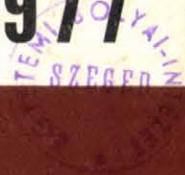


tanulmányok

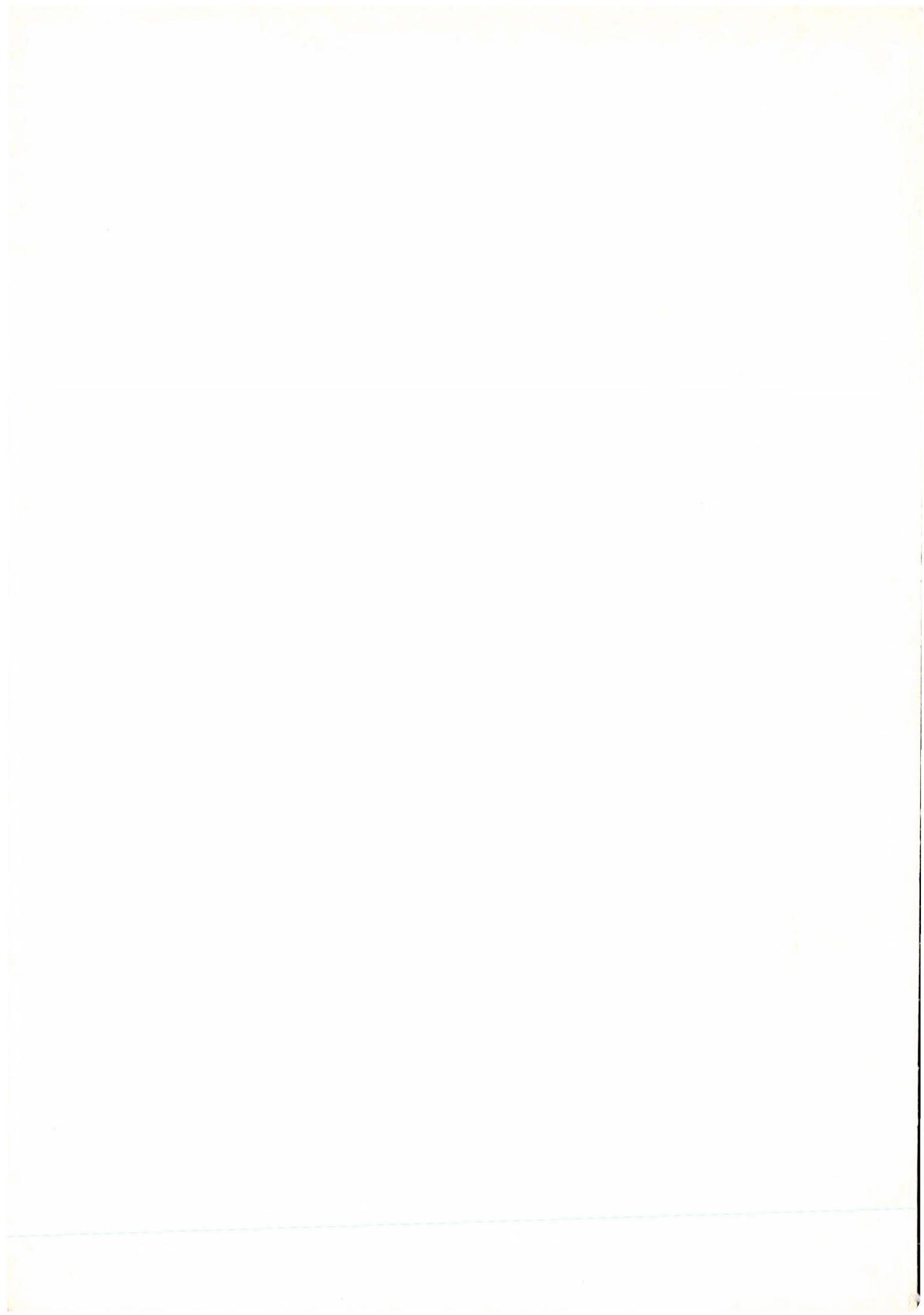
60/1977

1977. MÁRCIUS



MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest





COMPUTER AND AUTOMATION INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES

THE USE OF PIECEWISE FORMS FOR THE NUMERICAL
REPRESENTATION OF SHAPE

Malcolm Arthur Sabin

Reports 60/1977

*THE USE OF PIECEWISE FORMS FOR THE NUMERICAL
REPRESENTATION OF SHAPE*

Dissertation

submitted for the degree of
Candidate of Technical Sciences

by

MALCOLM ARTHUR SABIN

1976

Budapest

Responsible Publisher

T. VAMOS

ISBN 963 311 035 1

TABLE OF CONTENTS

| | |
|--|-----|
| 1. <u>INTRODUCTION</u> | 7 |
| 2. <u>TWO DIMENSIONAL RESEARCH</u> | 13 |
| 2.1 Background | 13 |
| 2.2 Storage of curves | 20 |
| 2.3 Interpolation | 41 |
| 2.4 Approximation | 50 |
| 3. <u>THREE DIMENSIONAL RESEARCH</u> | 64 |
| 3.1 Background | 64 |
| 3.2 Triangular polynominal surfaces | 73 |
| 3.3 Regular trinagular partitioning | 87 |
| 3.4 Regular tirangular partitioning of the sphere | 106 |
| 4. <u>REFERENCES AND BIBLIOGRAPHY</u> | 115 |

Jelen dolgozat a 499.sz
számu intézeti témában került
kidolgozásra

Acknowledgements

Much of the research described here was carried out while the author was employed by the British Aircraft Corporation. He wishes to thank the corporation for permission to publish the work in this form, and the many colleagues, both within the corporation and without, whose discussions helped to shape these ideas.

Thanks are also due to his director of studies for the continued nagging which finally got this document created, and to the authors family for putting up with the inconvenience of the typing.

INTRODUCTION

During the last two or three decades manufacturing technology has been profoundly influenced by two developments. One the automatic control of machines, in particular of the metal-cutting machine tool. The other the computer.

The numerically controlled machine tool, by taking much of the function of special fixtures into the control tape, allowed parts of greater complexity to be machined economically. The computer allowed some of the clerical task of translating the intentions of the part designer into particular instructions for a particular machine tool to be done electronically, thus enabling the control tapes for moderately complex parts to be produced in an acceptable time and at an acceptable cost.

There was still, however, a part programming problem. Although the cost of part programming was acceptable, because of the large reductions in manufacturing cost, particularly of more complex parts, it was an increasingly large fraction of the total cost.

Further, the difficulty of part programming a long machining sequence correctly caused tape prove-out to be a significant activity.

At the human level, part programming is a moderately unpleasant activity for full time employment, being demanding of complete accuracy, requiring little judgement when organised properly, and providing little pleasure in its correct performance.

The objective seemed a worthy one, therefore, to find ways in which the part programming task, that of choosing cutter movements which result in a particular shape being machined, might be reduced by further applications of computing.

If numerical algorithms are to generate cutter motions from the required shape, that required shape must be represented in numerical terms, accessible to the algorithms.

The research task chosen was to explore the field of numerical representations of shape, both of two-dimensional and of three-dimensional geometric entities; in particular searching for forms which would enable the shapes of actual mechanical engineering objects to be defined easily, to be stored in a small amount of computer store, and to be processed in the ways necessary for mechanical engineering by small, fast computer algorithms. Because a large proportion of parts can be described in terms of two-dimensional profiles the research was split into two thrusts, one into the easier two-dimensional area, where the problem was essentially that of improving the store efficiency and the applicability of methods already known, the other into the much more difficult area of those shapes which cannot be represented adequately by the engineering drawing.

The precursors of the work in the two-dimensional field were primarily developments in the languages for part programming numerically controlled machine tools.

Early languages had allowed as individual geometric elements unbounded straight lines and complete circles. General second degree curves were added in APT, and a general smooth curve through empirical data points (many different methods being used to implement this). More recently still, the idea of the Contour was introduced, being a piecewise curve whose pieces could in principle be parts of any of the unbounded primitives. In both the Ferranti Z-curves and the NEL NELAPT implementations each Contour was restricted to consist of straight line segments and circular arcs. In NELAPT empirical curves were allowed, because the BI-ARC interpolation method had been developed, which joined the data points by spans each consisting of two circular arcs.

The straight line/ circular arc composite curve thus fitted a large fraction of two dimensional parts. There was scope for considerable improvement in two areas, the storage required per span and the number of pieces used for empirical curves. Also necessary, to extend its applicability to all two dimensional parts, was a method of approximating, in a tolerably store-efficient way, curves defined by explicit equations, or by very dense empirical data. These improvements were the area chosen for two dimensional development.

The three-dimensional field has been investigated in two respects in the past. One approach deals with complex assemblages of simple faces; the other, with which I have been more concerned, deals with surfaces which are smooth, but which are without any single simple equation. This area,

of sculptured surfaces, has a good history of academic and industrial development, all of which treats each particular surface as a piecewise mapping from a parameter (or abscissa) plane into the space of three dimensions, each piece of the mapping operating on a unit square of the parameter plane. This approach is very successful for objects consisting of but few surfaces, but has difficulties when many surfaces conceptually distinct, but joining with continuity of slope are involved. The rectangular partitioning of the parameter plane is too restrictive.

The area chosen for research in the representation of three-dimensional shapes is that of finding a more fundamental view of the successful Bezier and B-spline sculptured surface techniques, and if possible to determine analogous methods which would not depend on the strict rectangular partitioning of the parameter plane.

The method adopted was that of theoretical study. In each area the first stage was to abstract from the situation the essential features, trying to put aside aspects which merely confused the issue. Parallels were then sought, wherever possible, in ideas which dealt successfully with analogous situations.

Once an appropriate thought framework had been found the development of the structure could take place, specific algorithms being worked out in detail, and checks being made that the results would be usable in practice.

In the two-dimensional work the first stage was to examine the range of requirements (and of possible future requirements) and to assess the variation for which it would be prudent to allow. When it became apparent that a particular representation would combine internal efficiency with apparent external open-endedness if certain problems were solved, those problems were identified and the necessary solutions worked out.

The material of chapter two describes first a new form for the representation of two dimensional curves formed from straight line segments and circular arcs, together with the algorithms for transforming data to and from this form. This form is optimal in the sense that no further reduction of storage is possible while still retaining the same generality of representation, and is numerically well-conditioned in that small changes of numerical values cause only small changes of shape, and vice versa.

It then describes a new algorithm for interpolating such a curve through empirical data points, using only half as many arcs as previous methods, but giving equally acceptable results.

Finally, a new algorithm for approximation of dense data is described, which can be applied either to empirical data or to points evaluated densely on explicitly specified curves.

With these three capabilities available it is now possible for a two-dimensional computer software geometric system to provide outwardly a wide variety of shape definition while using internally a single, simple, efficient canonical form. This economy of representation means that the system will make only very modest demands of computing power.

In the three-dimensional case the first step was to pick out the common features of the best existing surface description methods, trying to identify the necessary properties of these methods, distinguishing them from coincidental properties. Then examples were sought of systems with the necessary properties but with different coincidental properties. At first these examples were abstract and academic, but then concrete and practical.

The material in chapter three describes first the properties of generalisations of the Bezier/B-spline surface description methods to arbitrarily partitioned surfaces. Then a Bezier-like representation of individual triangular surface elements. And finally B-spline-like representations applicable to regular triangular partitionings of the parameter plane and of a parameter sphere.

2.1 Background

Two key threads run through the whole of this work. One is the external impetus of the problem of describing the shapes of mechanical engineering parts holistically. The other is the internal impetus given by the power and generality of spline theory.

It turns out that the latter gives a quite adequate, elegant and efficient solution to the former in the two-dimensional case. The pieces of this solution are described in the chapters of this section. The three dimensional case is rather more difficult. No complete solution is offered, but some contributions toward more effective handling are described in section three.

Early history

The virtue of having a representation of a composite geometric item was recognised at least as early as the early 1960s, when Ferranti Ltd. offered a system called Z-curves for flame-cutting applications of their numerical control equipment. This allowed a profile to be described in terms of a basic profile with notches and cutouts. Once defined the composite profile could be used as a single geometric entity in further definitions.

This early example tended to be forgotten, however, when, during the 1960s, the centre of advance in numerical control applications moved across the Atlantic to America. In the APT-like systems single but composite representations were for many years restricted to curves interpolated

through empirical data points.

Empirical curves

Many forms have been proposed either for fitting a smooth curve to empirical data points, or for designing such a curve to have the required qualitative features.

APT itself originally offered a 4-point cubic interpolation which fitted the span between each pair of consecutive data points by a cubic based on the chord between the points as abscissa axis. The cubic chosen was that which interpolated the nearest two points on each side of the span. This cannot have been very satisfactory, as the curve fitted has sharp changes of direction at the data points. By 1965 a spline option had been added which still used cubics based on the chord, but with slope continuity assured. This type of curve is explored in Butterfield [1969].

Other ad-hoc methods used during the 1960s included a method based on Cornu spirals, used by AEL, and another based on a form of Hermite interpolation, used by Ferranti Ltd. in their Cutting Sequence language. [Alexander 1966] In this latter tangents were calculated at all the data points by fitting circles through each point and its two neighbours. The spans were then filled in by parabolic arcs. McConalogue [1970] gives another form which uses the same tangent calculation (derived independently) and parametric cubic spans. This avoids some of the geometric singularities met by the Ferranti method.

Splines

Meanwhile, however, the theory of splines, started by Schoenberg's analysis of the approximation properties of piecewise cubics [1946] was developing fast.

The spline gains its name from a long thin pliable piece of wood used in shipyards for interpolating data points. The physical spline solves the variational problem of the curve of minimum bending strain energy passing through the data points. Although the resulting differential equation is in general non-linear [Mehlum 1969,1971,1974], the special case arising when the first derivative is small is linear and readily soluble. In this case the curve consists of a sequence of cubic spans abutting with continuity of both first and second derivatives. Calculation of the exact coefficients of the spans requires the solution of a set of simultaneous linear equations, but because of the band structure the time taken to solve these is only linear in the number of data points. The standard theory is to be found in Greville [1969] and Ahlberg, Nilson and Walsh [1967]. It is applied in Adams [1974].

The mathematicians soon found many useful generalisations. Other criteria can be minimised in place of strain energy. This gives spans which are no longer cubic, but which need the same structure of solution for calculating the coefficients to interpolate data. Splines in tension, for example, were studied by Schwiekert [1966], Spath [1969] and, later, Cline [1974] and Nielson [1974].

Higher and lower odd-order curves can also be derived from variational criteria. Other foundations can give the even order splines, as we shall see below.

One of the most fruitful concepts was that of the parametric spline, in which each of the coordinates x and y is a spline function of an independent variable t , say, the two spline functions sharing the same set of knots i.e. values of t at which each span joins the next. Because linear combinations of such functions are also splines with the same knots, the curve remains invariant in form under rotation (indeed, under all affine transformations). This natural axis-independence is ideal for shape representation. Manning [1974] makes good use of the extra degree of freedom, the assignment of the value of parameter at each data point, together with the fact that continuity of curvature does not demand continuity of second derivative with respect to parameter, to obtain the ultimate smoothness from a set of cubic arcs.

