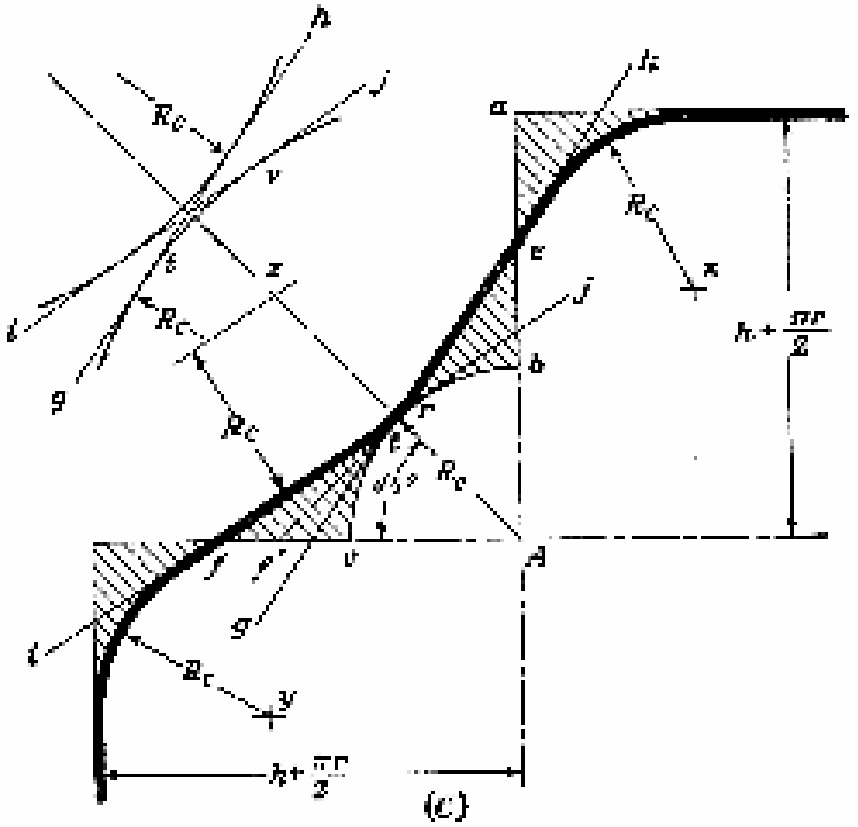


FIG. 10-24C. Correct development of blanks for rectangular draws: a concave sloping curve is produced if $R_c < 0.36(h + 1.57r)$.

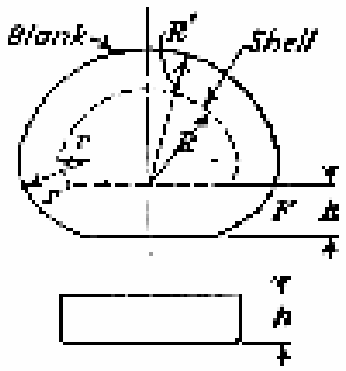


FIG. 10-25. Layout of a blank for a kidney-shaped shell.

Fig. 2.1 Non computational method of blank shape prediction.

Modern methods my be divided into two main categories; one type is Finite Element Modelling (F.E.M.) or Finite Difference Modelling and the other Geometric Modelling.

2.2.1.2. Finite Element Modelling.

Finite Element Modelling is a computer based numerical analysis technique that is widely used [Wood 1981]. Basically the method involves breaking down a complex problem into a number of simpler component problems from which the complex problem can be constituted according to a set of defined rules. This is similar to a brick building. Each brick is simple to construct but the building may have a very complex shape.

Automobile researchers have been trying to model metal forming for more than twenty years [Hatt 1993]. Early efforts employed 'implicit' finite element codes. Programs such as NASTRAN™, MARC™ or ANSYS™ solve a global sparse matrix to estimate stresses and deformations. In practical forming problems, this matrix can consist of 20 000 to 50 000 elements. Because metal forming simulations are time-dependent, the matrix must be solved a number of times to get the answer right. Implicit codes thus took too long.

F.E.M. is usually performed by modelling an actual or imaginary forming process moving through a series of small but discrete steps. [Chu, Soper 1985]. The calculations begin with set initial conditions (blank shape, lubrication, die geometry etc.) and continue until the desired part shape has been reached or some unacceptable result such as tearing, wrinkling, localised necking, or thinning is indicated. If the modelling shows that the part cannot be formed, the initial conditions must be changed until an acceptable solution is reached.

Initially F.E.M. was not very successful because of the complexity of "real" metal forming problems and the large amounts of computing time required. An example of the problems faced by F.E.M. is the modelling of a simple lift out panel by General Motors [Arlinghaus 1985]. Due to (i) the large number of elements and associated size of three dimensional stiffness matrix required for accurate modelling, and (ii) the non-linearities of the problem caused by material, geometric and boundary conditions, the problem required hundreds of hours on an IBM 3081 to be solved. To produce results inside a day, the team had to use a CRAY 1-S/2000 with a machine coded program.

During the eighties, computers became faster and F.E.M. more accurate and more efficient. Modern Finite Element packages such as ABAQUS™ are capable of analysing metal forming problems taking plastic flow, lubrication, friction and formability into account [Hibbitt, Karlsson and Sorensen 1989 I]. Recent work at Volvo Personvagnar Olofström has shown good results can be achieved by the use of dynamic F.E.M. using programs originally developed for motor vehicle crash simulation [Hyllander 1991]. Dynamic F.E.M. codes solve plastic deformation by 'explicit' codes. Explicit codes do not solve a global matrix and so require less computing time [Hatt 1993]. However the time required is still large. The explicit F.E.M. program PAM-STAMP™ requires up to two months to analyse a complex stamping [Hatt 1993]. A recently reported problem modelled an aluminium sheet metal

forming problem that involved 127 000 finite elements in the mesh [Clifford 1993]. It required 65 hours of CRAY supercomputer CPU time, probably over a week of real time.

F.E.M. is also no longer exclusively performed on mainframe computers. The advances in computing technology have produced Personal Computers with processors such as the 80386 which for smaller meshes (less than 60 000 nodes) can perform the required calculations as fast as mainframes [Bussler and Paulsen 1988].

Despite advances in algorithms and computer power, the basic problems of F.E.M. remain, namely, the critical importance of element selection, the accuracy of both the mathematical model and the input data and their expense.

The importance of element selection was stated by MacNeal and Harder [1985] who said "Nothing is as important to the success of Finite Element Analysis as the accuracy of the elements...it has been shown that almost every problem is capable of evoking results ranging from excellent to poor".

R.W.Clough [1990] states that "the results of a finite element analysis cannot be better than the data and judgement used in formulating the mathematical model, regardless of the refinement of the computer program that performs the analysis". He then examines the modelling of stress and strain by finite elements and adds "...an even more troublesome tendency among many engineers [is] to accept as gospel the stress values produced by a finite element computer program... clearly the assumption of specified strain patterns used in formulating the element stiffness makes it impossible for stress equilibrium to be satisfied locally at arbitrary points within elements". F.E.M. programs are also expensive e.g. ABAQUS™, a mainframe based system and ANSYS™, a PC based system are both over USD 10 000.

Designers of sheet metal parts need simple analytical tools to give them guidance about what might or might not work as they are developing part designs. Today's F.E.M. codes are too complex for this purpose [Hatt 1993].

This has led to the use of a simplified finite element analysis in which only selected regions of the part are analysed [Gloeckl and Lange 1983].

However the problems described above have lead to the development of alternative methods to perform the same tasks as F.E.M. or to provide better initial condition estimates, thereby reducing computing time.

2.2.1.3. Simplified Models of Forming Processes.

Because the analysis of actual sheet metal forming is so complex, tool designers have developed simple models of forming processes which can be solved and displayed on simple personal computer systems [Duncan and Sowerby 1981]. For example, if the sheet is clamped in the die ring, either as a flat surface or as a developable surface, it is assumed that, as the punch descends, the sheet is penetrated by the punch and at any instant, elements of the sheet are either unmoved or adhere to the punch. Clearly this is a gross simplification of the process but the analysis is not difficult as it is a simple interpenetration exercise. It is a useful aid and the designer can adjust the orientation of the punch and blank on the screen to obtain a suitable "foot-print" of the punch on the sheet.

Similar to this "foot-print" approach is the use of total strain theory. This considers the complex curves on a part to be a straight line. The change in length caused by the punch (or other deformation devices) is then used to calculate the strains, determined by the ratio of final to initial length. This information is used to predict blank shape and possible defect sites. This method has been used to good effect by both NISSAN [Furubayashi 1985] and MAZDA [Yamasaki, Nishiyama and Tamura 1985] motor companies.

Another modern technique which predicts blank shape and gives some information on the forming process is Slip line field analysis. Slip line fields are a method of solving problems relating to the plane plastic flow of a rigid-perfectly plastic solid [Johnson, Sowerby and Venter 1982]. The solution consists of a statically admissible stress field and a kinematically admissible velocity field that is related to the stress field according to the flow rule, there being positive plastic energy dissipation where deformation occurs. Slip-line fields and hodographs are graphical solutions of the plane flow equations. While successful for simple shapes slip line fields are not suitable for modelling complex re-entrant shapes.

2.2.1.4. Geometric Modelling and The Ideal Sheet Metal.

In forming, the sheet is transformed from a plane to a non-developable surface [Duncan and Sowerby 1981]. Some deformation is necessary but for practical purposes it may be preferable that this should be done by in-plane shear distortion rather than thinning. Hence the ideal sheet would be considered to have an infinite strength or resistance to deformation in the through thickness direction i.e. during deformation the thickness would remain constant.

An example of a sheet metal forming model of the deformation of this "ideal" sheet is the Constant Area Transformation model, which is detailed in the next section. If thickness is constant throughout deformation and volume is conserved (as it is in common metal forming processes), the area of the sheet material is invariant. Though in reality no such ideal sheet

metal exists, many traditional design rules use a constant thickness (and thus constant area) approach.

We would wish to form the sheet without force, hence the in-plane yield stress should be zero. Obviously a component of zero strength is of no use so we would also wish some subsequent hardening mechanisms.

While no such metals exist, modern drawing quality steels come close to this ideal. Their high normal plastic anisotropy (r values) [Lange 1985], mean that they deform in the plane more easily than in the through thickness direction. It is possible also to develop sheet that will gain strength after forming by strain ageing.

2.3. CONSTANT AREA TRANSFORMATION BLANK SHAPE PREDICTION.

Computer Aided Blank Shape Prediction is a computer modelling method that can be used to predict blank shape. It is a geometric transformation based on an assumption of constant area. Previous work by J.L.Duncan, R.Sowerby and E.Chu [1986] detailed a method of constant area transformation; this has been used to develop a computer modelling package [Templer 1987].

Computer Aided Blank Shape Prediction may be considered in two parts: Element Transformation and Boundary Specification.

2.3.1. The Constant Area Transformation.

The argument for applying the assumption Constant Area Deformation to 'real' sheet metal may be found by examining metal forming theory and practice as follows [Marcimak & Duncan 1992].

2.3.1.1. Justification of the Constant Area Transformation.

If we subject an element of metal to various tractions, as shown in Fig. 2.2 we will induce in it a two dimensional stress state.

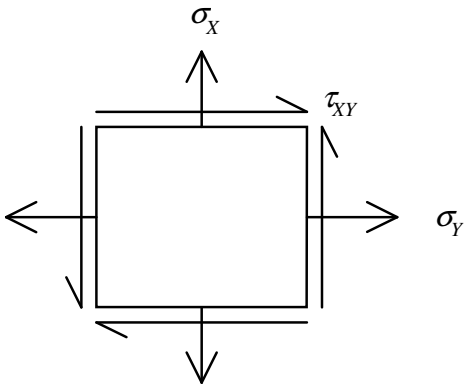


Fig 2.2 Two dimensional stress state on an element.

The stress state of the element can be transformed to the actions of two principal stresses using a Mohrs circle of stress as shown in Fig. 2.3. This is a good approximation of sheet metal stamping deformation as the third principal stress normal to the sheet which is created by the tooling is often less than a few percent of the in-plane stresses.

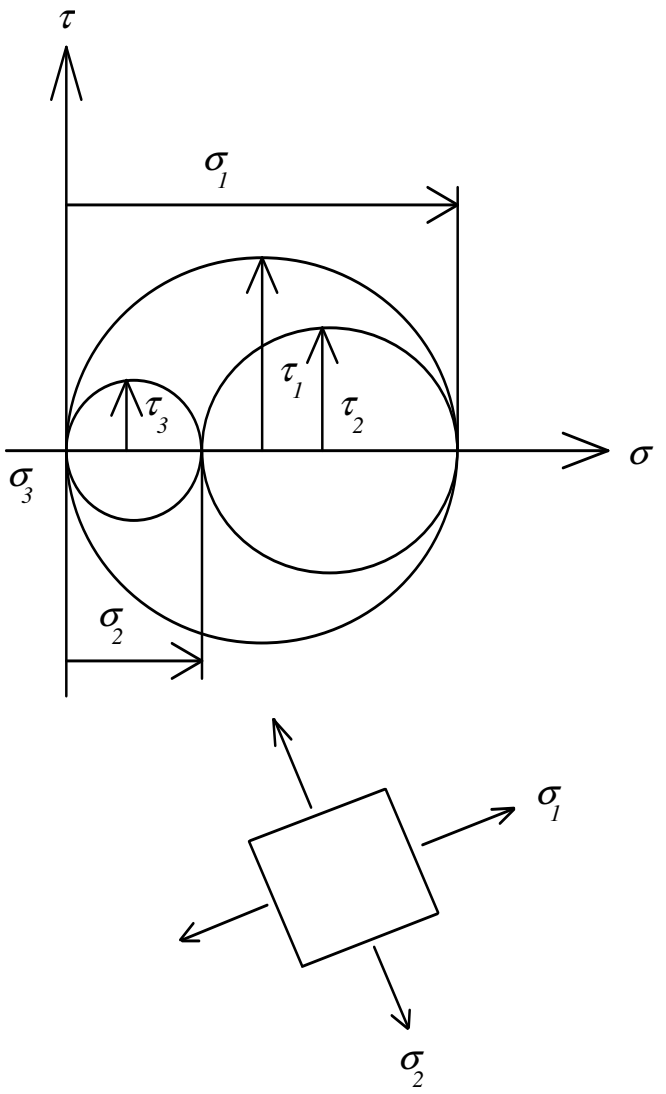


Fig 2.3 Mohr circle of stress and the calculated principal stresses.

If this stress state reaches a certain level governed by the yield criterion, the element will deform or strain. Two common yield criteria are the Von Mises and Tresca yield conditions, shown for a two dimensional stress state in Fig. 2.4.

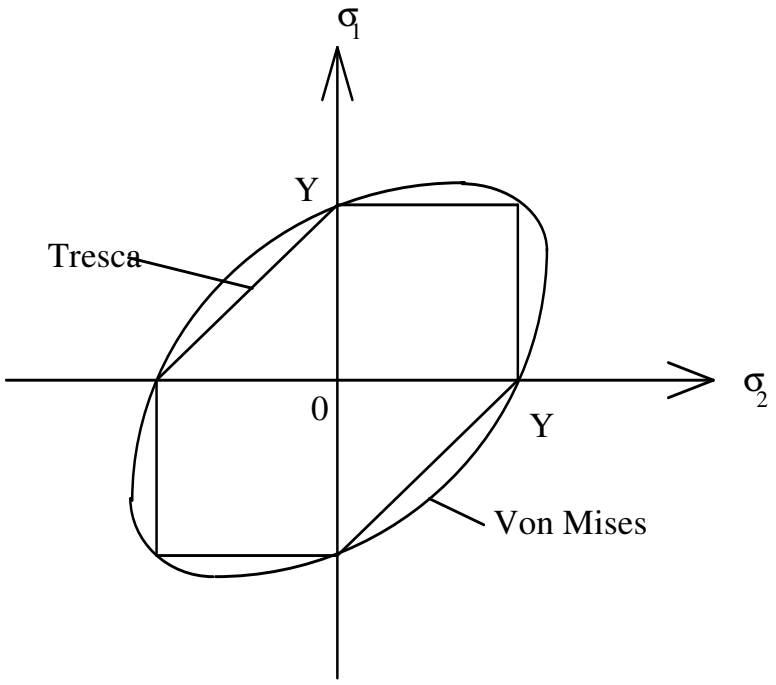


Fig 2.4 The Von Mises and Tresca yield criteria in the plane stress space ($\sigma_3=0$).

If we consider also the strains, ε_1 vs. ε_2 and plot the membrane strains, we obtain a forming limit diagram. A plane stress forming limit diagram for a typical simple sheet metal part is shown in Fig. 2.5. It is important to remember that stress states are limited by strength and yield criteria while strain states are limited by ductility or stability.

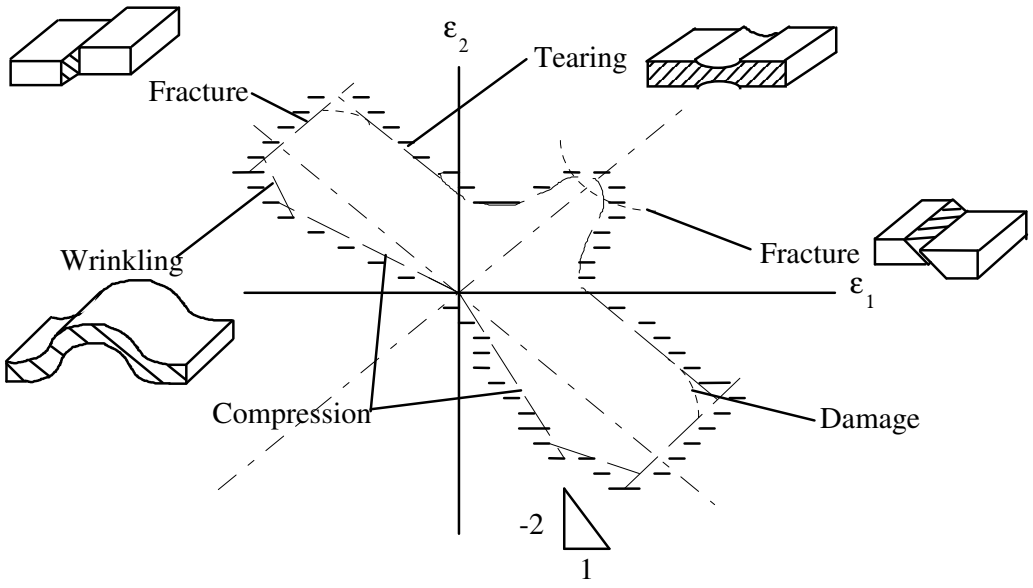


Fig 2.5 The various limits to a simple sheet forming process.

Ductile fracture may be caused by stresses exceeding the critical shear stress. In the strain space diagram this is illustrated by the two curves marked fracture on Fig. 2.5. In the left-hand side, shear occurs on a plane perpendicular to the surface as a result of high drawing stresses, $\varepsilon_1 = -\varepsilon_2$; on the right-hand side, shear occurs at 45° to the sheet surface, $\varepsilon_1 = \varepsilon_2$. Wrinkling failure is not governed solely by a material property, it is associated with the constraint imposed and buckling characteristics governed by the elastic modulus and thickness.

There is a further limit inherent in sheet metal forming. Sheet metal forming comes about as a result of tractions transmitted through the sheet; these tractions arise from the normal forces exerted by the tooling on the sheet. The tractions in the plane of the sheet are mostly tension. However if the material is drawn into a converging space, compressive tractions will arise. A typical example is in drawing the flange of a deep cup. The circumferencial forces and hence stresses are compressive while the radial stress varies from a high tensile value at the inner radius of the flange to zero at the outer edge. This is a limiting stress case in sheet metal forming; the membrane principal stresses in the sheet cannot both be compressive, and the limit is when σ_2 is negative and σ_1 is zero i.e. when $\varepsilon_2 = -2\varepsilon_1$. This is illustrated by the line of slope -1/2 in Fig. 2.5.

As can be clearly seen from the safe forming diagram, the basic slope of the window of safe deformation is elongated with the largest dimension having a slope of -1. This corresponds to $\varepsilon_2 = -\varepsilon_1$ i.e. $\varepsilon_3 = 0$, or constant area deformation. Since the strains that can be maintained in constant thickness deformation on the safe forming diagram are much larger than in other

strain paths, the majority of high strain sheet metal forming processes use this direction of straining. The diagram below (see Fig. 2.6) illustrates that this Constant Area direction is also a favourable stress state to reach yield.

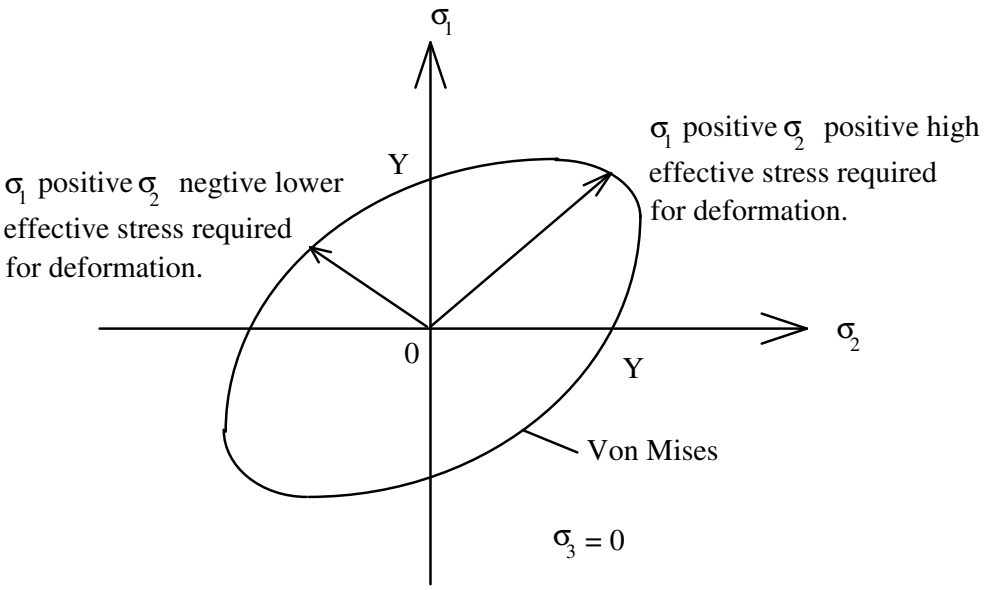


Fig 2.6 The effect of stress directions on effective stress (and therefore the energy) required for deformation.

If one remembers that the tensile forces transmitted through the sheet must obey an equilibrium condition then it seems likely that those regions in which the stresses required for deformation are at a minimum ($\sigma_1 = -\sigma_2 = Y/2$) will be more likely to deform than regions where one stress is maximised (plane strain) or both are large ($\sigma_1 = \sigma_2 = Y$).

Thus, although few metal forming processes are perfect Constant Area Transformations, the majority may be approximated by such an assumption.

2.3.1.2. Constant Area Transformation Method.

The constant area transformation is a method of transforming the elements of a deformed surface onto a flat blank. Each element is assumed to deform without change in thickness or area (plane strain conditions) and continuity between elements is assumed so that there are no gaps or overlap.

The deformed surface is first covered with a quadrilateral grid having nodes *I, J, K ...N* as shown in Fig. 2.7. This can be drawn on the part or be generated by a computer mesh generation system. The nodes of the grid are then digitised and when these co-ordinates are known, geometric calculations and transformations can be carried out. The mapping

procedure is illustrated in Fig. 2.7. Fig. 2.7(a) shows a region of mesh located on the deformed part.

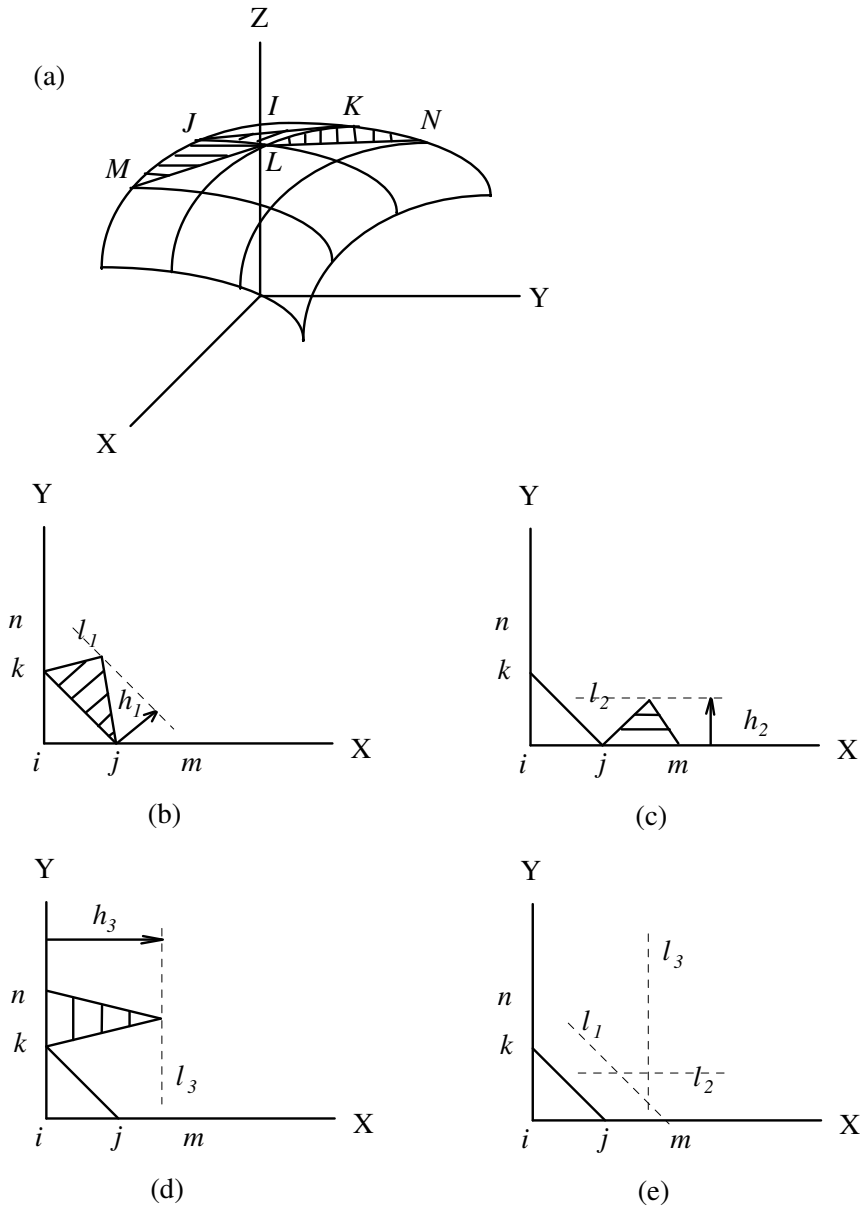


Fig 2.7 (a) Region of mesh located on the surface of a part. (b) - (e) The complete mapping procedure.

The boundary nodes *i,j,k* etc. are laid down on the flat sheet using some arbitrary rule. The choice of these rules and the effect of the different assumptions and rules regarding the placement of boundary nodes is discussed in detail in the next section. With known nodes, triangles *ijk, jkl* etc. can be determined.

An example of a simple boundary rule is to assume that the boundary nodes will lie along straight lines. The two dividing lines along nodes *IJM* and *IKN* are assumed in this case to be at right angles. The distances between *i,k,n* (on the flat plane) and *i,j,m* are calculated and plotted along the X and Y axes. This is performed by determining the distances between the nodes and 'laying them out flat' on the axes.

It now remains to calculate the position of *L* on the plane (*l*). This calculation is shown in Fig. 2.7(b)-(e). The first offset h_1 is calculated by dividing the area of the triangle *JKL* by half the length of the line *jk*. To conserve the area of the planar triangle point l_1 must lie on a line parallel to *jk* a distance h_1 away. Using the area of triangle *JLM* and the line length *jm*, l_2 can be calculated which must lie parallel to *jm*. Similarly l_3 can be calculated.

As shown in Fig. 2.7(e) the lines form a triangle within which *l*, the transformed position of *L* is assumed to lie. Duncan, Sowerby and Chu [1986] suggest the centroid of the triangle as the location of point *l*. Thus the quadrilateral *JKLM* has been mapped from a 3-D deformed surface onto a plane using the constant area transformation. After locating point *l* the rest of the transformed surface can be determined by repeating the procedure.

For computing the blank shape, a slightly different algorithm to the one detailed above is used as shown by Fig. 2.8.

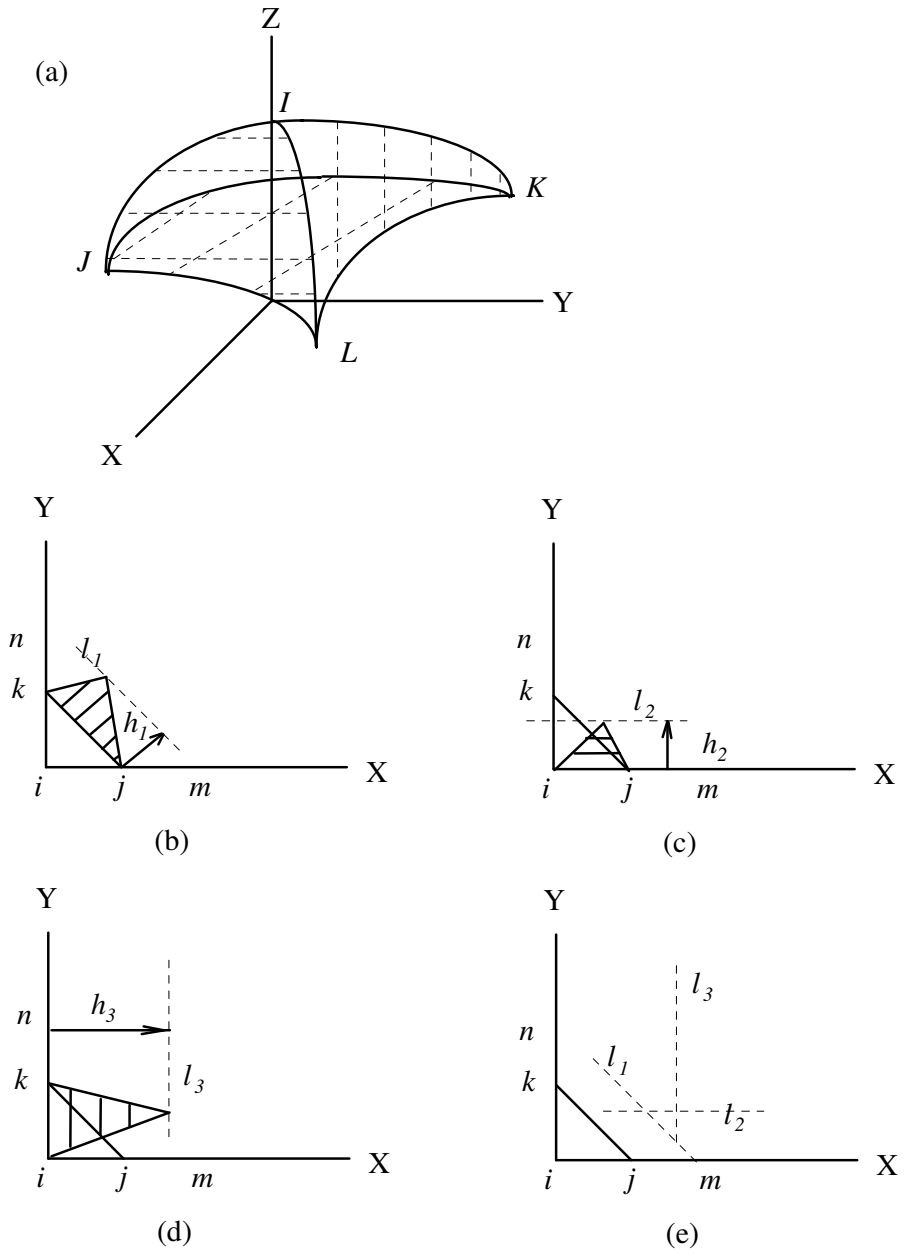


Fig 2.8 (a) Quadrilateral of a mesh located on the surface of a part. (b) - (e) The complete computer aided mapping procedure.

This method of transformation uses the information of the single quadrilateral element of interest. In all other respects it is identical. This method is adopted because of the finite nature of the meshes analysed and allows transformation right to the edge of the mesh rather than stopping one element back.

2.4. BOUNDARY SPECIFICATION.

The development of Computer Aided Blank Shape Prediction has been three fold,

(i) Investigation of new boundary definition methods,

(ii) Developing methods of assessing mapping performance,

(iii) Investigating possible forming information obtainable from mapping.

This section details the two existing methods of boundary specification and the two new methods that have been developed.

2.4.1. Existing Methods of Boundary Definition.

The effects of boundary definition on finished blank shape can be determined by considering a simple three-dimensional part (see Fig. 2.9) and transforming it to a flat blank using several boundary definition methods.

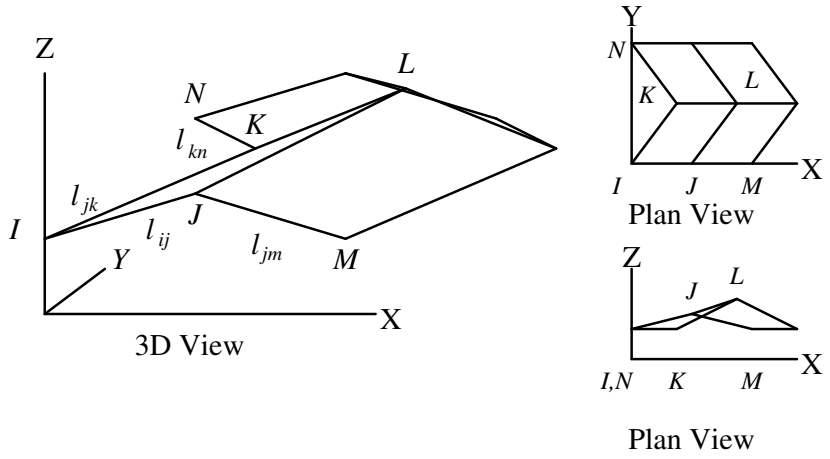


Fig 2.9 Test Shape.

The two existing boundary definition methods are (i) regular lengths laid at right angles, as shown in Fig. 2.10 and (ii) actual lengths laid at right angles, as shown in Fig. 2.11. These boundaries only accurately model forming conditions when there is very little change in shape during deformation and the boundaries were originally both straight and at right angles, thus restricting the range of parts which may be analysed. If we consider most real parts, this is rarely the case.

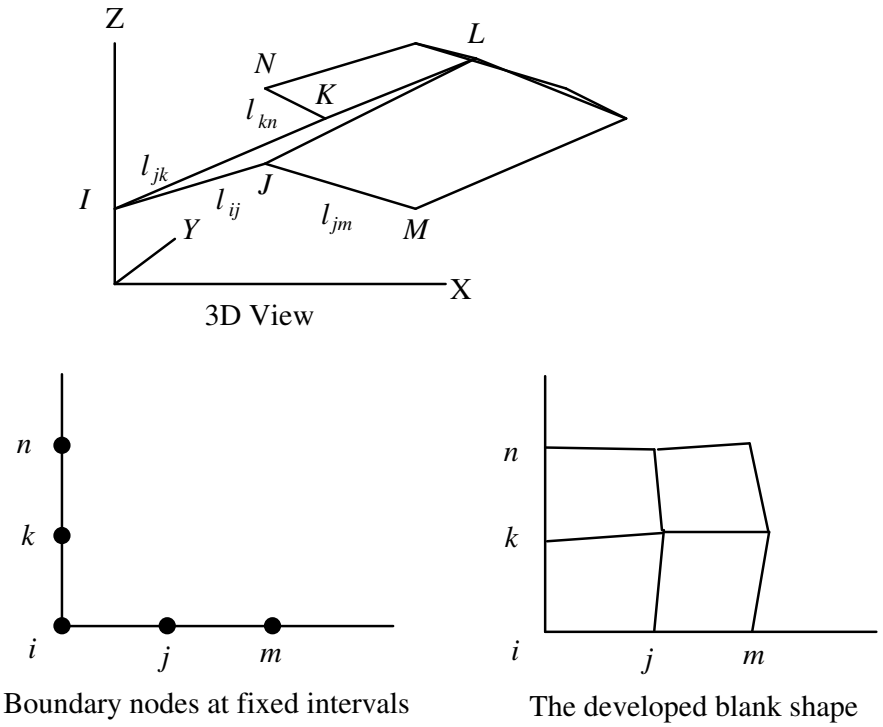


Fig 2.10 Regular lengths at right angles.

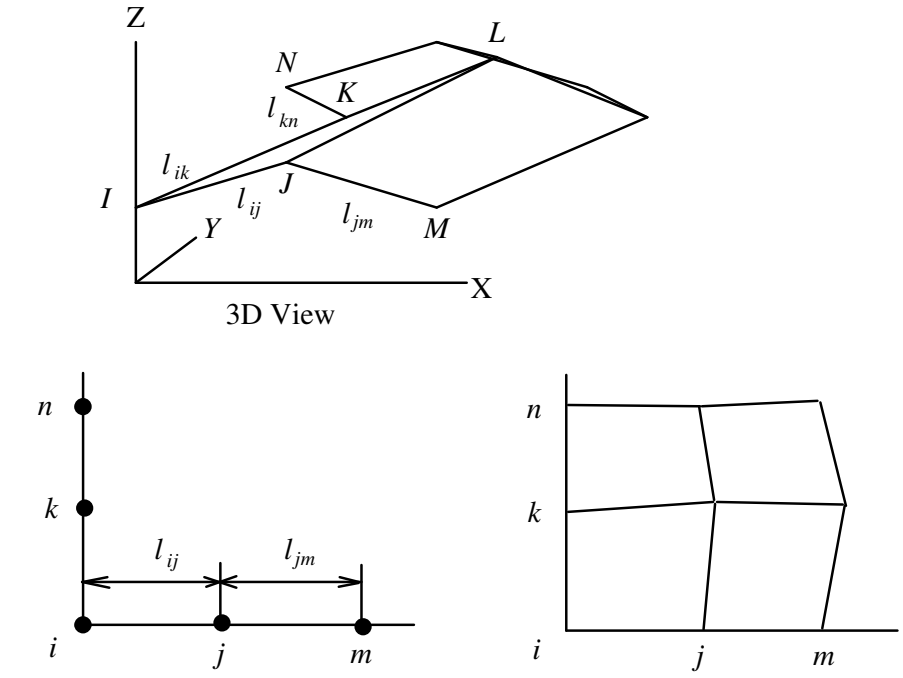


Fig 2.11 Actual lengths at right angles.

2.4.2. New Methods of Boundary Definition.

Common 'real' boundaries are planes of symmetry and free edges.

2.4.2.1 Plane of Symmetry.

If we consider the plane of symmetry shown in Fig. 2.12 we note that no shear occurs along the plane, therefore lines parallel and close to the plane of symmetry remain parallel, and lines close to and perpendicular to the plane of symmetry remain perpendicular. This is because the plane of symmetry is a principal stress plane.

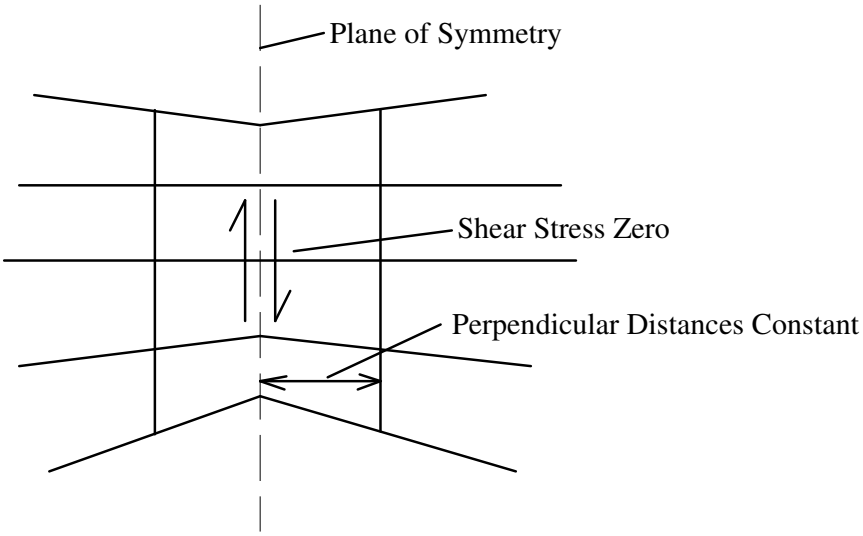


Fig 2.12 Elements adjacent to an axis of symmetry.