Even with such care, however, the spline curve tended to ripple, and those concerned with the design of curves, rather than their part-programming, tried other methods. The work on splines in tension, mentioned above, was one such alternative. Others were the work of Nutbourne [1972][Adams 1975] on intrinsic definition, that of Bezier [1971, 1972][Forrest 1972a] and of Riesenfeld [1973][Gordon and Riesenfeld 1974b][Forrest 1972b]. This last is the most fundamental. It uses the idea of B-splines, which are basis functions non-zero only over a finite interval. The coefficients of these functions are points which do not in general lie on the curve, but on a polygon whose shape exaggerates that of the curve. The B-spline curve controlled by a polygon in this way has two important qualities.

First, that the effect of any change in one of the control points is always local, and second, that the maximum change in the curve is never larger than the change in the polygon. Use of this basis gives a very natural control over the curve even though the control points do not lie on the curve. The finite support of each function also means that when the curve is being used to approximate dense data in a least squares sense the equations form a band matrix with good numerical properties and economic of computer time in solution. The B-spline approach also leads naturally to even order curves.

Mechanical parts

Great as these advances were for the design and description of individual smooth curves, none could be used alone for mechanical engineering parts. Designers equipped with straightedge and compass not only in terms of today's equipment but also in terms of their training have an understandable tendency to use circular arcs in their designs. When designing parts for manufacture by rotating cutters, the circle has an even more fundamental role. Although some of the curve design methods include circles as a special case, and all of them can approximate a circle arbitrarily closely, circles are in all cases held uneconomically. None of the curves can be precisely displaced to allow, for example, for a cutter radius offset. Furthermore, blending arcs are often used between other profiles (see examples in [Davies, K. 1973]), and for none of the curve methods described above is the accurate fitting of such a blend circle trivial. Either inaccuracy is introduced, or else a complex and expensive algorithm is necessary.

These considerations gave good reason for looking to see whether it might be possible to build an interpolating curve from circular arc segments. Sandel[1937] had shown how a Hermite-type interpolation could be performed, given the tangent direction at each point by using two circular arcs in each span of the curve. By some chain, which I have not been able to trace in detail, but which probably included Horn of AEG and Exapt-Verein, these results came together during the late 1960s with the ad-hoc methods of tangent determination which had been used by the Ferranti interpolation [Shippey 1967a,1967b]. Biarc interpolation seemed to indicate that a complete system might be built using only circular arcs and straight line segments. Indeed, the NELAPT system goes some way in this direction [Anthony 1973][McWaters 1974] [Wilkinson 1974] as does CKDAPT [Macurek and Vencovsky 1974]. In shipbuilding the AUTOKON system had used circular arcs as its final form , but only regarded as an approximation to a spline curve interpolation. The BSRA Britships system took up the Biarc idea and improved it by a better choice of the tangent directions[Bolton 1975].

Two further questions arose. What could be done with really dense data, of the kind resulting from automatic digitisation, or from paths computed across sculptured surfaces? What could one do with profiles defined in other terms?

If both of these could be solved, then a 2D system whose only internal primitives were straight lines, circles and piecewise combinations of these, would be possible. Such

a system would be extremely attractive, because the simplicity and unity of the representation would mean great economy of algorithms, because the unity of representation would give complete orthogonality - any facility would work for all profiles - and because curves defined in any way could be added to the system by the writing of a single routine.

Even if these problems were not solved, there are sufficient applications which actually use only straight lines and circles explicitly for it to be worth trying to find the optimum ways of storing such piecewise curves, so that as much as possible might be kept in the immediate access store of the computer. It is also worth while to try to improve the Biarc interpolation method in smoothness and store-efficiency.

These then were the challenges which, if met, could lead to a very efficient 2D shape handling system for mechanical parts:-

- 1 To find the most store-efficient forms for holding the straight-line/circular arc curves.
- 2 To improve, if possible, the smoothness and store efficiency of the Biarc approach to interpolation of empirical data points.
- 3 To find algorithms for approximating dense pointwise data and curves with given equations other than circles and straight lines.

Answers to these form the next three chapters.

2.2 Efficient storage of circular arcs

The question of efficient use of store is best approached from an information-theoretic viewpoint. A certain amount of information is necessary to distinguish each object represented from all other such objects representable. Once a universe of discourse is defined, there is a lower bound on the amount of storage space required. The most efficient representation will use exactly that amount of store, but we might choose to accept lower store-efficiency if by doing so we achieve other desirable properties; in particular we require algorithms for accessing the stored form which are reasonably fast and which do not themselves occupy a great deal of storage space.

We can identify easily certain undesirable features of representations: ambiguity is intolerable, the representing of the same object by more than one possible representation is inefficient. Every representation should correspond to some actual object.

Any representation fitting into a finite amount of computer store can distinguish only between a finite number of alternatives. This implies that questions of resolution must be considered; fine resolution demands additional storage. The number of alternatives might also be kept small by restricting the class of objects which can be handled; if the class is kept wide to give power and generality, a penalty has to be paid in the storage of all the simple objects normally dealt with.

A precise definition of the objects to be handled is thus an indispensable starting point.

The definition used here is :-

A profile is an ordered set of pieces of curve. Each piece is either a circular arc, or a segment of straight line. Each piece shares one end point with its predecessor and the other with its successor, so that the complete profile has continuity of position. The first piece does not have to share an end point with the last piece. Every point of every piece lies within a square region of the Euclidean plane of stated maximum dimension. Two profiles which have the property that every point of the first has a point of the second within a stated distance (tolerance) and vice versa need not be distinguished from each other numerically.

This definition has certain implications. One is that the length of a profile, in the sense of the number of pieces, is not bounded.

In this context we therefore assume that some variable length format will be used, so that short profiles can be stored in a small piece of store without placing any restrictions on the maximum number of pieces in a single curve.

Another is that the resolution is defined in terms of geometric precision. This obviously fits the actual requirements, and leads us to reject some of the more obvious possible representations.

The question of position continuity is arguable. Certainly we can accept position continuity, and certainly we need to represent shapes which are not tangent continuous, but it is

possible to force discontinuous tangents into a system with continuity of tangent by adding singular pieces at the corners.

We also need a list of the operations involved in the use of the representation. In this case we need to create the stored form from a variety of different geometric definitions, we need to calculate many different geometric properties, and we may need to transform the object represented by applying certain geometric transformations. In principle this last can be achieved by using the calculation of properties and the definition of a new profile, but it is advantageous if direct calculations can be made. The bulk of this chapter is taken up by demonstrating the economy of the operations required when acting on the proposed representation.

For the moment however, we note only that most of the calculations deal with local properties, and so it is convenient for the local shape to be defined by local data. An obvious example of this is that the endpoints, shared by two adjacent spans, should be accessible from either of them, and that the additional data defining the course of the profile between those points should be local too.

If possible the straight line should be a natural special case. The routines which work for arcs ought to work equally well when the curvature is zero, even if they are then slightly more timeconsuming than is strictly necessary.

The disadvantages of traditional forms

The most used representations of full circles are

(i) the centre and radius. This gives the potential function form $f(P) = 0$ very quickly and easily, but it is not easy to add the bounding information to deal with arcs instead of full circles.

(ii) the centre and a point on the circumference. The potential function is again easily calculated, but if we add a second point on the circumference to bound the arc, we find both ambiguity and redundancy. The ambiguity is because two arcs form the complete circle, the redundancy because the centre needs to be equidistant from both of the circumferential points.

(iii) two points and the radius. This still has the ambiguity but the redundancy is avoided. However, the radius is not a good measure because circles of very small curvature have a very large radius value, and because small perturbations of the radius value for arcs near a semicircle cause comparatively large changes in the actual shape. Radius values less than half the distance between the two points do not correspond to any arc.

The ambiguity is actually worse, because there are four arcs corresponding to a single radius value. Not by coincidence, an extra square root is necessary in the calculation of the potential function.

The ill-conditioning due to the use of the radius can be improved by the use of its reciprocal, the curvature, but there are still problems near a semicircle and the square root is still necessary.

Possible tangent-continuous representations

Analysis of the number of degrees of freedom shows that the minimum number of items of data for a tangent continuous profile is two per span plus three. If for convenience we store the endpoints of the pieces, there is only one degree of freedom left for the entire profile. This we can consider to be equivalent to the direction of the tangent at one of the data points. The directions at all of the others may be calculated, but only with the danger of rounding errors corrupting the accuracy as the calculation proceeds along the curve. If uncorrelated noise is added to all the point coordinates, the error in tangent direction will tend to grow as the square root of the number of steps taken. This effectively places a bound on the maximum number of pieces in a single profile.

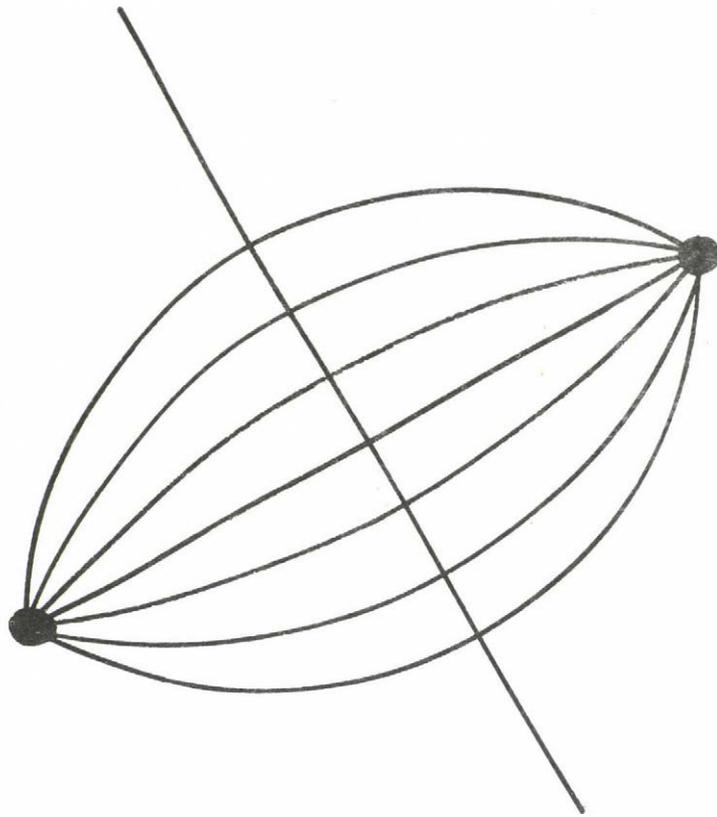
Another disadvantage is that the presence of singular pieces at the sharp corners prevents the calculation of the tangent direction proceeding past such points.

An alternative would be to store distances and angles to each of the data points, measured along the curve and from some datum direction to the tangent. This is equivalent to the curvature profile advocated by Nutbourne. While there is the advantage that relocation of the profile in the plane becomes extremely trivial, the disadvantage that the coordinates of the data points have to be calculated, by a process subject to build up of rounding error, rules this form out for mechanical engineering.

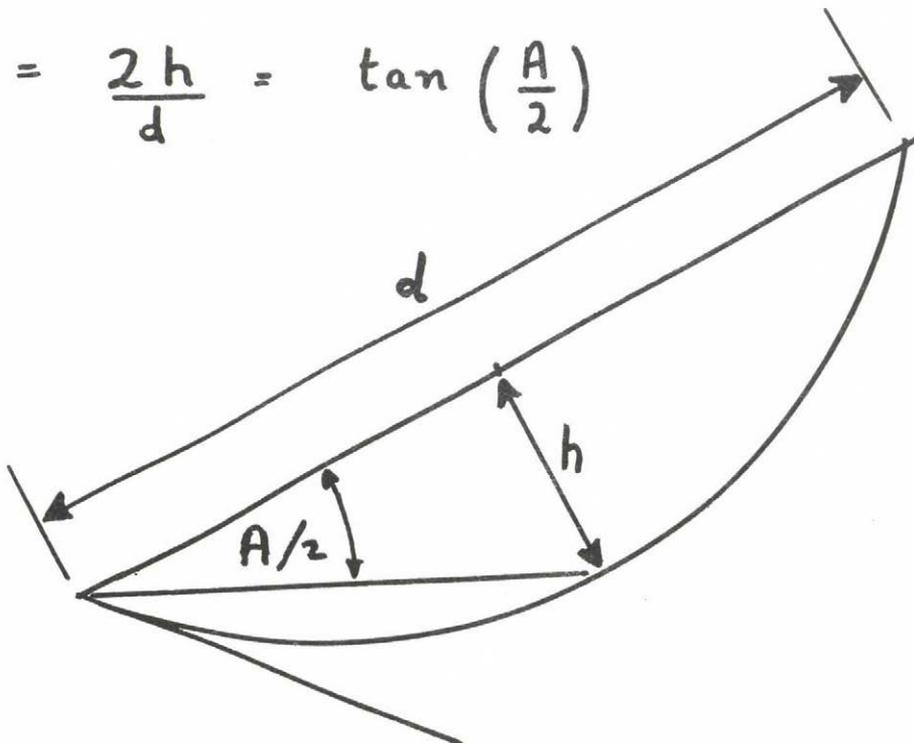
Efficient position continuous forms

Consideration of the number of degrees of freedom gives in the tangent-discontinuous case, three numbers per piece plus two, which may conveniently be distributed as the set of data points plus one number per span. The question is really what that one number should be. As we have already seen, the radius is unsuitable, and the curvature almost as bad. The best guide in this matter is to draw, on a sheet of paper, a set of arcs all sharing the end points. It is then quite obvious that to each arc corresponds a single point, in which that arc cuts the perpendicular bisector of the chord joining the two endpoints, and that the geometry of the arc is well conditioned with respect to the position of that point. The point is in turn well defined by a single number which describes its distance from the chord, the sign being positive on one side of the chord, negative on the other. There are then some very minor variations because we might hold the distance itself, or the ratio of the distance to the chord. The latter will be marginally better on a computer with division slower than multiplication, and with access to the data outweighing calculation of the stored form. The latter also has the advantage of being a function only of shape, not of size or position.

The form advocated here stores the ratio of the distance perpendicular to the chord to half the length of the chord, because that value is the tangent of half the angle between the chord and the arc at the endpoints, and from the tangent of half an angle there are simple algorithms for calculation of the sine and cosine, which are in turn often required in the calculation of geometric properties of the profile.



$$B = \frac{2h}{d} = \tan\left(\frac{A}{2}\right)$$



Machine representation

Once a logical representation is chosen it has to be implemented on a particular machine. The actual coordinate ranges and resolutions need to be considered. To a large extent the use of Fortran REALs disguises this question. REALs are there, so we might as well use them; equally, what else can one do in Fortran? While this is true, the Fortran programmer needs to be aware that inaccuracies of significant import can result from the use of only 32 bits for floating point quantities, particularly if a base 16 exponent is used. Double precision is often necessary, but it is expensive and so should not be applied generally.

Equally, the machine code programmer, writing for a minicomputer without floating point hardware, needs to know if he can use fixed point quantities, and whether single length arithmetic is adequate.

For the representation proposed these questions can be answered straightforwardly. If we take as our measure of resolution required the tolerance stated divided by the size of the maximum square in which all profiles lie, the machine resolution necessary is the same.

For example, if we wish to distinguish two profiles whenever they differ by more than 0.1 mm and the size of the object which the profiles bound is always less than 1 metre in total extent, then a precision of one part in 10000 is adequate for all the numbers in the storage representation, provided that no arc subtends more than 180 degrees. A resolution of

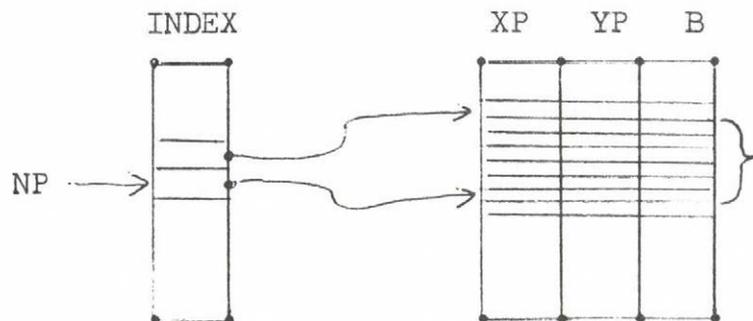
one part in 10000 is provided by 14 bits, and so a 16 bit minicomputer could use single length storage quite safely.

If , however, a tolerance of 0.001 mm was to be held over a dimension of 10 metres, one part in 10 to the 7 would be required, just demanding double precision floating point.

Proposed Storage Format

We propose two formats, one for use in a Fortran environment, the other for use in machine coding. The first consists of four arrays, one INTEGER with as many elements as the maximum number of profiles plus one, the other three either REAL or DOUBLE PRECISION as resolution demands and with as many elements each as the total number of endpoints in the complete set of profiles to be stored. The integer array is termed INDEX here, and the three others XP,YP, and B

The data for a particular profile is stored in a set of consecutive elements in XP ,YP, and B and the set of elements is bounded by two consecutive elements of INDEX. Typically the subscript of INDEX will be obtained from a symbol table, or by allocating the profiles serial numbers. Let the subscript be NP (for Number of Profile); then INDEX(NP) points to the first subscript of the profile data in XP, YP and B and INDEX(NP+1) contains the subscript one greater than that of the last element of the profile data.



Thus $XP(INDEX(NP))$ and $YP(INDEX(NP))$ contain the coordinates of the first point of the profile, $XP(INDEX(NP)+1)$ and $YP(INDEX(NP)+1)$ the coordinates of the second point, $XP(INDEX(NP+1)-1)$ and $YP(INDEX(NP+1)-1)$ those of the last point. $B(INDEX(NP))$ is not used, $B(INDEX(NP)+1)$ holds the bulge factor for the first span of the profile, and $B(INDEX(NP+1)-1)$ holds the bulge factor for the last span.

The machine code format is logically the same, but array `INDEX` holds the store addresses of the first X coordinate of the profiles, and the store allocation is transposed so that consecutive store locations (single or double length depending on the resolution required) hold `XP`, `YP`, `B`, `XP`, etc.

A legible format for the visual examination of curve data and for the transmission of profiles from one program to another uses the length of the profile (number of spans) in place of the `INDEX` array:

| | | | |
|------|------|----|---------------------|
| i.e. | NS | | identification data |
| | X1 | Y1 | 0.0 |
| | X2 | Y2 | B1 |
| | X3 | Y3 | B2 |
| | etc. | | |

It must be assumed in what follows that examples using the Fortran format apply equally to the other equivalent representations.

The reason for leaving the first `B` slot vacant instead of the last is that in a practical application it may be desirable to use that word of data for information relating to the whole profile (a flange thickness, for example) and in that situation it is more convenient to have such data available at the start of the data, so that it may be accessed in strict stream fashion.

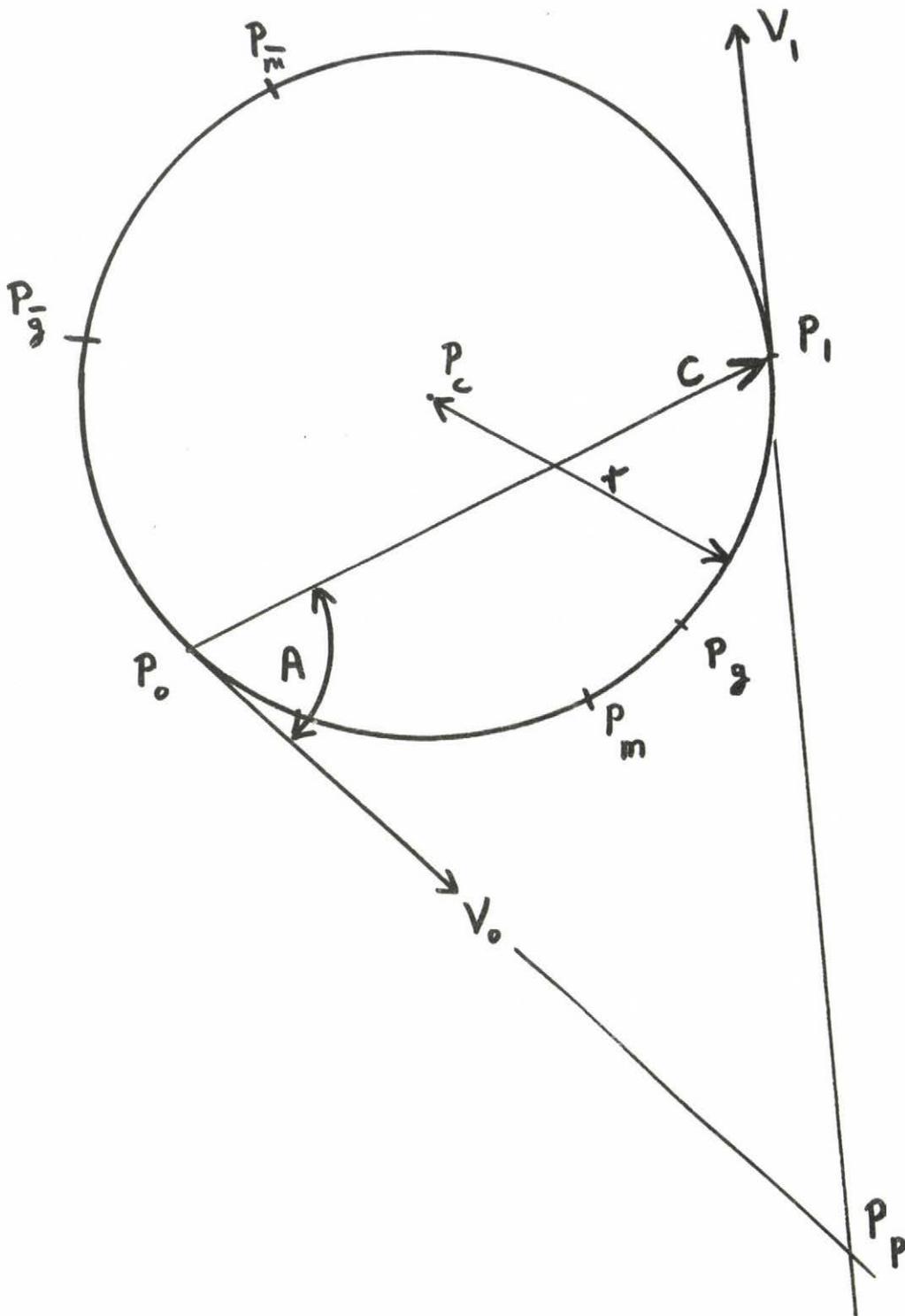
Algorithms

A numerical representation is meaningful only when appropriate algorithms are known which can generate it, and which can abstract from it the information needed. The remainder of this chapter deals with such algorithms: first those for calculating the B values from other geometric data (we may assume that the calculation of the XP and YP values poses no special difficulty), then those for calculating geometric properties, and finally those for transforming profiles.

Construction algorithms

The profile consists of a set of spans, and for the construction of the profile as a whole we must construct each of the spans individually. In each case we assume that the endpoints are already known, together with at least one further piece of data. The items which might be supplied are

- P_c centre of the arc
- r radius of the arc, signed to indicate the sense, together with some information to select either the large arc or the small one
- P_m midpoint of the arc
- P_g a general point of the arc
- $P_{\underline{m}}$ the midpoint of the remainder of the full circle
- $P_{\underline{g}}$ a general point somewhere on the remainder of the full circle



V_0 initial tangent vector, pointing into
into the arc

V_1 final tangent vector, pointing out of
the arc. V_0 and V_1 are not assumed
to be unit vectors.

P the pole point, where the tangents
intersect.

A the angle between the tangent and the chord.

s, c the sine and cosine of A . These are
assumed to be known together, but both
might be multiplied by some factor.

(i) Given A

This is the only construction requiring trigonometric functions to be calculated.

$$B = \tan(A / 2)$$

(ii) Given s and c

We assume that $s = x \sin(A)$

and $c = x \cos(A)$

where x is unknown.

$$\text{Now } s = 2B / (1 + B^2)$$

$$\text{and } c = (1 - B^2) / (1 + B^2)$$

$$\text{so that } s / c = 2B / (1 - B^2)$$

Solving this equation for B gives

$$B = -c/s + \sqrt{(c/s)^2 + 1}$$

This form is singular at $s = 0$, but the equivalent

$$B = -s / (c + \sqrt{c^2 + s^2})$$

$$B = -s / (c + \sqrt{c^2 + s^2})$$

is always well behaved. By inspection it can be confirmed that the positive square root is always the one required.

(iii) Given V_0

For this case we must introduce two vector operators, the dot and cross products, both of which give scalars in the two dimensional geometry context.

Let U and V be two two-dimensional vectors, held as their components U_x, U_y, V_x , and V_y

The dot product $U \cdot V$ is the scalar value

$$U_x V_x + U_y V_y$$

which has the value of the product of the lengths of U and V multiplied by the cosine of the angle between them.

The cross product $U \times V$ is the scalar value

$$U_x V_y - U_y V_x$$

which has the value of the product of the lengths of U and V multiplied by the sine of the angle between them, a counter-clockwise rotation from U to V giving a positive result.

Now the angle between V and C , the chord of the arc, is just A , and so we may calculate s and c by

$$s = V_0 \times C$$

$$c = V_0 \cdot C$$

and use construction (ii)

(iv) Given V_1

$$s = C \times V_1$$

$$c = C \cdot V_1$$

and again use construction (ii)

(v) Given P_p

This construction can only be used when the value of P_p given lies on the perpendicular bisector.

$$V_0 = P_p - P_0$$

and use construction (iii)

(vi) Given P_g

As P_g traverses the arc the angle subtended at P_g by the chord $P_0 P_1$ remains constant at the value $180 - A$

Thus

$$s = (P_g - P_0) \times (P_1 - P_g)$$

$$c = (P_g - P_0) \cdot (P_1 - P_g)$$

and use construction (ii)

(vii) Given $P_{\bar{g}}$ or $P_{\bar{m}}$

$$s = (P_0 - P_{\bar{g}}) \times (P_1 - P_{\bar{g}})$$

$$c = (P_0 - P_{\bar{g}}) \cdot (P_1 - P_{\bar{g}})$$

and use construction (ii)

(viii) Given P_c

This construction can only be used when the value of P_c given lies on the perpendicular bisector of $P_0 P_1$

$$s = C \cdot \left(\frac{P_c - P_0}{C} \right)$$

$$c = C \times \left(\frac{P_c - P_0}{C} \right)$$

(ix) Given r

It will be shown below (algorithm xvi) that the radius is given in terms of B by the expression

$$r = \frac{d}{4} \left(B + \frac{1}{B} \right)$$

where d is the length of the chord C

This equation may be solved for B giving

$$B = \frac{2r}{d} \left(1 \pm \sqrt{1 - \left(\frac{d}{2r} \right)^2} \right)$$

This is the preferred form, because d, r should not be zero, $2r$ should be greater than d , and the sign of the square root may be simply determined.

The positive sign gives the arc greater than half a circle, the negative sign gives the arc less than half a circle.

r and B will be positive for counterclockwise arcs and negative for clockwise arcs.

Property algorithms

(x) To determine A

$$A = 2 \arctan(B)$$

(xi) To determine s and c

Although it would be possible to determine s and c from the value of A determined by algorithm (x) this would be extremely inefficient. Far quicker is to use the half-angle formulae

$$s = 2B / (1 + B^2)$$

$$c = (1 - B^2) / (1 + B^2)$$

If only the ratio of s and c is required, the division by the common denominator term may be omitted for speed.

(xii) To determine V_0 and V_1

We introduce here the concept of the rotation operator R . $R(U)$ is a vector obtained by rotating the vector U through one right angle counterclockwise.

$$R(U) = \begin{matrix} -U \\ x & y \end{matrix}$$

$$R(U) = \begin{matrix} U \\ y & x \end{matrix}$$

This operator is linked to the dot and cross products by

$$U \cdot R(V) = -U \times V$$

$$U \times R(V) = U \cdot V$$

A vector of length d and direction V_0 may be set up

by
$$V_0 = (P_1 - P_0)c - R(P_1 - P_0)s$$

and similarly

$$V_1 = (P_1 - P_0)c + R(P_1 - P_0)s$$

If the unit vectors are required, they may be divided by the magnitude of d . Only one square root is required to normalise both. If the $1 + B^2$ term is ignored in the calculation of s and c the vectors calculated will be correspondingly longer.

(xiii) To determine P_m

$$P_m = (P_1 + P_0)/2 - B R(C/2)$$

(xiv) To determine $P_{\bar{m}}$

The B value corresponding to the other arc of the full circle is $-1/B$ and so

$$P_{\bar{m}} = (P_1 + P_0)/2 + R(C/2)/B$$

(xv) To determine P_c

P_c is the midpoint of P_m and $P_{\bar{m}}$ and so

$$P_c = \frac{(P_m + P_{\bar{m}})}{2} = \frac{(P_1 + P_0)}{2} - \frac{1}{2} \left(B - \frac{1}{B} \right) R(C)$$

(xvi) To determine r

r is half the distance from P_m to $P_{\bar{m}}$ and so

$$r = \frac{d(B + \frac{1}{B})}{4}$$

(xvii) To determine P_p

$$P_p = \frac{(P_1 + P_0)}{2} - \frac{B R(C)}{(1 - B^2)}$$

This is only useful if $B^2 < 1$

(xviii) To determine whether a given general point P lies to the left or to the right of the full circle traversed in the direction of the arc.

$$\text{Let } f(P) = c(P - P_0) \times (P - P_1) - s(P - P_0) \cdot (P - P_1)$$

Then the value of $f(P)$ will be positive if P lies to the right of the full circle, negative if it lies to the left, the circle being traversed in the direction of the arc.