To model this boundary the 'Curve Match' boundary specification method was developed. To simulate the presence of a plane of symmetry at the boundary, lines parallel to the boundary remain parallel; this is illustrated by Fig. 2.13.

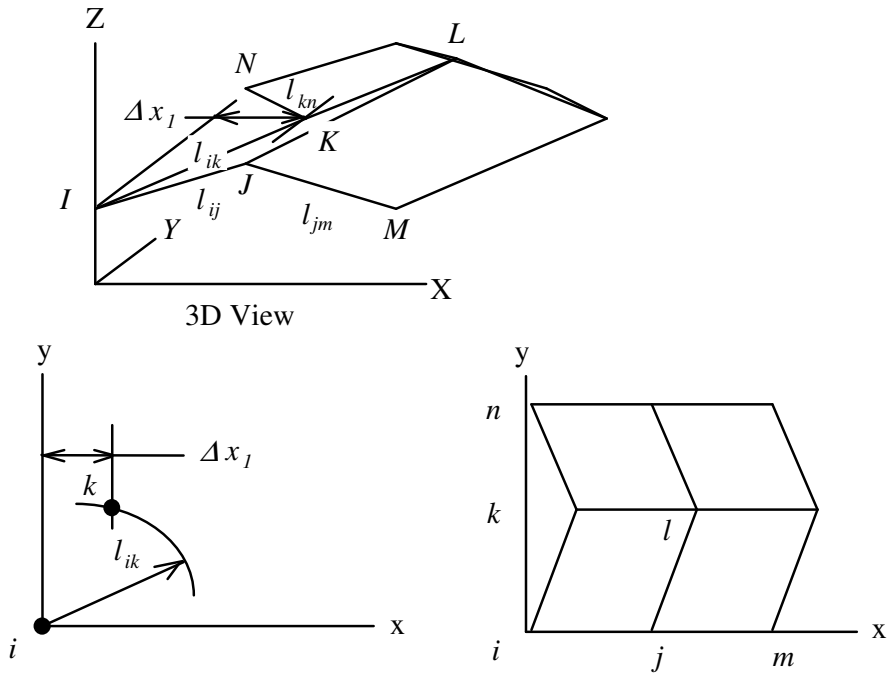


Fig 2.13 The 'curve match' method of blank shape prediction.

In the 'Curve Match' method shown in Fig. 2.13 the boundary closest to the Y axis is being laid down. Starting from the origin we are attempting to find the plane position of the three dimensional point *K*. First the true length of line l_{ik} is determined. The distance between the Y axis and the point *K*, Δx_1, is calculated and conserved. Point *k* is found by plotting a line parallel to the Y axis offset by the distance Δx_1 and then swinging the known length l_{ik} in an arc to intersect the line. This intersection gives the plane transformation of the point *K*, point k, as calculated by the 'Curve Match' method.

2.4.2.2 Free Edge.

We now consider the Free edge shown in Fig. 2.14.

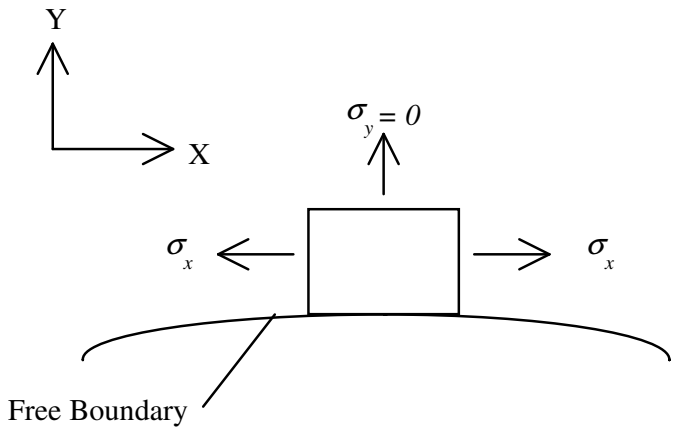


Fig 2.14 A free boundary. No stress can be maintained perpendicular to the boundary.

The only stresses that can act on the element are parallel to the free edge. Taking this further if we consider a small element in a corner, no forces can act on it, and the element is undeformed, with the internal angles remaining constant. If a free boundary has only a low or zero stress gradient perpendicular to it, the relative angles of the elements of the boundary will be approximately conserved.

To model this boundary the 'Angle Conservation' boundary specification method was developed. As an approximation of this in the Angle Conservation method three dimensional line lengths and internal angles are preserved, as shown in Fig. 2.15.

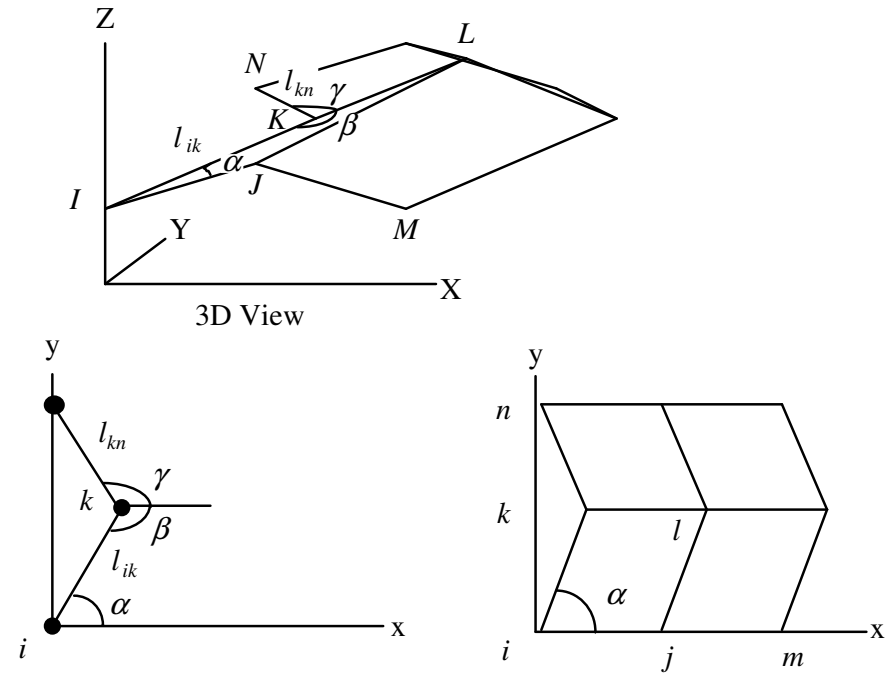


Fig 2.15 The 'angle conservation' method of blank shape prediction.

In the 'Angle Conservation' method shown in Fig. 2.15 the boundary closest to the Y axis is being laid down. Starting from the origin we are attempting to find the plane position of the three dimensional point *K*. First the true length of line l_{ik} is determined. The angle between the points *K I J,* α, is calculated and conserved. Point *k* is found by plotting a line of length l_{ik} at an angle of α to the *X* axis. This plot gives the plane transformation of the point *K*, point *k*, as calculated by the 'Angle Conservation' method. The position of point *N* and other points are found by calculating the internal angles β and γ. These internal angles can then be added to give the total angle change. This can then be used along with the true length l_{kn} to find the position of point *N*.

2.5. COMPUTER AIDED BLANK SHAPE PREDICTION ASSESSMENT.

The development of new methods of boundary specification prompted investigation of means of assessing the performance of the various algorithms.

2.5.1. Visual Assessment.

The most obvious means of assessing the correctness of the result is visual - does the shape produced look like a possible blank shape for the original three dimensional part; is the connectivity correct; are the boundaries smooth etc. This is an important method but is very difficult to quantify. Two methods that produce similar shapes can be visually hard to tell apart.

2.5.2. The Area Ratio.

Another method of assessment developed by the author is the comparison of areas. By definition in the constant area transformation we are attempting to conserve area. A simple possible measure of the accuracy and correctness of the transformation can be found by comparing the areas of the three dimensional and two dimensional quadrilaterals that make up the initial part and the final blank shape.

This gave rise to an area ratio for each quadrilateral,

$$AR = \frac{\text{Area of 3D element}}{\text{Area of element in blank}} \tag{2.1}$$

The information that may be obtained by considering individual quadrilaterals' area ratios is detailed in a later section. Only the average area ratios are considered in this section. The average area ratio may be defined as the total of all the area ratios divided by the number of quadrilaterals. The average area ratio, or $\overline{AR}$ provides a numerical measure of the accuracy of the constant area assumption over the entire part. The following results also indicate that as $\overline{AR}$ approaches 1.0 the accuracy of the blank shape prediction increases. Thus $\overline{AR}$ provides a convenient method of assessing Computer Aided Blank Shape Prediction assessment.

If we consider the simple test piece illustrated in Fig. 2.9, the average area ratios for the four boundary specification methods are shown in Table 2.16. The increase in $\overline{AR}$ corresponds to an increase in the visual accuracy of the mapping - it looks more like the original part shape.

Table 2.16 Average area ratio values.

Method of Boundary Specification	**Fig. No.**	$\overline{AR}$	**% Error**
Regular side length at 90°	2.10	1.039	3.9
Actual side length at 90°	2.11	1.036	3.6
Curve Match	2.13	0.994	0.6
Angle Conservation	2.15	0.994	0.6

The above results suggest that the two new methods of blank shape prediction are more accurate. Analysis of more complex components can now be attempted with confidence.

2.5.3 Auto body pressing Analysis.

One of the biggest users of pressed sheet metal is the auto industry. In this section part of an aluminium auto body panel is analysed by the preceding methods. The component was marked with a fine square mesh before pressing then digitised after forming [Z. Zhang 1993]. The auto body pressing is shown in Fig. 2.17.

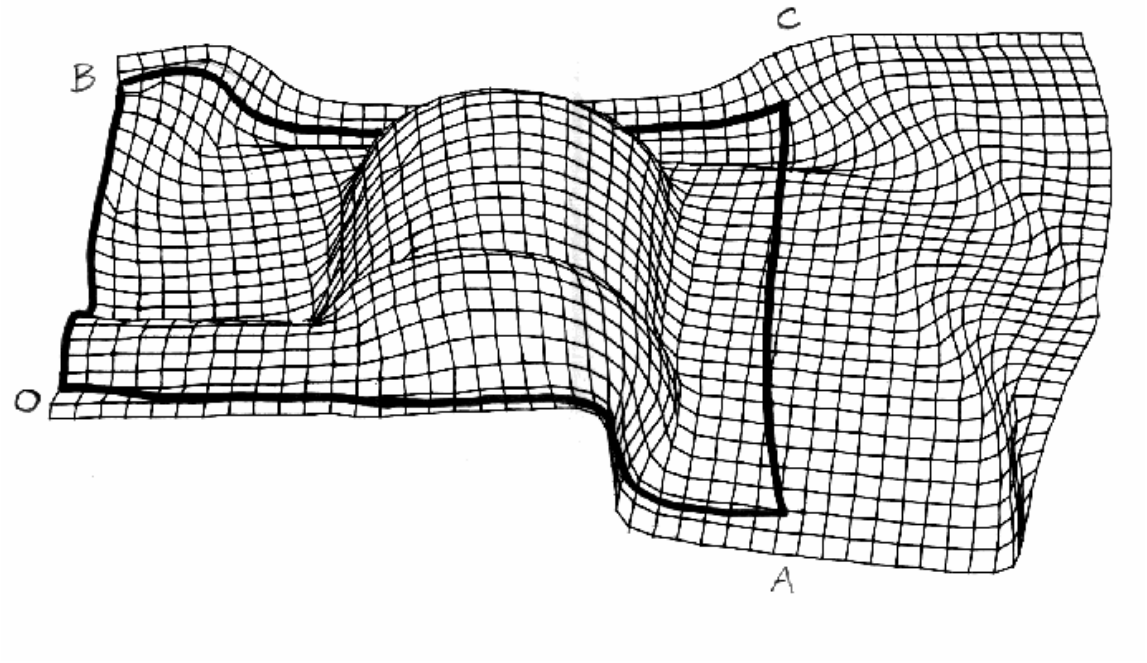


Fig 2.17 A fine map mesh of the 3D auto part. The area used for the blank shape prediction is outlined by the bold line.

In Fig 2.17 the boundary OB was towards the centre of the deformed sheet and was close to a plane of symmetry. Boundaries OA and BC were near to free edges and boundary AC was in the middle of a sizeable part of the sheet.

In Fig. 2.18 the bold outline borders the area used as a test for the blank shape prediction program. This area then had a quadrilateral mesh drawn onto it. The mesh was digitised and used as the input for the program, see Fig. 2.18. The deformed shape boundary corner nodes are labelled O, A, B, and C. The predicted blank shape corner nodes will be denoted by O, A', B', and C'. O is the reference node and is common to both deformed and blank shapes.

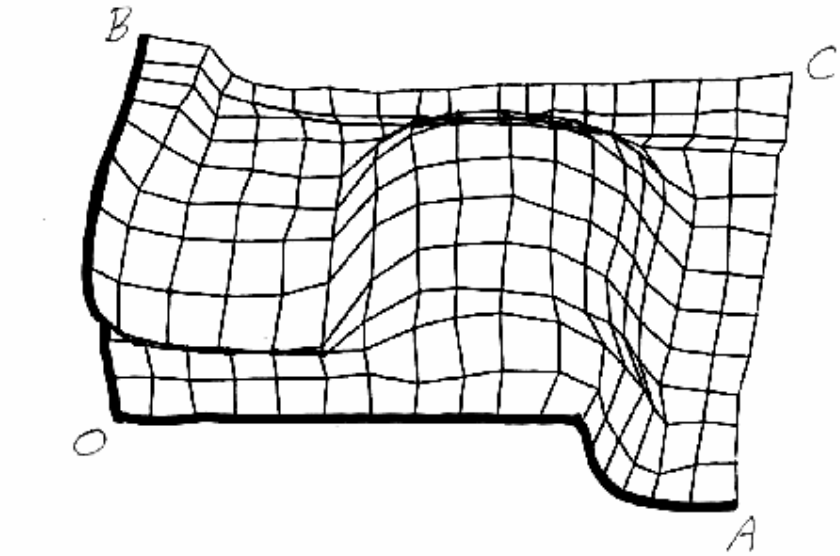


Fig 2.18 The 3D digitised mesh used as the input for the blank shape prediction. The bold line denotes the chosen boundary.

The section chosen had been, prior to deformation, marked with a regular grid mesh. This permitted large strain analysis to be performed on the part and also allowed the 'initial' shape of the drawn-on grid flat blank to be determined. The initial blank shape was determined by plotting the intersections of the drawn-on mesh (marked after deformation) and the scribed regular mesh (marked prior to deformation) as shown in Fig. 2.17.

The intersections with the regular mesh produce the 'initial' blank shape shown in Fig. 2.19. This is used as a comparison for the various predicted blank shapes.

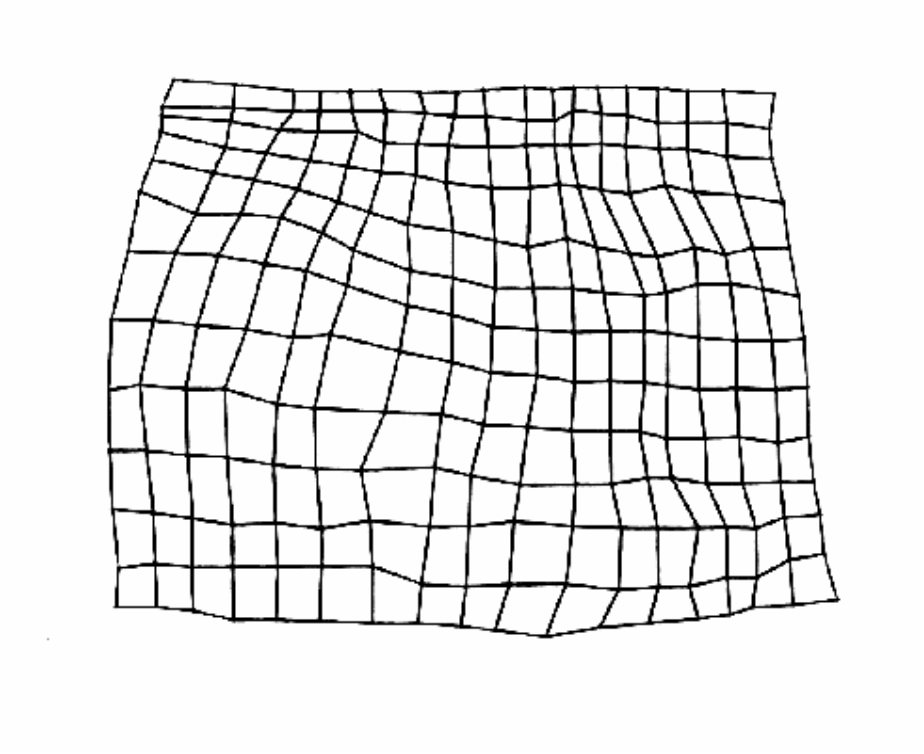


Fig 2.19 The 'actual' initial blank shape of the deformed section. This was determined by transforming the intersections between the fine mesh and the coarse mesh transformed back to the original rectangular grid.

Fig 2.20 illustrates the blank shape predicted using fixed boundary nodes as described in section 2.4.1 (line style ÄÄÄÄ) overlaid on the 'actual' blank shape (line style - - - - -). The average area ratio, $\overline{AR}$ is 1.059 which shows reasonable agreement between the area of the blank shape and the deformed part. Clearly there is little agreement along the boundary nodes. While A and A' are close the boundary between O and A is curved while O and A' are, by definition, straight. The same applies to OB. The large distortion along boundary BC' is caused by the large bump in the deformed part at this point; because the transformation uses a straight line approximation errors occur in areas of high curvature. The bump is also be an area of biaxial stretching, where the assumption of constant area is less valid. It is this that causes the distortion. While the mapping is generally good, the boundary match is poor.

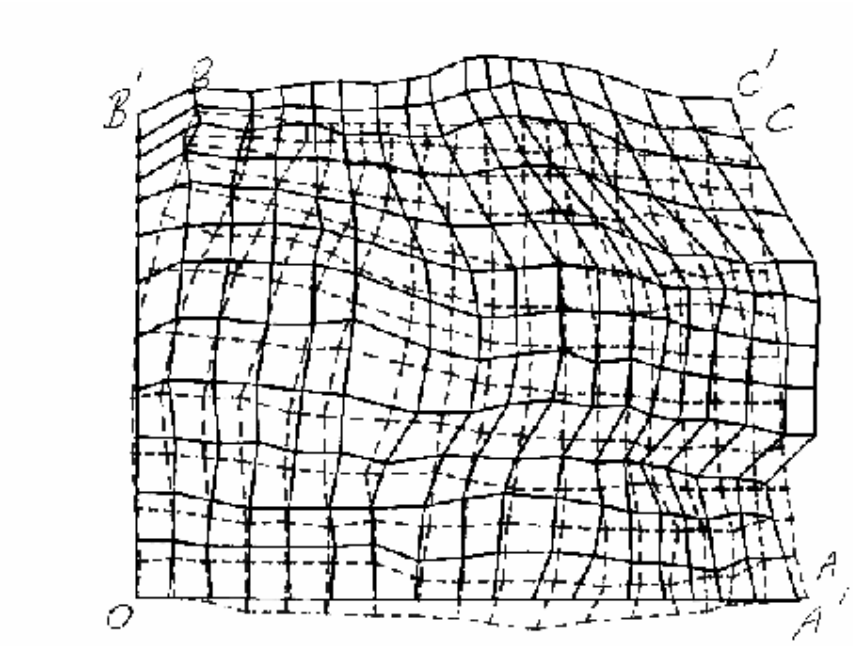


Fig 2.20 The blank shape predicted using fixed boundary nodes overlaid on the actual blank shape. The $\overline{AR}$ is 1.059.

Fig 2.21 illustrates the blank shape predicted using the angle match method of assigning boundary nodes, overlaid on the 'actual' blank shape. The average area ratio, $\overline{AR}$, is 0.9989 which shows excellent agreement between the area of the blank shape and the deformed part. Boundary OA is mapped with increased accuracy, OA' following the curve to produce a more accurate shape. OB is not well mapped by OB' as the angle conservation curves away at the top. C' however lies closer to the position of C than in the fixed side prediction. While the mapping is generally good, and the lower boundary accurate, the OB boundary match is poor.

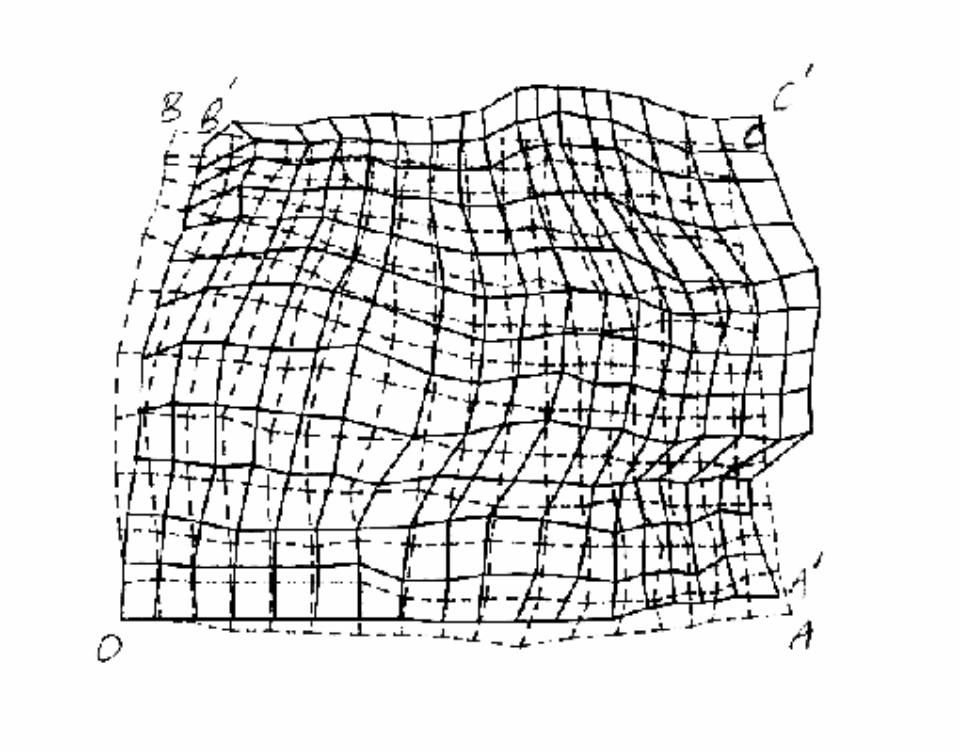


Fig 2.21 The blank shape predicted using the angle conservation method of placing boundary nodes overlaid on the actual blank shape. The $\overline{AR}$ is 0.9989.

Fig 2.22 illustrates the blank shape predicted using the curve match method of assigning boundary nodes, overlaid on the 'actual' blank shape. The average area ratio, $\overline{AR}$, is 1.024 which again shows good agreement between the area of the blank shape and the deformed part. Boundary OA is mapped with reasonable accuracy, OA' following the curve but at the end near A' curving in the wrong direction. OB' is extremely well mapped and it follows OB almost perfectly. C' however lies further from the position of C than the fixed side prediction. While the mapping is generally good, and the OB boundary accurate, the OA boundary match is only reasonable.

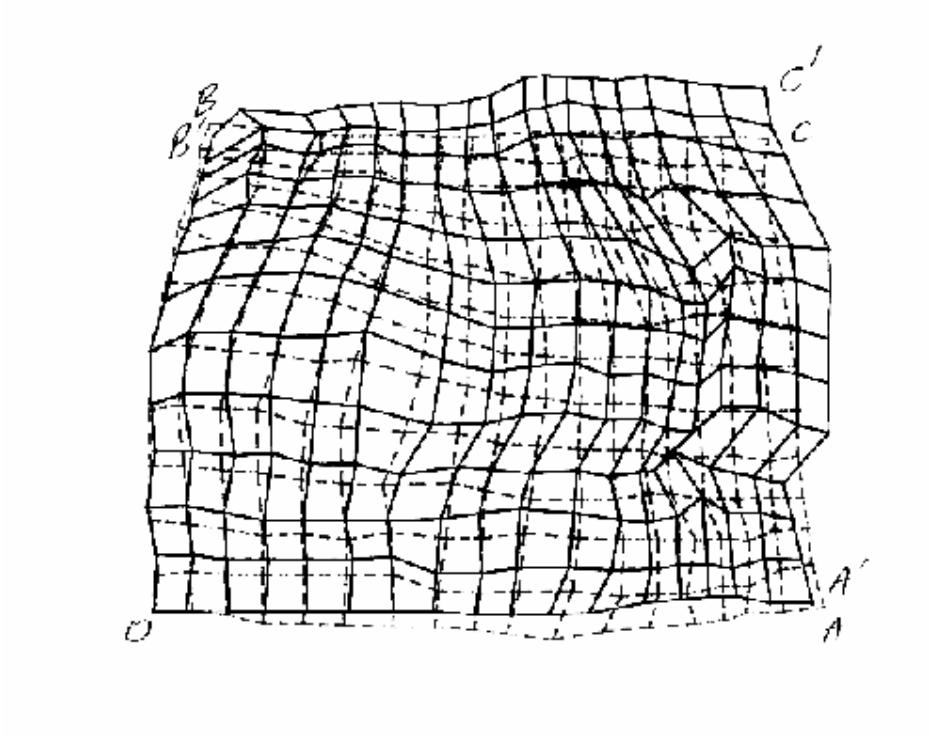


Fig 2.22 The blank shape predicted using the curve match method of placing boundary nodes overlaid on the actual blank shape. The $\overline{AR}$ is 1.024.

Obviously the best result would be to combine the boundary methods, allowing the appropriate boundary condition to apply to each side. The prediction program has been modified to allow this, the user being given three options for each boundary: fixed, angle match and curve match. Specifying a separate boundary condition for two boundaries produced the result shown in Fig. 2.23. In this case the OA boundary is modelled by angle conservation and the OB boundary by curve matching. There is an improved correspondence to the real part, as edge OA is free and boundary OB is in the middle of a wider part of metal and forms a rough plane of symmetry.

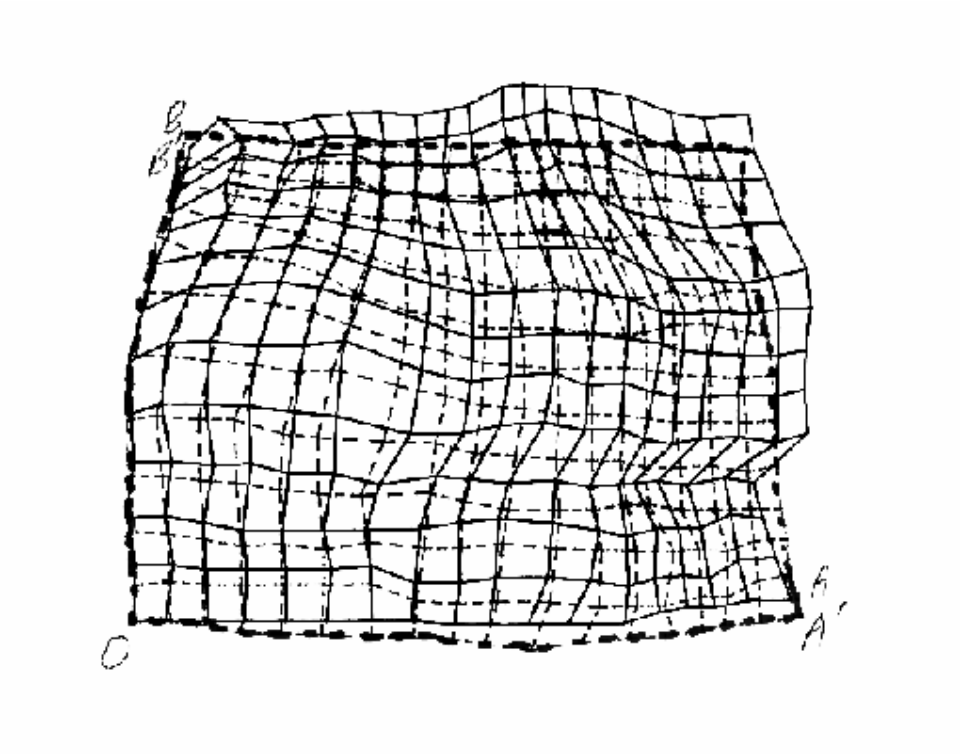


Fig 2.23 The blank shape predicted using the curve match method for boundary OB' and the angle conservation method for boundary OA'. The $\overline{AR}$ is 1.002.

The average area ratio, $\overline{AR}$ is 1.002 showing excellent agreement between the blank area and the area of the deformed shape. Boundary OA is mapped with good accuracy, OA' following the curve. OB' is extremely well mapped as it follows OB almost perfectly. C' lies closer to the position of C than any of the previous predictions. The general mapping is encouraging, with both boundaries accurately mapped.

Table 2.24 Blank Shape Transformation Results.

Boundary Mapping Method	**AR**
Fixed	1.059
Angle Conservation	0.9989
Curve Match	1.024
Combined Angle Conservation and Curve Match	1.002

These results, shown in Table 2.24, illustrate the benefits of increased accuracy in boundary node placement. As illustrated by both the average area ratio and visual inspection, the closer the boundary node match the more accurate the total mapping. Boundary node accuracy is markedly increased by the use of the two new methods, angle conservation and curve match. These provide a simple yet realistic method of modelling common real life boundary conditions. The inclusion of these methods in a Computer Aided Blank Shape Prediction program increases both the accuracy of the prediction and the range of shapes that can be mapped.

2.6. PROVIDING MORE THAN SHAPE INFORMATION WITH COMPUTER AIDED BLANK SHAPE PREDICTION.

Computer Aided Blank Shape Prediction involves a numerical analysis of a deformed three dimensional surface. Can this analysis provide us with further forming information about the part?

As previously mentioned (in section 2.5.2) an Area Ratio of the areas of the three dimensional quadrilateral to the plane quadrilateral is of use in determining the accuracy of the transformation. In the following section further uses for the information obtained by the individual area ratios (AR) as opposed to average area ratio ($\overline{AR}$)will be examined.

2.6.1. Effective Strain vs. Area Ratio.

If we compare plots of effective strain with plots of values of area ratio, there are several notable similarities. Effective strain is a measure of the work done during a non-strain-hardening deformation and may be defined as:-

$$\bar{\varepsilon} = \frac{2}{3}\sqrt{\varepsilon_{11}^2 + \varepsilon_{22}^2 + \varepsilon_{33}^2} \tag{2.2}$$

Figure 2.25 is a contour plot of effective strain calculated from the measured deformed small grid, using the large strain analysis method described in appendix 7.1. The regions of greatest effective strain are labelled A to E.

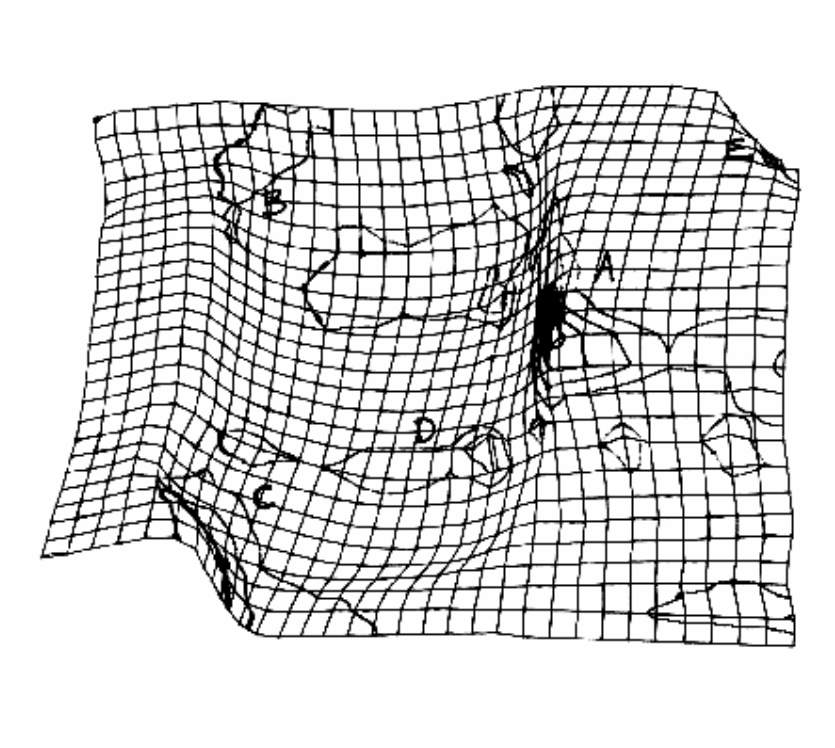


Fig 2.25 The effective strain ($\bar{\varepsilon}$) contour map. The contours range from 0.1 to 0.5. Peak strains are: A 0.67, B 0.43, C 0.51, D 0.44, E 0.63.

Figure 2.26 is a contour plot of the areas where the area ratio exceeds 1.1 (red) or is below 0.9 (blue). These regions are labelled a to e. It is immediately obvious that there is a good correlation between areas of high effective strain and areas where the area ratio deviates significantly from 1.0. In particular areas A and a, B and b and C and c are in very similar positions.

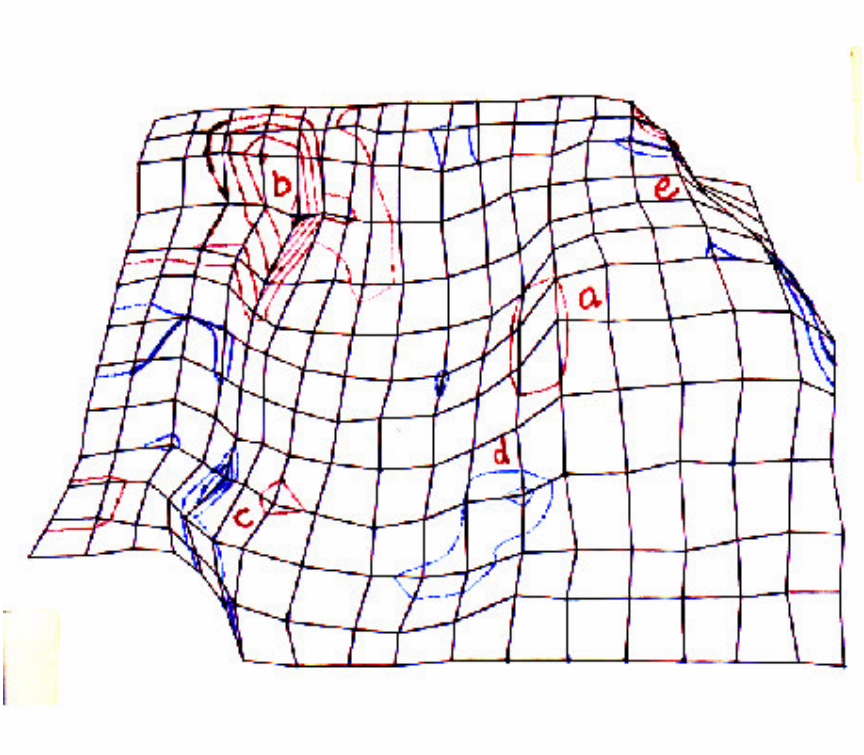


Fig 2.26 The area ratio contour map (AR). RED 1.1 <AR< 1.5, BLUE 0.5 <AR< 0.9.

The Area Ratio for individual quadrilaterals provides us with clues to the location of areas of high effective strain. Areas of high effective strain are also areas where failure, or forming difficulties are more likely to occur. Thus the 'local' area ratios provide us with some simple predictive forming information.

These results indicate that the Computer Aided Blank Shape Prediction program developed may not only provide a predicted blank shape, but also a clue to problem forming areas.

2.7. CONCLUSIONS.

This work on computer aided blank shape prediction has resulted in three major improvements.

The two new methods of specifying boundary shape, angle conservation and curve match, permit the experienced designer to improve the predicted blank shape. By using their knowledge of the component, what edges are free or approximate planes of symmetry the designer can input this information to the computational method. The results obtained can give a better approximation of the actual blank shape, both visually and numerically.

The area ratio can provide information on the accuracy of the constant area transformation.

An area with area ratios significantly greater or less than one may indicate possible areas of high strain, areas which may cause problems during the forming process.

These three developments may enable the designer to more accurately and usefully predict blank shape, thus speeding development and eventual production.

3. FOLDED DEVELOPABLES.

This chapter examines folded developables. Developables are defined and the mathematics that governs their behaviour is presented. The folding of developable surfaces along curved lines is introduced and discussed and a computational method for modelling folding is presented along with examples of its use.

3.1. DEFINITIONS OF A FOLDED DEVELOPABLE.

Before embarking on a mathematical description of folded developables several definitions are necessary. To define a folded developable, notation, developable surfaces and 'developments' must first be defined.

3.1.1. Notation.

Scalars

s	arc length parameter along a curve
β	angle between the principal normal and the tangent plane
γ	angle between the tangent and a generator
α	fold angle
$S_{1,2}$	developable surfaces 1 and 2
P	a point on the developable surface
C	a space curve
K	first curvature of a space curve, surface curvature in a particular direction
$K_{1,2}$	principal curvatures of developable surfaces $S_{1,2}$
K_g	geodesic curvature
τ	torsion of a curve
τ_g	geodesic torsion
ω	angle between the osculating plane and the surface normal
ρ	radius of curvature

Vectors

b binormal unit vector

n unit principal normal vector to a curve

N unit vector normal to a surface

r position vector of a point

t unit tangent vector of a curve

u unit vector in a tangent plane normal to a surface curve

This notation applies for Chapters 3, 4 and 5.

3.1.2. Developable Surfaces.

The Shorter Oxford Dictionary defines a developable surface as "a ruled surface in which consecutive generators intersect". The Encyclopaedic Dictionary of Mathematics for Engineers [Sneddon 1976] defines a ruled surface as "a surface that can be generated by moving a straight line with one degree of freedom". This is shown in Fig. 3.1, which also shows the developable surface being 'unwrapped' to form its development. The various positions of the line, moved with one degree of freedom, identify the "generators". Thus a developable surface or developable is a ruled surface with generators that intersect either at a point or intersect and are tangent to some arbitrary curve. This intersection point or curve is known as the edge of regression. Thus a developable surface is the surface described by a series of lines tangent to the edge of regression.

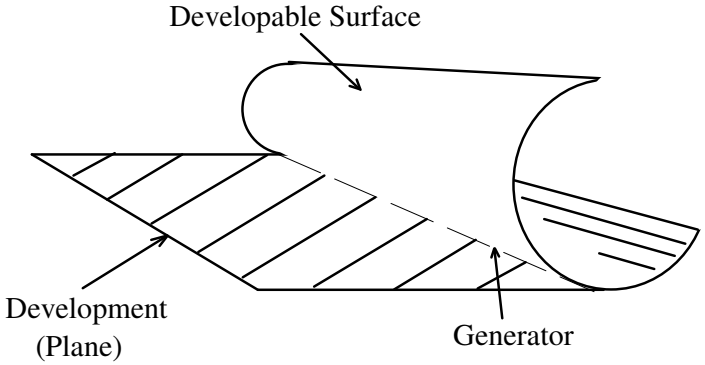


Fig. 3.1 A developable surface showing the plane development being 'unwrapped' from the surface.

Common developable surfaces have simple edges of regression. The two most common developable surfaces are the cone and the cylinder. The cone has an edge of regression at its vertex. This means that the generators that form a cone have the vertex of the cone as their origin. The cylinder has an edge of regression at infinity because the generators are parallel.

3.1.3. Development.

A development of a developable surface is formed when a developable surface is unwrapped onto a flat plane. In the process of unwrapping generator length and arrangement is preserved. The arc distance between two points on a developable is also conserved in the development. Because a development is flat its curvature in any direction is zero.

For a surface to be developable it must be capable of being unbent into a plane; i.e. a developable surface can only be deformed by bending about its generators. A cone may be unbent to form a segment of a circle and the radius of the circle will be the slant height of the cone. This is the development of a cone. Likewise the development of a cylinder is a rectangle with one side equal to the length of the cylinder and the other side equal to the circumference.

3.1.4. Folded Developables.

A folded developable is a developable surface that has been folded along an arbitrary curve. Folding a developable along an arbitrary curve will produce two separate developable surfaces joined along the fold curve. The total object is known as a folded developable.

Not all curves on a developable surface can be used as fold curves; for example a folded developable cannot be formed by folding one developable surface along a straight line which is not itself a generator (alternatively it might be stated that a straight fold folds the developable surface back on itself resulting in two identical surfaces occupying the same, or in a physical example where thickness is greater than zero, very close to the same, space). Some folding curves may contain points of inflection that cause the folded surfaces to change the sign of their curvature. An example of a folded developable and its development is shown in Fig. 3.2.