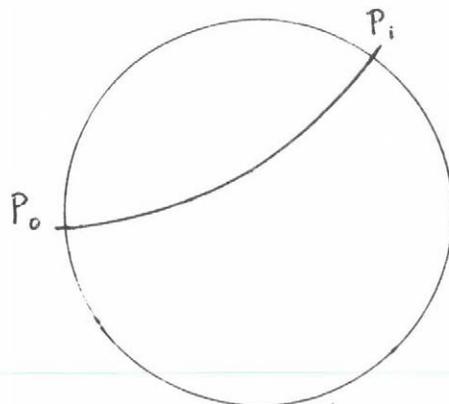
Providing that s and c are the true sine and cosine, and that P lies close to the circle, the magnitude of $f(P)$ will be approximately d times the distance of P from the full circle.

(xix) To determine whether a point lying close to the full circle lies near to the arc, or near to the other part of the full circle.

$$\text{Let } g(P) = s(P - P_0) \times (P - P_1) + c(P - P_0) \cdot (P - P_1)$$

Then the value of $g(P)$ will be negative if P lies on the arc, positive if it lies on the other part of the full circle.

The locus $g(P) = 0$ is in fact another circle cutting the arc orthogonally at P_0 and P_1



(xx) To determine a sequence of points lying on the arc

$$\begin{aligned} \text{Let } d_u &= 1 + B^2 \\ d_2 &= 4B^2 \\ d_1 &= -d_2 \\ N_u &= d_u P_u \\ N_2 &= 2BR(C) + 2B^2 (P_u + P_1) \\ N_1 &= d_u P_u - N_u - N_2 \end{aligned}$$

Then as a parameter t varies from u to 1 the point

$$P(t) = \frac{(N_2 t + N_1)t + N_u}{(d_2 t + d_1)t + d_u}$$

will traverse the arc from P_u to P_1

Transformation algorithms

The most important transformations are the solid body rotations. Note that because circles transform into ellipses under a general affine transformation, the most general linear transformation we can apply to these profiles is a solid body rotation, combined with an isotropic scaling and a mirroring in some line.

(xxi) To apply a solid body rotation

Transform each of the endpoints, and leave the B values unchanged.

(xxii) To apply an isotropic scaling

Apply the scaling to each of the endpoints and leave the B value unchanged.

(xxiii) To apply a mirroring

Apply the mirroring to each of the endpoints, and negate each of the B values.

(xxiv) To reverse the direction of traversal of a profile

Reverse the sequence of the endpoints, and reverse the sequence of the B values, also negating them.

It is not claimed that these twentyfour algorithms are the only ones which will ever be required. Rather, that all the other facilities which will be required in a practical system can be worked out in a similar style, probably using the above as building blocks. These twentyfour do, however, indicate that the proposed storage format is reasonably efficient in terms of access to the data actually used, as well as in terms of storage density.

Addition of redundancy

There are two operations which may consume an excessive amount of time in the above algorithms. The first is the calculation of the length of the chord of the arc. If this is embarrassing because it requires the calculation of a square root it would be a reasonable trade-off to store the values of d as a fourth field. If on the other hand it was the square root in the calculation of B from s and c which was the time problem, a possible solution would be to store s and c instead of B.

2.3 Interpolation of empirical data

In mechanical engineering curves are defined in four principal ways.

- a) by explicit straight lines and circular arcs, obtained by use of straight edges and compasses.
- b) by fairing some smooth curve through empirical points.
- c) by specific equations - for example, a radar reflector might have parabolic supporting elements.
- d) by dense data from automatic digitising or from computation of cross-sections through sculptured surfaces.

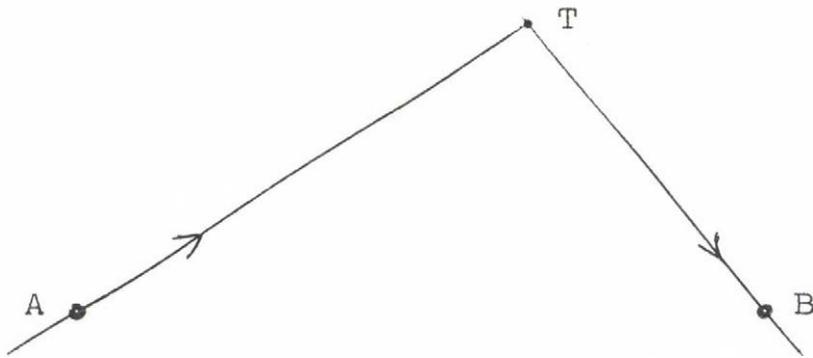
A general purpose shape handling system must be capable of accepting information in any of these forms. Provided that geometric accuracy is maintained, however, any internal representation may be used. This chapter and the next show how information of types b), c) and d) can be converted into profiles of the form discussed in chapter 2.2 above. Here type b) is considered.

Several authors have described methods for interpolating sparse empirical data points by curves consisting of circular arcs. [Sandel, Shippey, Bolton, Uveras] In each case the algorithm has the general form of first calculating the tangent directions at each of the data points. Sandel assumes them to be given, Shippey uses the direction of the circle through the point and its two neighbours, Bolton hints at the use of a cubic spline to determine directions, Uveras takes a particular linear combination of the three circles, each through three consecutive data points, which pass through the point in question.

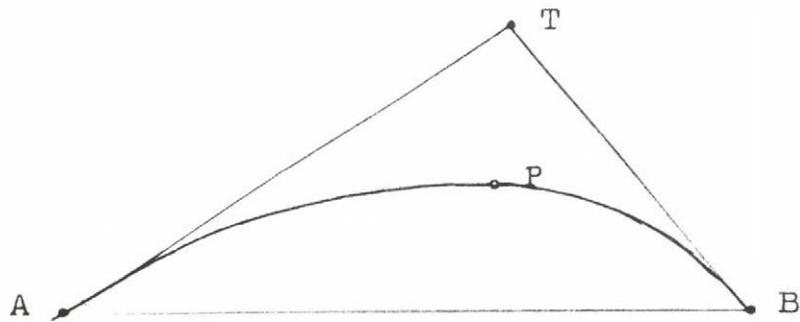
Once the directions are found at all the data points, the spans are then considered one at a time.

A convenient description of the situation of a typical span is in terms of a tangent triangle, the triangle formed by the chord and the two tangents at the ends of the chord.

Let the chord be AB so that two successive data points are A and B . The point T is at the intersection of the two tangents AT and TB



The next stage in each is to determine a point P such that the arc AP tangent to AT at A is tangent to the arc PB at P while the arc PB is in turn tangent to TB at B .



Sandel pointed out that for tangency to be achieved at A , P and B P has to lie on a circle which passes through A and B and also through the incentre of triangle ABT .

The question is then, which point of the locus to use?

Shippey showed that for the arcs AP and PB to have the minimum bending energy P should also lie on the perpendicular bisector of AB. Bolton selects his point to minimise the difference of the radii of AP and PB.

All these methods therefore require two circular arcs per data point supplied.

Two new results are now put forward.

The first is that if the curve is not to have an inflexion the point P must lie inside the triangle ABT, and that therefore a point which is known to lie both on the locus and inside the triangle will be a good choice in the sense that inflexions not demanded by the data will not appear. Such a point is the incentre itself.

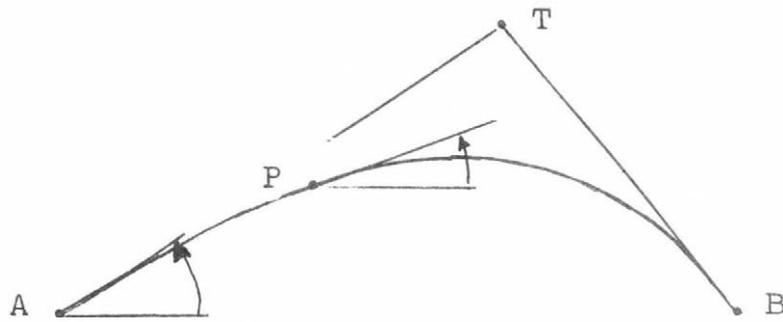
The solution proposed by Shippey can have extraneous inflexions, and will do if the ratio of the angle of A to that at B or vice versa exceeds three.

The second is that there is no magic about any of the methods proposed for finding the tangents. If instead we treat the tangent directions as spare degrees of freedom, we can use those degrees of freedom to reduce the number of arcs required per data point. It is a standard approach to spline theory to express the curvatures at left and right of a data point as functions of the data ordinates and the postulated slopes at the data points, and then to equate these two curvatures, giving a set of equations which are then solved for the slopes. We can take the exact analogy. The curvature at each end of a bi-arc span can be expressed in terms of the end coordinates and the end tangents, and the condition for continuity of curvature across each data point is the condition that the

arcs on the two sides of the data point are parts of the same arc. This idea is applicable whatever the strategy for fixing the point P on the locus. It results in a halving of the number of arcs necessary to interpolate any given set of data.

General calculation of curvatures

If P moves along the locus toward A , the tangent direction at P becomes the reflection in AB of the tangent direction at B . Similarly, if P moves along the locus toward B the tangent direction becomes the reflection in AB of the tangent direction at A . We can parametrise the position of P along the locus in terms of this tangent direction, and the end curvatures then become functions of the chord length s and the three directions only. In order to ease the ensuring of consistency later, we represent each tangent direction by the angle measured counterclockwise from the direction AB to the tangent in question. Note that the tangents are directed, so that the direction of BT is 180 degrees different from that of TB .



Let the tangent direction at A be $2a$, that at B be $2b$, and that at P be $2c$. In the situation illustrated the angle at B has a negative value.

The position of P may be calculated from a, b and c because the angle of AP is just a+c and that of PB b+c so that P is given by the intersection of two known straight lines.

The incentre is given by $c = 0$, and Shippeys bisector condition by $2c = -(a+b)$.

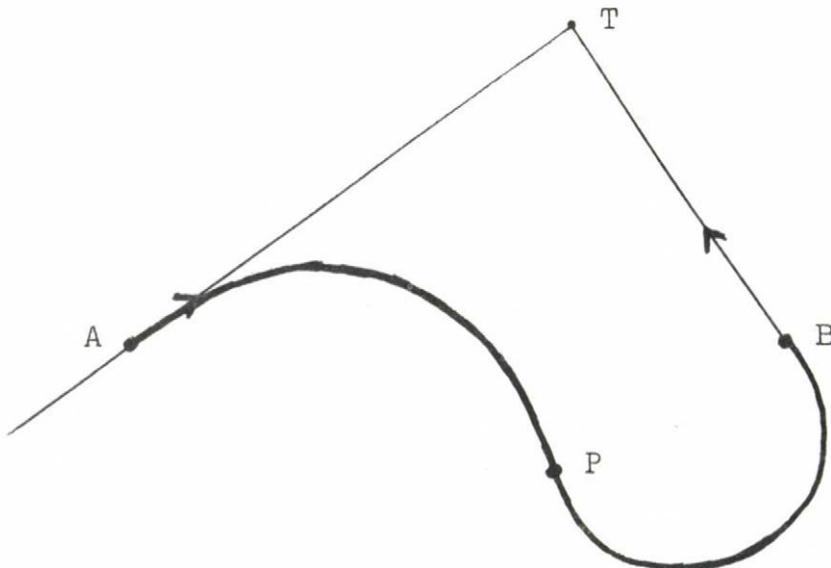
The curvatures at the ends of the span are given by

$$c_{rA} \text{ (curvature at right of A) } = - \frac{2 \sin(b-a) \sin(a-c)}{s \sin(b+c)}$$

$$c_{lB} \text{ (curvature at left of B) } = - \frac{2 \sin(b-a) \sin(b-c)}{s \sin(a+c)}$$

Inflexion case

If the data points themselves show a change in sign of the cross product of the second difference with the first, an inflexion in the curve itself is unavoidable. It will appear either at a data point, or in a span for which a and b have the same sign. If we are achieving continuity of curvature across the data points the former possibility is ruled out, and so it is necessary to consider the latter case.



Here the locus passes not through the incentre, but through two of the excentres, neither of which is suitable as a position for P . A reasonable solution is found, however, by letting the tangent direction at P be the reflection in AB of the median from T to AB . The justification for this is that in the bisector case the direction of the tangent at P is always the reflection of either the internal bisector of the angle ATB or the external bisector. We may consider the condition $c = 0$ as giving the tangent direction parallel to the reflection of the external median from T , and so this choice is described by similar terms. The limiting case behaviour as either tangent tends toward AB is also similar.

This construction gives

$$\tan(2c) = -2 \frac{\tan(2a) \tan(2b)}{\tan(2a) + \tan(2b)}$$

The expressions for the curvatures are, of course, equally valid for this case. Note that the two curvatures are now different in sign.

Calculation of tangents at the data points

Consider a complete set of data points. Let the length from the i^{th} point to the $(i+1)^{\text{th}}$ be s_i and let the angle of that chord from some convenient datum be t_i . Postulate tangent directions at each of the data points, that at the i^{th} point being at angle u_i to the datum.

Then the discontinuity of curvature there will be given by the difference of the two curvatures calculated one at the left hand end of span $i+1$ the other at the right hand end of span i . In each case the values of a and b , and thence c are functions of the u t and s , and thus so are the curvatures. We may therefore write the discontinuity value

f_i as

$$f_i = f(u_{i-1}, u_i, u_{i+1}, t_{i-1}, t_i, s_{i-1}, s_i)$$

Written out in full this looks fiercely non-linear, but the appearance is deceptive. f_i is definitely monotonic in u_i

and u_i is shown below to be the dominant variable in f_i

Provided that a good first approximation is available, and any of the published methods will give an adequate first approximation, we may use Newton-Raphson iteration to modify the u values until f_i is acceptably small at all of the data points.

$$\delta f_j = \sum_i \left(\frac{df_j}{du_i} \right) \delta u_i = -f_j$$

and so

$$\delta u_i = - \sum_j \left[\frac{df_j}{du_i} \right]^{-1} f_j$$

The matrix $\frac{df_j}{du_i}$ varies at each iteration, if only slightly, and so needs recomputation and reinversion into the residual discontinuity. The inversion process is by far the less onerous of the two, as the matrix is a triband matrix, invertible in a time linearly proportional to the number of

data points. The setting up of the matrix, however, requires the evaluation of many trigonometric functions. This is the least satisfactory aspect of this work.

Fortunately few iterations are typically required. Once an acceptable set of u values are known, the actual change points may be calculated, and algorithm vi of chapter 2.2 above applied to convert the curve to a standard form.

Note that the original data points become general points of the profile, the change points at which the pieces of curve abut being the newly calculated P points.

End conditions

Two forms of end conditions are typically encountered. In one case the end tangent direction is known (often by tangency required with an adjacent explicitly specified curve); in the other it is not.

In the first case we may set up the initial u value satisfying the constraint, and insert the equation

$$\delta u_1 = 0$$

as the first line of the matrix.

In the second case there is no reason why a single circular arc should not be fitted to cover the whole of the first span as well as half of the second. Thus u_1 is initially set to be

$2t_1 - u_1$ and the equation

$$\delta u_1 + \delta u_2 = 0$$

inserted as the top line of the matrix.

Perturbation behaviour

Consider the situation when a set of data points are evenly distributed around a circle. The distance from each point to the next is s and the angle between each chord and the tangent to the circle is $2a$.

By symmetry $c = 0$ and $b = -a$

Now $\frac{df}{du} = -2 \frac{dc}{da}$ and $\frac{df}{du} = 4 \frac{dc}{da}$

So $\frac{df}{du} = 4/s$

and $\frac{df}{du} = \frac{8 \sin(3a)}{s \sin(a)}$

As a tends to zero the coefficients of the matrix thus tend to the $1 \ 0 \ 1$ configuration familiar from the quadratic spline. We therefore expect the curve described to behave in a similar way to the quadratic spline under conditions of small curvature. The ratio of the terms becomes $1 \ 4 \ 1$ for $2a = 60$ degrees, and so for all reasonable configurations the ripple behaviour will be no worse than the cubic spline.

2.4 Approximation of dense data

We now consider the situations where a curve is specified either by some explicit equation, or by dense coordinate data obtained from automatic digitisation or from other computational processes. The term dense here implies that the distances between data points are considerably less than half the wavelength of the features characterising the curve, so that the information carried by each data point is relatively little.

It is not practicable to have routines for the handling of every possible curve equation, and so the approach is suggested of dealing with explicit equations by writing a routine for each new form as it arises, generating dense data along the curve, which can then be fitted using the same methods as apply to dense data from the other sources. This way it is possible to combine wide applicability with the economy so far achieved. The generation of points along any curve is usually a straightforward matter, not requiring any particular skill in geometry to program.

The problem is thus reduced to the finding of a profile with as few pieces as possible such that it passes within a specified maximum distance of every one of a dense set of data points.

As thus stated, the problem is very difficult to approach, and so the more tractable version is tackled here - find a profile with an acceptably low number of pieces passing within an acceptably small distance of all of the data points.

We may distinguish two variants of this problem. One allows discontinuities of tangent in the profile, the other does not. The former is relatively straightforward, the latter more subtle. If the tolerance permitted is very small compared with the distance apart of the data points the two should give very similar results, but it may be necessary to determine a slope-continuous profile for smooth machining, or to satisfy the onlooker that a good fit has been obtained to a slope-continuous explicit equation.

Tangent-discontinuous approximation

A profile with an acceptably low number of pieces (though obviously not minimal) can be obtained by starting at one end of the data stream. The first data point is treated as the first point of the profile. A tangent direction is computed, either from the original source of the data, or by some approximation to the first few points.

The first arc now has but two degrees of freedom, which we may take to be arc length and curvature. Whatever the arc length, provided it be great enough, each data point after the first places upper and lower bounds on the curvature which the arc may have while still passing within the permitted maximum distance of that data point. As successive data points are considered these bounds are cumulatively tightened, until a data point is found such that if the arc passes through that point it will violate one of the bounds. The previous point is then taken as the end point of the first arc, and the B value determined by algorithm iii of chapter 2.2 above. All previous data points are then

discarded, and the process repeated for the next piece, as many times as are necessary until the end of the data stream is reached. It is important that the determination of the slope at each new start point be determined looking ahead only, as otherwise too many spans may be generated at sharp corners in the data.

Tangent continuous approximation

This is somewhat less straightforward. Shippey suggests a modification of the previous algorithm, using as the start direction of each new span the final direction of the previous one. Clearly this gives a profile with slope continuity, and Kansy reports that its results are quite acceptable. In the situation where there are actual discontinuities of curvature in the curve which the dense data represents, one would expect it to ripple, however. Consider the example of a straight line joining a circle, approached along the straight line. Let the tolerance permitted be ϵ and the radius of the circle r . The the algorithm can continue some way past the actual change point before it discovers a point on the circle which cannot be incorporated in the first arc.



The distance will be $\sqrt{2er}$ when the straight line is very long. By this time the fitted arc makes an angle of $\sqrt{\frac{2e}{r}}$ with the true direction of the circle. The next arc fitted is therefore starting in the wrong direction, and can only extend a distance $2\sqrt{2er}$ round the circle. The whole of the rest of the circle, in fact, has to be fitted by pieces of that length, instead of in just one piece.

The behaviour of the interpolating profile, described in chapter 2.3 above, however, suggests a parallel between the tangent-continuous profile and the quadratic spline. The approximation of a function of one variable given by dense data by a quadratic spline is a relatively simple exercise, and so it looked promising to explore the less intuitive possibility further. This proved to give a very powerful way of reaching a good solution, provided that the positions of the change-points along the data could be determined reasonably well beforehand.

Arc spline approximation

It is convenient to approach this from the general curve fitting procedure. We take here the least squares fit although the generalisation to the minimisation of the sum of the $2n^{\text{th}}$ powers of the errors, as suggested by Fletcher, Grant and Hebden, works well in this particular context, giving a good approximation to the minimax fit.

General method

To apply this method it is necessary to have some means of evaluating the error of each data point from a

postulated curve. Let the error at the i^{th} point be e_i

It is also necessary for the postulated curve to be subject to a number of perturbations, each controlled by a variable

amplitude. Let the amplitude of the j^{th} perturbation be

w_j

The perturbation giving the least sum of the squares of the errors is that set of amplitudes which satisfies

$$\sum_i e_i \frac{de_i}{dw_j} = 0 \quad \text{for all } j$$

Let the value of any quantity at the postulated configuration be indicated by a prescript 0 , thus the postulated value of e_i is e_{0i}

Then the best estimate of the value of

$$\sum_i e_i \frac{de_i}{dw_j}$$

at the perturbed position corresponding to a particular set of amplitudes w is

K

$$\sum_i \left[e_{ij} \left(\frac{de}{dw} \right)_j \right] + \sum_k \left[\left(\frac{de}{dw} \right)_j \left(\frac{de}{dw} \right)_k w_k \right] + \sum_k \left[e_{ij} \left(\frac{d^2e}{dw^2} \right)_{jk} w_k \right]$$

Equating this to zero for each j and solving the resulting equations for the w_k then gives

$$w_k = \left[\sum_j \left[\sum_i \left(\frac{de}{dw} \right)_j \left(\frac{de}{dw} \right)_k + e_{ij} \left(\frac{d^2e}{dw^2} \right)_{jk} \right] \right]^{-1} \left[\sum_i e_{ij} \left(\frac{de}{dw} \right)_j \right]$$

If the system is linear in the w values (the normal case encountered in text-books) the second derivative term is zero and the w values thus calculated give the exact solution. In the non-linear case the w values thus calculated merely give what is hoped to be a better fit than the originally postulated one. When the non-linearity is small the convergence is rapid (quadratic once the solution is close enough), in common with other applications of Newton-Raphson iteration.

Application

Consider the situation that we have an initial set of change points P_0 to P_n with the tangent direction at P_i being V_i . In the case of a tangent continuous curve it is reasonable to expect V_0 and V_n to be fixed externally. We have to improve the positions of the P_i so that the fit to a dense set of data points is better.

To apply the general method outlined above we need to be able to evaluate the errors at the data points. In this context this can be done by using algorithm xix of chapter 2.2 above to apportion the data points to specific spans of the postulated profile, and then algorithm xviii to evaluate the actual errors.

We then need a set of possible perturbations.

The vector at P_i tangent to the circle through

P_{i-1} , P_i and P_{i+1} turns out to play a key role in this method

It is given by the expression

$$S_i = \frac{([P_{i+1} - P_i] \cdot [P_{i+1} - P_{i-1}]) [P_{i+1} - P_i] + ([P_{i+1} - P_{i-1}] \cdot [P_{i+1} - P_i]) [P_{i+1} - P_{i-1}]}{([P_{i+1} - P_{i-1}] \cdot [P_{i+1} - P_{i-1}])}$$

and has three properties relevant to this problem.

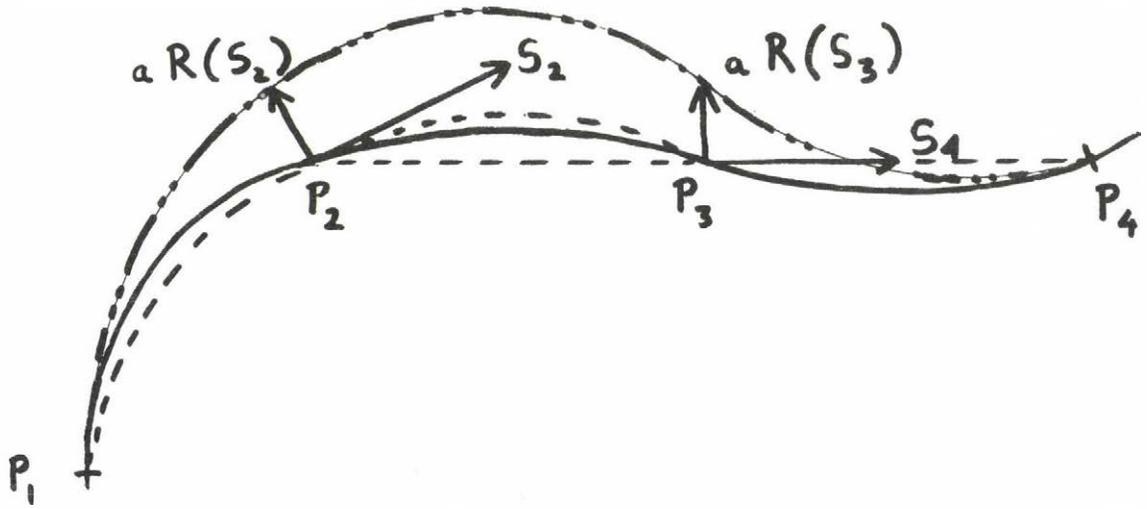
- 1) Small displacements of P_i along S_i have only a second order effect on the final direction of the profile V_n calculated from V_u and the P_i assuming tangent continuity. The circle is the locus in the tangent triangle based on P_{i-1} P_{i+1}
- 2) Small displacements perpendicular to S_i have the greatest effect on the final direction.

3) The effect of such small displacements is inversely proportional to the magnitude of the S_i as given by the above expression.

Thus if P_i is moved by the addition of $a_i R(S_i)$ the final direction V_n will be rotated through $\pm a_i$, the sign depending on the number of spans, odd or even, between P_i and P_n .

The first usage of this vector is that if we move P_i by $a_i S_i$ and P_{i+1} by $a_{i+1} S_{i+1}$ the net result outside the range P_{i-1} to P_{i+2} will be zero. Such a perturbation is the close analogue of the quadratic B-spline basis function. We can set up $n-2$ of these perturbations by setting $i = j$, $a_j = w_j$ for each j in turn from 1 to $n-2$. Each such perturbation leaves V_n unchanged (to a first order) and so does any linear combination. The general theory outlined above can therefore be applied, and does, in fact, work very well.

In practise, of course, the second order terms are not zero, and so V_n does get displaced at each step. A correction procedure is therefore necessary, and the S vectors are again applied.



The displacement pattern which gives maximum change of V_n for a minimum sum of squares of the displacements of the change points is to add to each P_i the correction $\pm h s_i / (S_i \cdot S_i)$ the signs being alternately plus and minus along the curve. The magnitude h is chosen by equating the total change in final tangent direction

$$\sum_i h / (S_i \cdot S_i) = h \sum_i 1 / (S_i \cdot S_i)$$

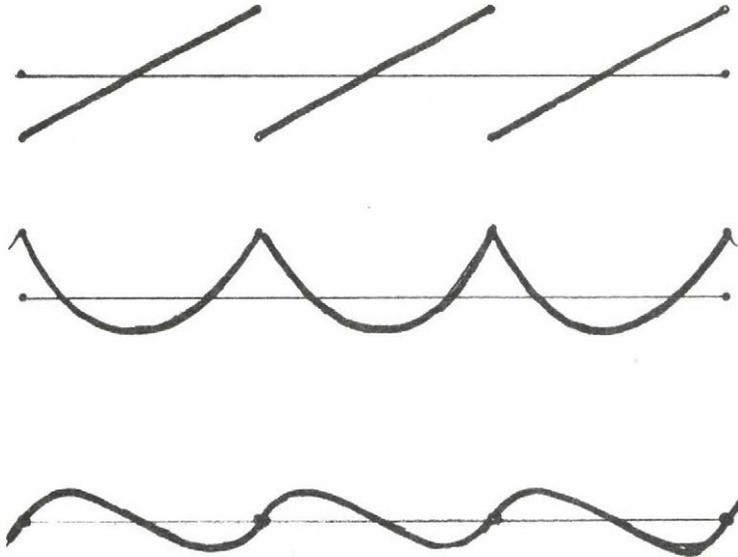
to the required change in angle. This correction is itself iterative, and must be repeated until the discrepancy of final direction is acceptably small. In practice this typically means one or two steps.

Initial estimate

The above depends heavily on having a reasonable initial estimate of the changepoint positions. Such an initial estimate needs to provide changepoints well distributed along the profile. It should not provide far too many, it must not provide too few.

The algorithm suggested above for fitting a tangent-discontinuous profile, run with half the actually required tolerance, is an obvious candidate, and can be applied quite generally. In the case where tangent directions are available at all the data points, however, as will be the case if explicit equations or other computations provide the data, the following method is recommended.

Consider a piece of Cornu spiral, in which curvature is a linear function of arc length, of length S , and varying in curvature from C to $C + \delta C$. Approximate the plot of C against S by a histogram, using n equal steps. The plot of error in C will be a sawtooth curve, that of error in angle (the integral of C with respect to S) will be a series of quadratics, and that of the lateral position error (the second integral) a series of cubics, joining with continuity of slope and position.



The maximum error may readily be evaluated to be

$$\frac{\delta C S^2}{72 \sqrt{3} n^3}$$

Thus to fit the spiral with a maximum error e the necessary number of pieces is given by

$$n^3 > \frac{\delta C S^2}{72 \sqrt{3} e}$$

The algorithm will arrive at the discontinuity with $\delta C = 0$, but with S quite large, and will therefore be triggered into creating a change point as soon as even quite a small change in C is encountered. Slight sophistication could look at the mean value and its two neighbours and determine almost exactly where the new change point should be placed. If the mean value lies outside the range of its neighbours, there is a very strong indication that a discontinuity of slope has been encountered, requiring two change points to be deposited.

Whichever initial estimate method is used, it will be necessary to apply the process for correcting the final direction before using the estimate for further refinement.

Longitudinal perturbation

It is a weakness of the method so far described that the distribution of the changepoints along the curve cannot be improved. There is a prima facie possibility of using displacements of the P_i along the S_i to provide perturbations which can be used to improve this aspect. When this was tried, the algorithm failed to converge at all rapidly. This may have been due to faulty analysis, to coding errors, or merely to the considerably greater non-linearity in the longitudinal perturbation case.

The absence of this facility is not significantly embarrassing in practical applications, since a small increase in the number of pieces is sufficient to reduce the variation of the profile from an optimum fit by the necessary factor, and the initial fit algorithm gives extremely good distributions.

It is, however, an aesthetic blemish.

Conclusions

The curve formed from circular arcs and straight lines is a very practical primitive for two-dimensional geometric computation in mechanical engineering. Such curves may be stored very economically, and may be generated as a canonical form from the other specifications which might be presented. This unified representation considerably simplifies the application of the stored geometry.