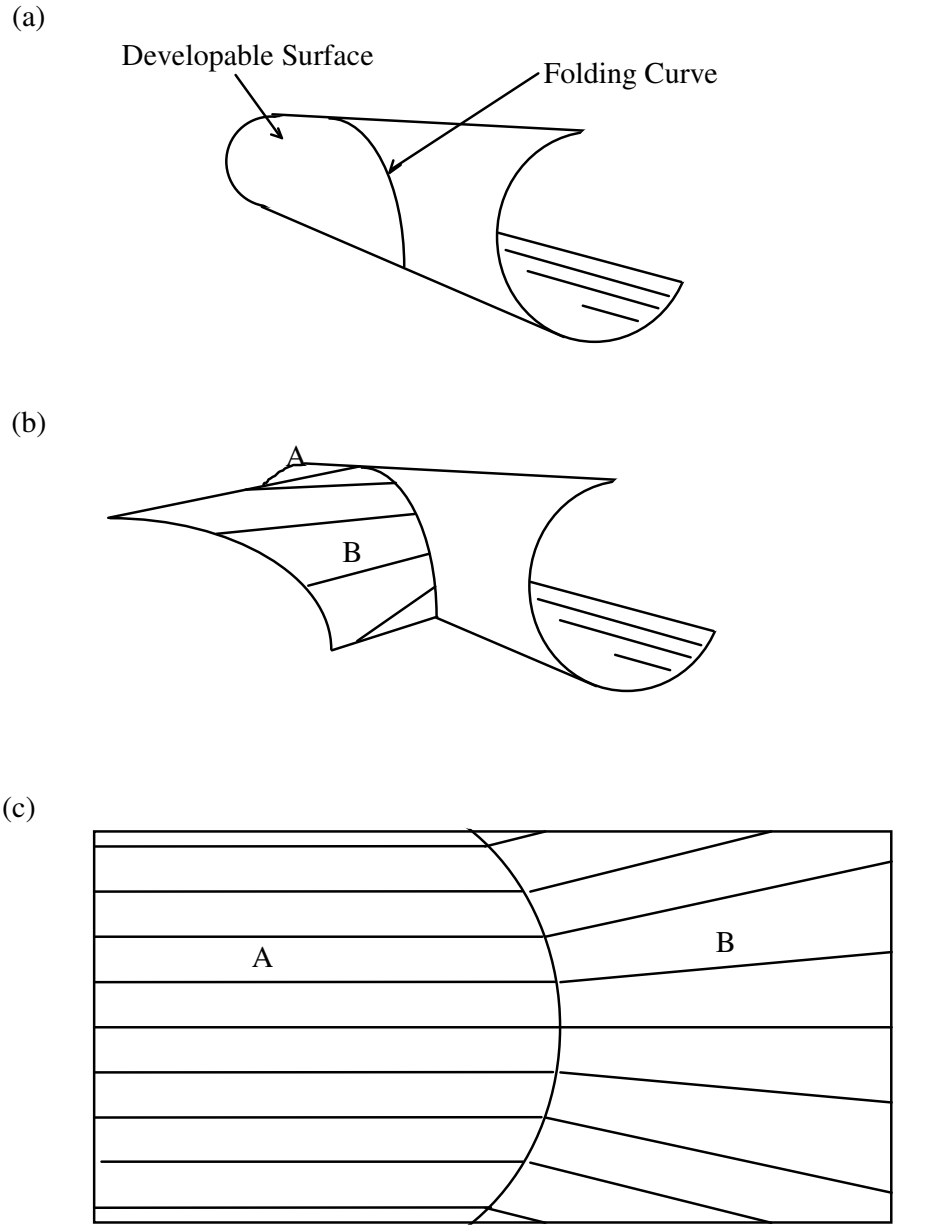


Fig 3.2 Curved line folding (a) a folding curve on a developable surface. (b) folding along the curve to produce the second developable surface B from the original surface A. (c) the plane development of both surfaces.

Folded developables can be made of sheet material - flat sheets of cardboard, sheet metal or plastic for example. Sheet metal represents the area of largest potential for folded developables due to its widespread use.

3.2 A REVIEW OF THE MATHEMATICS OF FOLDED DEVELOPABLES.

The mathematics of folded developable has seen two major areas of contribution;

1. the creation and description of developable surfaces.

2. elucidation of folded developables.

3.2.1 Developable Surfaces.

Developable surfaces have been detailed in many mathematical texts. Willmore [1959] comprehensively details developable surfaces, specifying their mathematical and geometric properties.

A large amount of the work on developable surfaces has been produced by Naval architects. Ship hulls are traditionally constructed from steel plates or timber planks. If the hull is formed from developable surfaces large savings in construction time can be achieved as the hull material does not need to be deformed.

Vickers, Bedi, Blake and Dark [1987] describe the use of computer generated developable surfaces to loft and fair hull shapes. Lofting is the process of laying out a full size working drawing of ship lines and contours, usually on the wooden floor in the loft above the plant. Fairing is the art of modifying a surface to achieve smoothness of form in all three views (plan, side and end elevation). Fairing is a skilled, time-consuming, iterative approach in which lines drawn in one view are projected into the other two views until all lines appear acceptably smooth.

To create a developable hull shape, the hull is initially described by a series of cross sections known as station lines. These cross sections are then joined by a series of polynomial splines known as profile lines. Developable surfaces are generated between pairs of profile lines as shown in Fig. 3.3.

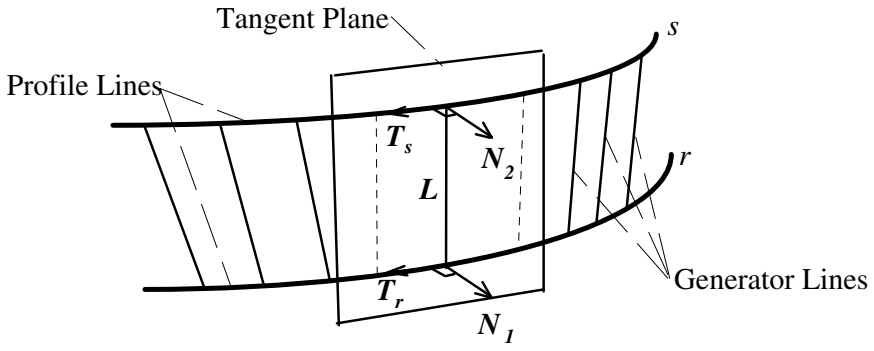


Fig 3.3 Straight line generators for developable surfaces.

The normals to the generator lines may be calculated from;

$$\begin{aligned} \mathbf{N_1} &= \frac{d\mathbf{r}}{dx}(x_i)\times\mathbf{L_i} \\ \mathbf{N_2} &= \frac{d\mathbf{s}}{dx}(x_j)\times\mathbf{L_i} \end{aligned} \quad . \tag{3.1}$$

The tests for developability are;

$$\begin{aligned} \mathbf{N_1}\times\mathbf{N_2} &= 0 \\ \mathbf{N_1}.\mathbf{N_2} &= 1 \end{aligned} \quad . \tag{3.2}$$

The use of such a system to design ship hulls proved extremely successful with an estimated cost saving of USD 15000 to 20000 per ship. This was due to the greatly reduced time taken for lofting and fairing and the more efficient use of material.

Two methods for creating simple developable surfaces between arbitrary curves are described by Weiß and Furtner [1988]. The first method, by Weiß, follows seven steps to create developable surfaces between two space curves p and q:

1. parameterise the curves p and q.
2. take an arbitrary point P on p and its left and right neighbour P_l and P_r on p.
3. on q pick point Q, Q_l and Q_r.
4. test if Q_l, Q_r, P_l and P_r are coplanar (using modified forms of equations 3.1 and 3.2) and if they are not,
5. pick out the 'next' series of points $Q_{\text{-new}}$, $Q_{l\text{-new}}$ and $Q_{r\text{-new}}$ on q as shown in Fig. 3.4 and test again; and if the four points are finally coplanar

6. store the co-ordinates of *P* and *Q*

7. go on with the 'next' *P* on *p* until finished.

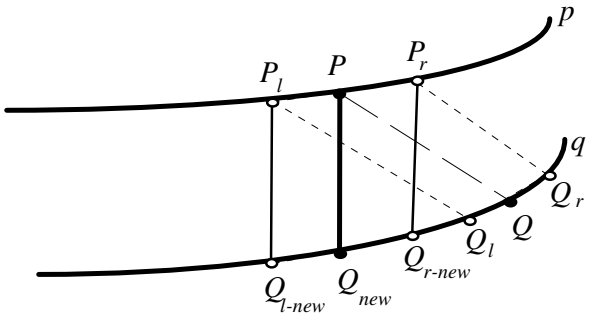


Fig 3.4 Developable generators connecting the curves *p* and *q*.

This method simply requires a small computer program to allow developable surfaces to be generated between two general curves.

The second method, developed by Furtner is more accurate, but depends on the ability of curves *p* and *q* to be defined parametrically. If the curves are expressed in the parametric form $p=p(\boldsymbol{u})$ and $q=q(\boldsymbol{v})$ then the point *Q* which defines the developable surface generator may be found by determining its corresponding parameter *v* from the equation

$$d(\mathbf{v})_{\mathbf{u}} = \left| p'(\mathbf{u}).q(\mathbf{v}) - p(\mathbf{u}).q'(\mathbf{v}) \right| = 0 . \tag{3.3}$$

This equation is solved numerically using either a change in tangent angle between developable surfaces or a maximum step distance as an aid to convergence on a correct solution.

The major advantage of Furtner's method, besides increased accuracy is the increased robustness of the algorithm. It can be successfully applied with less user intervention.

Naval Architects Letcher, Brown and Stanley [1988] also examine the use of developable surfaces for lofting and fairing ship hulls. In addition to the approach of Vickers et al. [1987] and the method described by Weiß they consider the end effects of developable panels.

At the end of a pair of profile lines it may not be possible to form developable surfaces. The creation of imaginary extensions of the profile lines, which allow the surfaces to 'run out', as shown in Fig. 3.5, ensures the entire panel is composed of developable surfaces.

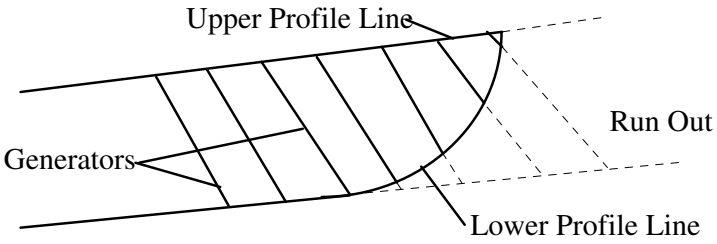


Fig 3.5 The end of a developable panel from a yacht with the 'run out' shown by the dashed lines. The run out allows the panel to be fully specified by developable surfaces that are then trimmed to size.

This method has been extensively applied in the commercial software packages FAIRLINE™, AutoYACHT™ and AutoSHIP™.

Aumann [1991] described the extensive mathematical conditions for the extension of Weiß and Furtner's methods to use developable Bézier patches. The patches are used in place of the flat quadrilaterals to form a developable surface between two curves.

Though complex, this method makes it possible for surfaces to be first order continuous (i.e. tangents continuous) and in some cases second order continuous (i.e. rate of change of the tangent along the surface continuous).

An unusual approach to developable surfaces was taken by Redont [1989] who looked at a novel method for representing and controlling developable surfaces. In Redont's method the developable is generalised by describing its spherical indicatix. This is the transformation of a representative curve on the developable onto a unit sphere, where the position of the curve $X(s)$ say, is transformed to a curve $N(\sigma)$ on the sphere where σ is a function of the arc length of the developable, s. A knowledge of this transformation, plus the angular relationship between the tangent to the representative curve and the normal to the surface, allows the developable to be extracted from the spherical indicatix. In simple cases the relation between the two vectors is often constant or a linear function.

In the special case of a conical developable the spherical indicatix reduces to a circle. Thus the developable can be created and manipulated by specifying the position and shape of a curve on the circular indicatix that represents the transformation space of the developable.

This method produces developable surfaces composed of conical sections that can be manipulated and deformed. Despite its mathematical interest Redont's method has little practical application as there is no easy physical or visual link between the line on the spherical indicatix and the developable surface it produces. Redont's deformation does not allow for the creation of folded developables.

Developable surfaces can today be easily created between two general curves. CAD packages such as Pro/ENGINEER™, CATIA™ and AutoSURF™ all allow their creation simply by selecting the two lines.

The challenge this work conquers is to give designers a tool with which they can easily create any general folded developable.

3.2.2 Folded Developables.

Duncan and Duncan [1982] produced the definitive work on folded developables. Their elucidation of the mathematics of folded developables is referred to in Section 3.3. While their work detailed the mathematical relationships of the two developable surfaces and the folding curve, it does not present a method for the creation of a general folded developable. However their inclusion of several special case folded developables, typically involving conic sections, allowed developables to be used as engineering structures. Such applications of folded developables, usually developed by trial and error are now common in every day life, especially in the packaging industry.

Staublin and Gerdeen [1986] investigated the springback of a particular folded developable. The special case chosen was a sheet metal conical developable with a plane fold line and constant fold angle. The folding curve radius, the radius between the generators of the two surfaces, was chosen as zero.

Staublin and Gerdeen used finite element analysis to model the springback. The folded developable was modelled as two elastic surfaces joined by a plastic hinge. The forces and moments caused by the forming of each surface can be determined for the special case and are then used to determine the springback. The springback is determined by allowing the stresses trapped after folding to relax producing a change in shape.

For the developable shown in Fig. 3.6 Staublin and Gerdeen found that, the spring back angle $\Delta\theta$ was at a minimum in the middle of the fold curve increasing to a maximum at the edges.

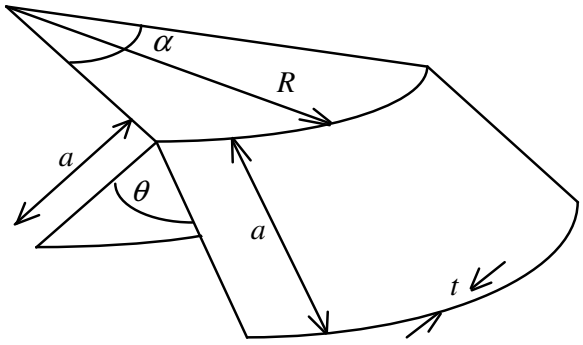


Fig. 3.6 Geometry of a surface resulting from curved line folding.

They also found the following approximate relationships for the springback angle;

$$\Delta\theta \propto \frac{1}{t^{\frac{3}{2}}} \tag{3.4}$$

$$\Delta\theta \propto a \tag{3.5}$$

$$\Delta\theta \propto R^2 \tag{3.6}$$

Recently Kergosien, Gotoda and Kunii [1994], published work on the virtual creasing of paper. Their investigation of creasing used a finite element style approach, successively solving a grid mesh for minimum energy, and then comparing results with models of creased paper. They have produced representations of zero or very small radius creases and straight line creasing. These results are similar to folded developables, in that they produce two developable or near developable surfaces joined by a crease or fold. However due to the lack of the precise geometric relationships developed later in this work, the representations are approximations that provide 'qualitative' results for a few special cases only.

This work presents a solution to the major barrier to the use of folded developables in engineering. Section 3.4 describes a systematic design theory for the creation of a general folded developable. The design theory and the computational tool based on it allow the designer to design folded developables rapidly and accurately.

3.3. THE MATHEMATICS OF FOLDED DEVELOPABLES.

3.3.1. Geometry of a Developable Surface.

A point on a general developable surface is shown in Fig. 3.7. A general developable surface consists of a series of generators that originate from an edge of regression.

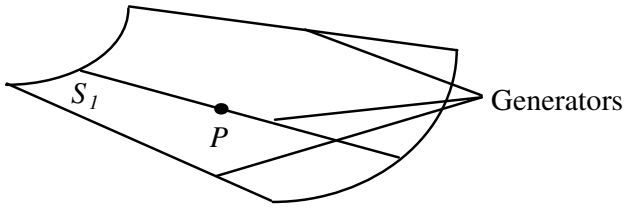


Fig 3.7 The general developable surface S_1.

The curvature of a surface is a measure of the rate of change of the slope of a surface with respect to the distance across the surface. It is a property of developable surfaces that in the direction of the generators the curvature is zero. Because generators are straight lines their slope does not change along their length.

However perpendicular to the generators the curvature of the developable surface is at a maximum. This curvature is known as the principal curvature and is shown in Fig. 3.8.

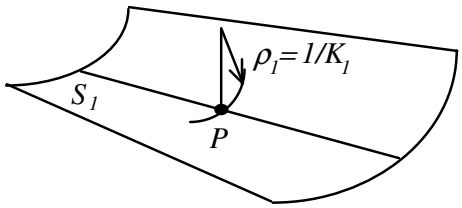


Fig 3.8 The general developable surface S_1, with the maximum principal curvature, K_1, at point P shown.

Note that,

$$\rho_1 = \frac{1}{K_1}, \tag{3.7}$$

where ρ_1, is the principal radius of curvature. Thus the developable surface at point P curves with a radius perpendicular to the generator of ρ_1. A common simple developable surface is a cylinder; the radius of a cylinder is equal to the radius of curvature and is the reciprocal of the curvature.

3.3.2. Geometry of a Space Curve.

Consider a space curve, C, as shown in Fig 3.9 [Kreyszig 1983]. If *C* is a continuously differentiable curve, defined by the vector ***C***, then the following relations apply,

$$\mathbf{t} = \frac{d\mathbf{C}}{ds}, \tag{3.8}$$

where *s* is the arc length along *C*. The curvature is;

$$K = \left|\frac{d\mathbf{t}}{ds}\right| = \left|\frac{d^2\mathbf{C}}{ds^2}\right|. \tag{3.9}$$

Provided the curvature, *K*, is not zero;

$$\mathbf{n} = \frac{1}{K}\frac{d\mathbf{t}}{ds}, \tag{3.10}$$

where ***n*** is the unit principal normal vector of *C*. ***n*** is perpendicular to ***t*** as the derivative of a vector ($\frac{d\mathbf{t}}{ds}$) is either zero or perpendicular to the original vector.

The vector;

$$\mathbf{b} = \mathbf{t} \times \mathbf{n} \tag{3.11}$$

is called the unit binormal vector of C.

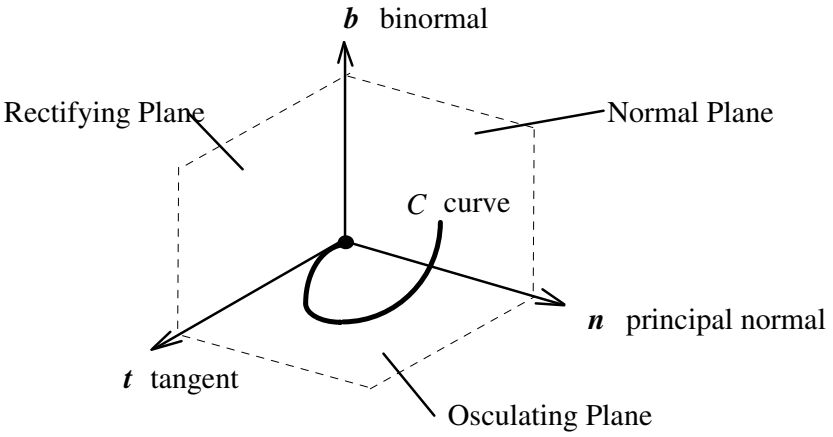


Fig 3.9 The space curve *C,* with the unit tangent, normal and binormal vector triad shown.

Since $\frac{d\mathbf{b}}{ds}$ is perpendicular to both ***b*** and ***t*** it must lie in the same direction as, and be proportional to, the normal ***n*** of the curve. Thus

$$\frac{d\mathbf{b}}{ds} = -\tau\mathbf{n}, \tag{3.12}$$

where the scalar function τ is known as the torsion of the curve *C*. The torsion is

$$\tau = -\mathbf{n} \bullet \frac{d\mathbf{b}}{ds}. \tag{3.13}$$

Using the results above the Frenet formulas can be developed:

$$\frac{d\mathbf{t}}{ds} = K\mathbf{n}, \tag{3.14}$$

$$\frac{d\mathbf{n}}{ds} = -K\mathbf{t} + \tau\mathbf{b}, \tag{3.15}$$

$$\frac{d\mathbf{b}}{ds} = -\tau\mathbf{n}. \tag{3.16}$$

3.3.3. Geometry of a Developable Surface and Curve.

A developable surface, with a curve that passes through a point *P*, as shown in Fig. 3.10, contains the vector triad ***t***, ***n***, and ***b***. ***t*** is the tangent to curve *C* at point *P*, ***n*** is the normal to curve *C* at point *P* and ***b*** the binormal to curve *C* at point *P*. The angle γ, is the angle between the tangent, ***t***, and the generator *GP*.

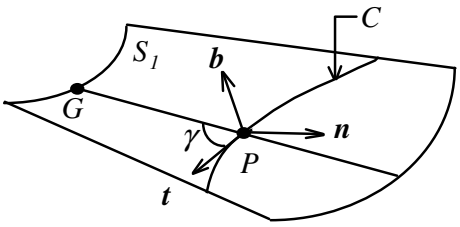


Fig. 3.10 A portion of a developable surface containing the curve *C* and the generator *GP*.

There exists a tangent plane that is co-incident with *GP* along its entire length and contains the tangent vector, ***t***, as shown in Fig. 3.11.

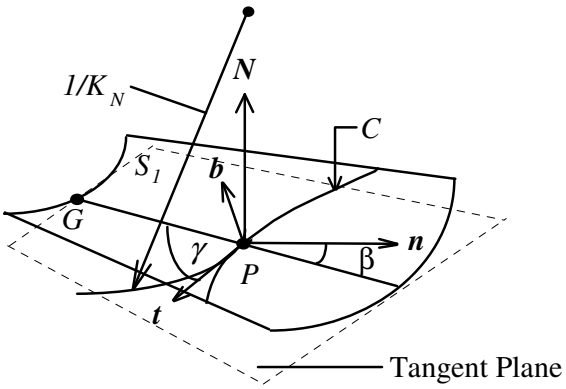


Fig. 3.11 A portion of a developable surface containing the curve *C* and the tangent plane to the generator *GP*. The curvature of the surface and the angles to the tangent and normal are shown.

The vector ***N*** is the surface normal of this tangent plane at the point P. The normal curvature of the surface in the direction of the curve C is known as the surface normal curvature, K_N. The angle β is the angle between the principal normal, ***n,*** at P and the tangent plane.

If the curve C is projected into the tangent plane it forms the curve *C′*, as shown in Fig. 3.12.

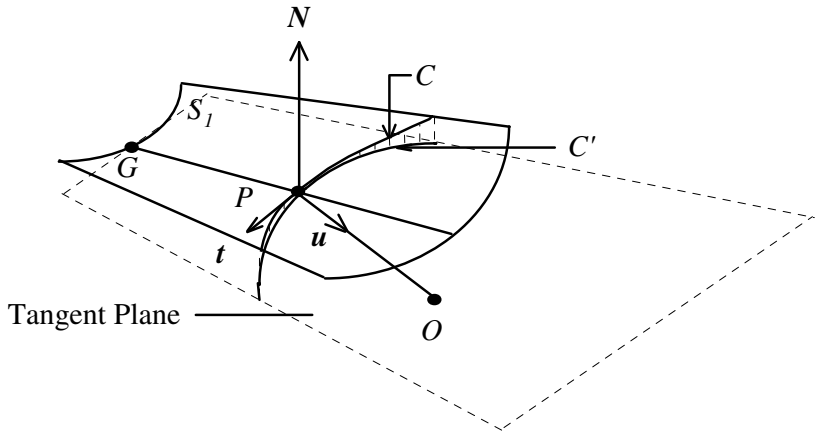


Fig. 3.12 A portion of a developable surface containing the curve *C* and *C′* the projection of *C′* into the tangent plane to the generator *GP*.

The curvature of *C′* is invariant with changes in the developable surface curvature, and is known as the geodesic curvature, K_g. The Geodesic curvature is in the direction of the vector ***u*** as shown in Fig. 3.12. From Fig. 3.12

$$K_g = \frac{1}{\overline{OP}}. \tag{3.17}$$

The geodesic torsion, τ_g, of the surface is the arc rate of rotation of the normal $\boldsymbol{N}$ along the curve C.

At a point P on the surface, there exists two orthogonal directions, in which the geodesic torsion of a surface curve is zero and the normal curvatures K_0 and K_1 attain maximum and minimum values known as principal curvatures. The curvatures may be transformed in a similar fashion to principal stresses. In the direction of the tangent to the curve C at point P, at an angle γ to the generator PG, as shown in Fig. 3.11, the properties are given by Euler's formulae,

$$\begin{aligned} K_N &= K_0 \cos^2 \gamma + K_1 \sin^2 \gamma \\ \tau_g &= \frac{1}{2}(K_0 - K_1)\sin 2\gamma \end{aligned} \quad . \tag{3.18}$$

In the developable S_1, the generator PG is a principal direction in which $K_0 = 0$. The other principal direction is perpendicular to PG along which the normal curvature is K_1.

Thus

$$K_N = K_1 \sin^2 \gamma \tag{3.19}$$

and

$$\tau_g = K_1 \cos \gamma \ \sin \gamma. \tag{3.20}$$

The curvatures are related by Meusnier's theorem, which may be expressed as

$$K_n \mathbf{n}_C = K_g \mathbf{u} + K_N \mathbf{N} \tag{3.21}$$

Where K_n is the normal curvature of the curve C at point P and K_g is the geodesic curvature. Fig. 3.13 illustrates the relation of these vectors in a plane normal to the tangen***t t.***

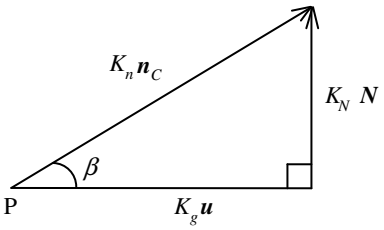


Fig. 3.13 A diagram of the relations between the curvatures shown in the normal plane at P. The tangent to the curve C, which passes through point P, points out of the page.

From Fig. 3.7

$$\cos\beta = \frac{K_g}{K_n}, \tag{3.22}$$

where β is the angle between the unit normal to the curve C, $\boldsymbol{n_C}$, and the tangent plane, as shown in Fig 3.11.

It is shown by Duncan and Duncan [1982] that using the triple scalar product $K^2\tau = [r', r'', r''']$ equation 3.19 can be expressed in terms of torsion, thus,

$$\tau = \frac{\left(K_g K_N^{'} - K_N K_g^{'}\right)}{K^2} - \tau_g \tag{3.23}$$

where primes (') denote differentiation with respect to distance s.

This may be defined as

$$\tau = -\left(\frac{d\beta}{ds} + \tau_g\right) \tag{3.24}$$

after the theorem of Bonnet [Willmore 1959].

3.3.4. Geometry of Folding.

If during the curving of surface S_1 by bending about the generators, the sheet is also folded or creased along the curve C, the sheet on the other side assumes a developable surface S_2 which is not a continuation of S_1. The curve C lies in each surface and the previous relations apply to both surfaces [Duncan and Duncan 1982].

The folded developable created by folding along the curve C is shown in Fig. 3.14.

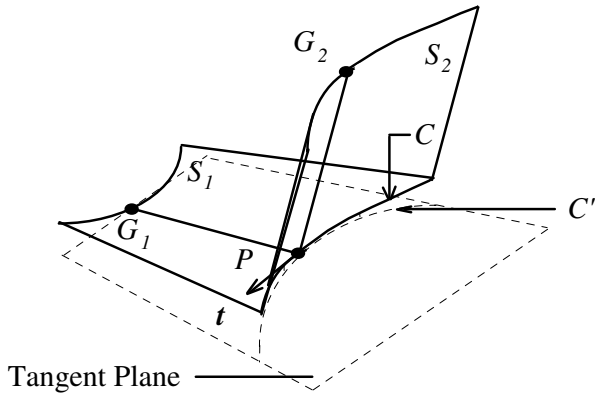


Fig. 3.14 The two developable surfaces S_1 and S_2 formed by folding along curve C.

Fig. 3.15 illustrates the curvature relations for the two surfaces.

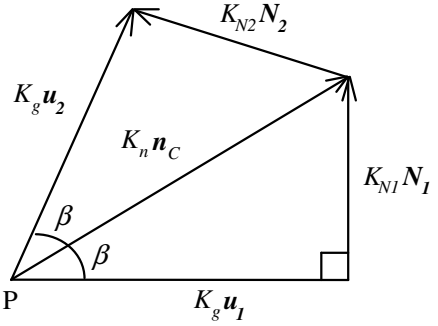


Fig 3.15 A diagram of the relations between the curvatures of the two developable surfaces S_1 and S_2. The tangent to the curve C that passes through the point P points out of the page. The subscripts *1* and *2* refer to the two developable surfaces.

As,

$$\left|K_g\mathbf{u_1}\right| = \left|K_g\mathbf{u}_2\right|, \tag{3.25}$$

the osculating plane, within which lies the unit normal $\boldsymbol{n_C}$, bisects the tangent planes for each surface. The tangent planes are defined by the unit tangent vector $\boldsymbol{t}$, and the two unit direction vectors $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ that are both at an angle β to the normal $\boldsymbol{n_C}$,.

From Fig. 3.15 it follows that

$$K_{N2} = -K_{N1} \tag{3.26}$$

The process of curved line folding produces two surfaces having equal and opposite surface normal curvatures in the direction of the fold. When viewed from one side, a curved-line fold

can only connect a concave and convex surface as in Fig. 3.14. From Fig. 3.15 we note that for this to be possible the fold line must be curved i.e. if $K_g = 0$, $\beta = \pi/2$ if $K_{N1} > 0$. This would indicate that S_2 would fold back upon S_1. Therefore, in a folded developable, $\boldsymbol{n_C}$, and $\boldsymbol{N_1}$, cannot be collinear.

Considering torsion we obtain,

$$\frac{d\beta}{ds} = -\frac{\left(K_g K_N^{'} - K_N K_g^{'}\right)}{K^2}, \tag{3.27}$$

and since, from equation 3.26, $K_{N1} = -K_{N2}$, $K'_{N1} = K'_{N2}$ we also obtain

$$\frac{d\beta_1}{ds} = -\frac{d\beta_2}{ds}. \tag{3.28}$$

Thus the geodesic torsion of the surface S_2 in the direction of the fold is, from equation (3.22), given as;

$$\tau_{g2} = \tau_{g1} + 2\frac{d\beta}{ds}. \tag{3.29}$$

The surface S_2 is completely defined by the surface S_1 and the curve C. In some special cases the generators of the surface S_2 can be determined from these relations.

However, the generators of the second surface, created when a general developable surface is folded along a general space curve, cannot be determined. The following section details a computational method to find the generators of surface S_2, for any general developable and general folding curve. This has not previously been developed.

3.4. A COMPUTATIONAL METHOD FOR THE GENERATION OF DEVELOPABLES FOLDED ALONG AN ARBITRARY CURVE.

3.4.1. Introduction.

Section 3.4 describes a new method for computationally modelling folded developables. The theories behind the method, the details of the numerical techniques and a description of a computer program that implements the method are presented. Results obtained from the program are also discussed.

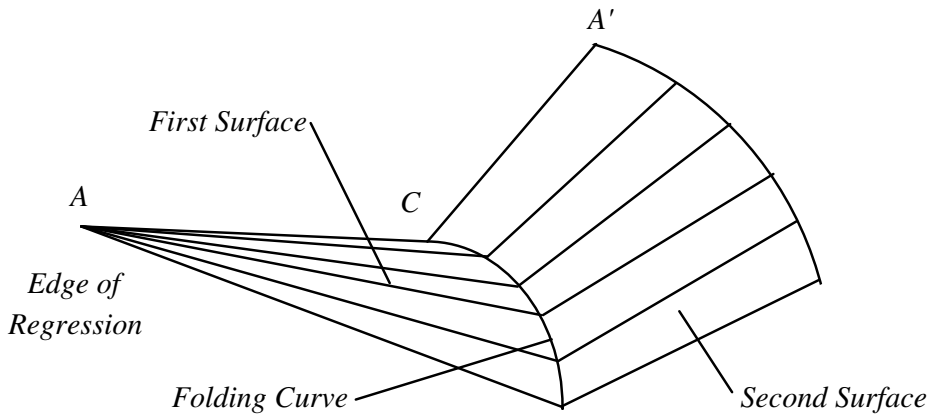
Previous design of folded developables was by trial and error. While the theorems governing curved folded developables have existed for some time [Duncan and Duncan 1982], the mathematics to predict the shape of a general folded developable have not previously been given. Prior to this work the only solutions for folded developables were two special cases; a right circular cone and a cylinder and cone [also Duncan and Duncan 1982]. The mathematics to predict the result of folding a developable surface along a curved line has been determined in this work and forms the basis of a computer package, 3FD, that allows rapid and systematic graphical design of folded developables.

A folded developable is composed of two developable surfaces joined by a common folding curve, and to create it, three major steps must be followed. The first surface must be defined, the folding curve must be defined and from this the second surface must be constructed.

The first surface is defined by the edge of regression and the curvature (or radius of curvature) of the surface. This surface may then be unwrapped onto a flat plane to form the development of the surface. The folding curve is most easily defined as a curve on that development.

Two simple folded developables are illustrated in Fig. 3.16.

(a)



(b)

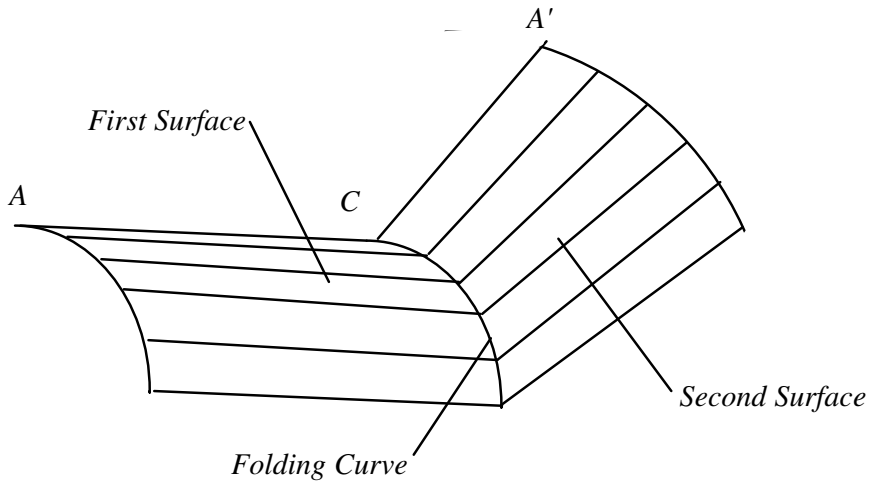


Fig 3.16 (a) Conic Developable. The 'first' surface is the surface between the edge of regression and the folding curve. (b) Cylindrical Developable. The edge of regression for a cylindrical surface is an infinite distance to the left.

3.4.2. Definition of the First Surface.

In this work it is convenient to define the first surface in terms of its development and the curvature about the generators.

3.4.2.1. Development of the First Surface.

The first surface can be physically formed by cutting the development from a sheet of material and bending it about the generators. As the plane development is curved about its generators, it passes through a family of developable surfaces and the edge of regression becomes increasingly twisted. In Fig. 3.17 a simple developable surface, with the generators limited by a folding curve is shown with its development, which has been unwrapped onto the plane.

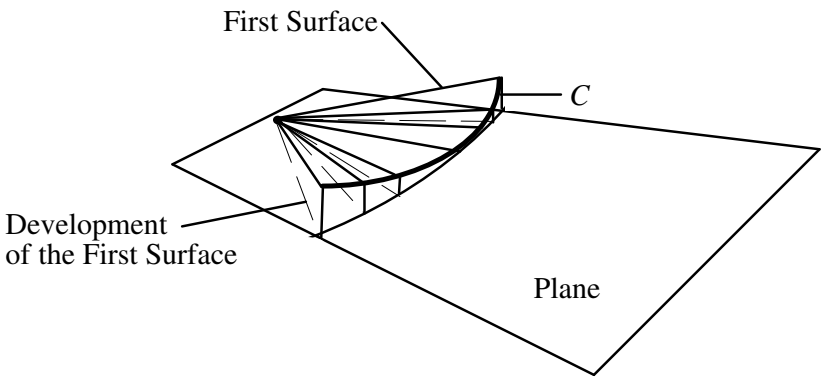


Fig 3.17 The development of the first developable surface. The first surface is bounded by the folding curve *C*.

3.4.2.2. The Edge of Regression.

The edge of regression, which contains the starting points of the surface generators, can be defined in the development in several different ways. In this work there are three classifications;

(i) edge of regression is a point - a cone.

(ii) edge of regression a point at infinity - a cylinder.

(iii) edge of regression is a mathematical function - the general case.

Generators of the surface initially lie in the plane of the edge of regression and are tangential with the edge of regression. Generators are projected either from points that are separated by regular intervals on the edge of regression or at regular angular intervals from the edge of regression. For practical purposes the generators are limited in length, but mathematically they may extend infinitely on one side of the edge of regression. Examples of edges of regression are shown in Fig. 3.18.

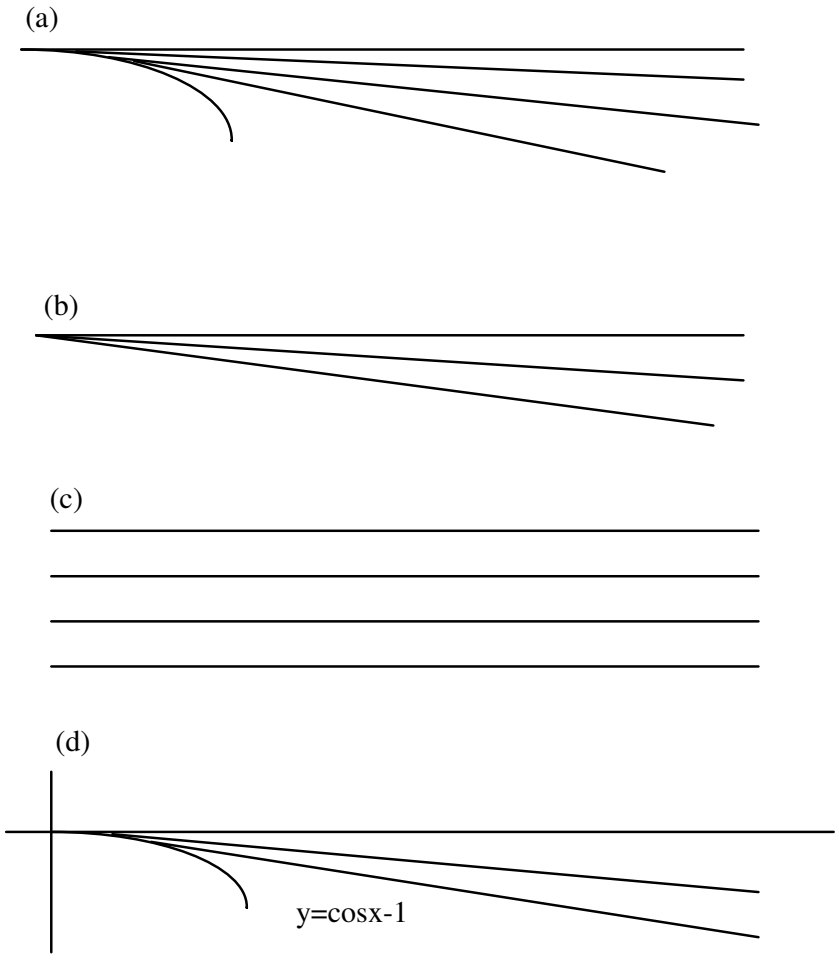


Fig. 3.18 Examples of Edges of Regression.
(a) General Curve
(b) Point of Regression i.e. a cone.
(c) Edge of Regression at Infinity i.e. a cylinder
(d) Mathematical Function, (y=cos$x$-1).

3.4.2.3. Specifying the Curvature.

If the development of the edge of regression of a developable surface is given by some function in a flat plane, the curved developable surface is formed by torsion, or 'twist' of the edge of regression [Willmore 1959]. This torsion is related to principal curvature K of the surface (in a direction normal to the generator). As stated in Section 3.3.4 the principal curvature parallel to the generator is zero. Therefore the curvature of a folded developable element is defined by one principal curvature. It is a property of a developable that the curvature varies inversely with distance, l, along the generator [Willmore 1959], i.e.

$$\frac{l_D}{l_B} = \frac{K_B}{K_D} \tag{3.28}$$

where lengths and curvatures are defined by Fig. 3.19. Hence, by specifying the curvature K_B at some point B as shown in Fig. 3.19, the surface curvature is specified. Curvature at point B may be specified by several methods. The simplest methods involve the specification of the radius of curvature (the inverse of curvature) at that point. The specification of curvature is detailed in Section 3.4.3.4

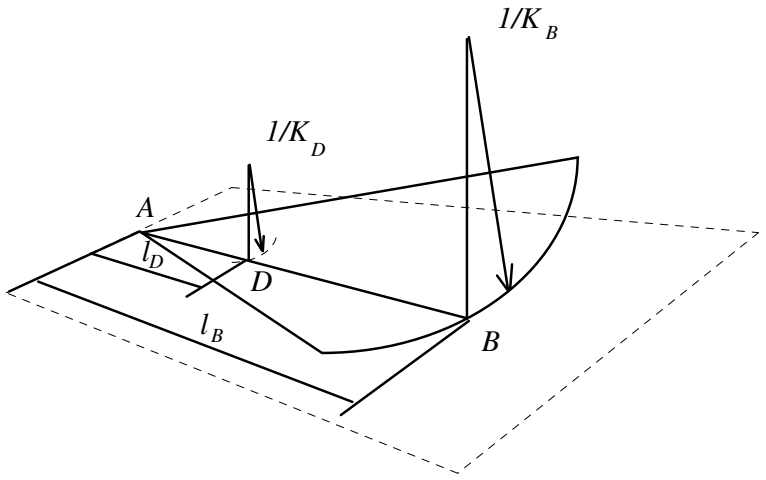


Fig 3.19 The relation between curvature and generator length.

3.4.3. Specifying the Folding Curve and the Curvature at the Intersection of the Folding Curve and the Generators.

To construct the second developable surface the intersections between the folding curve and the first surface generators must be found. Because the folding curve is most easily defined in the development, the curvature of the first surface at the points of intersection with the folding curve must be determined.

3.4.3.1. The Folding Curve.

In the development, the folding curve may be defined either by a sequence of discrete points or by a simple mathematical function such as a circular arc or bezier curve. Several restrictions apply

(i) the curve cannot be straight for any length.

(ii) the curve may not fold back on itself.

These conditions are consequential to the relationship

$$\cos\beta = \frac{K_g}{K_n}, \tag{3.29}$$

that was developed as equation (3.22) in section 3.3.

3.4.3.2. Determining the Intersection between the Plane Folding Curve and the Generators.

The intersection of the generators with the folding curve can be found by several methods. For a piecewise curve or spline the intersection may be found by interpolation as shown in Fig 3.20 below.

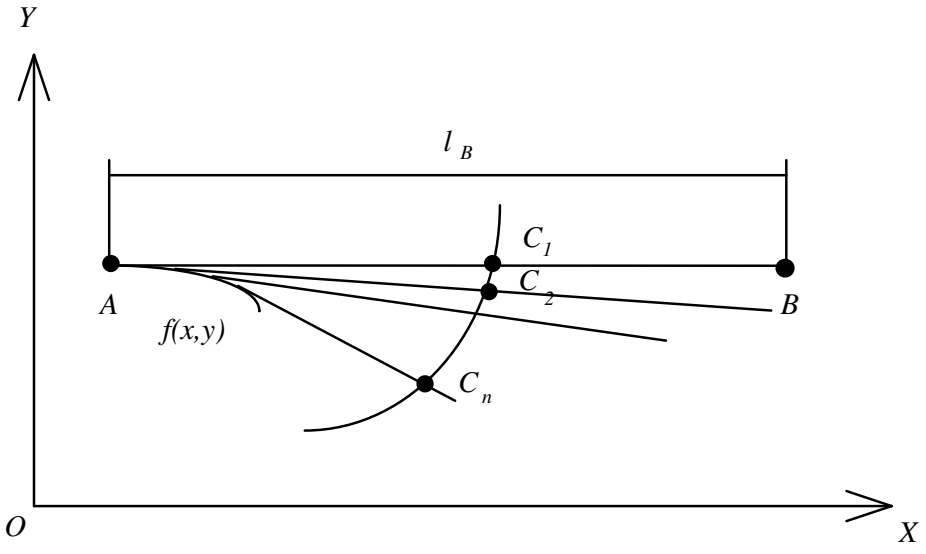


Fig. 3.20 Plane development of a general developable. The generators intersect a folding curve at points C_1 to C_n.

However for a mathematically defined curve such as the arc of a circle an exact intersection may be determined. This is demonstrated below.

Consider a generator defined by the points *A*, on the edge of regression, and *B* some length *l* away from *A*, that intersects a folding curve that is the arc of a circle. The centre of the circle is a distance x_c from an arbitrary *X* axis and a distance y_c from an arbitrary *Y* axis. The radius of the arc is *R*. This is shown in Fig 3.21.

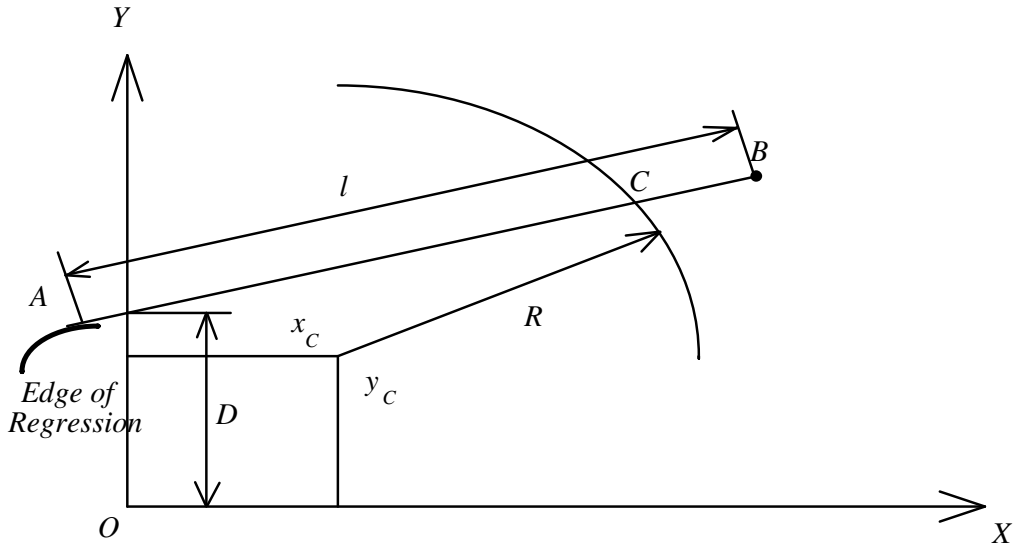


Fig. 3.21 Intersection between a generator and a circular arc.

In the X,Y co-ordinate system defined, a line co-incident with the generator is described by

$$y = mx + D \tag{3.30}$$

where *m* is the gradient of the generator and *D* is a constant determined by the edge of regression's offset from the co-ordinate axis. *D* may be determined by finding its intersection of the line with the y axis. The circular arc can be defined by the equation

$$(x - x_C)^2 + (y - y_C)^2 = R^2 \tag{3.31}$$

where *R* is the radius and x_c and y_c the offsets from the origin of the centre. Equating equations 3.30 and 3.31 will give the point of intersection.

Equation 3.31 can be expanded to

$$y^2 - 2yy_C + y_C^2 = R^2 - x^2 + 2xx_C - x_C^2$$

$$y^2 - 2yy_C = R^2 - x^2 + 2xx_C - x_C^2 - y_C^2 \tag{3.32}$$

Substituting this into equation (3.30) produces,

$$(mx+D)^2 - 2y_C(mx+D) = R^2 - x^2 + 2xx_C - x_C^2 - y_C^2,$$

or

$$x^2\left(m^2+1\right) + x\left(2Dm - 2my_C - 2x_C\right) + \left(D^2 - R^2 + x_C^2 + y_C^2 + 2y_C D\right) = 0 \quad (3.33)$$

Since equation 3.33 is a quadratic equation the solution for x is simply,

$$x = \frac{-\left(Dm - my_C - x_C\right) \pm \sqrt{\left(Dm - my_C - x_C\right)^2 - \left(m^2+1\right)\left(D^2 - R^2 + x_C^2 + y_C^2 + 2y_C D\right)}}{m^2+1} \quad (3.34)$$

Only the positive solution is valid as the generators extend from the edge of regression in the positive direction.

3.4.3.3. Finding the Position of the Folding Curve in the First Surface.

When the intersection of the folding curve with the generators in the development of the first surface has been found, the folding curve in the curved first surface may be easily determined. The length *AC*, from the edge of regression to the folding curve along a generator, is the same in both the development (two dimensional) and the curved developable surface (three dimensional). Thus the position of *C* in the curved surface can be determined by moving along the generator *AB* in the curved surface a distance of *AC*.

3.4.3.4. Finding the Curvature of the First Surface at the Folding Curve.

Using the property of developables described in Section 3.2.2 that curvature is linearly related to length along a generator, the curvature at the folding curve intersection point *C*, can be determined from the known curvature at point *B* using the relation,

$$K_C = \frac{\overline{AB}}{\overline{AC}} K_B. \quad (3.35)$$

3.4.4. Construction of the Generators of the Second Surface.

The direction of a generator in the second surface can be specified by determining,

1. The fold angle (α),

2. The angle (γ_2) the generator makes with the tangent of the folding curve.

These angles are shown in Figs 3.22 and Fig. 3.23 below

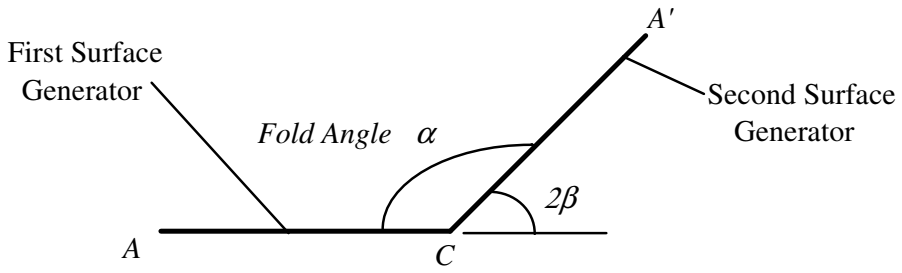


Fig 3.22 Part of a simple folded developable showing the fold angle between the generators in the normal plane. The tangent to the folding curve points out of the page.

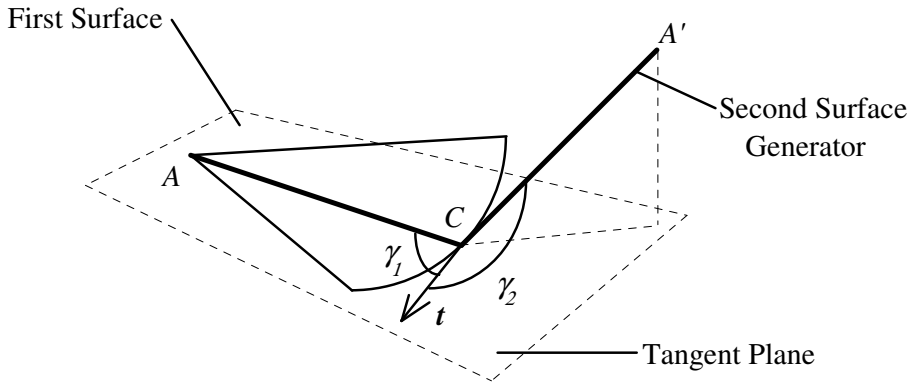


Fig 3.23 Part of a simple folded developable with the two angles between the tangent vector and the generators in the tangent plane shown.

3.4.4.1. The Fold Angle.

The fold angle is the angle between the generators and is $\pi - 2\beta$. To determine the fold angle the angle β must be found.

From Section 3.3,

$$\cos\beta = \frac{K_g}{K_n} \tag{3.29}$$

The above equation uses the first curvature of the fold curve and the geodesic curvature of the fold curve to determine β. It is more useful, to express β in terms of the principal normal curvature of the first surface, K_{N1} or K_N for ease of notation, and the geodesic curvature of the folding curve, K_g,

$$\tan\beta = \frac{K_N}{K_g} \tag{3.36}$$

The implications of this result on the kinematics of folded developables are discussed later in Chapter 5.

To find β at the intersection points between the generators of the first surface and the fold curve the local values of the surface's normal and geodesic curvatures must be found. For a known first developable surface the first principal curvature, K_1, at the point C can be determined as in Section 3.4.3.4. Also known from the geometry of the folding curve, is the geodesic curvature K_g of the folding curve, as shown in Fig. 3.24. The geodesic curvature of the folding curve is the same for both the first developable surface and for the second developable surface created by folding about the folding curve, as proven in Section 3.3 equation 3.17.

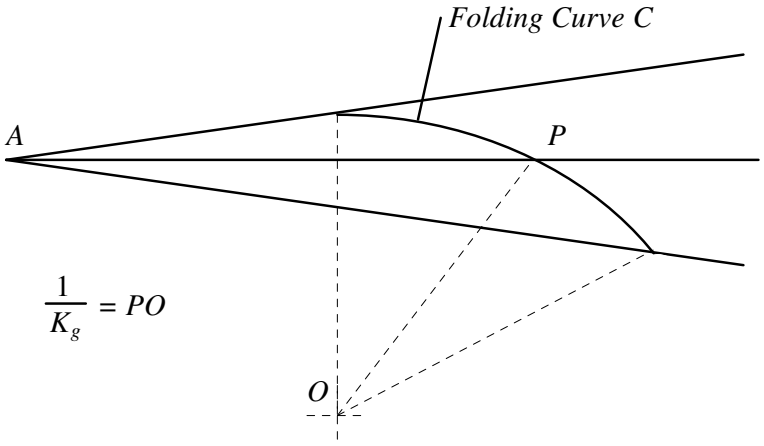


Fig. 3.24 Part of the development of a developable showing the geodesic curvature at the point P of the folding curve.

3.4.4.2. The Angle of the Second Generator to the Folding Curve Tangent.

The angle, γ_2, between the second generator and the tangent vector of the folding curve can be determined algebraically as detailed below, using the results of Section 3.3.

Given

$$K_N = K_1 \sin^2 \gamma, \tag{3.37}$$

$$\tau_{g1} = K_1 \cos\gamma_1 \sin\gamma_1, \tag{3.38}$$

$$\tau_{g2} = \tau_{g1} + 2\frac{d\beta}{ds}, \tag{3.39}$$

and

$$K_{N2} = -K_{N1}, \tag{3.40}$$

substituting (3.26) into (3.27) leads to

$$K_2 \cos\gamma_2 \sin\gamma_2 = K_1 \cos\gamma_1 \sin\gamma_1 + 2\frac{d\beta}{ds}. \tag{3.41}$$

Using (3.38), (3.39) and (3.41)

$$K_2 \sin^2 \gamma_2 = -K_1 \sin^2 \gamma_1$$

or

$$K_2 = \frac{-K_1 \sin^2 \gamma_1}{\sin^2 \gamma_2} \tag{3.42}$$

Substituting (3.41) into (3.42) produces

$$\frac{-K_1 \sin^2 \gamma_1 \cos\gamma_2}{\sin\gamma_2} = K_1 \cos\gamma_1 \sin\gamma_1 + 2\frac{d\beta}{ds} \tag{3.43}$$

which can be rearranged,

$$\frac{-K_1 \sin^2 \gamma_1}{\tan\gamma_2} = K_1 \cos\gamma_1 \sin\gamma_1 + 2\frac{d\beta}{ds} \quad ,$$

or

$$\tan\gamma_2 = \frac{-K_1 \sin^2 \gamma_1}{K_1 \cos\gamma_1 \sin\gamma_1 + 2\frac{d\beta}{ds}}. \tag{3.45}$$

$\frac{d\beta}{ds}$ is a measure of the twist of the folding curve *C*. It is determined by the rate of change of the fold angle as it moves along the folding curve as

$$\frac{d\beta}{ds} = -\frac{d\alpha}{ds}. \tag{3.46}$$

Equation 3.45 can be used to determine the second tangent angle, γ_2. This angle together with the fold angle, α, fully defines the direction of the generator in the second developable.

This key new result means that the second surface of any general folded developable can be constructed from any developable first surface and a general folding curve.

In general however it is not possible to know the local values of the first tangent angle, γ_1 and the rate of angular change of the osculating plane $\frac{d\beta}{ds}$. In the general case these must be determined by a numerical method detailed in the next section??

3.4.5. A Numerical Method for Determining the Tangent Angles and Rate of Change of Fold Angle and Finding the Image of the First Generator.

Because in the general case it is not possible to know the local values of γ_1 and $\frac{d\beta}{ds}$, they must be determined by a numerical method. The relevant approximations are described below??

3.4.5.1. Generator Tangent Angle.

The desire to create a numerical method to find the tangent angle requires an elemental approach. An element of a general developable surface, shown in Fig. 3.25 below is now considered. To determine the angle γ_2, the angle that the existing developable surface generator makes with the tangent vector of the folding curve, γ_1, must be found. If $AC >> C_{i-1}C_{i+1}$ the tangent, ***t***??can be assumed to be parallel to the chord $C_{i-1}C_{i+1}$ and passes through the point C_i, as shown in Fig. 3.25. γ_1 can then be easily determined and is given by,

$$\gamma_1 = \arctan\left(\frac{y_{C_i} - y_{A_i}}{x_{C_i} - x_{A_i}}\right) + \pi - \arctan\left(\frac{y_{C_{i+1}} - y_{C_{i-1}}}{x_{C_{i+1}} - x_{C_{i-1}}}\right) \tag{3.47}$$

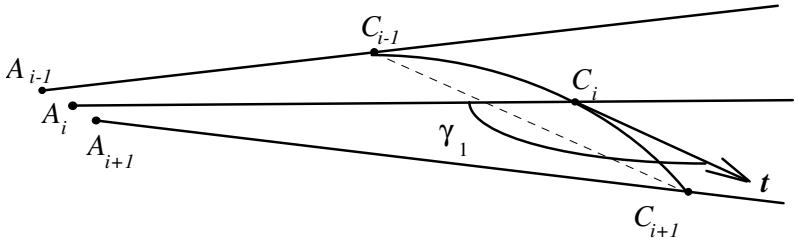


Fig 3.25 The tangent angle γ_1, the angle between the tangent vector ***t***, and the generator A_iC_i.

3.4.5.2. Rate of Change of the Fold Angle.

To determine $\frac{d\beta}{ds}$, since $AC >> C_{i-1}C_{i+1}$ and if the rate of change of β is also small then the following approximation may be used;

$$\frac{d\beta}{ds} = \frac{\beta_{C_{i-1}} - \beta_{C_{i+1}}}{C_{i-1}C_{i+1}}. \tag{3.48}$$

By making these two numerical approximations it is possible to determine β and γ_1. γ_2 is calculated from equation 3.45 while β can be found using for each generator

$$\beta_i = \arctan\left(\frac{K_{Ni}}{K_{gi}}\right) \tag{3.49}$$

With these angles known the fold angle, α, and γ_2 can be calculated and the direction of the generator in the second surface is known.

By repeating this process, as shown by the flow diagram in Fig. 3.26, for all generators, surface two can be constructed.

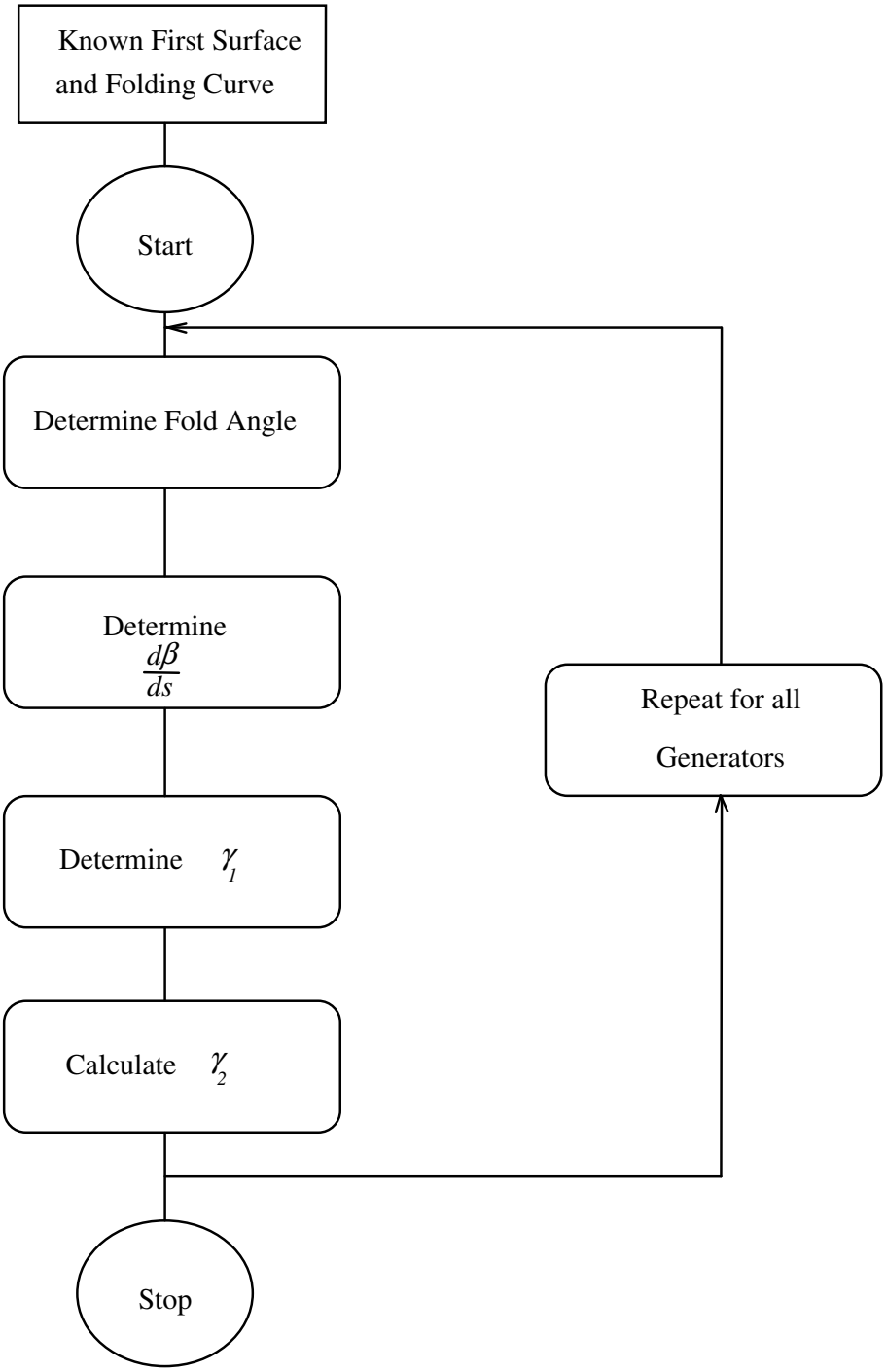


Fig. 3.26 Flow Diagram for the construction of the second surface.

3.4.5.3. Creating the Second Surface Generators.

It is useful, particularly for graphical display, to know the generator of the second surface that connects to a generator in the first surface at the folding curve. It is also useful to limit the generator's length, so it fits in the desired display space.

This second generator is found by conserving length. Since the fold angle and angle to the tangent of the folding curve, relative to the first generator, are known for the second generator all that is required is to limit its length. This is done simply by measuring along the second generator the same distance as the length of the first (AC) to a point A', the end of the second generator. This is shown in Fig 3.27 below

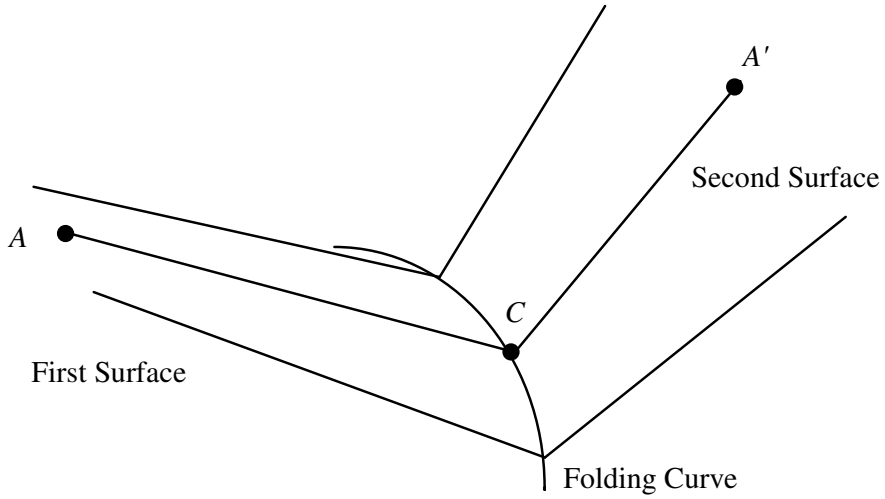


Fig 3.27 The second generator CA' in the second developable surface that connects to the generator in the first surface AC.

3.4.5.4. The Plane Development of the Complete Developable.

Once the folded developable has been defined, its plane development may be calculated. Since the development of the initial surface is known, the development of the second surface needs to be found.

For simple folded developables such as those formed from two conic surfaces, the plane development is easily found. The development is simply a sector of a circle with a radius equal to the generator total length of both surfaces.

For more complex shapes the plane development can be found by two methods. Since all generator lengths and angles are known the blank shape can be produced directly or it can be approximated by assuming that the area between two generators is constant, whether the generators are plane or part of a curved surface. By definition a developable surface cannot undergo deformation other than simple bending. Since simple bending does not result in any area change, the constant area assumption is valid.

The development of the second surface can be determined using the constant area transformation blank shape prediction method described in Chapter 2.

3.4.6. Computer Program Functionality.

The program 3FD uses four steps to create a folded developable.

1. The development of the first developable surface must be defined.

2. The surface curvature must be specified.

3. The folding curve must be specified.

4. The second developable surface must be constructed.

The program also allows the user to view the created developable from any angle, to produce a colour shaded image and to plot both the three dimensional image and the flat development.

To illustrate this process, the steps required to produce the simple cone developable shown in Fig. 3.28 below will be detailed.

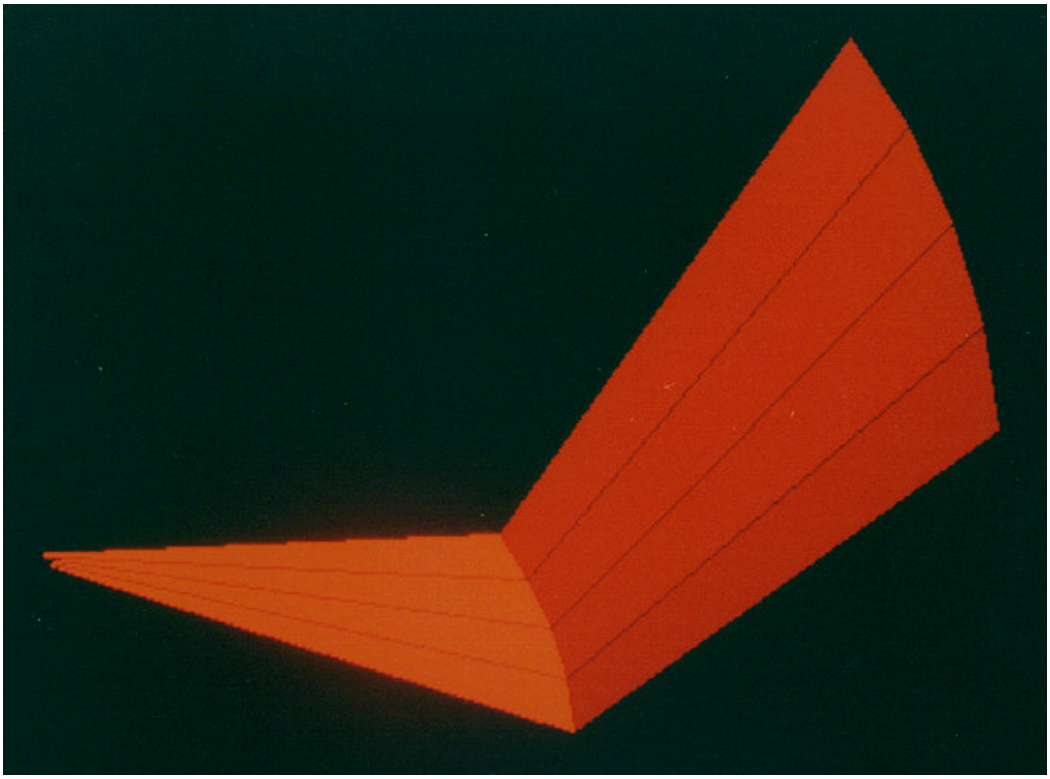


Fig 3.28 Simple Cone Developable.

3.4.6.1. Definition of the Development of the First Surface.

A developable surface is defined by a series of generators and related curvatures, as described in Section 3.4.2.1. The program has four options for the edge of regression:

(i) **Cone.** The edge of regression is a point. The number of generators must be specified and these are calculated as radii at 5° intervals. The resulting plane surface is a segment of a circle.

(ii) **Cylinder.** The edge of regression is a point at infinity. The number of generators must be specified and these are horizontal lines at regular intervals. The resulting plane surface is a rectangle.

(iii) **Function.** The edge of regression is a predefined mathematical function. A simple function is y = cos x-1 used in the program. The number of generators must be specified and these are tangent to the edge of regression at regular intervals.

(iv) **Bezier.** The edge of regression is a Bezier curve. A Bezier spline is a mathematical function that can be described by four points. Two points form the ends of the spline and the other two indicate tangent directions at the ends. This permits a general curve to be simply described. The generators are tangent to the Bezier curve (edge of regression) at regular intervals.

In the case of the cone developable the definition of the plane elements is a simple procedure. The program user selects *CONE* from the *E.O.R.* (Edge of Regression) menu, then specifies the number of generators, the generators are spaced at five degree intervals. The following development, shown in Fig. 3.29, is displayed. In this case the number of generators chosen is five.

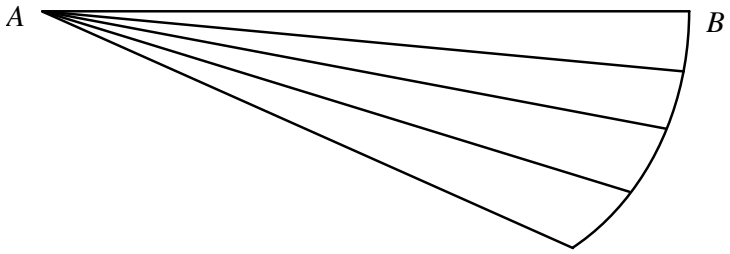


Fig. 3.29 A Simple Cone Developable's Generators.

The program also creates two imaginary generators that are either side of the developable surface, as shown in Fig. 3.30.

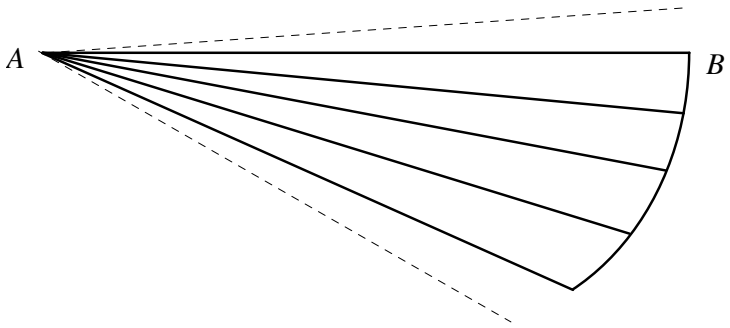


Fig. 3.30 Imaginary Generators, shown as dashed lines, which are used during calculations.

These are used as aids to calculate the positions of the generators after folding.

3.4.6.2. Curvature Specification.

With the edge of regression and generators defined in the plane, curvature can be added to the surface. The program has four options for specifying this curvature.

(i) **Cone Fixed.** A cone can be described by the length of the generators and the radius of its base. The radius of curvatuue and base radius are related by equation (3.49) following. There is no variation of curvature at points on different generators that are the same length. This is shown in Fig. 3.31.

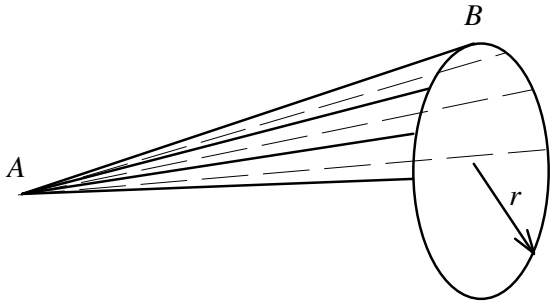


Fig. 3.28 Cone curvature definition.

(ii) **Normal Fixed.** The radius of curvature (the reciprocal of curvature), is defined as being normal to the generator. The normal radius of curvature of a developable surface, is the radius of curvature perpendicular to the generators.

The relationship between the cone base radius and the normal radius of curvature is geometric and given by

$$|r_{Cone}| = \sqrt{\frac{|r_{Normal}|(|r_{Normal}|+1)^2}{|r_{Normal}|+1}} \tag{3.49}$$

Curvature depends only on the distance along generators from the apex of the cone. This is shown in Fig. 3.32.

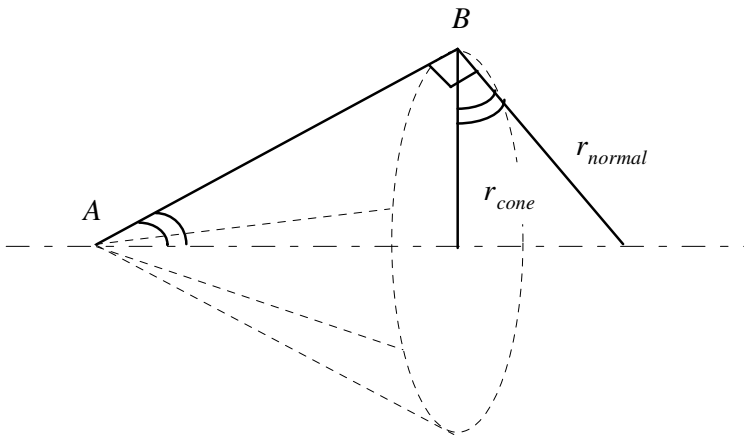


Fig. 3.32 The definition of curvature normal to the generator.

(iii) **Bezier.** A bezier curve can be created in a plane perpendicular to, and at the end of, the first generator. The generators are then 'draped' over the bezier curve. The end of the developable's surface then has similar curvature to the bezier curve, as shown in Fig. 3.33. The radius of curvature, and hence the curvature, of the developable surface is found by intersecting the perpendicular bisectors of adjacent generators.

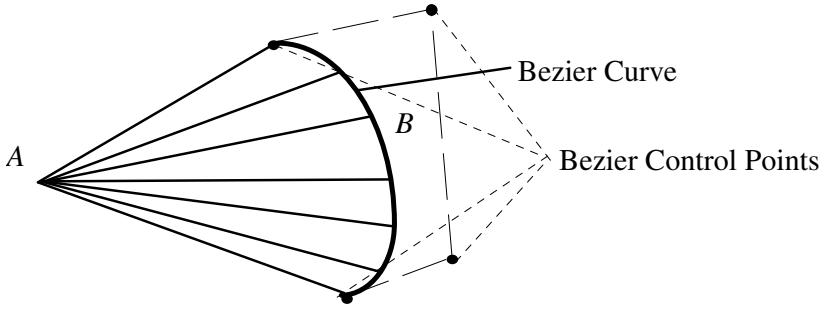


Fig. 3.33 Bezier curvature definition.

In the simple cone case the curvature is simply specified by selecting *NORMAL* from the *RAD of CURV* (Radius of Curvature) menu. The user is then requested to enter the normal radius of curvature for the first developable surface. If 0.5 is entered the result, shown in Fig. 3.34, is displayed.

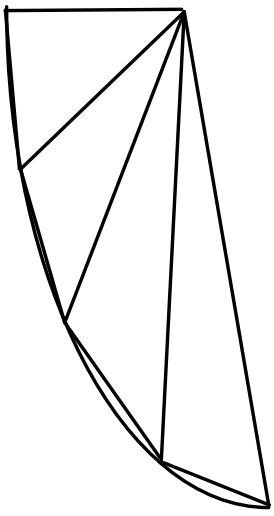


Fig. 3.34 End view of a simple Cone Developable, after surface curving.

3.4.6.3. Folding Curve Specification and Folding.

The initial surface is now fully specified. To determine the second surface, the folding curve must be defined. Following folding curve definition the generators of the first developable surface are folded about the folding curve using the method described in Section 3.4.1. Each generator is considered as the mid-line of a triangular element formed by the intersections of its two adjacent generators and the fold curve, as shown in Fig. 3.35.

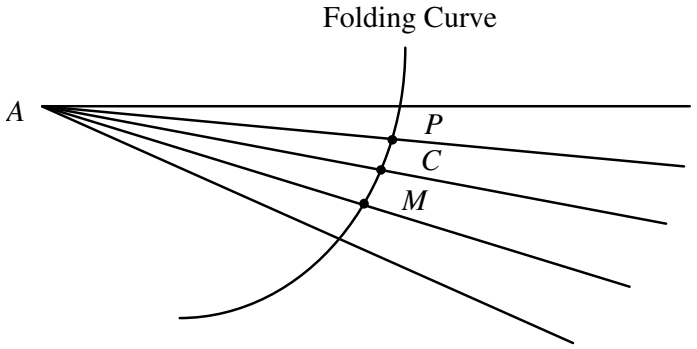


Fig 3.35 The triangular element *APM*, that includes the generator *AC*, that is used to calculate the position of the generator *CA′* in the second surface.

The program has four options for fold curve definition.

(i) **Axis Arc.** The fold curve is the arc of a circle. The centre of the circle lies on the horizontal line made by the first generator. The centre of the arc is the point, A, of the first generator on the edge of regression. The intersection of the folding curve and the plane developable is calculated as in Section 3.4.3.3, shown below in Fig. 3.36. The new lengths of each generator are calculated and the lengths of the generators in the 3D developable surface are reduced to match them. The second surface of the folded developable is then calculated following the procedures described previously.

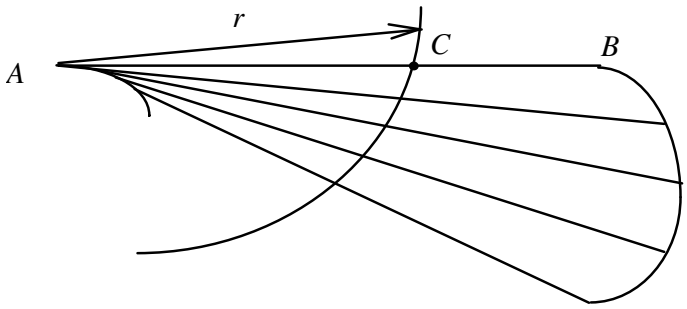


Fig. 3.36 Axis arc folding curve.

(ii) **Centre Arc.** The fold curve is the arc of a circle. The centre of the circle and its radius must be specified. The intersection of the folding curve and the plane developable is calculated as in Section 3.4.3.3, as shown in Fig. 3.37. The new lengths of each generator are calculated and the lengths of the generators in the 3D developable surface are reduced to match them. The second surface of the folded developable is then calculated following the procedures described previously.

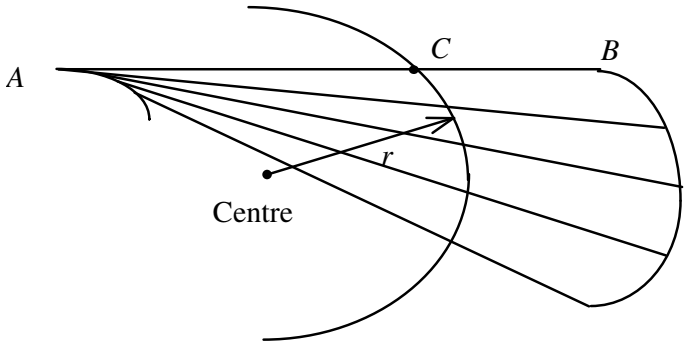


Fig. 3.37 Centre arc folding curve. Centre point and the radius of the fold curve must be defined.

(iii) **Bezier.** The fold curve is a Bezier curve. The curve may be anywhere on the plane development provided it crosses all the generators. The intersection of the folding curve and the plane developable is calculated as in Section 3.4.3.3, as shown in Fig. 3.38 below. The new lengths of each generator are calculated and the lengths of the generators in the 3D developable surface are reduced to match them. The second surface of the folded developable is then calculated following the procedures described previously.