Recommendations for further research

The methods described above are felt to be adequate for application as they stand, without any further development. Improvements, however, might well be made by work in three areas. One is the formulation of the calculation of directions in the interpolation case, where at present the setting up of the triband matrix requires the calculation of many trigonometric functions, and is therefore relatively time-consuming.

The second is the search for a strategy for locating the changepoint in the tangent triangle which has the advantages of the incentre, but which avoids the singular case encountered when three data points lie on one straight line.

The third is the longitudinal perturbation of the change points when approximating dense data.

The application of the methods described should not wait, however, on these improvements, which, though desirable, will have only a marginal effect on the usefulness of the methods now available.

3.1 Background

Again we find the same two key points, the need for a representation of an object as a whole, and the idea that spline theory might help toward this.

There have been two distinct streams of development in the representation of 3D shape. One, typified by the numerical control part programming languages, deals with items whose complications stem from the number of faces and the patterns of their connectivity rather than from the complexity of the individual faces. Planes, cylinders and cones, with toroids and, possibly, blends between two cylinders form the bulk of all shapes defined today.

Current NC languages deal with the individual surfaces, leaving their interactions to the part programmer, but a number of experimental systems have been devised, Braid[1973],Engeli[1974],Hosaka[1974], Okino[1973] and Voelcker[1974] being among the most significant. All these concentrate on the relationships between the faces at the expense, to some extent, of face complexity, whereas previous attempts to handle this class of part [Comba 1965,1967],[Luh and Krolak 1965],[Weiss 1966] and [Woon 1971] tended to regard face complexity as the problem.

Sculptured surfaces

The other stream of development deals with smooth surfaces of which the qualitative features, rather than the forms of the equations, are of primary interest. Such surfaces are termed sculptured surfaces.

Graphical methods of dealing with these shapes were developed by Monge [Robertson 1966] in the late 1770s and are still widespread in the design of boats and small ships. They involve drawing out a series of the cross sections of the shape cut by parallel planes. These cross sections are also the positions of structural members, and so the method gives much of the required manufacturing data automatically and directly.

The principal disadvantage of graphical methods is that of accuracy, particularly when applied to larger shapes. If the scale is to be kept high enough for adequate precision large areas of working space need to be dedicated to the purpose, which becomes increasingly expensive.

During the 1940s a direct numerical analogue of the graphical techniques was developed for aircraft design, which used second degree curves (conic sections) for the cross section curves, and which was therefore termed conic lofting. The earliest document I have seen is Shelley[1946] but I believe that similar work had been done earlier in the U.S.A. These methods have also been reinvented many times since, in other contexts. [Lidbro 1956,1961],[den Hartog 1970]

In shipbuilding, where conic lofting could not be applied because second degree curves were not so appropriate for the shape as for aircraft, the development of numerical methods came later. [Clenshaw 1965] and [Theilheim and Starkweather 1961] are fairly typical of the methods being developed around 1960. The Autokon system, which uses a true spline for interpolation and for smoothing [Mehlum 1969] is probably the most widely used. Autokon is, in fact, a little unusual, in that it does not use functions of distance aft (x) and up (z) for the half-breadth (y) of the hull form. Many proprietary systems used linear splines to express the functional dependence of y on x along each horizontal section (waterline), and that of y on z along each frame.

This approach shares two disadvantages with conic lofting as originally devised. First, the shape is defined only at a set of curves in space, and second, the shape is dependent on the axes chosen for the definition. Rotation of the basic data through a small angle before fitting the surface gives a different answer from rotating the surface afterwards, and points with infinite derivative (vertical slopes) need special treatment.

Fully defined surfaces

The former problem may be avoided in two ways. One is to use a set of surface patches to fill in the holes in the net of curves. This approach has been suggested by Birkhoff and Garabedian[1960], by Coons[1967] and by Bezier[1971]. The Coons approach has the noteworthy feature that the edges of the patches may have any equation, and are not constrained by the patch equation. Most other methods of this class place restrictions on the curve equations which can be matched by the patches.

The Coons patch and the Birkhoff and Garabedian patch are both designed to give slope continuity across the patch boundaries automatically. Veron[1973] and Ris[1975] have investigated the constraints for continuity of Bezier patches.

The other way to avoid the holes is to define a complete surface by having, instead of a finite set of curves a continuum. When this is done either the conic lofting method or the linear spline shipbuilding method becomes a directrix-generator method. This class of methods has three elements:-

- (i) a set of directrix curves in space
- (ii) a correspondence rule whereby, for any point on any one of the directrices we can determine a unique point on each of the others
- (iii) an interpolation rule for passing a generator curve through each set of corresponding points.

Each of these three elements may be piecewise defined, and in general a discontinuity of any order in any of the three will cause a discontinuity of the same order in the surface. Discontinuities will normally lie along either directrices or generators, and one may regard such lines of discontinuity as patch boundaries.

Viewing the surface globally as being the result of piecewise generating functions tends to give either better continuity or better controllability than sewing patches together to make the whole. The two-way interpolating spline technique used in the British Aircraft Corporation Numerical Master Geometry system [Sabin 1971] may be regarded as a directrix-generator method giving continuity of both first and second derivative, whereas the Ferguson [1964] method used in APTLEFT-FMILL which fits identical bicubic patches into a net of spline curves, gives only first derivative continuity. The Gordon[1970] spline blending methods are a similar advance on the Coons patch method, and the B-spline techniques developed by Gordon and Riesenfeld may be regarded as the proper context for Bezier patches.

Parametric representation

The latter problem, of the axis dependence of the numerical definition, disappeared with the application of parametric methods in the early 1960s. Although the fundamental idea, of letting all three coordinates of a general surface point be functions of two free variables or parameters had been worked

out in some detail and used by Gauss in the 1820s, it appeared to be applied first to shape description by Ferguson[1964] and Coons[1967].

Parametric representation gives complete axis-independence because a coordinate in any axis is given by a linear combination with finite coefficients of the coordinates in any other axis set. Any surface of the form

$$\begin{aligned} x &= \sum_i a_{xi} f_i(u,v) \\ y &= \sum_i a_{yi} f_i(u,v) \\ z &= \sum_i a_{zi} f_i(u,v) \end{aligned}$$

(a very appropriate form for many reasons) will retain that form under any affine transformation. This axis-independence then avoids all problems of vertical slopes, because a point of vertical slope in one axis system is a perfectly normal point in most others.

The form above may be written in vector notation as

$$P = \sum_i A_i f_i(u,v)$$

the f being termed basis functions and the A the coefficients. In a computer based shape representation system the f will be written as subroutines and the A will be stored, different values corresponding to differently shaped surfaces.

The set of functions chosen for the f controls how the values of the A influence the shape represented.

At the end of this chapter we shall be examining the effects of various properties of the f on the way the A control the shape.

The two possibilities, of putting patches together to make the complete shape, or of having a global, but piecewise definition, are still available in the parametric representation. Practical systems have been built using each, and have been applied to the representation of aircraft [Sabin 1971],[Walter 1973], cars [Bezier 1971], ship hulls [MacCallum 1970,1972][Yuille 1970,1972], marine propellers [Thorne 1973][Klein 1975], turbine blades [Pochop 1974], impellers [Smith 1973] and experimental multivariate data [Altham 1970].

All these applications, however, have used formulations in which the patch boundaries lie always along lines of constant parameter. This places a certain amount of restriction on the design of some shapes. The remaining chapters of this section explore the possibility of having a patch structure other than a set of four-sided tiles with four tiles meeting at every corner.

Necessary properties

In looking for possibilities to develop, it is essential to retain the best features of existing methods. In particular the natural control of the Bezier and B-spline methods is extremely valuable. In these systems we have surface equations of the form described above where the coefficients are a set of control points lying near, but not usually on, the surface.

Displacement of any control point causes each surface point to move either not at all, or else in the same direction. This corresponds to the condition

$$f_i \geq 0$$

If all control points are equally displaced, the result should be a displacement without distortion of the entire surface. This gives

$$\sum f_i = 1$$

Beyond this there is only the intuitive condition shared by both Bezier and B-spline methods, that each basis function should have a single maximum (the mode centre), at which that function is considerably larger than any other. For interpolation through data points to work, the matrix of all function values at all mode centres needs to be well-conditioned, which implies that the basis functions should be linearly independent.

As the B-spline method shows, the basis functions do not need to be analytic. They can have piecewise definitions, when the continuity of the surface will in general be the same as that common to all the basis functions. (It may be higher, if the discontinuity vector lies in the tangent plane of the surface, and may be lower if the first derivative in any direction vanishes) If the definitions are piecewise the condition of axis independence demands that every patch boundary is shared by at least two of the basis functions.

The localisation property, that each basis function should be non-zero only inside a single connected region appears to be advantageous.

Although the Bezier and B-spline basis function sets are both Cartesian products of univariate bases this is not a necessary feature. Indeed, it is not strictly necessary for the basis functions to be defined over a parametric plane. Any 2-manifold in a higher space can act as domain. The condition $\sum f_i = 1$ need only hold over the domain itself, rather than throughout the higher space. Consider the manifold in (u,v,w) space

$$u^2 + v^2 + w^2 = 1$$

The six basis functions

$$(1 \pm u)^2, (1 \pm v)^2, (1 \pm w)^2$$

all divided by 8 form a set defining a distorted sphere. The shape of the sphere can be made exact by placing the coefficients at the vertices of an octahedron.

We will consider here, however, more general situations by looking at piecewise definitions with triangular pieces. The next chapter describes a numerical characterisation of triangular pieces of polynomial surface, the last two the ways in which these may be used in a global definition.

3.2 Triangular Polynomial Surfaces

We wish to be able to represent a triangular piece of surface. There is complete symmetry between the three sides, and so we look for a symmetric form. If we use not two, but three parameters u, v, w say, with the relationship between them $u + v + w = 1$ the region $u, v, w \geq 0$ is a triangle. This can be viewed either as a plane 2-manifold in 3-space or as a set of normalised homogeneous coordinates in the triangle [Maxwell 1963].

Just as the terms of $(u + (1-u))^n$ give the Bezier basis functions of order n , so the terms of

$$(u + v + w)^n$$

give basis functions for the triangular piece of surface.

A typical term is $f = \frac{u^i v^j w^k}{i! j! k!}$

where $i + j + k = n$

Clearly, because $u, v, w \geq 0$ inside the triangle this function is also not less than zero, thus satisfying our first necessary condition. Also the sum of all the terms

is just by definition $(u + v + w)^n = 1^n = 1$ thus satisfying the second condition also.

Differentiating the function with respect to position in the parameter plane gives

$$\delta f = \frac{i f}{u} \delta u + \frac{j f}{v} \delta v + \frac{k f}{w} \delta w$$

but $\delta w = -\delta u - \delta v$ and so for any $\delta u, \delta v$

δf will be zero if $\frac{i}{u} - \frac{k}{w} = 0 = \frac{j}{v} - \frac{k}{w}$

Thus $u = \frac{i}{n}$ $v = \frac{j}{n}$ $w = \frac{k}{n}$ gives a stationary value which is in fact a single maximum. The mode centres are therefore distributed evenly on a triangular lattice within the triangle $u, v, w \geq 0$ thus satisfying the third condition.

If i, j or $k = 0$ the conditions $u = i/n$ etc. still give the maximum value within the triangle $u, v, w \geq 0$ although the point is not stationary.

Notation

For curves the Bezier polygon corners will be denoted by $B_{i, n}$ it being understood that i runs from zero to n for the full set. This notation will also be used for the individual rows of a rectangular or triangular network.

A complete rectangle has its coefficients denoted by

$$B_{ij}^{mn} \quad (0 \leq i \leq m, 0 \leq j \leq n)$$

and the coefficients of the basis functions defined above for a triangle are denoted by

$$B_{ijk}^n \quad (0 \leq i, j, k \leq n \quad i + j + k = n)$$

Thus the equation of a general surface point is

$$P(u, v, w) = \sum_{i, j, k \geq 0}^{i+j+k=n} B_{ijk}^n \frac{n! u^i v^j w^k}{i! j! k!} \quad u+v+w=1 \quad u, v, w \geq 0$$

Diagrammatic Representation

The properties of the more familiar Bezier curves and rectangles are mirrored very closely by this formulation.

The B_{ijk}^n transform as points under affine transformations

of the surface. The points B_{n00}^n , B_{0n0}^n and B_{00n}^n are the positions

of the corners of the surface $P(1,0,0)$, $P(0,1,0)$ and $P(0,0,1)$.

The rows B_{ij0}^n , B_{0jk}^n and B_{i0k}^n are the Bezier polygons rep-

resenting the sides $P(u,1-u,0)$, $P(0,v,1-v)$ and $P(u,0,1-u)$.

The complete set of B_{ijk}^n may be visualized as a

triangular network exactly analogous to Beziers reseau for rectangular pieces of surface. Movement of any one of the

B_{ijk}^n influences most the nearest parts of the triangle. The

surface is a smoothed out version of the network, and the network exaggerates the qualitative behaviour of the surface.

Scalar Functions on a Triangle

If a scalar polynomial function on a triangular domain is represented in this form, the coefficients (which are now scalars instead of vectors) may be laid out at the appropriate points of the triangle. Both the precise numerical definition and the broad qualitative features of the shape will then be visible.

The individual basis functions themselves are represented by having a 1 at one of the lattice points and 0 at each of the others.