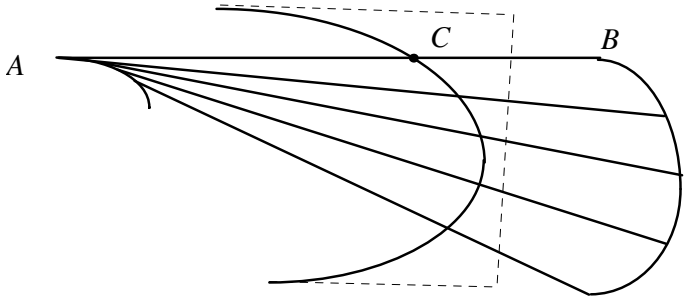


Fig. 3.38 Bezier folding curve. The four bezier control points must be specified.

In the case of the simple cone developable the fold curve is the arc of a circle with its centre at the vertex of the cone and having a radius of 0.5. After folding the developable can be displayed as a three dimensional shaded image as shown in Fig 3.39 below.

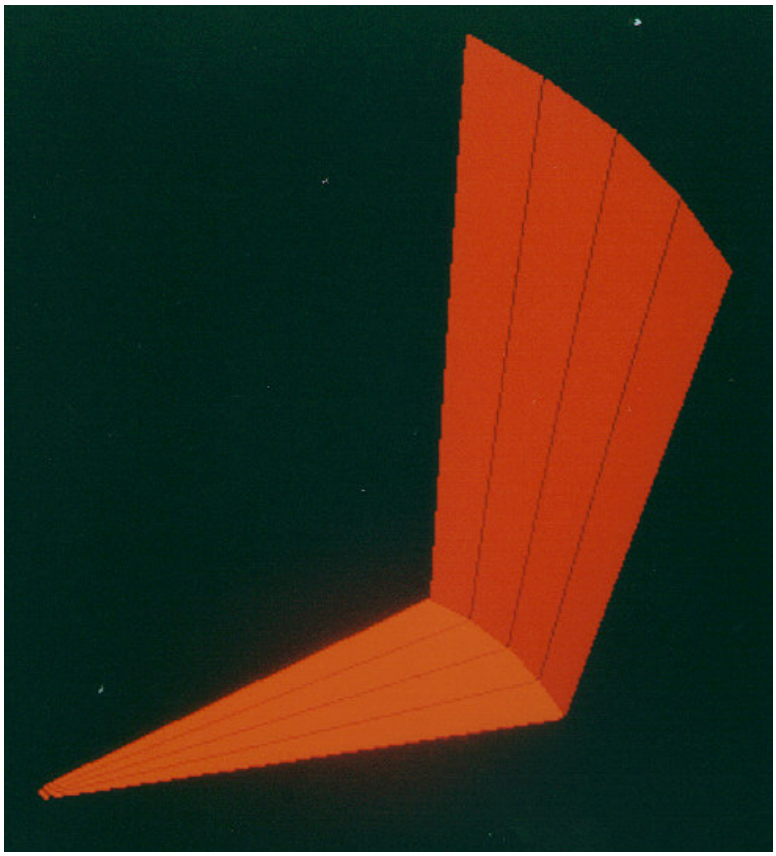


Fig 3.39 Shaded Image of the Folded Developable.

3.4.6.4. Visual Information and Output.

The program that has been developed permits viewing of the created developable from any angle. Four fixed views (Right Elevation, Left Elevation, Main Elevation and Plan) are set plus a three dimensional view that can be fully rotated. The default shows the wire frame structure of the folded developables but any of the views can be colour shaded.

The program can produce three dimensional plots of the folded developable as well as two dimensional plots of the development. Examples of these will be given in the following sections.

3.4.7. Results.

The results of the 3FD program may be examined by comparing folded developables generated by the program with similar physical models.

In each case, the basic shape of the folded developable will be compared with the physical model. The predicted fold angle and the angles predicted for the generators relative to the folding curve will also be compared.

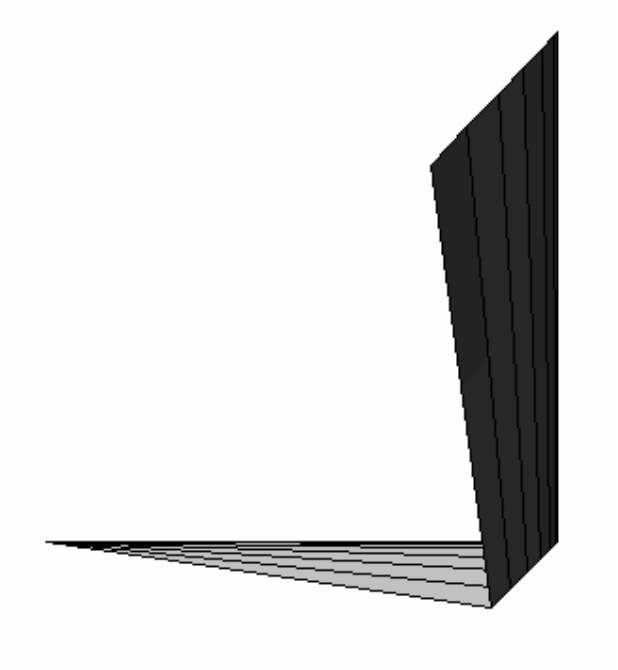
The physical models are made from thin cardboard sheet. The development is drawn on the sheet and it is cut out then folded. The surfaces are folded about the generators until the radius of curvature of the first surface at the fold curve, ρ_N matches that desired. The radius of curvature match is assured by using cardboard templates that match the desired curvature. These are glued to the underside of the first surface to fix curvature.

3.4.7.1. Simple Cone Developable.

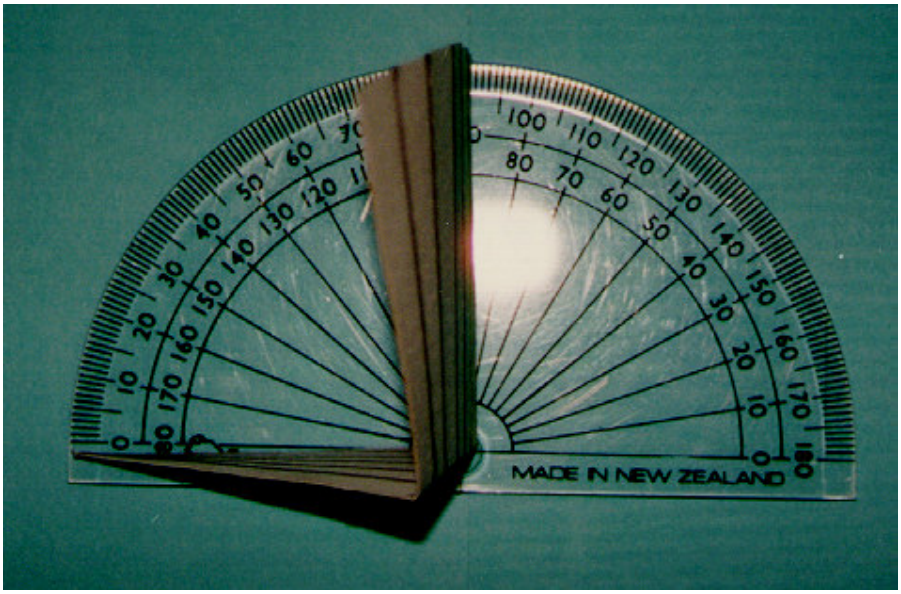
This simple cone folded developable was used as an example of the operation of the 3FD program in the previous section. It is represented by seven equally spaced generators.

The radius of curvature of the developable surface prior to the folding curve is equal for all generators at equal distances from the edge of regression (the cone point). The radius of curvature increases in proportion to the distance from the point of the cone. If the generators are of unit length the principal radius of curvature, normal to the surface, ρ_N, at the folding curve is 0.5.

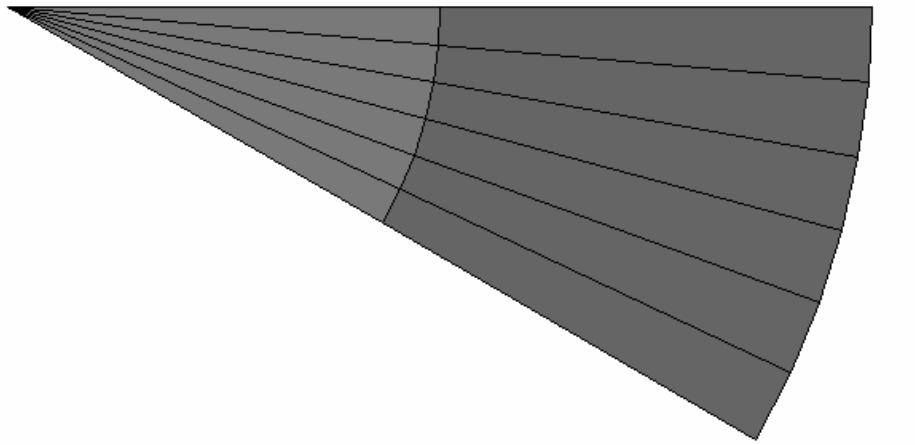
The plane projection of the folding curve is the arc of a circle with its centre at the vertex of the cone. The radius of this arc, which is also the geodesic radius of curvature of the folding curve, ρ_g, is 0.5.



(a)



(b)



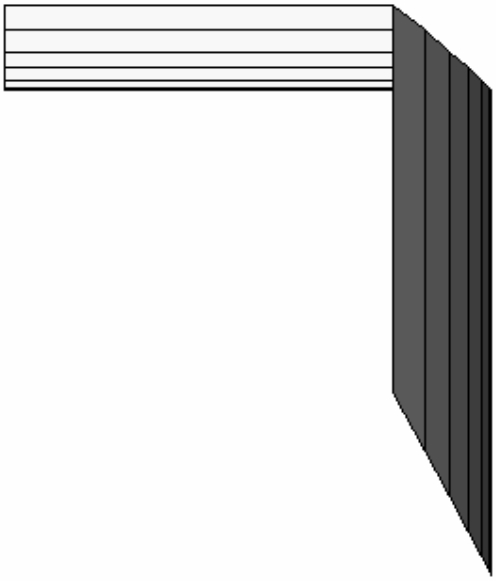
(c)

Fig. 3.40 Simple Cone Developable (a) 3FD Prediction, (b) Physical Model and (c) the 3FD predicted development.

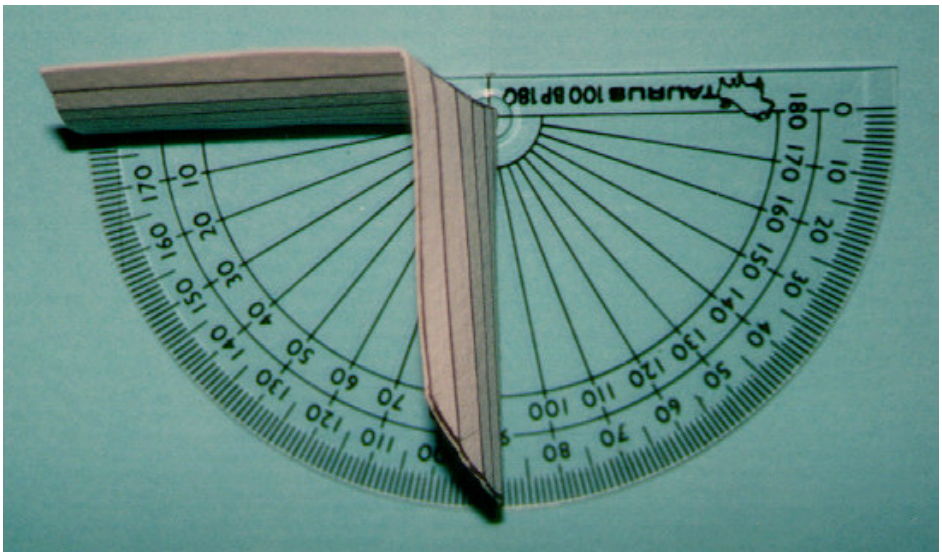
As shown in Fig. 3.40(a) and Fig. 3.40(b) the shape of the two models, computer generated (a) and physical (b) is very similar. The predicted fold angle is 90° and the measured angle is equal. The angles of the generators relative to the tangent of the folding curve are also identical.

3.4.7.2. Simple Cylindrical Developable.

A simple cylindrical surface represented by seven equally spaced generators has a constant radius of curvature ρ_N of 0.5 if the generator distance to the folding curve is one unit in length. The folding curve in the plane development is the arc of a circle of radius, ρ_g, of 0.5 with a centre at the start of the first generator.



(a)



(b)

(c)

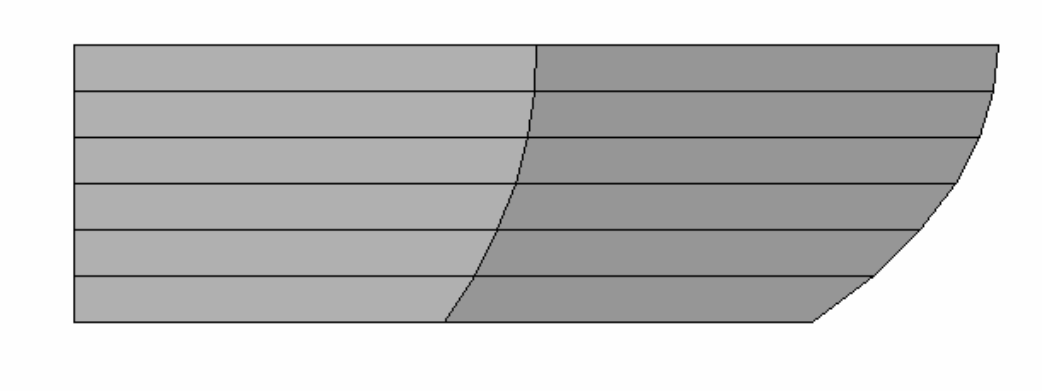


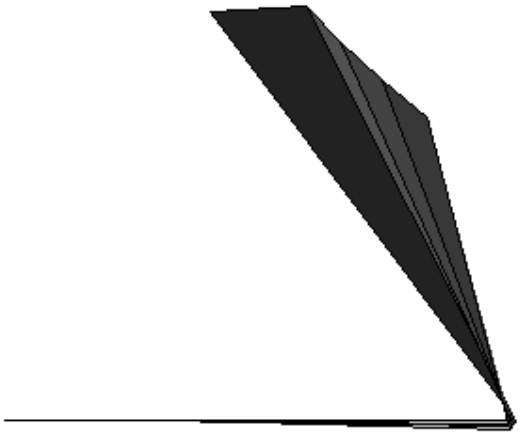
Fig. 3.41 Simple cylindrical developable a) 3FD Prediction, (b) Physical Model and (c) the 3FD predicted development.

Fig 3.41 (a) and (b) shows that the shape predicted by the 3FD program closely matches the physical model. The predicted fold angle of 90° and the angles of the generators relative to the folding curve are also similar for the 3FD model and the physical model.

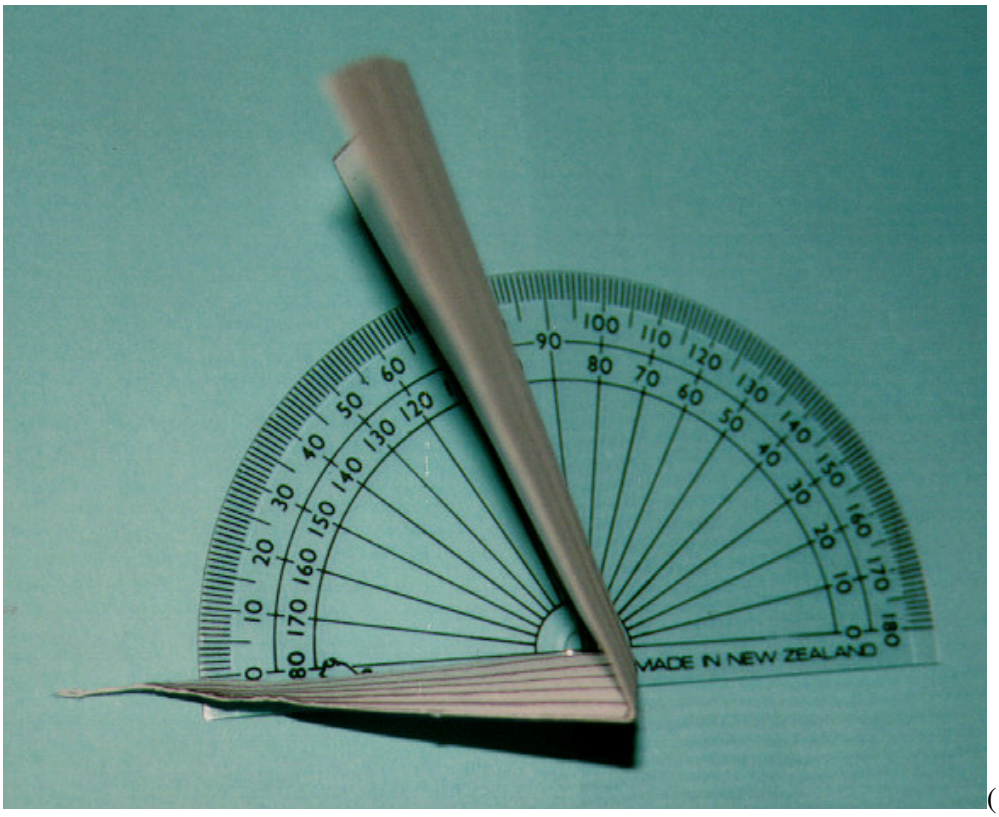
3.4.7.3. Developable Defined by a Function.

This folded developable's first surface has an edge of regression defined by the equation *y=cosx-1*. The surface is represented by seven generators at equal angular spacing.

The normal radius of curvature of the surface, ρ_N, at the end of each generator is fixed at 0.5. The folding curve is the arc of a circle, radius and geodesic radius of curvature, ρ_g, 0.5 with the centre at the start of the first generator.

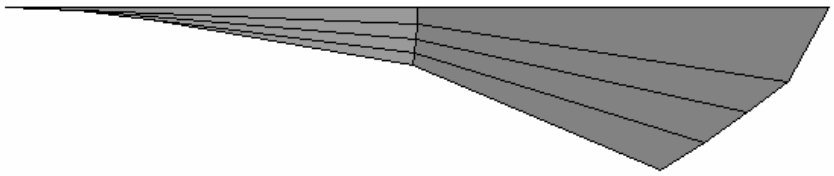


(a)



(

b)



(c)

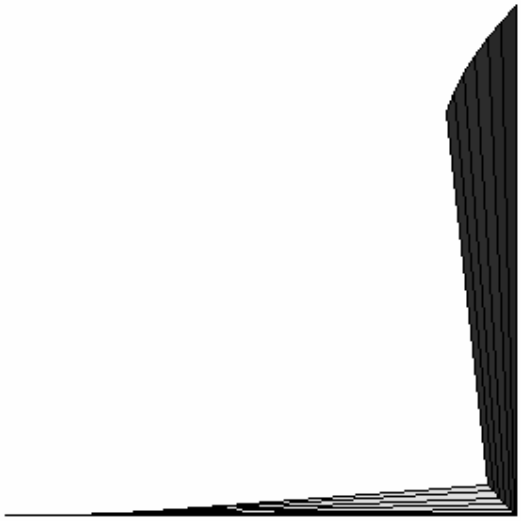
Fig. 3.42 Function defined developable (a) 3FD Prediction, (b) Physical Model and (c) the 3FD predicted development.

A comparison of Fig 3.42 (a) and (b) shows the shapes are almost identical. Due to the changes in surface normal curvature, ρ_N, at the folding curve the predicted fold angles range from 31° to 53°. The measured fold angles range from 33° to 54°. The difference is due to the difficulty in exactly making and folding the physical model. The thickness of a line on the cardboard can vary the radius of curvature and thus the fold angle.

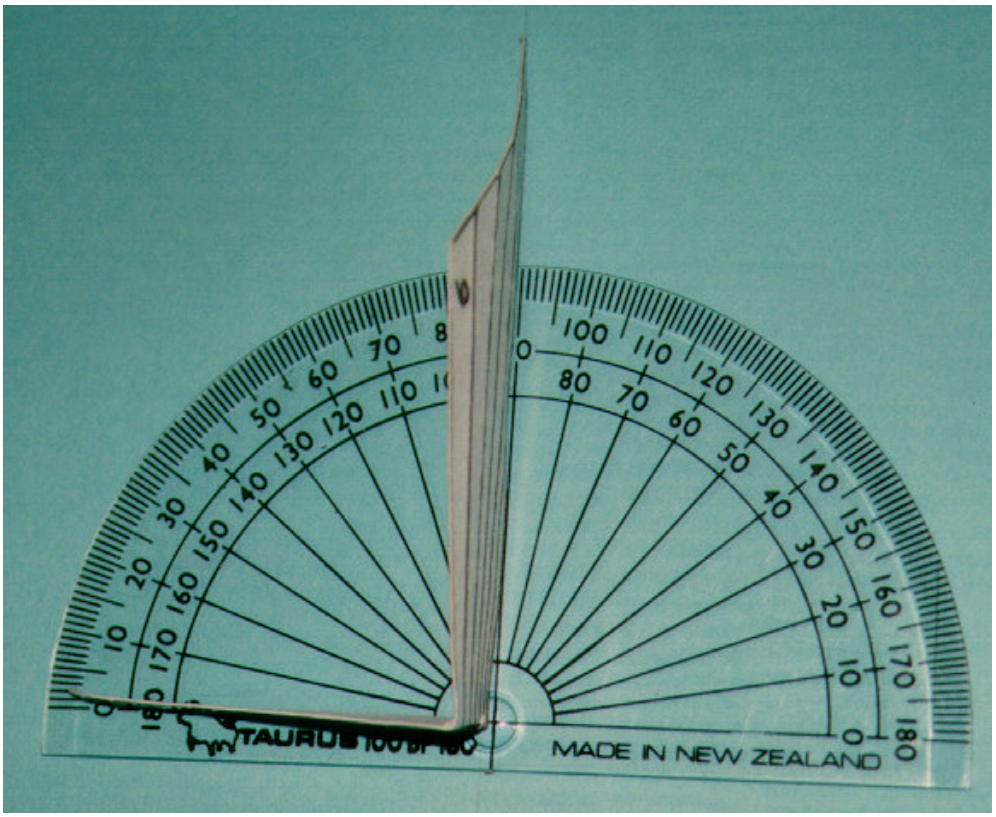
The generator angles with the folding curve for the physical model match those predicted by the 3FD program.

3.4.7.4. Bezier Defined Developable.

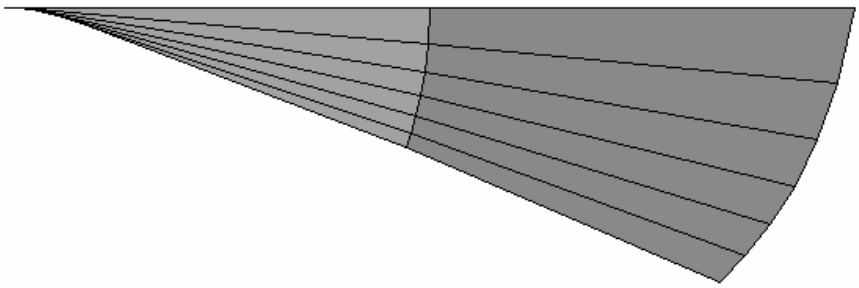
This folded developable has an edge of regression which is a bezier curve. The first surface is represented by seven generators which are projected tangent to the edge of regression at regular angular intervals. The plane projection of the folding curve, ρ_g, is a circular arc of radius 0.5.



(a)



(b)



(c)

Fig. 3.43 Bezier Defined Developable a) 3FD Prediction, (b) Physical Model and (c) the 3FD predicted development.

Once again the physical and computational models match in shape as shown in Fig. 3.43. The fold angles ranging from 90° for the first generator to 73° for the seventh generator match.

The generator alignment to the folding curve is also identical for both models.

3.4.7.5. Complex Conical Shape.

This example is a more complex folded developable because of the large changes in generator angle relative to the folding curve, on both sides of the folding curve.

In this case to physically model the 3FD results, the development predicted by the program must be used to make a physical model. If the two models match shape and fold angles we may assume that the program is accurately modelling a physically realisable situation.

The first surface of the folded developable is a cone surface with the normal radius of curvature, ρ_N, fixed at the ends of the generators at 0.5. The folding curve is the arc of a circle with the radius, in plane projection (ρ_g), of 0.6.

The centre of the folding curve is <u>not</u> located at the edge of regression; its location is shown in Fig. 3.44.

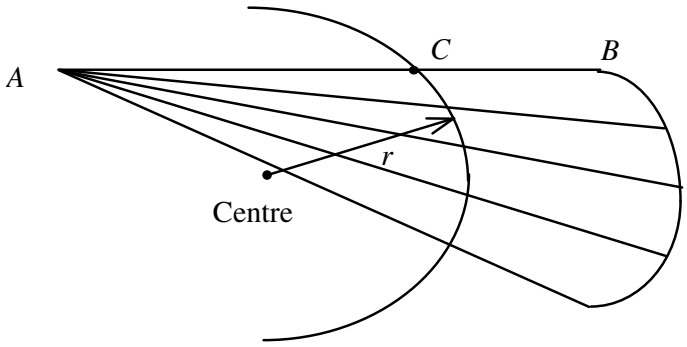


Fig 3.44 Centre of folding curve location

Because the normal radius of curvature of the cone, ρ_N, varies with generator length, it varies markedly along the folding curve as the generators are intersected at differing lengths. This results in large changes in the fold angle, which results in a large change in angle with respect to distance along the folding curve. This, from the theory described in Section 3.1, causes large changes in the generator angles. The changes in the angles are shown in Fig. 3.45. The physical model compared with the computational is shown in Fig 3.46.

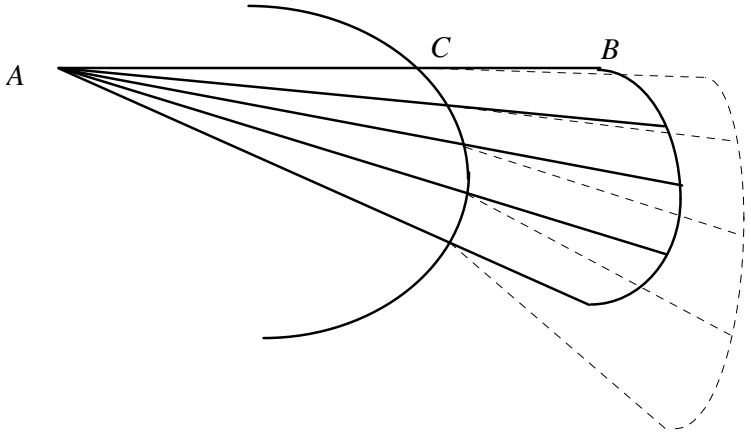
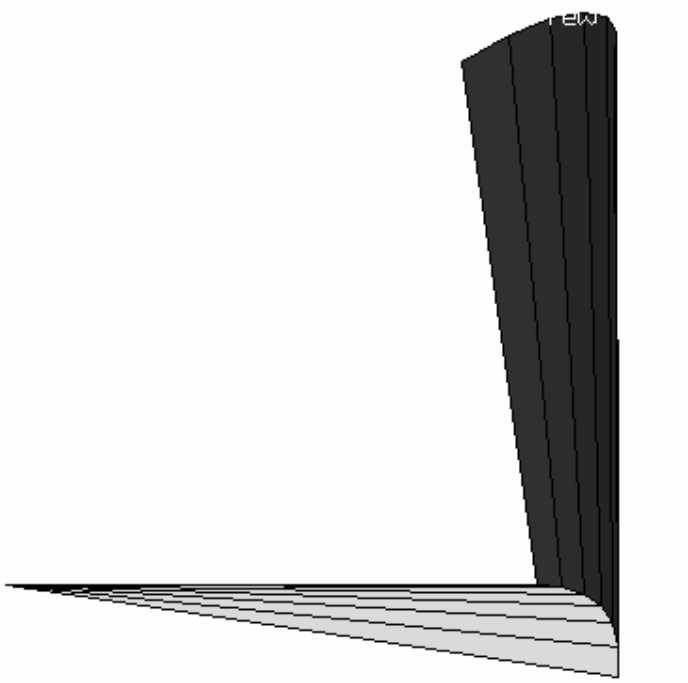
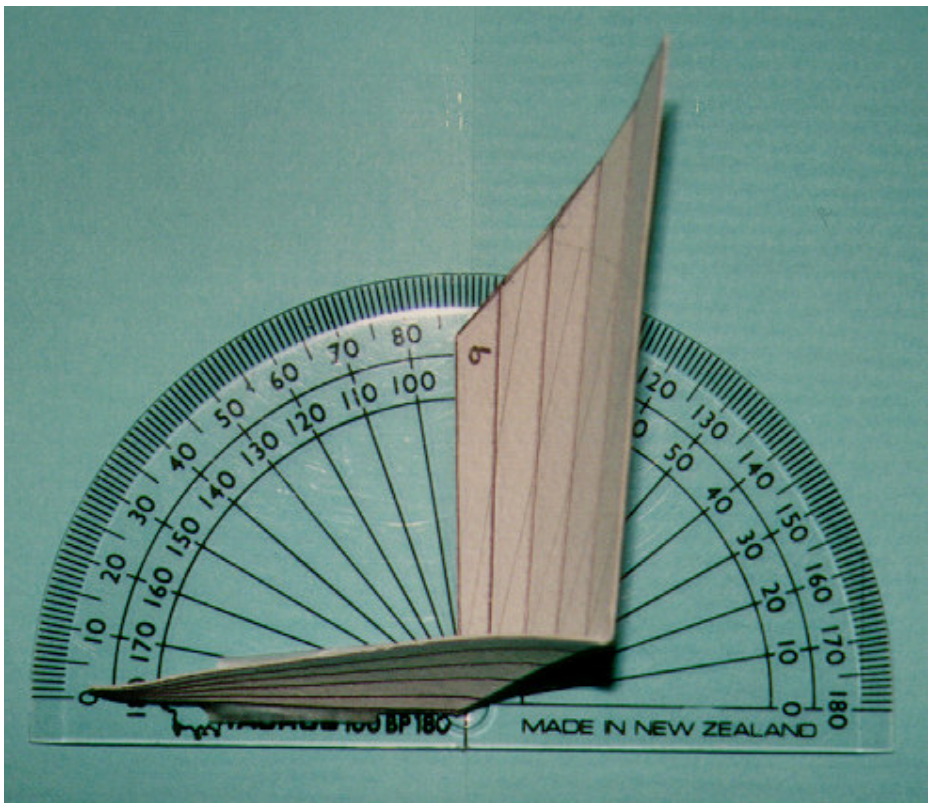


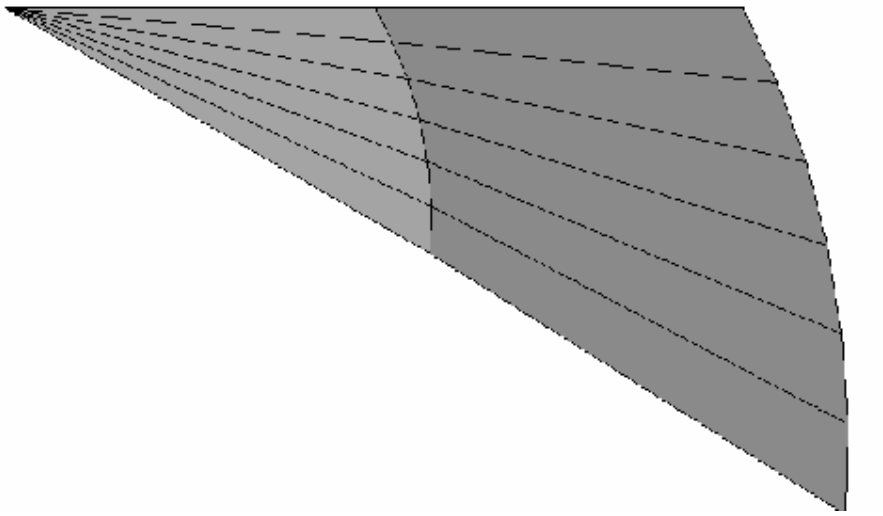
Fig. 3.45 The solid lines represent the development of the initial cone surface. The dashed lines represent the development of the second surface after it has been folded.



(a)



(b)



(c)

Fig. 3.46 Complex Coneal Shape (a) 3FD Prediction, (b) Physical Model and (c) the 3FD predicted development..

From Fig. 3.46 (a) and (b) above the shapes clearly match. Thus the shape predicted by the 3FD program accurately reflects the physical process of creating a folded developable.

3.4.7.6. Roofing Tile Shape.

An application of folded developables currently in production is metal roofing tiles. A simple developable roof tile can be modelled in seconds using 3FD. The resulting model is shown in Fig. 3.47.

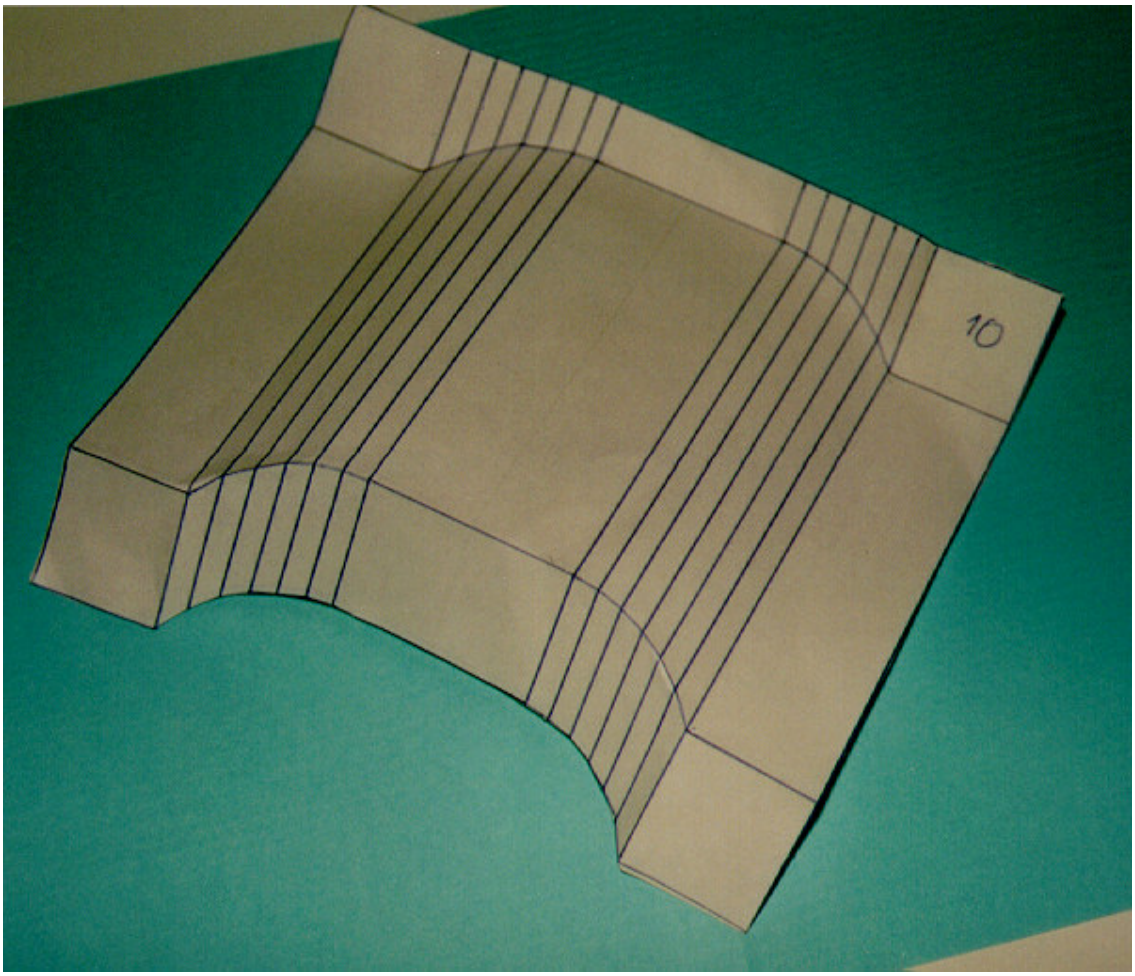


Fig. 3.47 Cardboard model of the roof tile developable .

The curved sections of the roof tile are simply the folded cylindrical surface described in example 2 The development of the folded developable shown in Fig. 3.48 below illustrates that the roof tile can be made purely by folding.

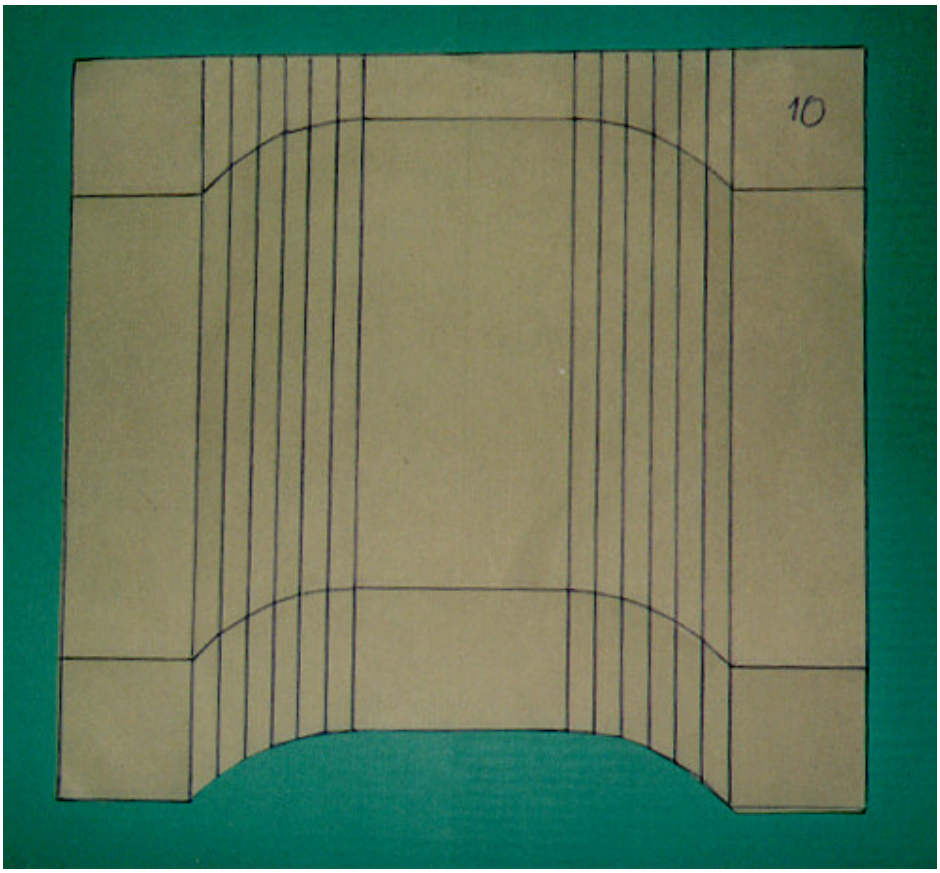


Fig. 3.48 Roof Tile Development.

The 3FD program allows the modelling of this, and more complex sheet forms, quickly, easily and as the examples show - accurately.

3.4.7.7. Convergence Check for Increasing Generator Numbers for an Identically Sized Developable.

To examine the accuracy of the central difference approximation used to determine the tangent angle, a convergence check on a sample result has been performed.

A conical developable with an angle of 30° between edge generators in its development has been created. The generators are one unit in length. The normal radius of curvature of the first developable surface is 1.0, at the generator ends. The folding curve is a circular arc radius of 0.6, with the centre at (0.0,0.3) if the first generator stretches from (0,0) to (1,0).

The developable was first created with three generators and the fold angle and tangent angles of the 'mid' (15°) generator recorded. The developable was then re-created with an increasing

number of generators, but identical curvature and folding curve. In each case the fold angle and tangent angles were recorded. The developable is shown in Fig. 3.49.

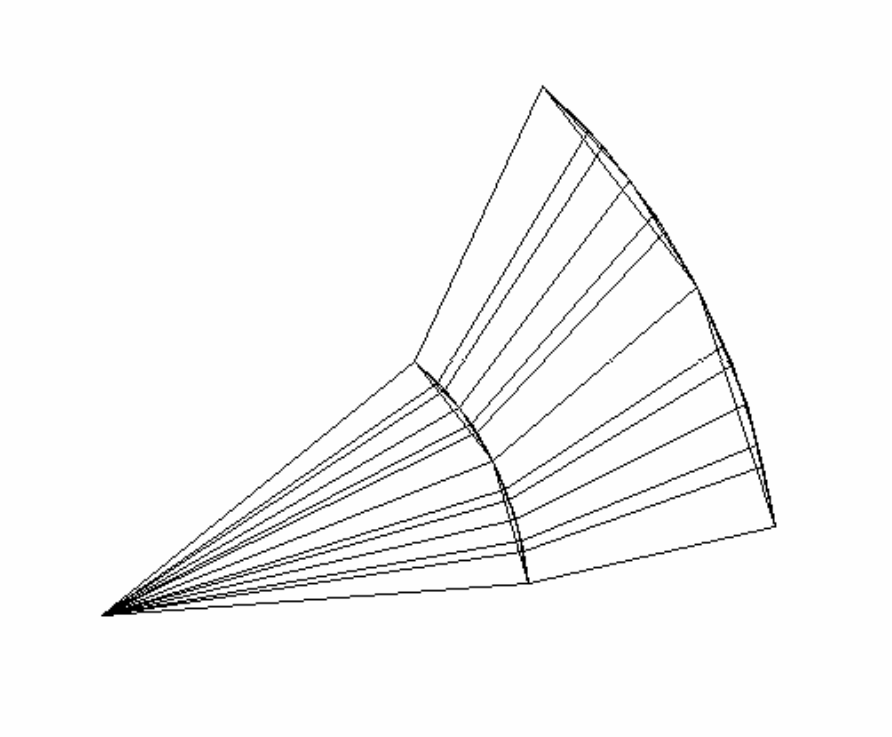


Fig. 3.49 A conical developable created with an increasing number of generators (3,5,7,9).

The fold angle for each developable is shown in Table 3.50.

Table 3.50 Fold Angle for the mid generator vs. Number of Generators that make a complete, but identically sized, developable.

Number of Generators	Fold Angle °
3	90.3
5	90.3
7	90.3
9	90.3
11	90.3
13	90.3
15	90.3
23	90.3

As expected fold angle is independent of the number of generators. The fold angle is determined purely by curvature relationships and does not use any central difference approximations.

The tangent angles of the generators are shown in tabular form in Table 3.51 and in graphical form in Fig. 3.52.

Table 3.51 Tangent Angles for the mid generator vs. Number of Generators that make a complete, but identically sized, developable.

Number of Generators	First Tangent Angle (γ_1)	Second Tangent Angle (γ_2)
3	109.9	65.1
5	114.5	60.7
7	116.0	59.1
9	116.7	58.6
11	117.2	58.2
13	117.4	57.9
15	117.6	57.8
23	118.1	57.3

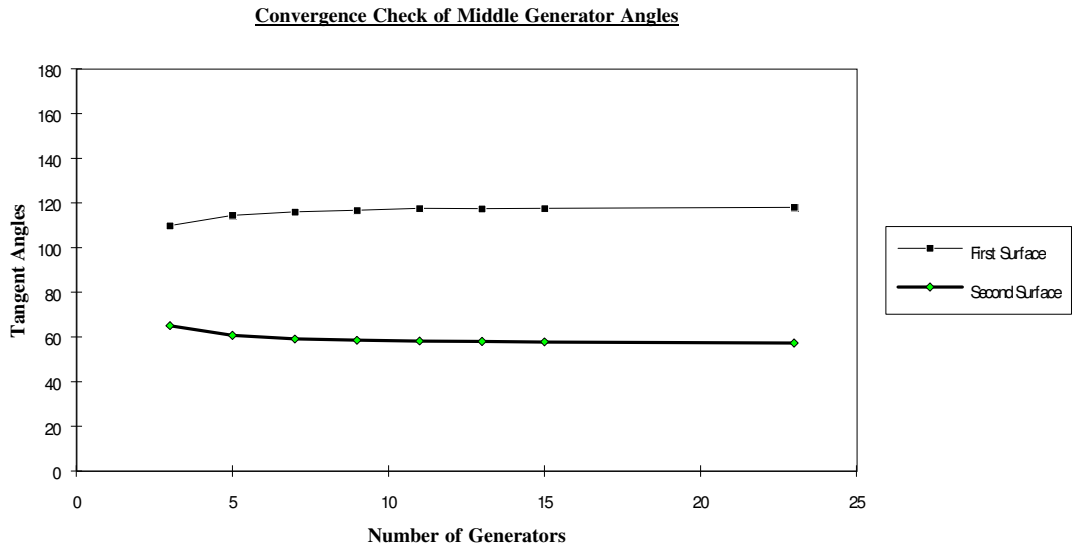


Fig. 3.52 Graphical Convergence Check.

The results converge rapidly, asymptotically approaching the final solution. This is the expected and desired result for a central difference approximation.

3.4.8. Conclusions.

Presented in the preceding section are the theory and equations required to construct the second surface of any general folded developable from any developable first surface and a general folding curve.

This theory forms the basis of the computer program 3FD. This program uses the above equations to allow the user to create three dimensional computational models of folded developables.

The 3FD program:

- Allows the user to define the first surface, by defining the first surface edge of regression and curvature.
- Allows the user to define the folding curve as a specific curve, such as the arc of a circle, or as any general curve.
- The program then constructs and displays the second surface.
- The developable can be rotated three dimensionally on screen and can be displayed as a wire frame or colour shaded image.
- The development of the folded developable can also be constructed using the Computer Aided Blank Shape Prediction method described in Section III.
- The programs Graphical User Interface allows the above process to be performed quickly and intuitively.

The results of the 3FD program have been compared with physical models. All the results, from simple conical developables to complex general curved developables, have shown excellent agreement between predicted and measured generator position.

The results produced by the 3FD program also show rapid and consistent convergence when tested with an increasing density of generators.

The 3FD program and the theory behind it accurately model folded developables. It is hoped that this will lead to a greater use of folded developables in science and industry.

4. DIE DESIGN - FOLDED DEVELOPABLES.

An important part of Engineering is the transition from theory to practice. This involves the engineer translating a design idea to a finished design. This may take the form of a detailed design drawing or a computer generated solid model, but the most common and most useful 'finished' design is a design prototype.

In the previous chapter a theoretical and computational method for the creation of curved folded developables was presented. This method proved to be accurate for the modelling of materials such as paper that can be easily creased or formed with very small fold radii. To form a sheet metal folded developable the simplest method is the use of a folding die. A description of a method that can be used to define such a die set is described in this chapter.

This method allows a user to design theoretical folded developables and to design sheet metal dies to manufacture them.

4.1. THEORY

This section describes the mathematics behind the creation of the folding dies. The transition between the theoretical folded developable shape and the folding dies involves the addition of a folding radius along the folding curve. At each section on the curve, the following steps must be performed:

1. the two generators that make up the section of folded developable of interest, must be translated to a reference plane

2. a folding radius must be added between the two generators, as shown in Fig. 4.1.

3. the modified generators and folding radius must be translated back to their original position.

These generators can be used to define the new folded shape. The final step in the creation of the folding dies is offsetting the top and bottom dies to allow for material thickness.

Springback is not considered in the creation of the dies as it is a material dependent phenomenon. The springback of folded developables is discussed by Staublin and Gerdeen [1986].

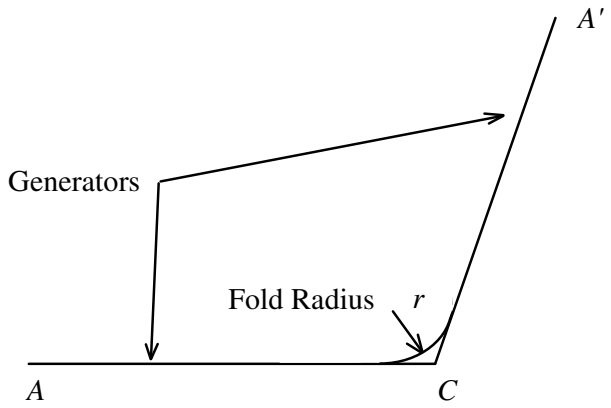


Fig 4.1 Two generators of a folded developable linked by a folding radius.

4.1.1. Co-ordinate Transformation.

The reference plane used for the addition of the folding radius is the *XY* plane with the *C* point at the origin. Consider the generator pairing shown in Fig. 4.2.

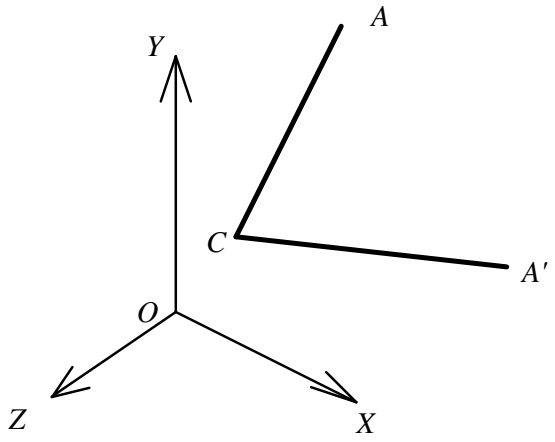


Fig 4.2 A generator pairing shown relative to the reference axes.

Point *C* and the associated generator pair, may be translated to the origin by;

$$\mathbf{A}_2 = \mathbf{A}_1 - \mathbf{C}_1$$

$$\mathbf{A'}_2 = \mathbf{A'}_1 - \mathbf{C}_1, \tag{4.1}$$

where **A**, **C** and **A'** are vector terms and the subscript 1 denotes the original vector, 2 the transformed vector. The transformed generator pair is shown in Fig. 4.3.

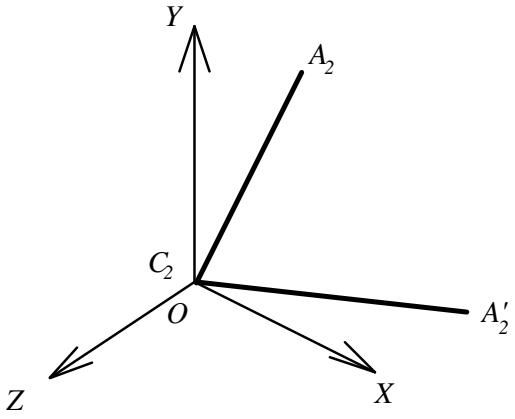


Fig 4.3 The generator pairing after the first transformation.

The plane containing A_2 and A'_2 has a unit direction vector $\boldsymbol{n}$, as shown in Fig. 4.4.

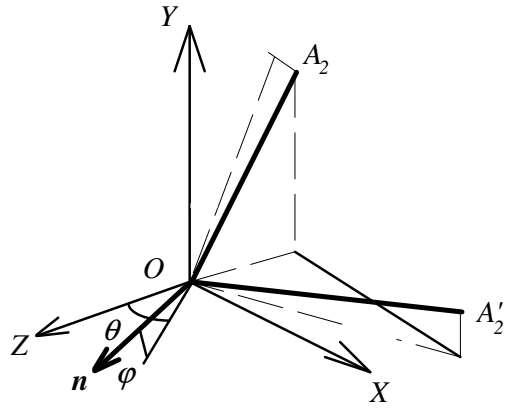


Fig 4.4 The generator pairing and the normal to the plane they lie in.

The unit normal vector is,

$$\mathbf{n} = \frac{\mathbf{A'}_2 \times \mathbf{A}_2}{\left|\mathbf{A'}_2 \times \mathbf{A}_2\right|}. \tag{4.2}$$

The rotation angles are defined by,

$$\sin\theta = \mathbf{n}_X ., \tag{4.3}$$

and

$$\sin\varphi = \mathbf{n}_Y . \tag{4.4}$$

The rotation can be expressed as two components. Rotation about the Y axis, $\boldsymbol{R_Y}$ is

$$\mathbf{R_Y} = \begin{pmatrix} n_Z & 0 & -n_X \\ 0 & 1 & 0 \\ n_X & 0 & n_Z \end{pmatrix}. \tag{4.5}$$

Rotation about the X axis, $\boldsymbol{R_X}$ is,

$$\mathbf{R_X} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -n_Y \\ 0 & n_Y & 1 \end{pmatrix}. \tag{4.6}$$

The combined rotation $\boldsymbol{R}$ is,

$$\mathbf{R} = \begin{pmatrix} n_Z & 0 & -n_X \\ -n_X n_Y & 1 & -n_Y n_Z \\ n_X & n_Y & n_Z \end{pmatrix}. \tag{4.7}$$

Following this rotation the generator pair will lie in the XY plane, as shown in Fig. 4.5.

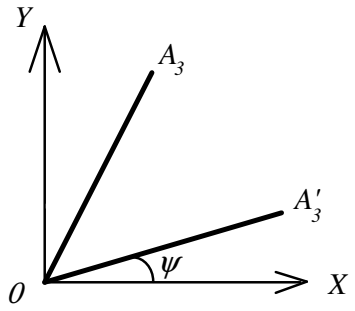


Fig 4.5 The transformed generator pairing after rotation into the XY plane. The angle ψ is the angle between the X axis and the closest transformed generator.

Following this transformation the folding radius is added. The generator pair with the addition of the folding radius is then translated back to its original co-ordinates using the same process described above but with the rotation matrix , $\boldsymbol{R^T}$, the transpose of $\boldsymbol{R}$.

4.1.1.1. Numerical Example.

Consider the generator pairing composed of the points:

A = (4, 4, 4)

C = (2, 2, 2)

A' = (4, 2, 4).

Using equation 4.1,

A_2 = (2, 2, 2)

A_2' = (2, 0, 2).

The normal to the generator pair is, from equation 4.2,

$$\mathbf{n} = \frac{(0-4, 4-4, 4-0)}{\sqrt{4^2+0^2+4^2}}$$

n = (-0.707, 0.0, 0.707).

The rotation matrix is, from equation 4.7,

$$\mathbf{R} = \begin{pmatrix} 0.707 & 0 & 0.707 \\ 0 & 1 & 0 \\ -0.707 & 0 & 0.707 \end{pmatrix}.$$

Rotating points A_2 and A_2' by the above matrix gives;

A_3 = (2.82, 2, 0)

A_3' = (2.82, 0, 2).

These points are clearly in the XY plane, and both generator length and the angle between the generators have been conserved. The points can be returned to their original position by rotation using the matrix,

$$\mathbf{R}^{\mathbf{T}} = \begin{pmatrix} 0.707 & 0 & -0.707 \\ 0 & 1 & 0 \\ 0.707 & 0 & 0.707 \end{pmatrix},$$

and addition of the original point C.

4.1.2. Addition of the Folding Radius.

If a folded developable is folded in a die with a regular folding radius, the radius forms the arc of a circle connecting the generators of the two surfaces. This is shown in Fig. 4.6.

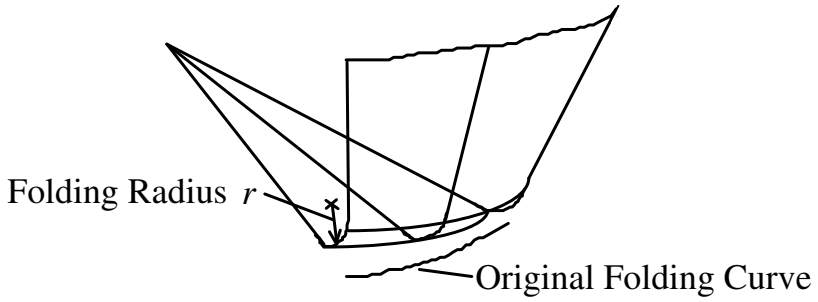


Fig 4.6 Folded Developable with fold radius shown.

In the *XY* reference plane a generator pair and folding curve radius, *r,* is shown in Fig. 4.7.

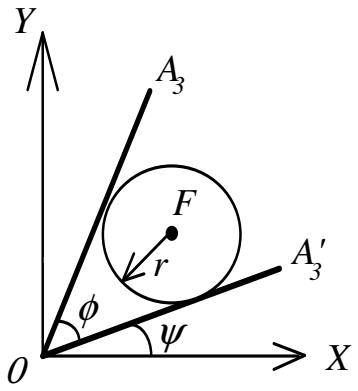


Fig 4.7 The generator pair and folding radius in the reference plane.

The folding curve radius centre is denoted by the point *F*. To define the curve described by the folding radius, the position of the centre of the folding radius must be found.

If the generator pairing is rotated through the angle ψ, the generator OA'_4 will be co-incident with the *X* axis. The centre of the folding radius, *F*, then lies at the point *(l,r)* as shown in Fig. 4.8.

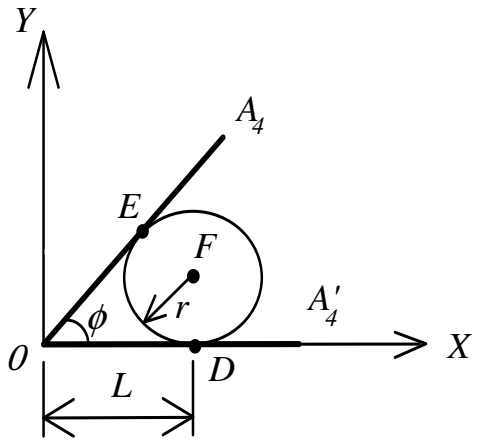


Fig 4.8 The generator pair and folding radius in the reference plane, rotated so the generator OA'_4 is co-incident with the X axis.

From the geometry the distance L is given by,

$$L = \frac{r}{\tan\left(\frac{\phi}{2}\right)}. \tag{4.8}$$

From Fig 4.8 the point D is $(L,0)$ and E $(L\cos(\phi/2),L\sin(\phi/2))$. The equation of the folding radius curve is,

$$(x - F_x)^2 + (y - F_y)^2 = r^2. \tag{4.9}$$

From equation 4.9 any point on the folding radius may be calculated.

Thus in the reference plane, the new shape of the generators with folding radius is fully described.

The final shape is achieved, by the generator pairing and fold radius being rotated by the angle $-\psi$, before translation by $\boldsymbol{R^T}$ and addition of the original point C.

4.1.3 Die Face Offset.

A Die Set typically consists of two dies. During folding the dies close completely on the metal folding it to match the shape of the dies. In folding the deformation should be limited to the folding curve, so the dies must be at least the thickness of the sheet apart.

To design the die shapes the steps described in section 4.1.2 must be followed, but with the following changes.

1. For the top die the A and A' points must be offset from the middle surface by t/2 and the folding radius reduced by t/2, where t is the thickness of the sheet.

2. For the bottom die, the A and A' points must be offset from the middle surface by t/2 and the folding radius increased by t/2.

4.2. THE DIE DESIGN PROGRAM

The design of dies for the forming of folded developables is included in the design program 3FD. The GUI menu system allows the user to create a die set with a specified folding radius using an already created folded developable. The program can also display the blank shape required to produce such a die formed folded developable. The major subroutines are described in the following sections.

4.2.1. Die Fold

This subroutine adds a folding radius to an existing folded developable. The program currently limits die set creation to folded developables based on cylinders and cones, but the theory and program can be extended to general folded developables. The major steps in the subroutine are illustrated in Fig. 4.8.

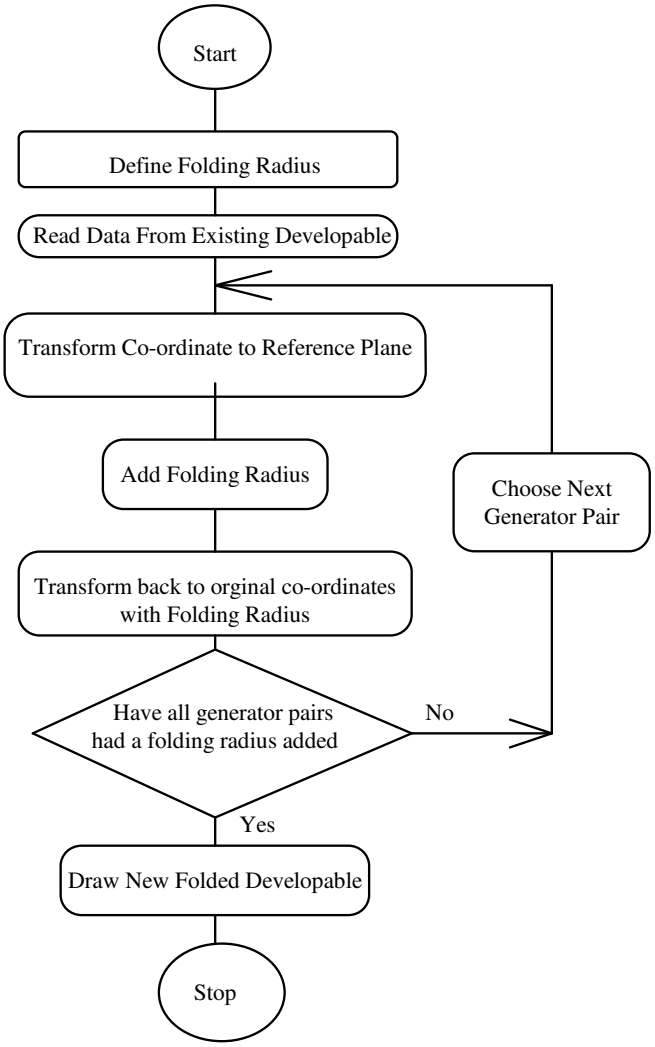


Fig 4.8 Schematic of the Die Fold Subroutine.

The folding radius is defined in terms of a fold radius to thickness ratio. This ratio is typically specified for common folding materials. It gives an indication of how tight a fold can be sustained by a given material. The use of the ratio ensures that the output from the program is dimensionless and generally within -1 to 1 in all dimensions. The output can then be scaled to match the desired sheet size or thickness.

The mathematics of Die Fold follows the theory described in Sections 4.1.1 and 4.1.2.

4.2.2. Die Draw

The die draw subroutine uses the same structure and subroutines as the draw subroutine used in 3FD to display folded developables, with the addition of drawing the folding radius. The points on the folding radius are stored in an array, D similar to the arrays A, C and A' that store the points of the start of the first surface generators, the folding curve and the end of the second surface generators respectively.

The die draw subroutine follows a different drawing process to the draw subroutine, this process is shown in Fig. 4.9.

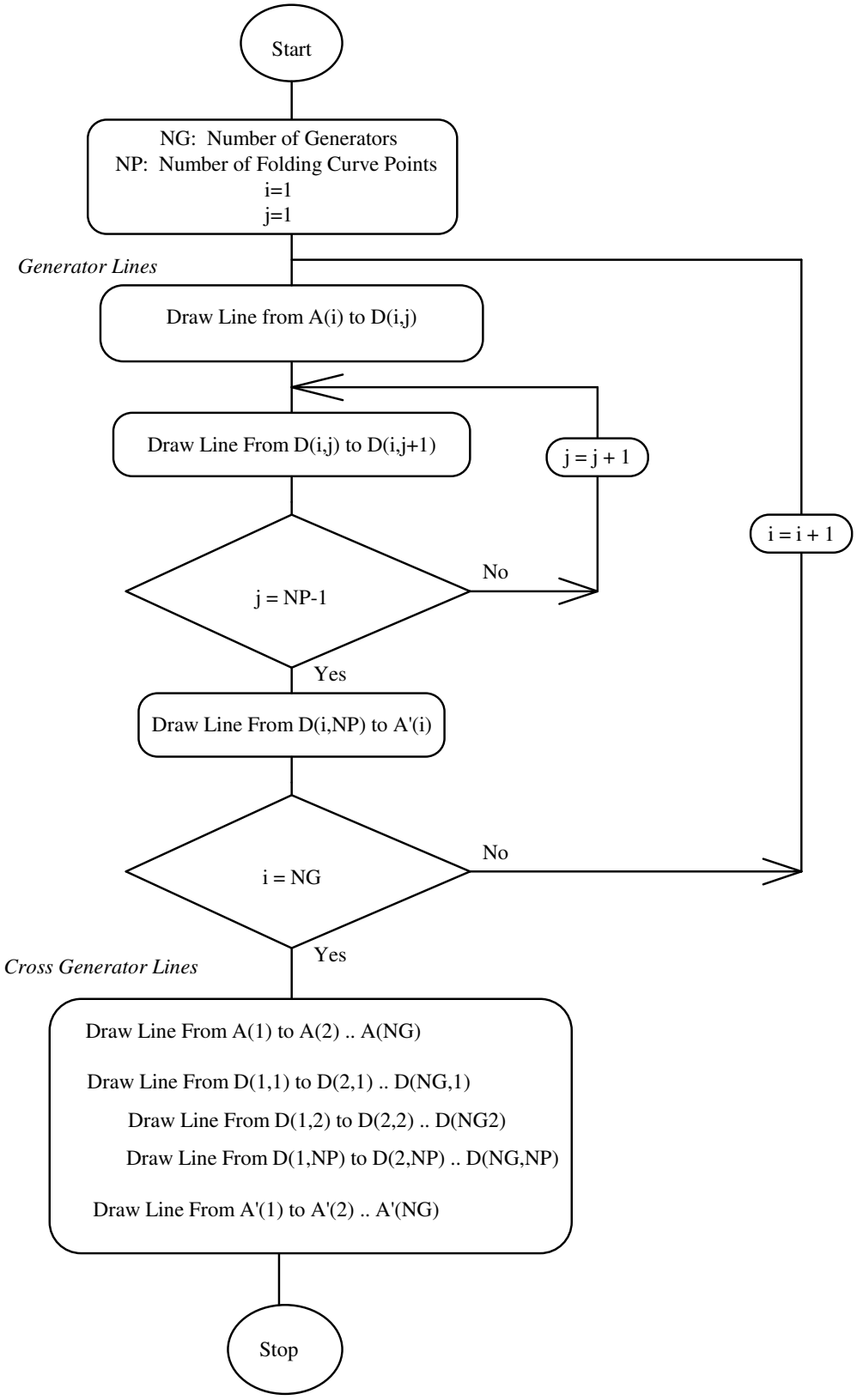


Fig. 4.9 Schematic of the Procedures of Subroutine Die Draw.

4.2.3. Die Set

This subroutine repeats the die fold subroutine but with modified input values. The top die is displaced upwards and the bottom die downwards. Fig. 4.10 illustrates the procedures followed.

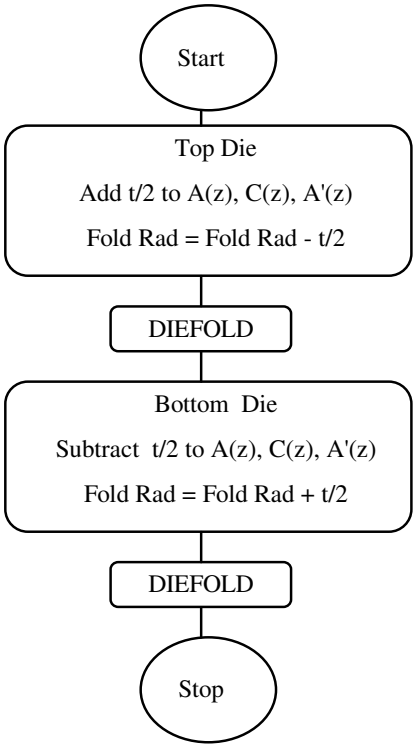


Fig 4.10 Schematic of the procedures of the die set subroutine.

4.2.4. Die Blank

This subroutine predicted the blank shape required for the die set created by the die set subroutine. The blank shape is predicted using the Computer Aided Blank Shape Prediction (C.A.B.S.P) method described in Chapter 5.

The blank shape prediction method uses the constant area transformation to determine the blank shape. The three dimensional folded developable, with or without folding radius may be represented by a continuous set of adjacent quadrilaterals, as shown in Fig 4.11 below.

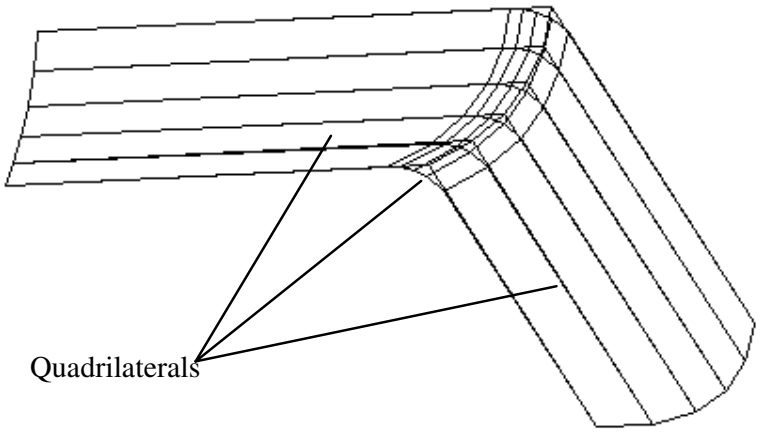


Fig 4.11 The Quadrilateral representation of a folded developable with folding radius.

The blank shape is mapped to a flat plane, quadrilateral by quadrilateral preserving the area from the three dimensional quadrilaterals to the planar quadrilaterals.

The subroutines that perform this transformation are described in Appendix 7.2.

4.3. EXAMPLES

4.3.1. Numerical Model of a Die Set and Die Blank Based on a Simple Cylindrical Developable.

The folded developable shown in Fig 4.12 consists of a simple cylindrical developable, with a first surface principal radius of curvature of 0.5 and a folding curve of radius 0.4, thus the fold angle is 103°.

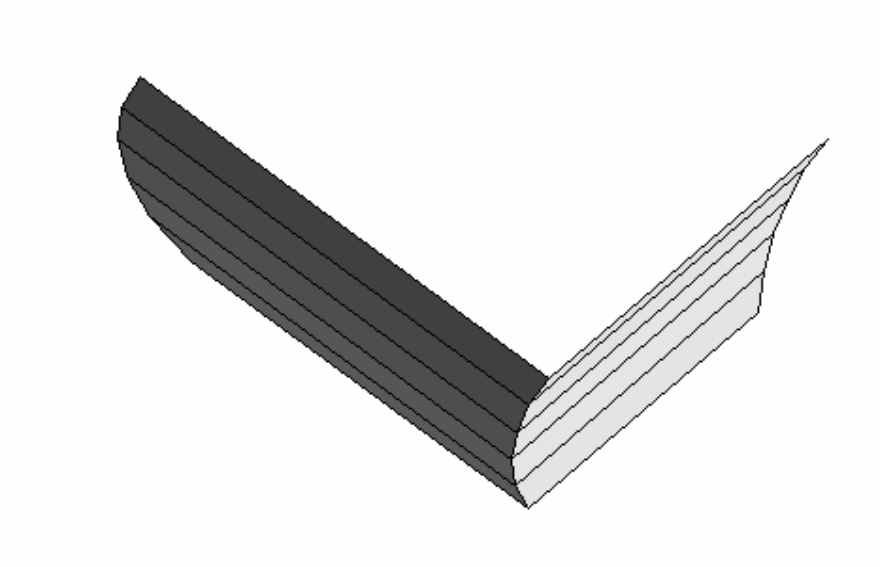


Fig. 4.12 Cylindrical Folded Developable.

A folding radius equal to 10 times the thickness of the sheet is added using the Die Fold subroutine to produce the developable shown in Fig 4.13.

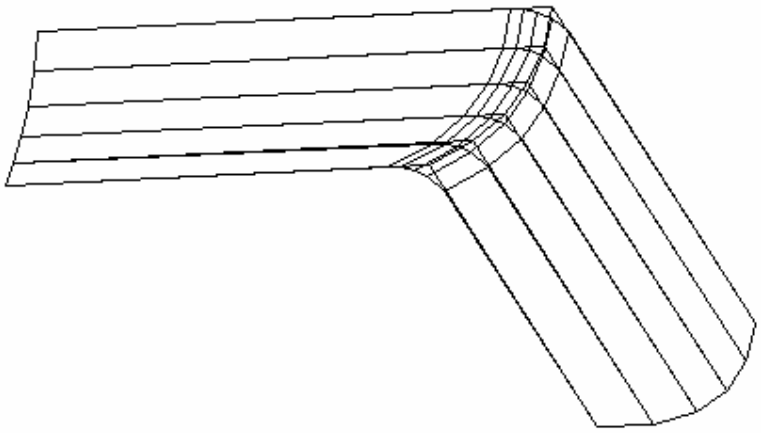


Fig. 4.13 Cylindrical Folded Developable with folding radius added.

The die set, for the folded developable is shown in Fig. 4.14. The top and bottom dies are displaced some distance from the developable for ease of viewing.

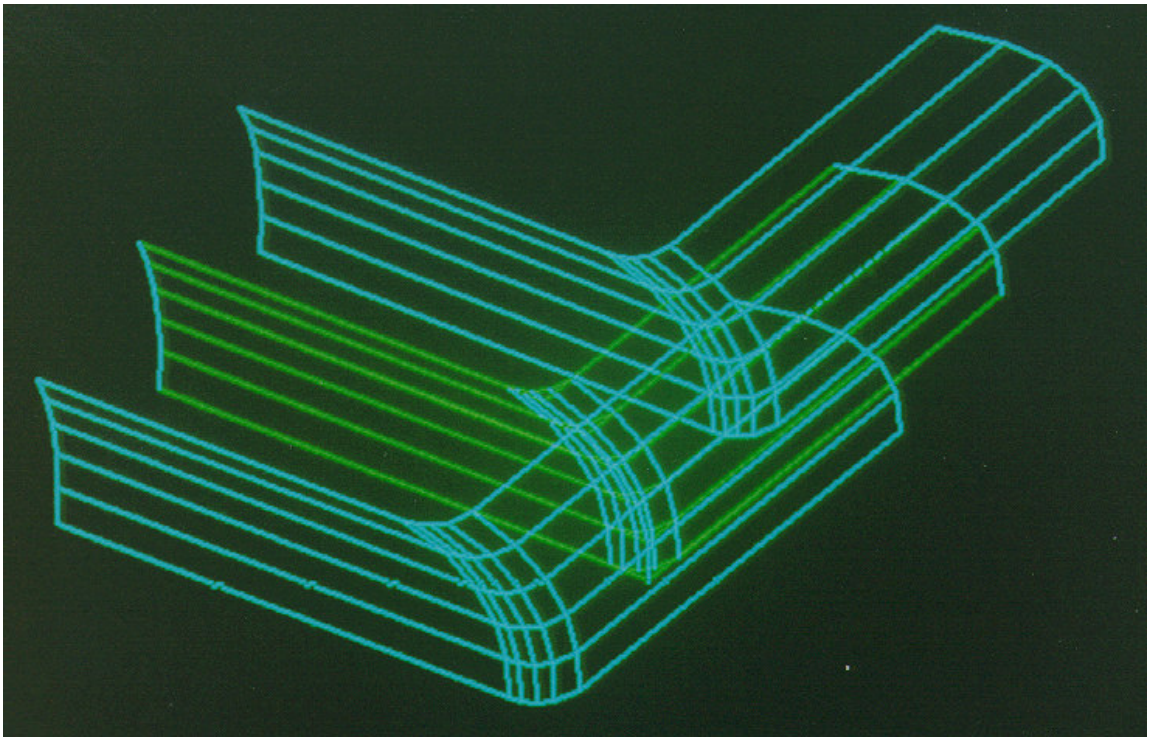
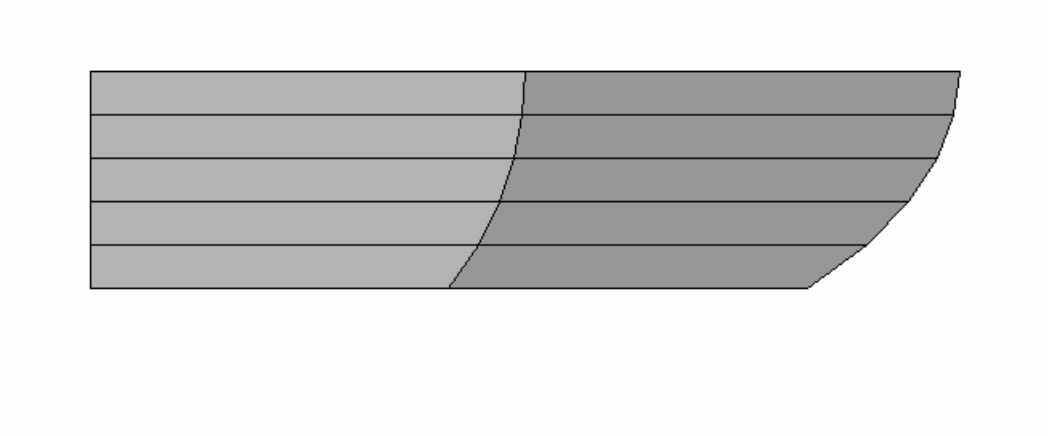


Fig. 4.14 Cylindrical Folded Developable Die Set.

The blank shape predicted by the 3FD program for the die set is shown in Fig 3.15.

(a)



(b)

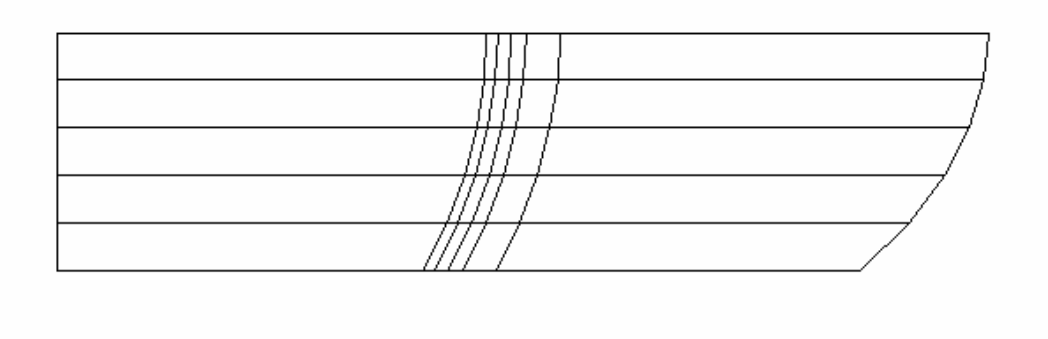


Fig. 4.15 (a) Cylindrical Folded Developable Blank Shape. (b) Blank Shape with the addition of a folding radius.

The blank shape with the folding radius is very similar to the blank shape formed without the folding radius. This is an expected result as the folding radius is small in relation to the generator lengths. The lines that make up the segments of the folding curve are clearly shown on the blank shape.

4.3.3. Physical Model of a Die Set and Die Blank Generated from 3FD Output.

The physical model was generated by the following steps.

1. Die Set generated in 3FD
2. Numerical Output passed to CATIA
3. Numerical Output used to create a NURBS surface model in CATIA
4. Cutter Paths generated in CATIA
5. CATIA output sent to Five Axis Mill for NC Machining
6. Blank Shape cut from sheet by hand.
7. Folded Developable formed between the two dies pressed.

The physical folded developable is shown in Fig 4.16. The physical developable is a close geometric match to the computational model shown in Fig. 4.12.

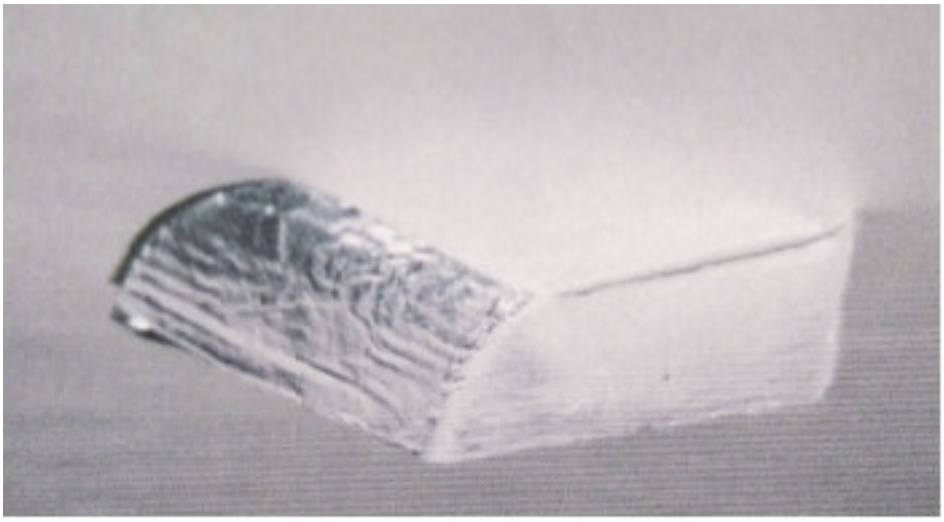


Fig. 4.16 The Physical Cylindrical Folded Developable.