For example:-

$$\begin{array}{cccc}
 & 0 & & \\
 & 0 & 1 & \equiv u \\
 & & & \\
 & 0 & & \\
 & 0 & 0 & \equiv u^2 \\
 0 & 0 & 1 & \\
 & & & \\
 & 0 & & \\
 & 1 & 0 & \equiv 2vw \\
 0 & 0 & 0 & \\
 & & & \\
 & 0 & & \\
 & 0 & 0 & \\
 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 \equiv 12u^2vw \\
 0 & 0 & 0 & 0 & 0
 \end{array}$$

Derivatives

For any function P of three variables it is true that

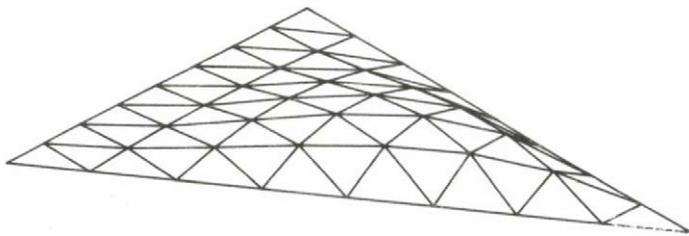
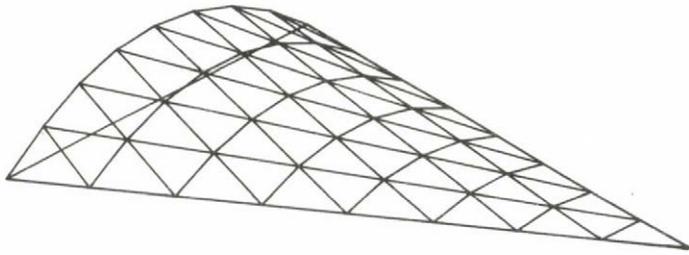
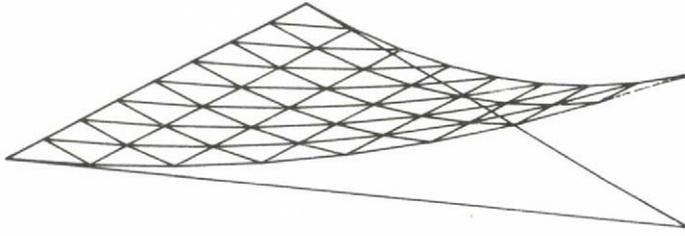
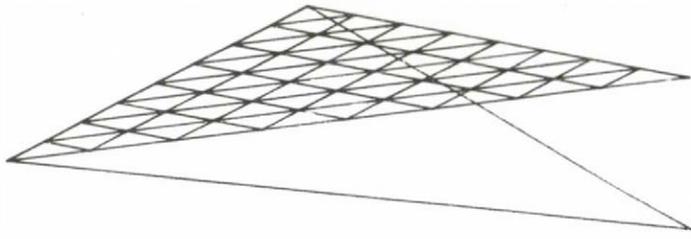
$$\frac{dP}{dt} = \frac{\partial P}{\partial u} \frac{du}{dt} + \frac{\partial P}{\partial v} \frac{dv}{dt} + \frac{\partial P}{\partial w} \frac{dw}{dt}$$

Let $\frac{du}{dt}$, $\frac{dv}{dt}$ and $\frac{dw}{dt}$ be written as a, b and c

Because $u + v + w = \text{const}$ $a + b + c = 0$

$$\begin{aligned}
 \text{Now } \frac{\partial P}{\partial u} &= \sum_{i>0; j, k \geq 0}^{i+j+k=n} B_{ijk}^n \frac{i^{i-1} j^j k^k}{i! j! k!} \\
 &= n \sum_{i, j, k \geq 0}^{i+j+k=n-1} B_{(i+1)jk}^n \frac{i^i j^j k^k}{i! j! k!}
 \end{aligned}$$

by changing $i-1$ for i as the summation variable.



Similarly

$$\frac{\partial P}{\partial v} = n \sum_{i,j,k \geq 0}^{i+j+k=n-1} B_{i(j+1)k}^n \frac{n-1! u^i v^j w^k}{i! j! k!}$$

$$\frac{\partial P}{\partial w} = n \sum_{i,j,k \geq 0}^{i+j+k=n-1} B_{ij(k+1)}^n \frac{n-1! u^i v^j w^k}{i! j! k!}$$

and so $\frac{dP}{dt}$ may be represented by a similar function

whose coefficients may be expressed as

$$D_{ijk}^{n-1} = n \left(a B_{(i+1)jk}^n + b B_{i(j+1)k}^n + c B_{ij(k+1)}^n \right)$$

This general equation has certain special cases of importance. In particular the slope parallel to any of the edges can simply be expressed. The slope parallel to $v = \text{const}$, for example, is given by the case $a = 1$
 $b = 0$ $c = -1$

$$D_{ijk}^{n-1} = n \left(B_{(i+1)jk}^n - B_{ij(k+1)}^n \right)$$

Continuity with adjacent surfaces

Basic properties

Two results from curve and rectangle theory will be of use in developing the continuity properties of triangles.

- 1) A Bezier curve of order m may be expressed as one of order n ($n > m$) by multiplying the expression for a general point by $((1-u) + u)^{n-m}$ and regrouping the terms.

$$\begin{aligned}
 \text{i.e. } & \sum_0^n \frac{{}^n B_i \cdot n! (1-u)^i u^{n-i}}{i! (n-i)!} \\
 &= \sum_0^m \frac{{}^m B_j \cdot m! (1-u)^j u^{m-j}}{j! (m-j)!} ((1-u) + u)^{n-m}
 \end{aligned}$$

Two special cases of this general equality are of particular significance.

(i) If $n = m+1$

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

where the undefined ${}^m B_{-1}$ and ${}^m B_{m+1}$ have zero coefficients

and are thus not actually used.

(ii) If $i = 1$ for any m, n , $n > m$

$${}^n B_1 = \frac{(n-m) {}^m B_0 + m {}^m B_1}{n} \quad \text{and} \quad {}^n B_0 = {}^m B_0$$

2) For continuity of tangent plane between two adjacent patches (P and Q, say) along their common edge ($u = 0$, say) we must have

$$Q = P$$

$$dQ/dv = Q_v = P_v$$

$$dQ/du = Q_u = f_1(v)P_u + f_2(v)P_v$$

If P , P_u and Q_u are all of order n in v , P_v will be of order $n-1$ and we have as a possible solution

$$f_1(v) = \text{constant} = a$$

$$f_2(v) = \text{linear in } v = b(1-v) + cv$$

where a , b and c may be arbitrarily chosen without spoiling the continuity.

Suppose that the common edge has the Bezier

polygon B_i^n , that the surface P has as its next

row A_i^n , and that the surface Q has as its next row

C_i^n .

$$\text{Then } (C_i^n - B_i^n) = a(A_i^n - B_i^n) + \frac{1}{n}b(B_i^n - B_{i-1}^n) + \frac{(n-1)}{n}c(B_{i+1}^n - B_i^n)$$

In the rational case there is a fourth term, but this does not add any geometric degrees of freedom to the system because it merely corresponds to reparametrisation of Q .

We shall apply these results in finding conditions for triangles to be continuous in slope by transforming each triangle locally into a rectangle.

There are two operations which form a rectangular surface from a triangular one. The first is extension, whereby an additional piece of surface extends the triangle into a rectangle. The second is singularity, whereby the triangle is interpreted as a singular parametrisation of a rectangle in parameter space.

Triangle to Rectangle Transformations

1) Extension.

Set $w = 1 - u - v$ in the equation for a general point of the triangular surface, and then regroup the equation in powers of u and v . The result is a bi-n-ic in u and v , and may itself be reexpressed in terms of the rectangular Bezier basis functions. The coefficients of these basis functions are expressions for the

$$B_{ij}^{nn} \text{ in terms of the original } B_{ijk}^n$$

In the linear case, for example, we can extend the unit triangle on the $w = 0$ side by the transformation

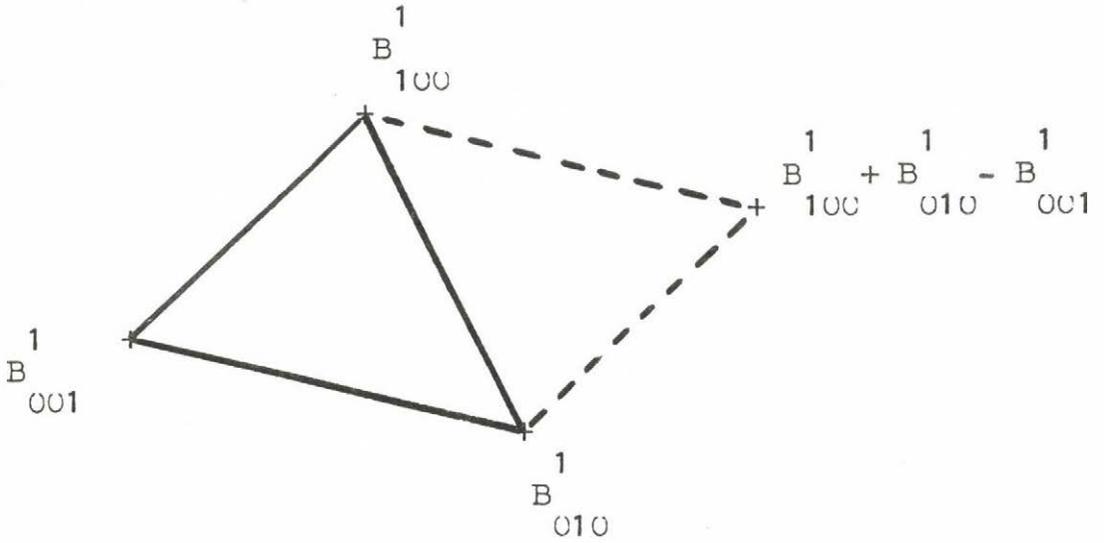
$$B_{00}^{11} = B_{001}^1$$

$$B_{01}^{11} = B_{010}^1$$

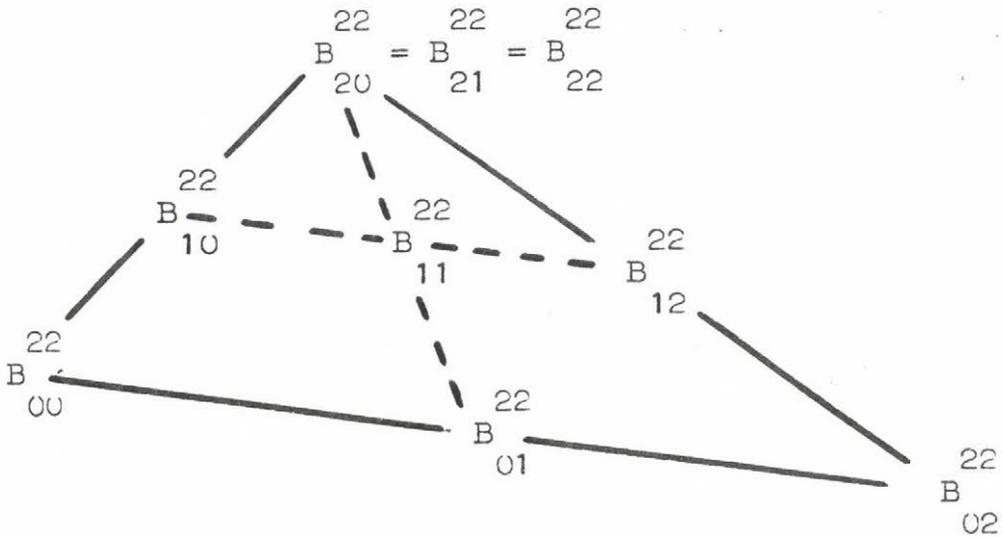
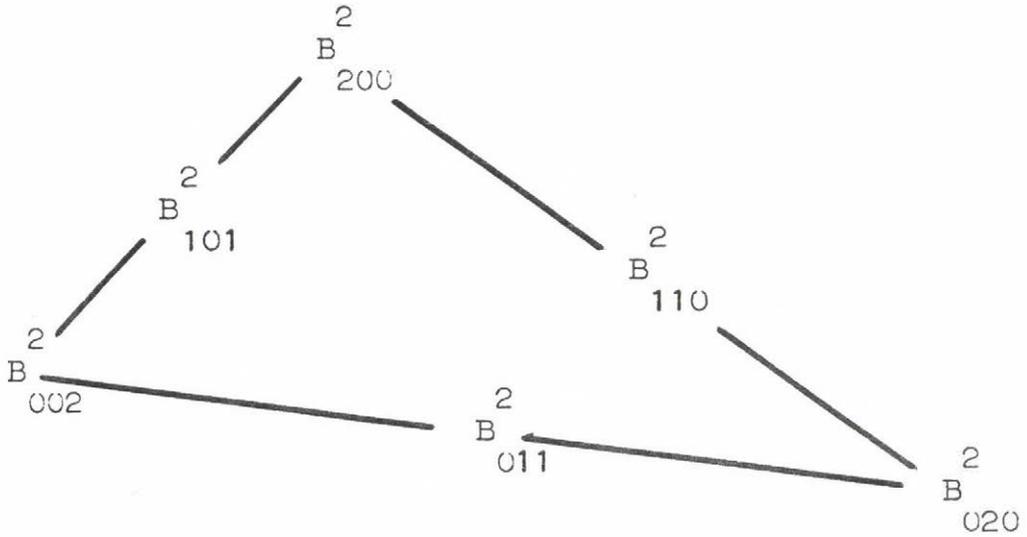
$$B_{10}^{11} = B_{100}^1$$

$$B_{11}^{11} = B_{100}^1 + B_{010}^1 - B_{001}^1$$

Similar transformations can be applied extending either of the other sides. The union of the results of all three applied independently to the same triangle forms a triangle rather than a hexagon.



Extension of triangle into rectangle.



Triangle and equivalent singular rectangle.

2) Singularity.

$$\begin{aligned} \text{Set} \quad u &= s \\ v &= (1-s)t \\ w &= (1-s)(1-t) \end{aligned}$$

and expand directly in terms of $s, 1-s, t$ and $1-t$. Where necessary multiply terms by $(t + 1-t)$ to make the result homogeneous in the four variables. Again, the coefficients of the rectangular Bezier basis functions are

expressions for the B_{ij}^{nn} in terms of the original B_{ijk}^n

This transformation does not extend the surface, but maps the unit triangle into the unit square in the $s t$ plane. The edge $s = 1$ is mapped into the single point $u = 1$ of the triangle, and so the rectangle is a singular one.

In the linear case we have

$$B_{00}^{11} = B_{001}^1$$

$$B_{01}^{11} = B_{010}^1$$

$$B_{10}^{11} = B_{100}^1$$

$$B_{11}^{11} = B_{100}^1$$

More generally

$$B_{0j}^{nn} = B_{0jk}^n$$

$$B_{1j}^{nn} = \frac{j B_{1(j-1)k}^n + k B_{1j(k+1)}^n}{j+k}$$

$$j+k$$

and subsequent rows B_{2j} , B_{3j} etc. are similarly given

by raising the order of the corresponding rows of the original B_{ijk}^n until all the rows are of order n , up to

$$B_{nj}^{nn} = B_{n00}^n \quad \text{for all } j$$

Similar transformations exist which place the singularity in either of the other corners of the triangle.

Continuity between adjacent Triangles

It is the singularity transformation which gives the most elegant condition for slope continuity between triangles.

Let B_i^n be the common boundary between P and Q and let

A_i^{n-1} and C_i^{n-1} be the next row in each as in the case of the rectangular surface considered above.