The folded developable was formed from thin aluminium foil, as this cannot easily be stretched. Thus the folded developable could only be formed by folding. The female section of the die set was machined from timber and is shown in Fig. 4.17

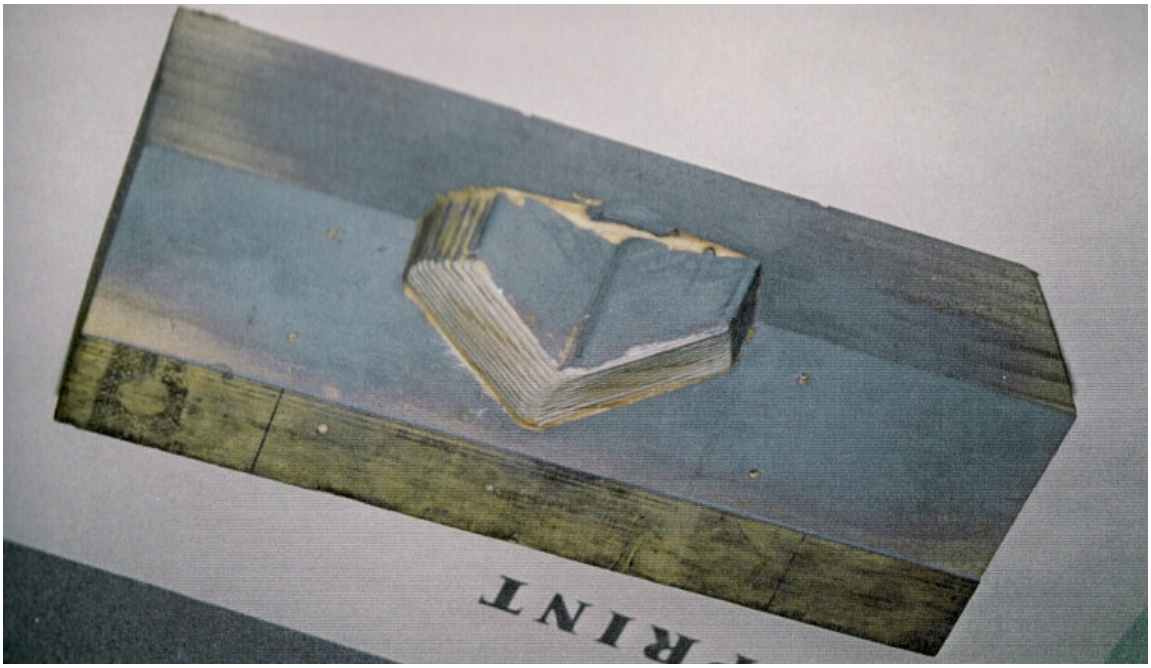


Fig 4.17 The female die for forming the cylindrical folded developable.

The male die was made from plaster of Paris using the female die as a mould. The complete die with the folded developable between the dies is shown in Fig. 4.18.

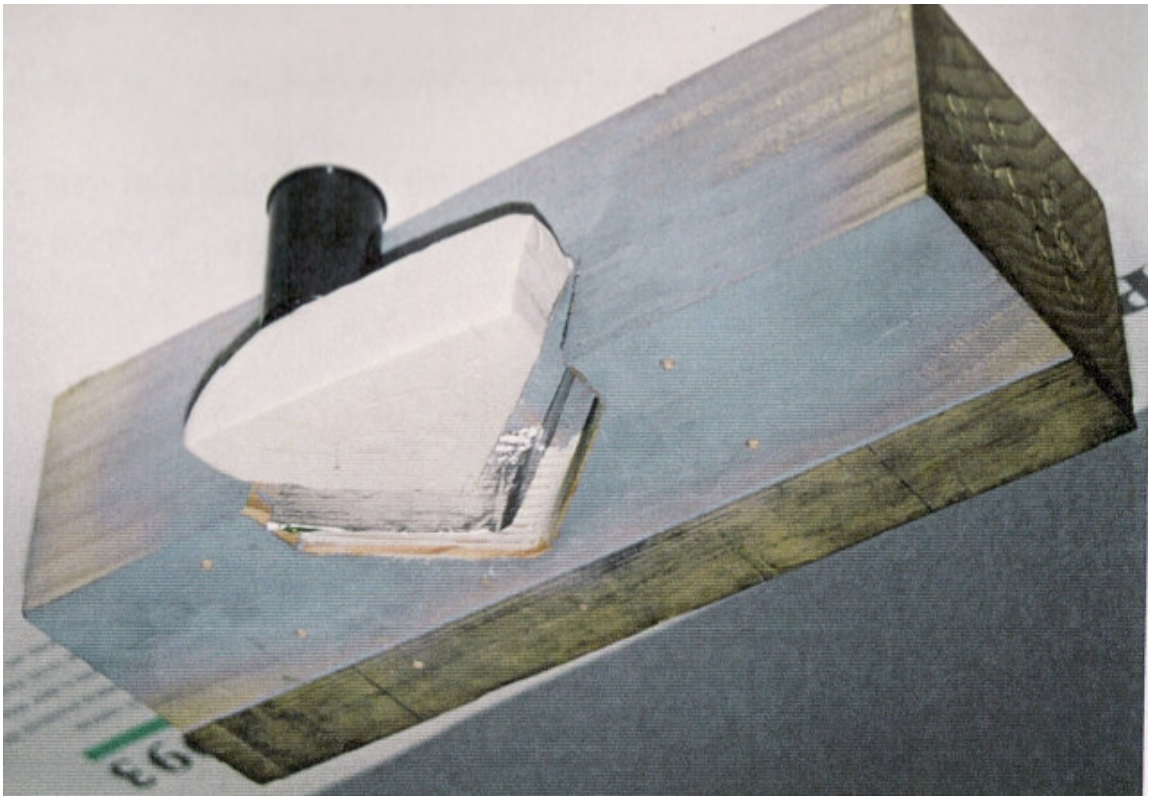


Fig. 4.18 Cylindrical Folded Developable with folding radius added.

While the die set created was simple, it aptly illustrates that the die shapes created by the design program can be used to manufacture actual parts.

4.4. CONCLUSIONS

The Die Design section of the 3FD program provides the important step from idea to reality.

It allows the rapid design and manufacture of dies for the forming of folded developables.

It provides the final step in taking curved folded developables from interesting mathematical entities to physically useful design parts.

5. THE KINEMATICS OF FOLDED DEVELOPABLES.

Kinematics is the study of motion. In this section the kinematics of folded developables, their motion as they fold, is investigated. Previous works on Folded Developables have not covered kinematics in detail. The development of a computational model of folded developables, as discussed in Chapter 3, allows the kinematics of folded developables to be modelled as well.

5.1. INTRODUCTION.

Folded developables are formed from sheets of material. Thus the kinematics of folded developables involves an investigation of sheet kinematics and mechanisms. A good introduction to the definition and design of mechanisms is Molian [1982]. Mechanisms may be classified as components that translate one mechanical motion into another predictable motion.

Sheet mechanisms include common components such as snap fits and bimetalic switches. Less common components such as a Spencer disk and a Belleville spring are also sheet mechanisms and their usage provides an insight into possible applications for a folded developable mechanism [Tuttle 1967].

A Spencer disk, shown in Fig. 5.1(a), is a snap action mechanism actuated by heat. Two metals with widely different coefficients of thermal expansion are fused together to form a convex or spherical disk. When heated the disk tends to reverse its curvature but this is resisted by the tension in the rim. When the heat and consequent force exceed the resisting force the disk snaps through the centre. Upon cooling the reverse takes place. Spencer disks are used as thermal cut outs on current limiting relays.

A Belleville spring, shown in Fig 5.1(b), has a snap action as it nears the centre position. The load can be a few grams or several hundred kilograms, dependent on its exact geometry.

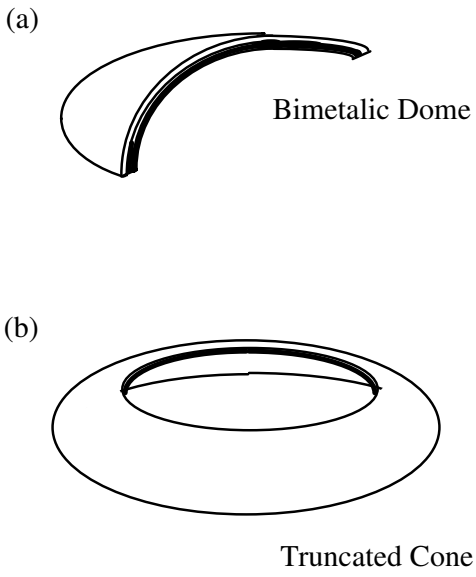
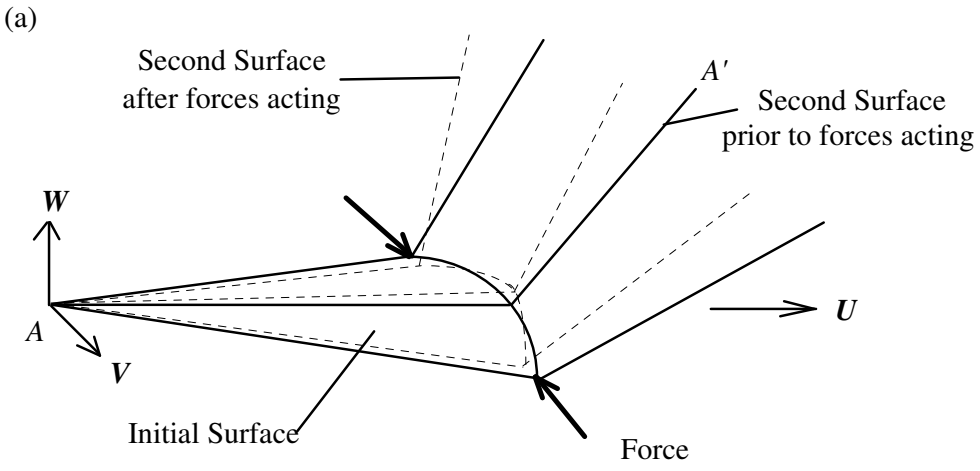


Fig 5.1 (a) a Spencer disk. (b) a Belleville spring.

However a folded developable mechanism generally does not 'snap' from one position to another. It provides a continuous range of motion within defined limits.

To determine the mechanisms of folded developables, the way they react to imposed forces and the theory behind this must be examined. A folded developable formed from a thin flexible sheet, such as cardboard can have forces applied to it in two distinct ways. The first involves changing the curvature of one of the surfaces, hereafter referred to as the first or initial surface. The second involves increasing or decreasing the twist of the folding curve.



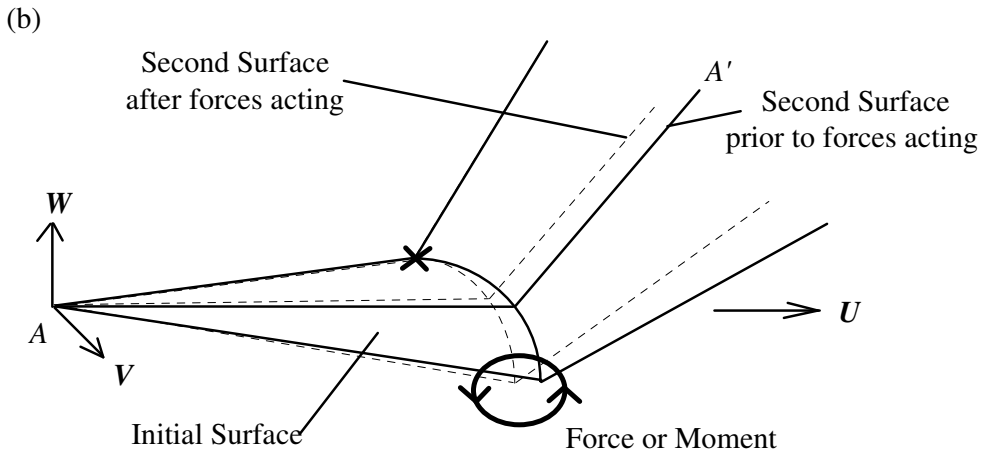


Fig 5.2 (a) forces that alter the curvature of a folded developable. (b) forces that change the torsion of the folding curve.

These two mechanisms are described in the following sections.

5.2. FOLD ANGLE KINEMATICS

The motion of a folded developable during folding is dependent on three major parameters:

(i) the normal surface curvature of the initial, or first, developable surface

(ii) the curvature of the plane projection of the folding curve

(iii) the twist of the folding curve.

The first two parameters effect the fold angle of the folded developable. Changing either of them results in the fold angle changing and the folded developable moving.

In this examination of the kinematics of folded developables, the generators are assumed to remain fixed to the folded surfaces.

5.2.1. Parameters That Govern Generator Position.

The position of the generator in the 'second' developable surface of a folded developable relative to the 'initial' surface is dependent on the fold or dihedral angle, α, and the two tangent angles γ_1 and γ_2. These angles are shown in Fig. 5.3. The fold angle is equal to $\pi-2\beta$ where β is the angle between the tangent plane and the normal to the folding curve and 2β is known as the bend angle.

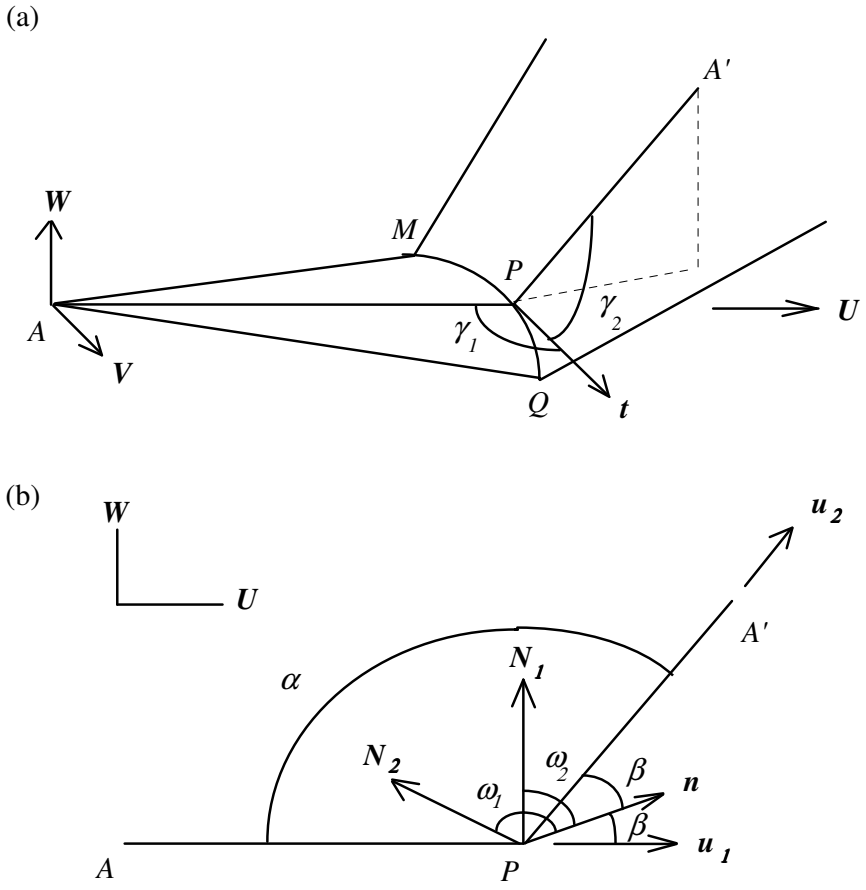


Fig. 5.3 (a) The generator tangent angles γ_1 and γ_2 .(b) The definition of the fold angle α.

5.2.2. The Effects of Changing the Normal Curvature of the First Surface on the Fold Angle.

The most important condition on the fold angle is described in Section 3.3. During the forming of a folded developable

$$\tan\beta = \frac{K_N}{K_g}, \tag{5.1}$$

where K_N is the principal normal curvature of the developable surface at the folding curve and K_g is the geodesic curvature of the folding curve. This may be expressed in terms of radii of curvature as

$$\tan\beta = \frac{\rho_g}{\rho_N}, \tag{5.2}$$

where, β is the angle between the normal to the folding curve and the surface normal, ρ_N is the non-zero principal radius of curvature of the developable surface at a given point and ρ_g is the geodesic radius of curvature of the plane projection of the folding curve.

As the fold angle, α, is $\pi-2\beta$, the fold angle may be expressed as,

$$\alpha = \pi - 2\arctan\left(\frac{\rho_g}{\rho_N}\right). \tag{5.3}$$

The fold angle is influenced purely by the geodesic radius of curvature and the normal radius of curvature of the first surface.

As described in section 3.3, when the fold curve is straight, $\rho_g \to \infty$. When this occurs α is either 0 or 2π, the sheet either folds back on itself or no fold occurs.

If the geodesic radius of curvature is fixed, the fold angle,

$$\tan\alpha \propto -\frac{1}{\rho_N}. \tag{5.4}$$

For this case, $\tan\alpha$ is dependent purely on the normal radius of curvature of the developable surface.

5.2.3. The Effects of Changes in the Geodesic Curvature of the Fold Curve, on the Fold Angle.

From equation (5.2) if the principal radius of curvature of the surface is fixed, the fold angle, α, is dependent purely on the geodesic radius of curvature of the folding curve. Thus,

$$\tan\alpha \propto -\rho_g. \tag{5.5}$$

The accuracy of the equations used by the 3FD program can be checked by measuring the fold angle of a large physical model and comparing it with the fold angle predicted by 3FD.

5.2.4. Experimental Verification.

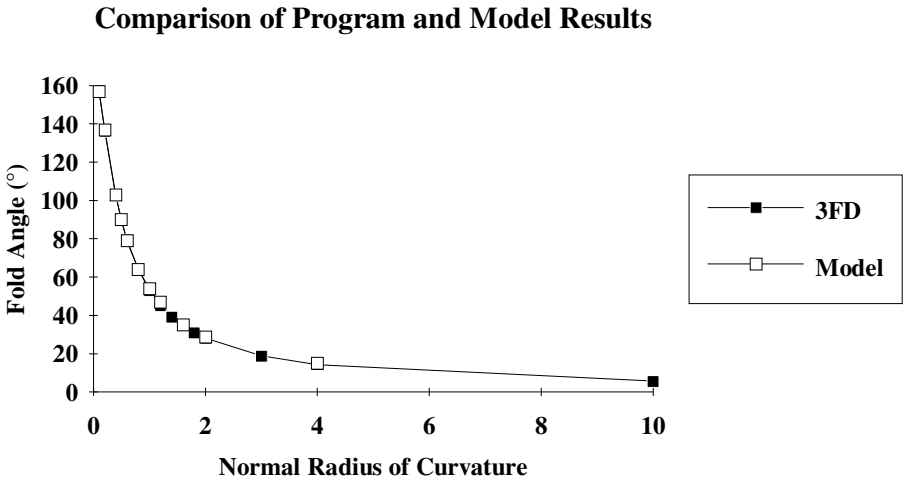
The models used for the validation of the kinematics results were a large card folded developable. The folded developables were formed from two cylindrical surfaces. Templates were used to ensure that changes in the curvature of the first surface were as close as possible. Different models were created for each of the examples that had a different folding curve radius of curvature. In all cases the models were made from templates created by the 3FD program.

5.2.4.1. Changes in the Normal Curvature of the First Surface.

The results and comparisons are shown below in Fig. 5.4(a) and (b). Generator Length is 1.0.

Normal Radius of Curvature	3FD Fold Angle	Model Fold Angle Range
0.1	157.4	156-158
0.2	136.4	136-138
0.4	102.7	102-104
0.5	90.0	90
0.6	79.6	78-80
0.8	64.0	63-65
1.0	53.1	53-55
1.2	45.2	46-48
1.4	39.3	
1.6	34.7	33-36
1.8	31.1	
2.0	28.1	27-31
3.0	18.9	
4.0	14.25	13-17

10.0	5.73	



5.4 (a) Fold Angle (in Degrees) vs. Normal Radius of Curvature with Geodesic Radius of Curvature fixed at 0.5. (b) Graph of Results

From Fig. 5.4 it is clear that the 3FD program is accurately modelling curved line folding.

Graphically the folded developable 'opens' as the normal radius of curvature increases. This is illustrated by Fig. 5.5.

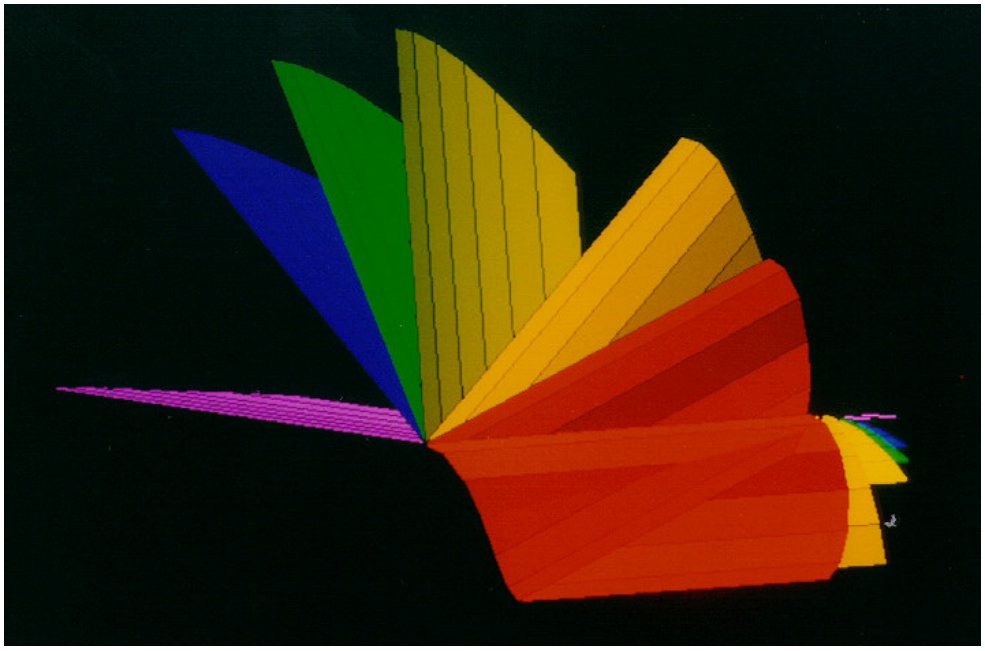


Fig. 5.5 Cylindrical folded developable. Geodesic Radius of Curvature Fixed at 0.5. Normal Radius of curvature ranges from 0.1 to 10.0. Note: The artifacting of the first surface is caused by an error in the graphics system, not the program.

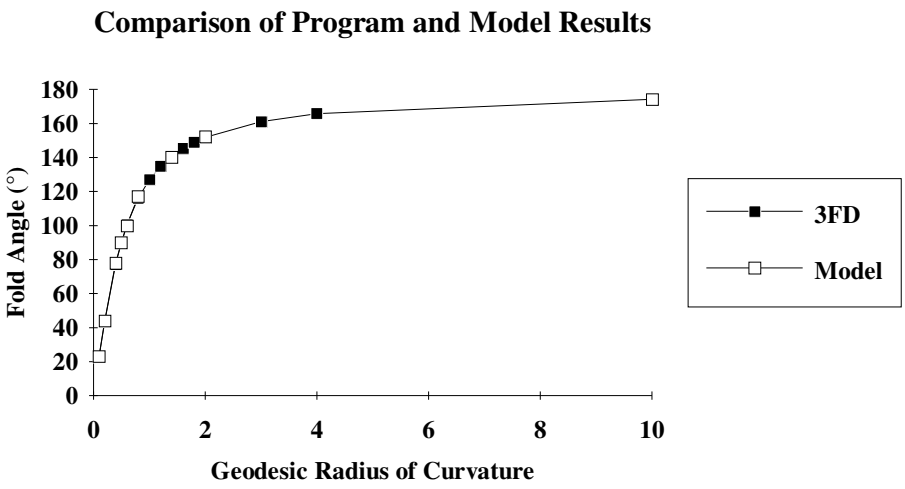
The results are as expected from theory, as the normal radius of curvature of the developable surface increases, the developable approaches a flat surface and the fold angle asymptotically approaches 180°. If the fold angle is 180° there is no fold and the surface is continuous.

5.2.4.2. Changes in the Geodesic Curvature of the Folding Curve.

The results and comparisons are shown below in Fig. 5.6(a) and (b).

Geodesic Radius of Curvature	3FD Fold Angle	Model Fold Angle Range
0.1	22.6	22-24
0.2	43.6	43-45
0.4	77.3	77-79
0.5	90.0	90
0.6	100.4	99-101
0.8	116.0	116-118
1.0	126.9	
1.2	134.8	
1.4	140.7	138-141
1.6	145.3	
1.8	149.0	
2.0	151.9	150-154
3.0	161.1	
4.0	165.8	
10.0	174.3	172-176

5.6 (a) Fold Angle (in Degrees) vs. Geodesic Radius of Curvature with Normal Radius of Curvature fixed at 0.5.



5.7 (b) Graphical Results

Graphically the folded developable 'closes' as the geodesic radius of curvature increases. This is illustrated by Fig. 5.8.

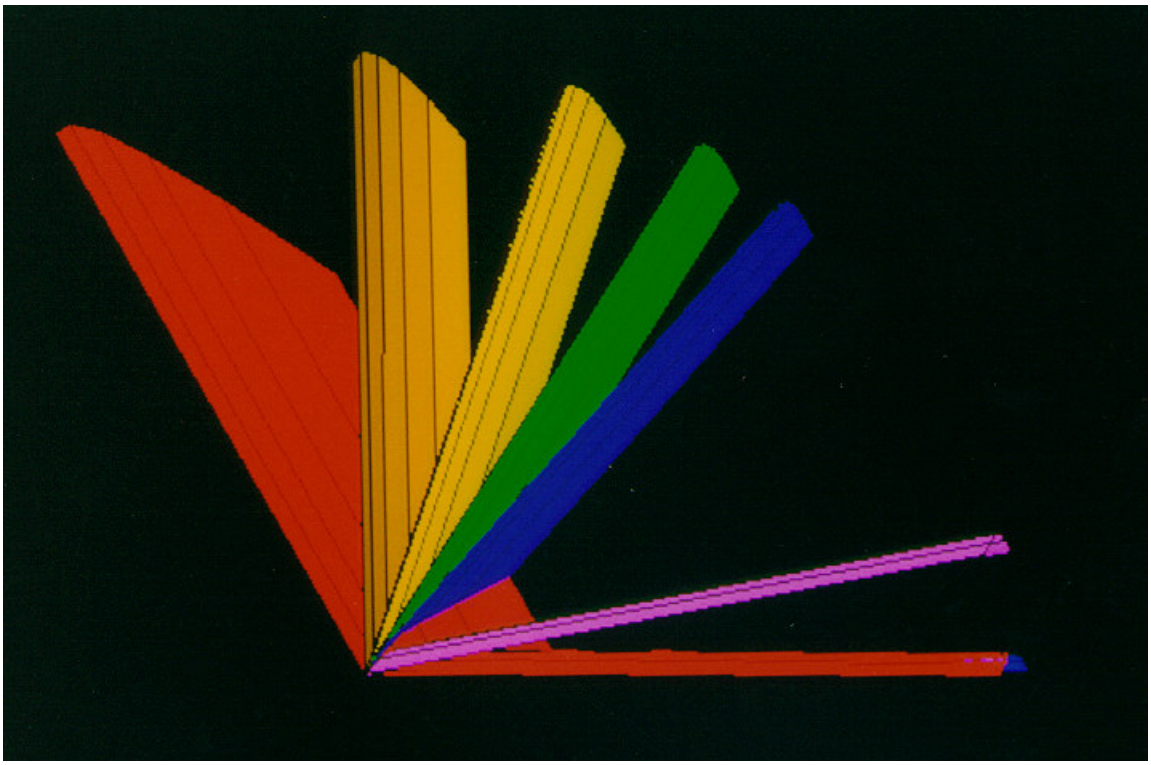


Fig. 5.7 Cylindrical folded developable. Normal Radius of Curvature Fixed at 0.5. Geodesic Radius of curvature ranges from 0.1 to 10.0.

Plane developments with folding curves shown. In all cases $\rho_N = 0.5$

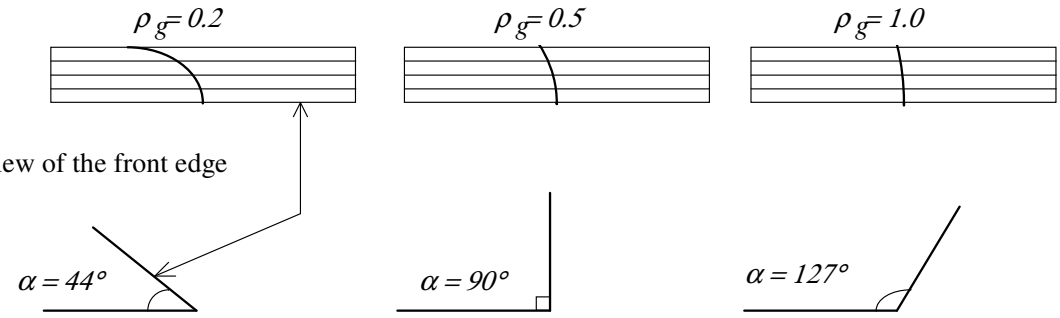


Fig. 5.8 Detail from Fig. 5.7, the effect of increasing the folding curve radius. The changed folding curves results in three different first surfaces.

From Fig. 5.6 and Fig 5.7 it is clear that the 3FD program is accurately modelling curved line folding.

The results match the theory, as the geodesic radius of curvature of the folding curve increases, the folding curve approaches a straight line and the fold angle approaches zero. If the folding curve is straight the radius of curvature is infinite and the fold angle will be zero. If the fold angle is zero the developable folds back on itself.

The results show that the folded developable acts as a predictable, non linear mechanism. For a known change in the radius of the fold curve a predictable movement of the generators occurs.

5.2.5. The Fold Angle Mechanism.

A folded developables' fold angle can be changed by applying a force to change the curvature of its first surface. The change in the fold angle causes the folded developable to act as a mechanism, translating a force to a predictable motion.

The mechanism may be actuated by 'squeezing' the edges of a developable at the folding curve. Assuming the generators remain fixed to the surface, this action reduces the distance, *D*, which in turn reduces the normal radius of curvature of the surface, as shown in Fig. 5.9.

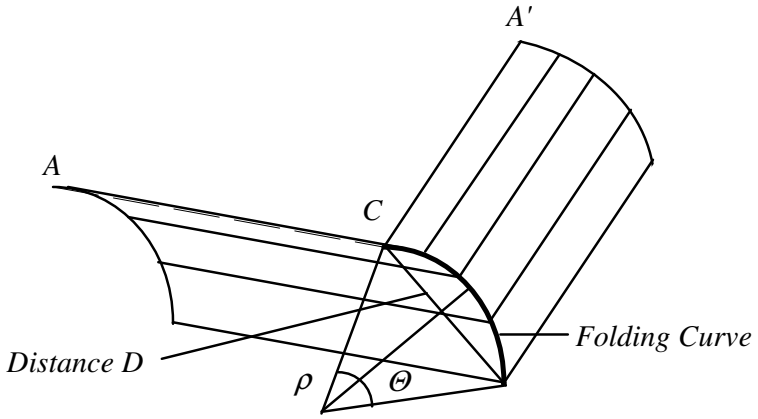


Fig 5.9 Relationship between the radius of curvature and the distance *D*.

The mechanism may be described by the response of the fold angle to changes in *D*. The distance *D* is related to the normal radius of curvature by equation (5.6),

$$D = 2\rho_N \sin\left(\frac{C}{2\rho_N}\right), \tag{5.6}$$

where *C* is the folding curve chord length.

The response of such a mechanism with a folding curve *C* length of p/2 is shown in Fig. 5.10.

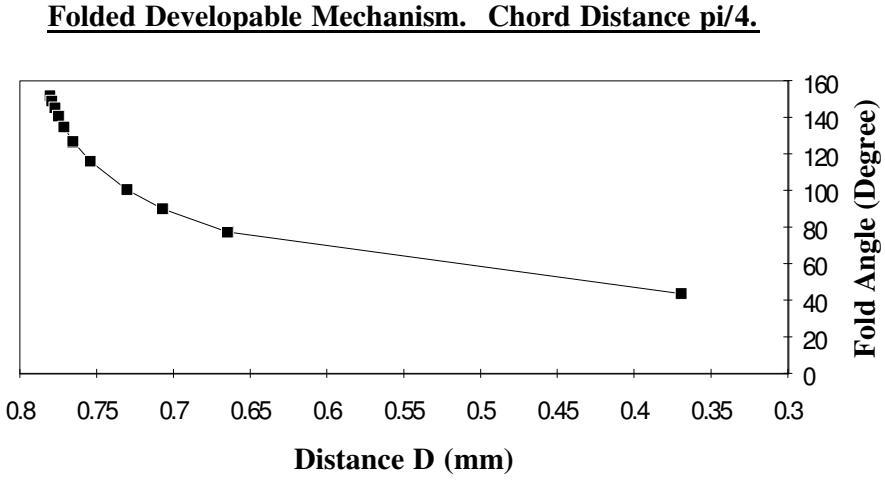


Fig 5.10 The folded developable mechanism.

5.3. THE EFFECTS OF THE PRINCIPAL CURVATURE OF THE FIRST SURFACE ON THE INCLINATION OF THE SECOND SURFACE GENERATOR.

The angle, γ_2, that a generator in the second surface makes with the tangent of the fold curve can be determined by equation (5.7), developed in section 3.4.4,

$$\tan \gamma_2 = \frac{-K_1 \sin^2 \gamma_1}{K_1 \cos \gamma_1 \sin \gamma_1 + 2\frac{d\beta}{ds}} \tag{5.7}$$

where γ_1 is the initial surface generator tangent angle, K_1 is the curvature of the initial surface and $\frac{d\beta}{ds}$ is a measure of the rotation rate of the osculating plane of the folding curve C. These three parameters determine the angle the second generator makes with the tangent.

If the folding curve has a constant fold angle $\frac{d\beta}{ds} = 0$ then $\gamma_2 = -\gamma_1$ unless $\pi < \gamma_1 < \frac{3}{2}\pi$.

Kinematically γ_2 is affected by changes in K_1 only if $\frac{d\beta}{ds}$ is not zero. Thus γ_2 changes only if the fold angle changes along the length of the fold curve.

If the change in the fold angle is negative, $\frac{d\beta}{ds} < 0$ then the second generator angle is of greater magnitude than the first, $\gamma_2 > \gamma_1$:

The effects of changing the three parameters are shown in the numerical examples in the following section.

5.3.1. Numerical Examples.

1. No change in fold angle (no osculating plane rotation).

   if $\gamma_1 = 60°$

   $\frac{d\beta}{ds} = 0.0$

   and $K_1 = 0.5$

   then $\gamma_2 = \arctan\left(\frac{-0.5\sin^2 60°}{0.5\cos 60° \sin 60° + 2\times 0.0}\right)$

   $\gamma_2 = -60°$ or $120°$.

The arctan solution produces two roots. In the computations only the positive root is used. As predicted if there is no change in fold angle, the two tangent angles are equal and opposite in sign.

2. Positive change in the fold angle (negative osculating plane rotation).

If $\gamma_1 = 60°$

$$\frac{d\beta}{ds} = -0.1$$

and $K_1 = 0.6$

then $$\gamma_2 = \arctan\left(\frac{-0.6\sin^2 60°}{0.6\cos 60°\sin 60° + 2\times -0.1}\right)$$

$$\gamma_2 = -82.4° \text{ or } 97.6°,$$

If the fold angle increases along the folding curve, the second tangent angle will be of greater magnitude than the first but of opposite sign.

3. Negative change in the fold angle (positive osculating plane rotation).

If $\gamma_1 = 60°$

$$\frac{d\beta}{ds} = 0.1$$

and $K_1 = 0.6$

then $\gamma_2 = -44.4°$ or $135.6°$.

If the fold angle decreases along the folding curve, the second tangent angle will be of lesser magnitude than the first but of opposite sign.

5.3.2. Computational Example.

If the first surface is part of a cone and the folding curve's centre is not at the apex of the cone, then the folding curve intersects the surface's generators at different distances from the apex of the cone. This results in different curvatures occurring at the folding curve, the fold angle is generally not constant, the fold curve is twisted and $\frac{d\beta}{ds}$ does not equal zero.

This is illustrated by Fig. 5.11 (a) (b) and (c) below. The three figures are three views of two developables. Both have identical first surfaces but one has a twisted folding curve, while the other does not. The comparison with a physical model of this process is illustrated by example 5 in the results section of chapter 3.

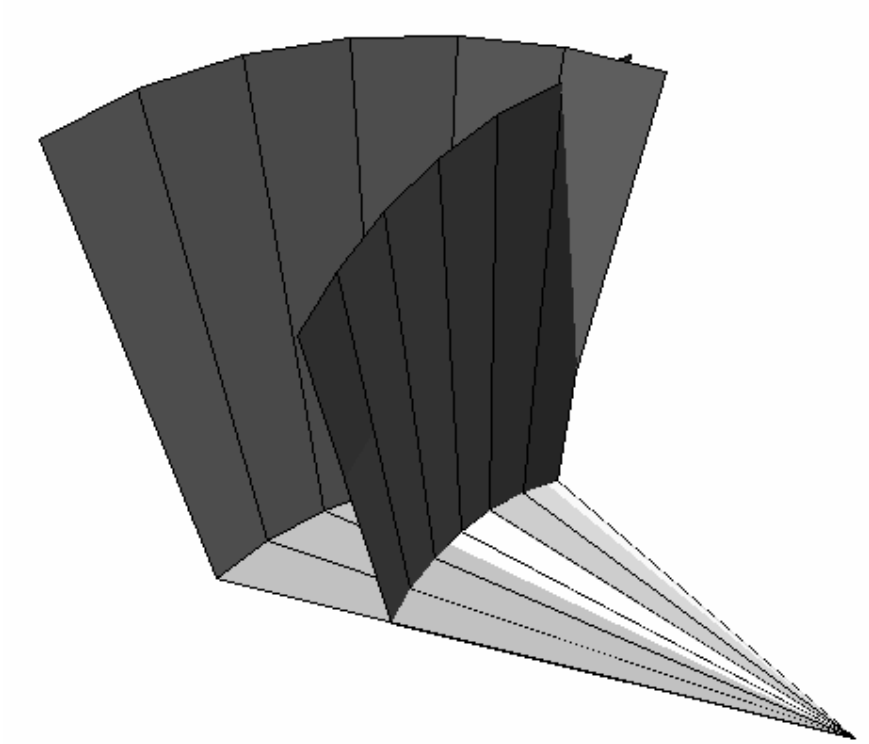


Fig. 5.11 (a) Cone developable. Two identical initial surfaces folded differently. For one surface the fold angle is constant because the normal radius of curvature does not change along the fold line. For the other the fold angle varies so the angles the generators make with the folding curve vary. In each case the normal radius of curvature is 0.9 at the base of the cone and the folding curve radius is 0.5.

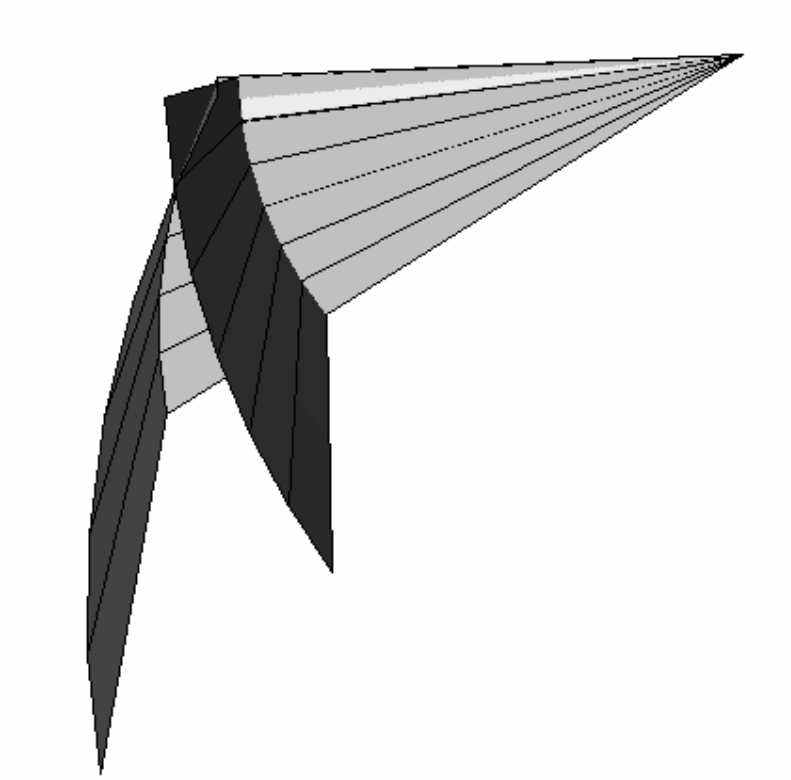


Fig. 5.11 (b) Cone developable. The same developable in a plane view.

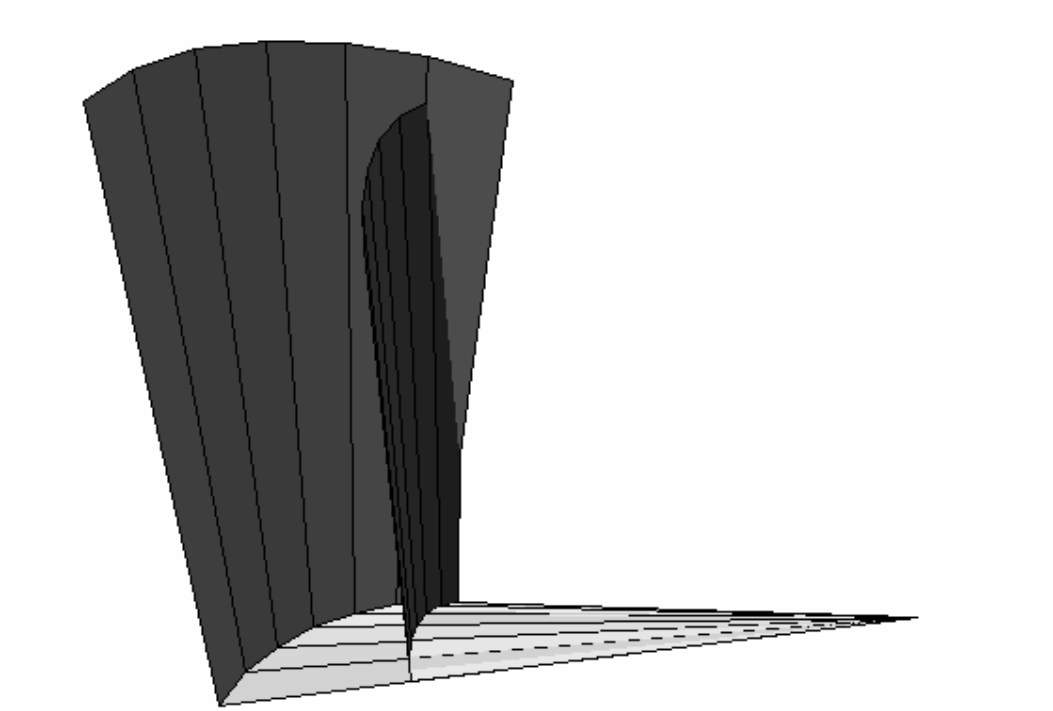


Fig. 5.11 (c) Cone developable. The same developable side view

The effects of the twisted folding curve, on the fold angle and the tangent angles are numerically quantified in Fig. 5.12.

Generator Number	Plane α	Twisted α	Plane γ_1	Twisted γ_1	Plane γ_2	Twisted γ_2
1	96.74	78.73	86.88	135.07	86.88	29.24
2	96.74	84.13	85.63	132.79	85.46	33.02
3	96.74	89.20	84.40	130.18	83.88	37.49
4	96.74	93.80	83.20	127.29	82.33	42.15
5	96.74	97.93	82.02	124.17	80.80	46.91
6	96.74	101.60	80.88	120.85	79.30	51.69
7	96.74	104.80	79.80	117.39	77.86	56.44

Fig. 5.12 A comparison of fold and tangent angles for a developable with (i) a plane folding curve and (ii) a twisted folding curve.

The developable with the plane folding curve has a constant fold angle. This is because the curvature of the first surface is equal at all, the generator intersections with the folding curve. If there is no change in the fold angle, from equation 5.7, the tangent angles will be equal. The tangent angles, in Fig 5.12, are close to equal, the slight increasing error due to rounding errors in the calculations (Max error less than 2.5%).

A twisted folding curve causes a changing fold angle. This is due to a variation in the curvature of the first surface at the intersections of the folding curve with the generators. From equation 5.3, this change in fold angle causes a change in fold angle. If the fold angle changes along the folding curve, from equation 5.7, the tangent angles will not be equal. This is shown in Fig. 5.12.

5.4. CONCLUSIONS.

The conclusions drawn from the modelling of the kinematics of folded developables are:

If the folding curve has a fixed radius of curvature in the plane projection, ρ_g, then as the radius of curvature of the initial surface increases the fold angle increases and the folded developable opens. Conversely if the folding curve has a fixed radius and the radius of curvature of the initial surface decreases the fold angle reduces and the folded developable closes.

If a developable surface, with a fixed principal surface curvature (fixed radii of curvature) is folded along a curve, that is part of the arc of a circle in its plane projection, the tangent of the fold angle is inversely dependent on the radius of the folding curve. As the plane projection of the radius increases the fold angle reduces. The relationship is not linear and is described by equation 5.3.

Folded developables act as predictable, non linear mechanisms. For a known change in the radius of curvature of the surface a predictable movement of the generators occurs.

The mechanism may be described by considering a distance , D, where D is the chord distance of the folding curve. Reducing the distance, D, reduces the normal radius of curvature of the surface and this causes a change in the fold angle that results in a change in generator position. The description for this motion is shown in Fig 5.6.

The angle, γ_2, that a second surface generator makes to the tangent of the folding curve is governed by three parameters. The first is the angle, γ_1, the corresponding generator in the first surface makes with the tangent curve. The second, K_1, is the principal curvature of the first surface. The third, $\frac{d\beta}{ds}$, is the rate of rotation of the osculating plane along the folding curve. This is inversely related to the rate of change of the fold angle along the folding curve. The exact relation is described in equation 5.7.

If there is no change in fold angle, the two tangent angles are equal and opposite in sign. If the fold angle increases along the folding curve, the second tangent angle will be of greater magnitude than the first but of opposite sign. If the fold angle decreases along the folding curve, the second tangent angle will be of lesser magnitude than the first but of opposite sign.

6 REFERENCES.

Arlinghaus F.J.et al. (1985), "Finite Element Modelling of a Stretch Formed Part", Computer Modelling of Sheet Metal Forming Processes", Edited by N.M.Wang and S.C.Tang, p65, Metallurgical Society Inc. 1985.

Aumann G. (1991), "Interpolation with Developable Bezier Patches", Computer Aided Geometric Design, Vol 8, pp409-420, Elsevier Science Publishers, July 1991.

Berry D.T. (1987), "Beyond Buckling - A Nonlinear FE Analysis", Mechanical Engineering, pp40-45, August 1987.

Berry D.T. (1988), "Stamping Out Forming Problems With FEA", Mechanical Engineering,, pp58-62, July 1988.

Bussler M.L. and Paulsen W.C. (1988), "Is FEA Faster On a PC?", Mechanical Engineering, pp64-67, July 1988

Chu E., Soper D. et al. (1985),"Computer Aided Geometric Simulation of Sheet Metal Forming Processes", Computer Modelling of Sheet Metal Forming Processes", Edited by N.M.Wang and S.C.Tang, pp65-76, Metallurgical Society Inc, 1985.

Clifford G. (1993), "System Simulation Cuts Time, Manufacturing Costs", Design News, Vol 49, #3, pp55-56, Cahners Publication Ltd., January 1993.

Clough R.W. (1990), "Original Formulation of the Finite Element Method", Finite Elements in Analysis and Design, Vol 7, pp 89-101, Elsevier Science Publishers. 1990.

Duncan J.L. and Altan T. (1980), "New Directions in Sheet Metal Forming Research", Annals of the CIRP, p71, 1980.

Duncan J.L. and Sowerby R. (1981), "Computer Aids in Sheet Metal Engineering", Annals of the CIRP, pp541-546, February 1981.

Duncan J.P.and Duncan J.L. (1982), "Folded Developables", Proc. R. Soc. Lond., A 383, p191-205, 1982

Duncan J.L. and Sowerby R. (1987), "Review of Practical Modelling Methods for Sheet Metal Forming", Proc. 2nd International Conference on Technological Plasticity, pp615-624, August 1987.

Furubayashi T. et al. (1985),"The Simulation of Forming Severity on Autobody Panels using a CAD system - Behaviour of Materials After Passing the Drawbead", Computer

Modelling of Sheet Metal Forming Processes", Edited by N.M.Wang and S.C.Tang, pp363-372, Metallurgical Society Inc, 1985.

Gloeckl H. and Lange K. (1983), "Computer Aided Design of Blanks for Deep-Drawn Irregular Shaped Components", Proceedings 11th N.A.M.R.I. Conference, pp 243-251, 1983.

Hatt F. (1993), "Simulating Stamping", Computer Aided Design Report, Vol 13, # 2, pp 11-12, CAD/CAM Publishing Ltd., February 1993.

Hibbitt, Karlsson and Sorensen (1989), "ABAQUS Theory", ABAQUS Documentation Version 4-6, 2.0, Hibbitt, Karlsson and Sorensen Inc. 1989

Hibbitt, Karlsson and Sorensen (1989 II),"Stretching of a Thin Sheet with a Hemispherical Punch", ABAQUS Documentation Version 4-6, 4.2.14-1. Hibbitt, Karlsson and Sorensen Inc. 1989.

Hyllander A. (1991), Personal Communication. 1991.

James G. and James R. (1959), Mathematics Dictionary, p111, D Van Nostrand Comp., Princeton. 1959.

Johnson W., Sowerby R. and Venter R.D. (1982), Plane Strain Slip-line Fields for Metal Deformation Processes, pp 50 -53, Pergamon Press, 1982.

Karima M. (1989), "A Methodology for Computer Aided Stamping Engineering", Steel Stamping Technology: Applications and Impact, SP-779, p75, Society of Automotive Engineers Inc., Feb 1989.

Keeler S. and Backofen W. (1963), Trans. ASME, 56, p25, 1963.

Keeler S. and Stine P. (1989), "Simulating the Sheet Metal Forming Process- An Optimization Exercise on the PC or Engineering Work Station", Steel Stamping Technology: Applications and Impact, SP-779, p25, Society of Automotive Engineers Inc., Feb 1989.

Kergosein , Gotoda and Kunii (1994), "Bending and Creasing Virtual Paper", IEEE Computer Graphics and Applications, p40, IEEE, January 1994.

Kokkonen V. (1985), "Modelling of Forming Processes for Tool Design and Manufacturing at Volvo", Computer Modelling of Sheet Metal Forming Processes", Edited by N.M.Wang and S.C.Tang, pp13-20, Metallurgical Society Inc, 1985.

Kreyszig E. (1983), Advanced Engineering Mathematics, Section 8.5, John Wiley and Sons Inc. 1983.

Lange K. (1985), Handbook of Metal Forming, Section 3.7, McGraw-Hill, 1985.

Letcher J., Brown J. and Stanley E. (1988), "Developable Surfaces in the Fairline System", RINA Small Craft Group International Conference on Computer Aided Design of Small Craft, Sail and Power, Vol 1, May 1988.

MacNeal R.H. and Harder R.H. (1985), "Proposed Standard Set of Problems to Test Finite Element Accuracy", Finite Elements in Analysis and Design, 1 1985 pp 3-20, North-Holland. 1985.

Marcimak Z. and Duncan J.L (1992), Mechanics of Sheet Metal Forming, Edward Arnold, UK, 1992.

Molian S. (1982), Mechanism Design, pp1-15, Cambridge University Press, 1982.

Redont P. (1989), "Representation and deformation of Developable Surfaces", Computer Aided Design, Vol 21 No. 1, pp13-20, Butterworth & Co, January/February 1989.

Sowerby R., Duncan J.L. and Chu E. (1986), "The Modelling of Sheet Metal Stampings", International Journal of Mechanical Science, Vol 28, No.7, pp415-430, 1986.

Staublin D. and Gerdeen J. (1986), "Springback of Doubly Curved Developable Sheet Metal Surfaces with Plane Bend Lines and Constant Dihedral Bend Angles", Proceedings of Near Net Shape Manufacturing Conference, ASM International, 1986.

Templer R. (1987), "Computer Aided Modelling of Sheet Metal Deformation", Mechanical Engineering Report, PME 87.53, pp6-17, 1987.

Tuttle S. (1982), Mechanisms for Engineering Design, pp126-134, Wiley and Sons, 1967.

Vickers G., Bedi S., Blake D. and Dark D. (1987), "Computer-Aided Lofting, Fairing and Manufacturing in Small Shipyards", The International Journal of Advanced Manufacturing Technology, Vol 2 No. 4, pp79-90, IFS (Publications) Ltd, 1987.

Weiß G. and Furtner P. (1988), "Computer Aided Treatment of Developable Surfaces", Computers and Graphics, Vol 12 No. 1, pp39-51, Pergamon, 1988.

Willmore, T.J. (1959), An Introduction to Differential Geometry, pp95-119, Oxford: Clarendon Press. 1959.

Wood R.D. (1981), "The Finite Element Method and Sheet Metal Forming", Sheet Metal Industries, pp561-567, August 1981.

Yamasaki H., Nishiyama T. and Tamura K. (1985), "Computer Aided Evaluation Method for Sheet Metal Forming in Car Body", Computer Modelling of Sheet Metal Forming Processes", Edited by N.M.Wang and S.C.Tang, pp373-382, Metallurgical Society Inc, 1985.

Zhang Z. (1993), "Computer Method of Die Design for Sheet Metal Forming", PhD Thesis, pp79-94, Mechanical Engineering Department, University of Auckland, 1993.

7. APPENDICES

7.1. LARGE STRAIN ANALYSIS

In Fig. 7.1 the homogeneous deformation of the square *OABC* into the deformed rhomboid *O'A'B'C'* is shown.

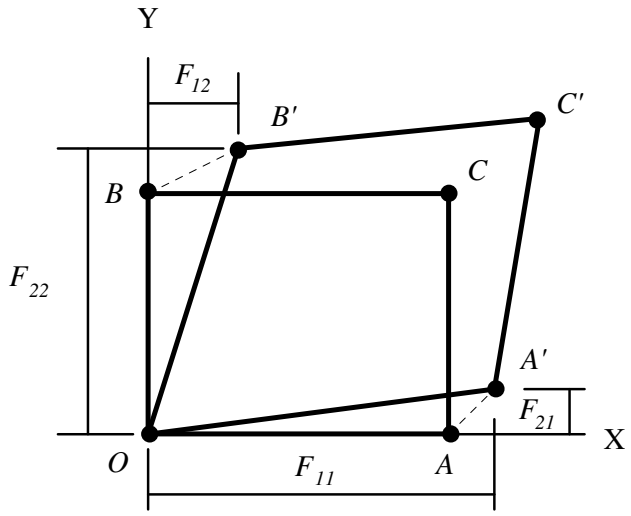


Fig. 7.1 Homogeneous deformation of a rhomboid.

The transformation can be mathematically described as

$$\begin{aligned} x &= F_{11}X_0 + F_{12}Y_0 \\ y &= F_{12}X_0 + F_{22}Y_0 \end{aligned} \tag{7.1}$$

where X and Y are the initial coordinates and x and y the transformed coordinates. The coefficients F_{ij} can be evaluated from three points after deformation, typically from *O', A'* and *C'*. Equation 7.1 can be written in matrix form

$$\begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & 0 & 0 \\ X_2 & Y_2 & 0 & 0 \\ 0 & 0 & X_1 & Y_1 \\ 0 & 0 & X_2 & Y_2 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{21} \\ F_{22} \end{bmatrix} \tag{7.2}$$

or in tensor notation

$$\mathbf{x} = \mathbf{F}.\mathbf{X} \tag{7.3}$$

where **x** represents the deformed state and **X** the initial state. **F** is the deformation gradient tensor with coefficients **Fij.**

$$\mathbf{F} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \tag{7.4}$$

F is usually a non-symmetric tensor, $F_{12} \neq F_{21}$. It is always possible to find a positive symmetric tensor **U**, $U_{12} = U_{21}$, which produces the same shape change as **F** and which is related to **F** by

$$\mathbf{F} = \mathbf{R}.\mathbf{U} \tag{7.5}$$

where **R** represents rigid body rotation. The symmetric tensor **U** produces symmetric deformation of the original square as shown in the bottom diagram of Fig. 7.1. It must be rotated by **R** in order to reach the configuration of *O'A'B'C'*.

Thus it can be concluded that **F** can be divided into a pure deformation **U** and a pure rotation **R**. In most realistic forming processes the full transformation **F** through the simultaneous action of rigid body rotation and pure deformation. This means the principal directions rotate during forming and the strain path becomes coaxial in the general case. This is different from the idealised situation where the pure deformation and rigid rotation are consecutive events. In that case the principal directions are fixed and the strain path is both straight and coaxial. In this case of 'pure homogeneous deformation' U alone deforms the initial square.

The inversion of the matrix in equation 7.3 will give the values of the four **F** coefficients

$$\begin{bmatrix} F_{11} \\ F_{12} \\ F_{21} \\ F_{22} \end{bmatrix} = \mathbf{D} \begin{bmatrix} Y_2 & -Y_1 & 0 & 0 \\ -X_2 & X_1 & 0 & 0 \\ 0 & 0 & Y_2 & -Y_1 \\ 0 & 0 & -X_2 & X_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} \tag{7.6}$$

where

$$\mathbf{D} = \frac{\mathbf{1}}{\left(X_1 Y_2 - X_2 Y_1\right)} \tag{7.7}$$

The deformation tensor derived from equation 7.14 is unsymmetrical, so a symmetric second order tensor, known as the Cauchy-Green deformation tensor, **C**, can be evaluated from

$$\mathbf{C} = \mathbf{F}^{\mathbf{T}}.\mathbf{F} \tag{7.8}$$

where $\mathbf{F}^{\mathbf{T}}$ is the transpose of **F**.

Expanding Equation 7.8, the components of **C** are

$$\begin{aligned} C_{11} &= F_{11}^2 + F_{21}^2 \\ C_{12} &= C_{21} = F_{11}F_{12} + F_{21}F_{22} \\ C_{22} &= F_{12}^2 + F_{22}^2 \end{aligned} \tag{7.9}$$

The principal values (eigenvalues) of **C** can be obtained by the transformation rule of a second order tensor. Alternatively a Mohr's circle construction could be employed. In this case the principal values are the elongation values squared, ie. λ_{11}^2 and λ_{22}^2,where the elongation ratio is the ratio of final line length to initial line length. The principal values obtained from equation 7.9, are given by

$$\begin{aligned} \lambda_{11}^2 &= \frac{C_{11}+C_{22}}{2} + \sqrt{\left(\frac{C_{11}-C_{22}}{2}\right)^2 + C_{12}^2} \\ \lambda_{22}^2 &= \frac{C_{11}+C_{22}}{2} - \sqrt{\left(\frac{C_{11}-C_{22}}{2}\right)^2 + C_{12}^2} \end{aligned} \tag{7.10}$$

The orientation of the principal axes is determined from

$$\tan 2\theta = \frac{2C_{12}}{C_{11}-C_{22}} \tag{7.11}$$

It follows that the principal logarithmic surface strains are

$$\begin{aligned} \varepsilon_{11} &= \ln(\lambda_{11}) \\ \varepsilon_{22} &= \ln(\lambda_{22}) \end{aligned}. \tag{7.12}$$

The third principal logarithmic strain, the thickness strain, may be determined using the incompressibility assumption

$$\varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33} = 0 \tag{7.13}$$

or

$$\lambda_{11}\lambda_{22}\lambda_{33} = 1 \tag{7.14}$$

The representative or equivalent strain $\bar{\varepsilon}$ is given by

$$\bar{\varepsilon} = \sqrt{\frac{2}{3}\left(\varepsilon_{11}^2 + \varepsilon_{22}^2 + \varepsilon_{33}^2\right)}. \tag{7.15}$$

7.2. 3FD COMPUTER PROGRAM SUBROUTINE AND ALGORITHM DETAILS.

The previous section described the functions performed by the 3FD program. 3FD is a FORTRAN program, and uses PHIGS (Programmers Hierarchal Interactive Graphics System) to provide the Graphical User Interface, (GUI). Although developed on an IBM RS6000 computer with IBM's implementation, graPHIGS, of PHIGS it can be transferred to other computers and implementations of PHIGS with few modifications.

This appendix describes in greater detail the structure of the program, the menus' structure, the algorithms behind the major subroutines and some detail of the graphics routines used to create the user interface.

7.2.1. Overall Program Sequence.

The sequence of the 3FD program is shown in Fig. 3.38. When the program is started it initialises the GUI, displays the title and introduction sequence, then moves into an event driven program loop. The program waits for the user to select one of the menu items and then executes the corresponding subroutines. This continues until the user decides to terminate the programs operation. The program then shuts down the GUI and terminates. This structure is shown in Fig. 7.2.

Program Start.
Initialise Graphics.
Display Title Graphics and Introduction.
Start Menu Driven Loop.
    Await Menu Item Selection
    Read Menu Item Selection
    Call Appropriate Subroutines
    Repeat unit 'Quit' Selected
Shut Down Graphics.
Terminate Program.

Fig. 7.2 Sequence of events controlled by the program 3FD. The event driven loop is controlled by the subroutine PHLOOP.

The general structure of the program PHLOOP and the data flows between the major modules is shown in Fig. 7.3.

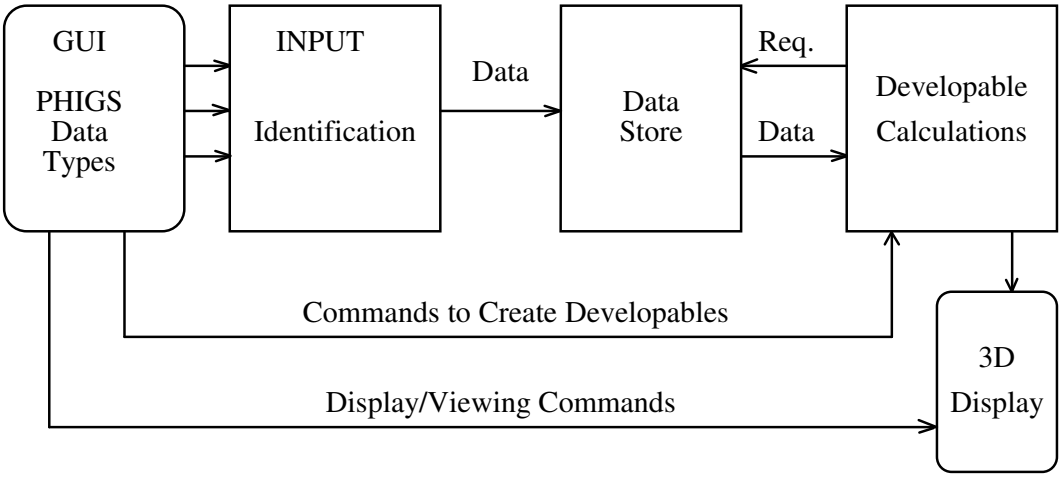


Fig 7.3 General Structure of the program and the data flows between major modules.

The details of the GUI and 3D display modules is shown in Fig. 7.4

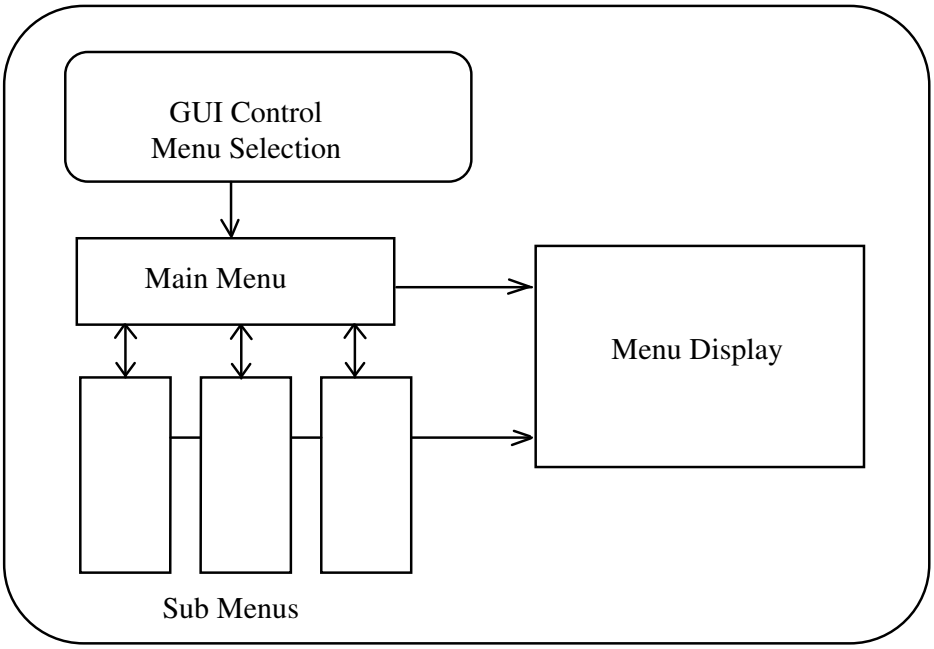


Fig 7.4 (a) Expansion of 'GUI' in Fig 7.3

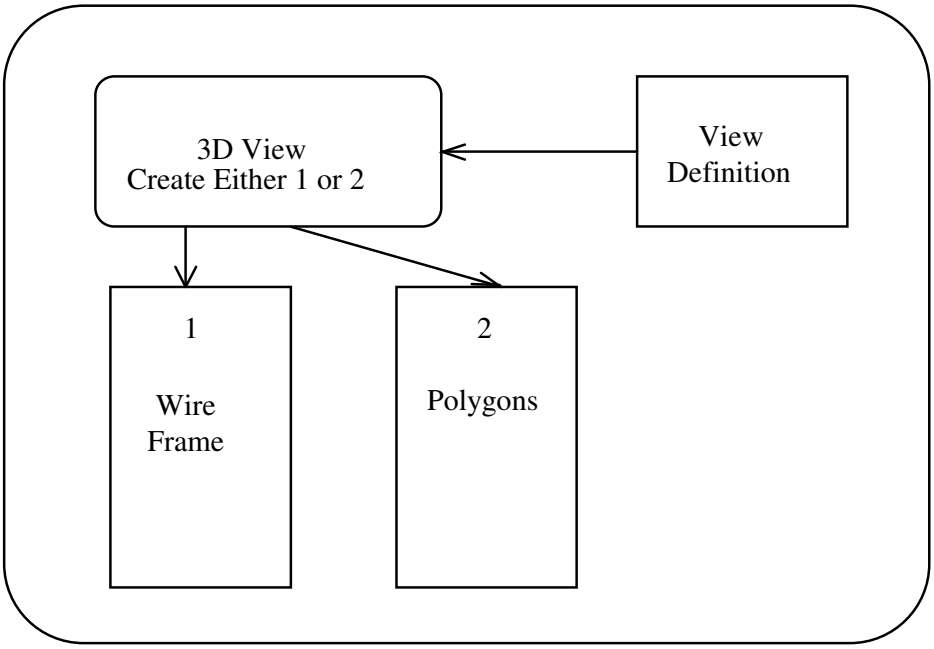


Fig 7.4 (b) Expansion of '3D Display' in Fig 7.3

The user interfaces with the 3FD program through the 'pop-up' menus. The main menu of the program is displayed at the bottom of the graphics screen. The user selects one of the items of the main menu by moving the cursor over the menu name and clicking a cursor button. A secondary menu then appears on the screen. In a similar fashion the user may select an item from the secondary menus. The program then calls the appropriate subroutines. The menus are designed to allow the user to intuitively design a folded developable and are structured as shown in Fig. 7.5.

The major subroutines used in 3FD are described in the following sections.

Set View	**Edge Reg**	**Rad Curve**	**Fold**	**Quit**
Main View	Conic	Cone Fixed	Axis Arc	
Right Elevation	Cylinder	Bezier	Centre Arc	
Left Elevation	Function	Normal Fixed	Bezier	
Main Elevation	Bezier		Flip	
Plan			Flatwrap	
Shade			Dieset	
Wire Frame			Dieflat	
Plot				

Fig. 7.5 Menu Tree for program 3FD.

7.2.2. The INPUT Subroutine.

The input subroutine is an interrupt driven subroutine. Upon interruption (some action by the user such as a menu pick, keyboard entry or dial movement) the subroutine determines the type of data that has been received and calls the relevant subroutine. The data type is determined by the data input class. Classes primarily relate to the device used for input, for example a string type input is data entered by the keyboard. The structure of INPUT is described by the Pseudo Code in Fig. 7.6.

```
Subroutine Input
    Await Input
    When Input Occurs Determine Type of Input
    If Input = Locator (Indicate)
        Call GPGTLC (Get Locater)
    else if Input = Pick (Select)
        Call GPGTPK (Get Pick)
    else if Input = Valuator (Dials)
        Call VALIN (Read Dials)
    else if Input = Choice (Function Keys)
        Call CHOIN (Which Key)
    else if Input = String (Keyboard)
        Call GPGTST (Get String)
    Endif
    RETURN
```

Fig. 7.6 The INPUT Subroutine which accepts data from the GUI and then passes it on to the processing subroutines.

7.2.3. The Set View Subroutine.

Changing the views of the Folded Developable is controlled by the OPTNA subroutine. This accepts the menu selection of the user and displays the corresponding view of the folded developable. The view is changed by moving the view port around the three dimensional object using the conventions shown in Fig. 7.7.

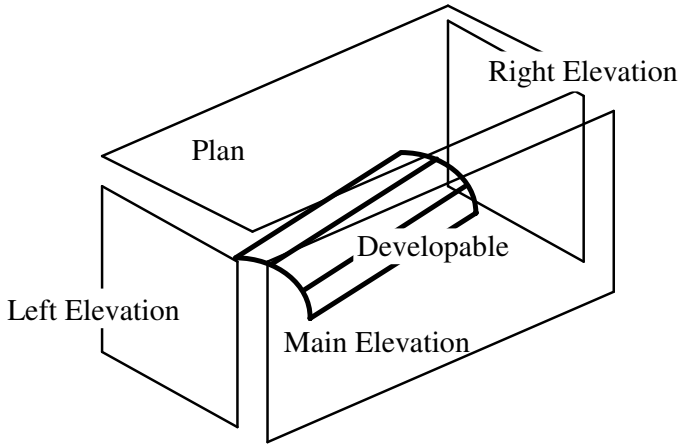


Fig. 7.7 Viewport Definitions

7.2.4. The Edge of Regression Subroutines.

As described in section 3.4.2.2 there are four possible methods of defining the edge of regression and hence the plane development of the first surface. The algorithms for the creation of a flat cone surface and for a surface defined by an edge of a regression defined by the function y = cos x -1 are shown in Figures 7.8 and 7.9.

*Subroutine AIN (*Arc Input*)*

Prompt user to INPUT number of Generators.
NP = INPUT
i = -1
Start Generator Creation Loop
A(i,x) = -0.5
A(i,y) = 0.0
A(i,z) = 0.0
*Theta = i * 5°*
B(i,x) = A(i,x) + cos (Theta)
B(i,y) = A(i,y) - sin (Theta)
B(i,z) = 0.0
i = i+1
IF i < NP + 1 go to start of loop
RETURN

Fig. 7.8 The AIN Subroutine which creates the plane development of the first surface of a cone developable.

The loop starts at -1 and continues to the number of generators +1 to create all the generators used in the folding calculations.

The function edge of regression is created in the subroutine INPUTE described in Fig. 7.9.

```
Subroutine INPUTE
    Prompt user to INPUT number of Generators.
    NP = INPUT
    i = -1
    Start Generator Creation Loop
        Ang = i*5°
        A(i,x) = Ang-0.5
        A(i,y) = Cos(Ang) - 1.0
        A(i,z) = 0.0
        dy/dx = -sin(Ang)
        Theta = Arctan(dy/dx)
        B(i,x) = A(i,x) + cos (Theta)
        B(i,y) = A(i,y) + sin (Theta)
        B(i,z) = 0.0
        i = i+1
    IF i < NP + 1 go to start of loop
    RETURN
```

Fig 7.9 The subroutine INPUTE that creates the plane development of the first surface of a developable with the edge of regression defined by a function.

7.2.5. The Radius of Curvature Subroutines.

As described in section 3.4.3.3 there are three possible methods of defining the radius of curvature of the first developable surface; cone, normal and bezier. The algorithm for a fixed normal radius of curvature is shown in Fig 7.10.

```
Subroutine FNIN (Fixed Normal In)
    INPUT Radius of Curvature
    RHO = INPUT
    i = -1
    Start of the Curvature Addition Loop.
        r = sqrt ((RHO*(RHO+1))/(RHO+1))
        L = Generator Length
        theta = i*5°
        alpha = theta/r
```

```
        beta   = arcsin(r/L)
        XX = sqrt (L*L - r*r)
        YY = -rsin(alpha)
        ZZ =  rcos(alpha)
        B(i,x) = XXcos(beta) + ZZsin(beta) -0.5
        B(i,y) = YY
        B(i,z) = XXsin(beta) - ZZcos(beta)
        i = i+1
IF i < NP + 1 go to start of loop
RETURN
```

Fig 7.10 The subroutine FINN that defines fixed normal curvature.

7.2.6. The Folding Subroutines.

As described in section 3.4.2.4 there are three methods of defining the folding curve. The example algorithm, Fig 7.11, is for a bezier curve folding curve. After the folding curve has been defined in the plane development, the surface is then folded. This algorithm, Fig. 7.12, follows the theory developed in Section 3.4.

The Bezier fold curve is created by the user inputting the four control points for the bezier curve. The first and the fourth points define the start and end of the bezier curve, the second and third points define the direction and magnitude of the tangents at the start and end of the curve. The curve is then drawn and intersected with the generators. At each intersection point, the radius of curvature of the folding curve is determined. This information is then passed to the Folding Subroutine CURV2.

Subroutine Bezcurv

Control Point Input

User to INPUT Four Control Points

u(1) to u(4) & v(1) to v(4) = INPUT

t = 0

Start Bezier Curve Creation Loop

*i = t*50 + 1*

$XX(i) = (1-t)^3u(1) + 3t(1-t)^2u(2) + 3t^2(1-t)u(3) + t^3u(4)$

$YY(i) = (1-t)^3v(1) + 3t(1-t)^2v(2) + 3t^2(1-t)v(3) + t^3v(4)$

t = t + 0.02

If t < 1.0 go to start of loop

Draw Curve

Move_to (XX(1),YY(1))

i = 2

Start Draw Loop

Draw_Line_to(XX(i),YY(ii))

i=i+1

If i < 50 Go to loop start

Find Generator Intersections

A x,y,z = Generator Start Point

B x,y,z = Generator End Point

C x,y = Plane Generator Fold Point

For Each Generator

beta = arctan((B(y) - A(y))/(B(x) - A(x))

gap = 10 000

i = 1

Start Intersection Gap Loop

xgen = XX(i)

ygen = A(y) + (xgen - A(x))sin(beta)

gen_gap = YY(i) - ygen

*diff = sqrt(gen_gap*gen_gap)*

IF diff < gap then

gap = diff

istore = i

ENDIF

If i < 50 go to loop start
C(x) = XX(istore)
C(y) = YY(istore)

Calculating Radius of Curvature.
The Radius of Curvature is calculated by finding the intersections of the normals of the bezier curve. The normals originate from a point on the folding curve C_i and a point either side of it, C_{i-1} and C_{i+1}.

$m_1 = (C(i,y) - C(i-1,y))/(C(i,x) - C(i-1,x))$
$D_1 = C(i,y)/(m_1*C(i,x))$
$m_2 = (C(i+1,y) - C(i,y))/(C(i+1,x) - C(i,x))$
$D_2 = C(i,y)/(m_2*C(i,x))$

Intersection Point O(x),O(y)
$O(x) = (D_2 - D_1)/(m_1-m_2)$
$O(y) = m_1O(x) + D_1$

Radius of Curvature RHO
$RHO(i) = sqrt([O(x)-C(i,x)]^2 + [O(y)-C(i,y)]^2)$
RETURN

Fig. 7.11 Detail of the subroutine BEZCURV used to create a bezier curve folding curve.

The subroutine that folds the developable surface along the specified folding curve is CURV2. The subroutine follows the numerical algorithm described in section 3.4. The structure of the subroutine is shown in Fig. 7.12.

Subroutine CURV2
NP = Number of Generators
i = -1
Start Generator Folding Loop
Determine Tangent to Folding Curve
Determine the tangent Angle Gamma1

$$\gamma_1 = \arccos\left[\frac{\left(A_1C_1^2 + C_1C_2^2 - A_1C_2^2\right)}{\left(2 * A_1C_1 * C_1C_2\right)}\right]$$

Determine the fold angle Beta

$$\beta = \arctan\left(\frac{\rho_i}{r_i}\right)$$

Determine the rate of Change of Fold Angle

$$\frac{d\beta}{ds} = \frac{\beta_i - \beta_{i-1}}{C_1 C_2}$$

Determine Gamma 2

$$\tan \gamma_2 = \frac{-K_1 \sin^2 \gamma_1}{K_1 \cos \gamma_1 \sin \gamma_1 + 2\frac{d\beta}{ds}}$$

Determine the location of A', the end of the generator

i = i+1

IF i < NP + 1 go to start of loop

RETURN

Fig. 7.12 The structure of the CURV2 subroutine that folds the first developable surface along the folding curve.

7.2.7. The Display Subroutines.

The folded developables created by 3FD can be displayed as either wireframe or shaded polygon models. The subroutine that displays the wireframe is C2 and its structure is detailed in Fig. 7.13.

Subroutine C2

Open Graphics Structure

Empty Graphics Structure

i=1

Start Wireframe Generator Creation Loop

Move3D(A(i,x),A(i,y),A(i,z))

Draw3D(C(i,x),C(i,y),C(i,z))

Draw3D(ADASH(i,x),ADASH(i,y),ADASH(i,z))

i = i+1

If i < Number of Generators+1 go to loop start

Start Loop to Draw Curves Across the developable

i=2

Move3D(A(1,x),A(1,y),A(1,z))

Start A Curve Loop

Draw3D(A(i,x),A(i,y),A(i,z))

i = i+1

if i < No. Gens +1 go to loop start

i=2

Move3D(C(1,x),C(1,y),C(1,z))

Start C Curve Loop

```
                Draw3D(C(i,x),C(i,y),C(i,z))
                i = i+1
                if i < No. Gens +1 go to loop start
        i=2
        Move3D(ADASH(1,x),ADASH(1,y),ADASH(1,z))
        Start ADASH Curve Loop
                Draw3D(ADASH(i,x),ADASH(i,y),ADASH(i,z))
                i = i+1
                if i < No. Gens +1 go to loop start
    Close Graphics Structure
    Update Workstation
    RETURN
```

Fig. 7.13 The C2 subroutine structure that displays the wireframe developables.

The colour shaded model is generated in a similar fashion. To shade the polygons additional information is required. The normal to the surface of the polygon must be specified. The subroutine that creates the shaded polygon model is CPOLY and its structure is described in Fig. 7.14.

Subroutine CPOLY

Open Graphics Structure

Empty Graphics Structure

Define Lights

Define Polygon Attributes such as colour, etc.

i=1

Start First Surface Polygon Creation Loop

Surface Definition (A(i),C(i),C(i+1),A(i+1))

Normal Calculation using Cross Products

$d1 = C(i+1,x) - A(i,x)$

$d2 = C(i+1,y) - A(i,y)$

$d3 = C(i+1,z) - A(i,z)$

$e1 = A(i+1,x) - C(i,x)$

$e2 = A(i+1,y) - C(i,y)$

$e3 = A(i+1,z) - C(i,z)$

$d1xe1 = d2e3 - d3e2$

$d1xe2 = d3e1 - d1e3$

$d1xe3 = d1e2 - d2e1$

$de\_av = sqrt((d1xe1)^2 + (d1xe2)^2 + (d1xe3)^2))$

$normal\_x = (d1xe1)/de\_av$

$normal\_y = (d1xe2)/de\_av$

$normal\_z = (d1xe3)/de\_av$

Draw Polygon

i =i+1

If i < No. of Gens go to loop start

Repeat for Second Surface

Update Workstation

RETURN

Fig. 7.14 The CPOLY subroutine that displays the developable as colour shaded polygons.

7.2.8. The Graphics Structure Subroutine.

The subroutine OPPHE opens the PHIGs graphics systems and sets up the graphics structures shown in Fig. 7.14. The basic structure of this subroutine is detailed in Fig. 7.15.

Subroutine OPPHE

Read Program Defaults File

Open PHIGS System

Open Workstation

Define Colours

Set Up Views

View = Screen Dims & AR

Main View (3D)

Right Elevation

Left Elevation

Main (Front) Elevation

Plan

Draw Structures

Menu Structure

Initial Box Structure.

Associate Roots with Views

Give each view an identifier and setup empty graphics structures in the views.

Initialise Input Devices

Locator (mouse)

Valuator (dials)

Choice (function keys)

Pick (mouse)

String (keyboard)

RETURN

Fig 7.15 The OPPHE subroutine that initialises the GUI and creates the PHIGS structures used by the program.