We can transform both triangles into rectangles with the singularities at the far corners, i.e. at the corners not on the common boundary. This will transform the representations so that we now have rows A and C given by

$$A_i^n = \frac{i}{n} A_{i-1}^{n-1} + \frac{(n-i)}{n} A_i^{n-1}$$

$$C_i^n = \frac{i}{n} C_{i-1}^{n-1} + \frac{(n-i)}{n} C_i^{n-1}$$

These are now substituted into the equation for continuity of two adjacent rectangles noted above

$$(C_i^n - B_i^n) = a (A_i^n - B_i^n) + \frac{ib}{n} (B_i^n - B_{i-1}^n) + \frac{(n-i)c}{n} (B_{i+1}^n - B_i^n)$$

This gives

$$\left\{ \begin{array}{l} \frac{1}{n} \binom{n-1}{i-1} \binom{n}{i} \\ + \\ \frac{(n-1)}{n} \binom{n-1}{i} \binom{n}{i} \end{array} \right\} = \left\{ \begin{array}{l} a \frac{1}{n} \binom{n-1}{i-1} \binom{n}{i} + b \frac{1}{n} \binom{n}{i} \binom{n}{i-1} \\ + \\ a \frac{(n-1)}{n} \binom{n-1}{i} \binom{n}{i} + c \frac{(n-1)}{n} \binom{n}{i+1} \binom{n}{i} \end{array} \right\}$$

Because there are only $n-1$ vector degrees of freedom with which to satisfy this equation, it is necessary for the two parts to be consistent. If $a + c = 1 + b$ the terms in $\frac{1}{n}$ are consistent with those in $\frac{(n-1)}{n}$ and the equation

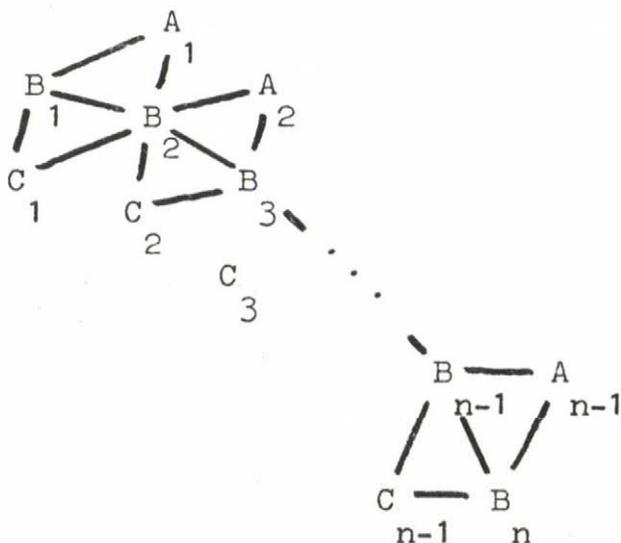
reduces to

$$\binom{n-1}{i} \binom{n}{i} = a \binom{n}{i} \binom{n-1}{i} + c \binom{n}{i+1} \binom{n}{i}$$

This equation may be interpreted geometrically as

saying that each configuration $\binom{n-1}{i} \binom{n}{i} \binom{n}{i+1} \binom{n-1}{i}$ should be

planar, and that all $n-1$ such configurations along the edge B should be affine transformations of each other.



The case $a = -1$ $c = 1$ gives a regular triangular lattice, which we shall examine further in the next chapter. It is not obvious that it is ever possible to choose the C_1^{n-1} to satisfy the requirements of continuity across more than one edge at once.

Continuity between a triangle and rectangle

Let B be the common edge of a triangle and an adjacent rectangle. Let the row C be the next row in the rectangle and the row A be the next row in the triangle.

When the triangle is transformed into a rectangle the slope continuity equation is obtained

$$\frac{1}{n} A_{i-1}^{n-1} + \frac{n-1}{n} A_i^{n-1} - B_i^n = a \left(\frac{1}{i} B_i^n - \frac{1}{i} C_i^n \right) + b \frac{1}{n} \left(\frac{1}{i} B_i^n - \frac{1}{i-1} B_{i-1}^n \right) + \frac{cn-1}{n} \left(\frac{1}{i+1} B_{i+1}^n - \frac{1}{i} B_i^n \right)$$

This system has n vector equations but only $n-1$ vector unknowns A_i^{n-1} and so these equations cannot simply be solved for the A . They can of course be used to calculate the rows B or C , but this is equivalent to transforming the triangle into a rectangle and then using the standard rectangle to rectangle constructions.

In order to achieve full freedom in fitting triangles to existing rectangles it is necessary to use a triangle of order $n+1$ whose boundaries, however, are of order only n . This can then be transformed into an $n \times n+1$ rectangle and have its coefficients chosen to match an existing order n rectangle.

3.3 B-Spline basis functions over a regular triangular lattice

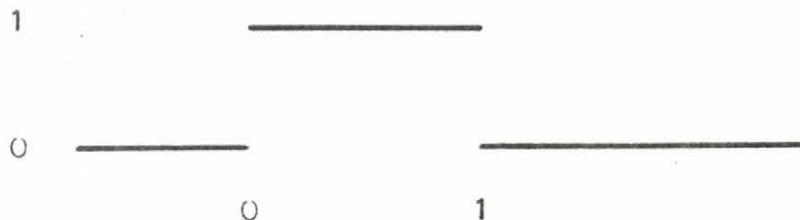
In the previous chapter we described a representation of triangular pieces of polynomial surface, and the conditions for such pieces to have continuity of slope. This is analogous to Bezier theory of curves and rectangular pieces of surface. We look here for an analogue to the B-spline theory of curves and rectangular pieces of surface.

There are a number of formulae by which the univariate B-spline basis functions may be generated. One of the most elegant algebraic methods is based on the convolution with a unit pulse.

The basis function of order n is smeared out by sliding it through one lattice unit, the mean being taken of the functions at all intermediate positions. This gives the prototype basis function of order $n+1$

$$S_{n+1}(t) = \int_{-\infty}^{\infty} H(t-u) S_n(u) du$$

where H is the unit pulse.



This can be integrated by parts to give

$$S_{n+1}(t) = \int_{-\infty}^t (S_n(u-1) - S_n(u)) du$$

i.e. The convolution operator which gives the successive members of the family of basis functions can be regarded as

the combination of two operators, each of which can be handled numerically. The first is a shift and subtract; the second is an integration.

The basis functions over a regular rectangular lattice are normally considered as being generated by the Cartesian product of two univariate families of basis functions. It is equally possible to regard them as being generated by two independent convolution operators.

It would be difficult to generate functions over a triangular lattice by taking products; the three way symmetry of the lattice does, however, allow a set of three independent convolution operators.

The question arises then as to the exact form of the numerical operations corresponding to the shift and subtract and the integration operators.

The shift and subtract is relatively obvious. The shift merely involves applying the coefficients of a particular triangle to the part of the parameter plane displaced by one lattice unit. Because the form developed in the last chapter for the representation of individual triangles is linear in the coefficients, the subtraction of the basis function corresponds to subtraction of the coefficients.

It turns out that the integration operator also has a simple numerical form. The integral of a surface of order n is a surface of order $n+1$. The actual integral needs a constant of integration, which is a function along the starting edge of the triangle. This is represented by the $n+1$ coefficients along that edge. There remain the coefficients of the triangle of side n which complete the order $n+1$ triangle to compute. These are determined by successive addition of the coefficients of the function being integrated.

Suppose that the integration is along the direction parallel to $u = \text{const}$, from $v = 0$ towards $w = 0$. The constant of integration is the curve

$$B_{10k}^{n+1}$$

and the integration algorithm is

$$B_{i(j+1)k}^{n+1} = B_{ij(k+1)}^{n+1} + B_{ijk}^n$$

The factor $1/n+1$ is omitted here in order to keep the numbers integral. The formula for integration should be compared with that for differentiation in chapter 3.2 above.

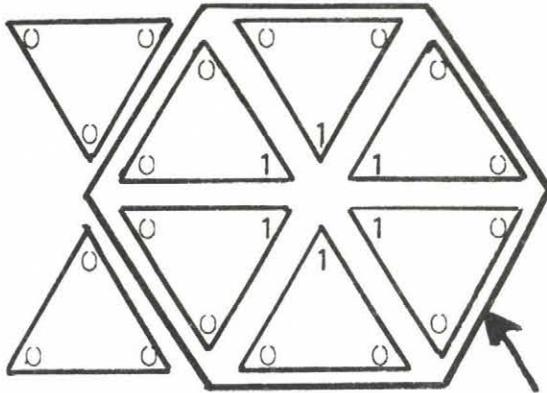
The constant of integration for each triangle in the lattice is determined from the triangle immediately upstream in the integration over the complete lattice. Far enough upstream we can assume a zero curve, because we are looking for basis functions with a bounded non-zero region.

Although the integration should be taken from minus infinity we need only start at the boundary of the non-zero region of the integrand. We term the shape of this boundary the Planform of the function.

These operators are best illustrated by example. Take the first order basis function, which can be set up intuitively to be a hexagonal pyramid. Each face is represented by the coefficients of the functions u , v and w , and the numeric values of those coefficients are associated with the points in the triangle at which each function has its maximum value, and we may use this association in a diagrammatic representation by drawing the triangular lattice with the values entered at the appropriate points.

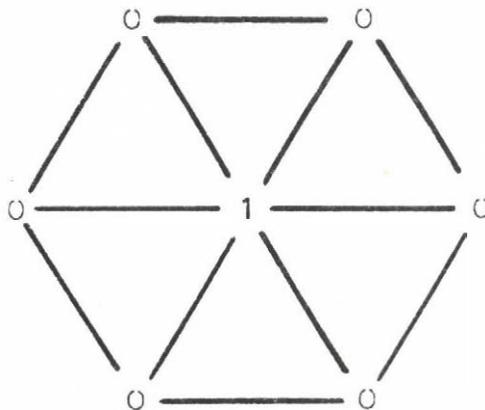
Because we have continuity of position the edge and corner coefficients are common to adjacent triangles, and need only be written once, actually on the edge or corner.

Diagrammatic Notation



Planform. All triangles outside this have all coefficients zero

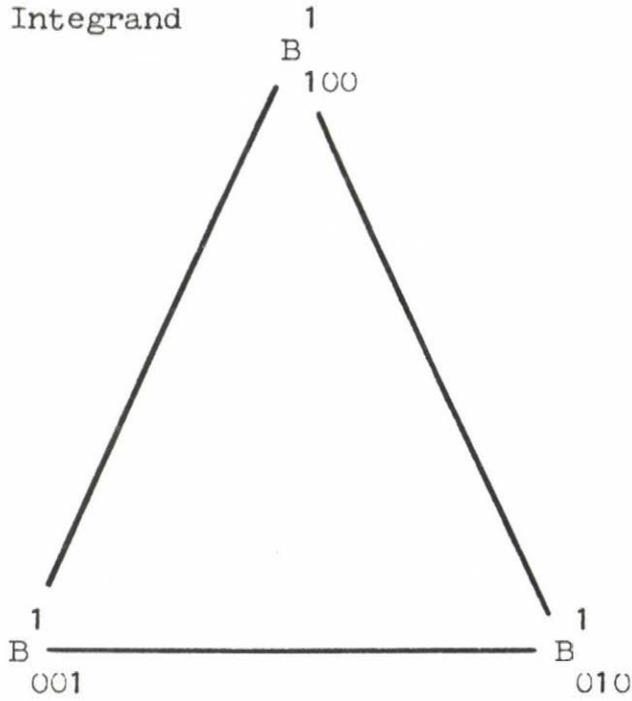
Because of the position continuity this picture could be drawn as



it being understood that regions not shown are all zero.

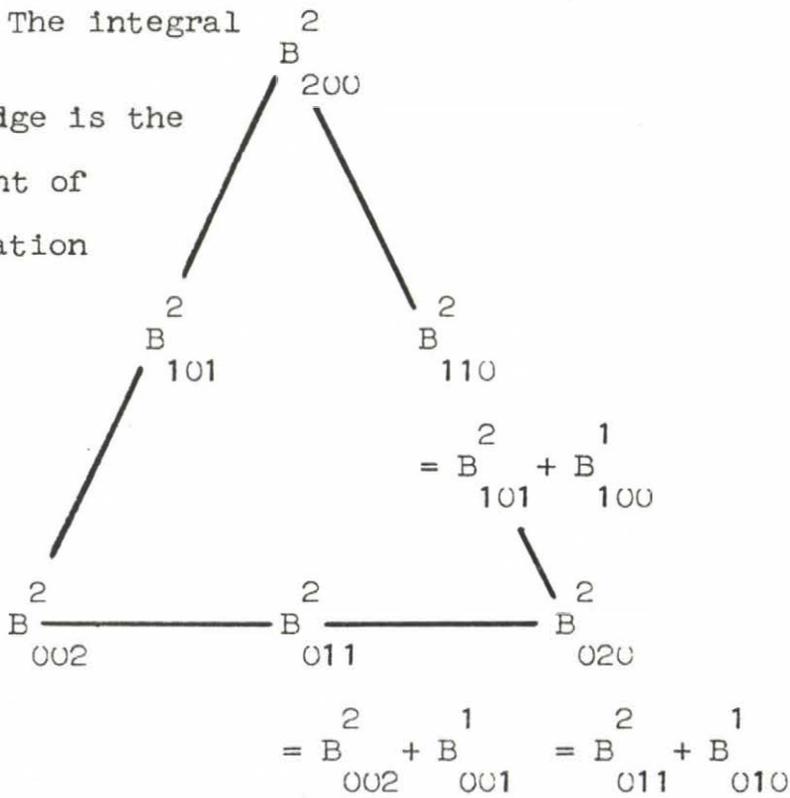
Integration

The Integrand

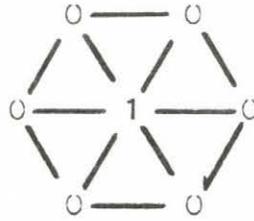


The integral

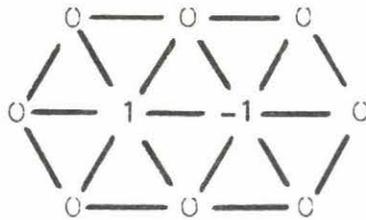
This edge is the constant of integration



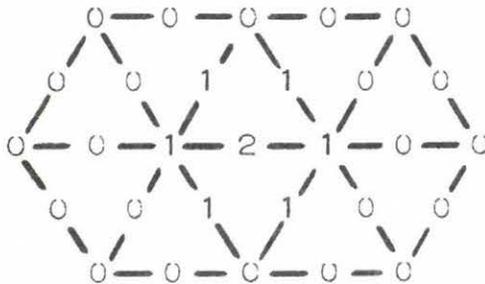
Using this diagrammatic notation we can follow through the generation of successive basis functions, as a sequence of diagrams. Starting from the first order hexagonal pyramid



the shift and subtract gives



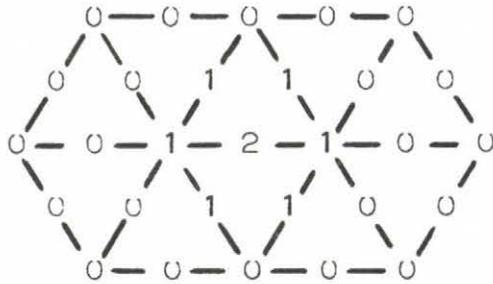
and integration increases the order of the triangular pieces giving



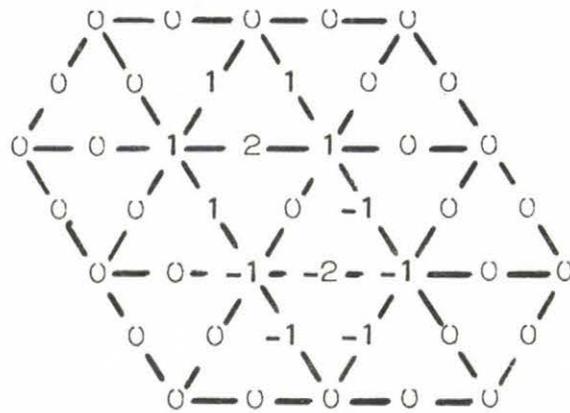
It may easily be confirmed that we now have continuity of slope over all the edges across which we have integrated, but only continuity of position across the edges along which we integrated.

Applying a second convolution operator in the same direction would increase the order of continuity still further across those edges where it has been increased already. It is more useful therefore to apply a convolution in a different direction.

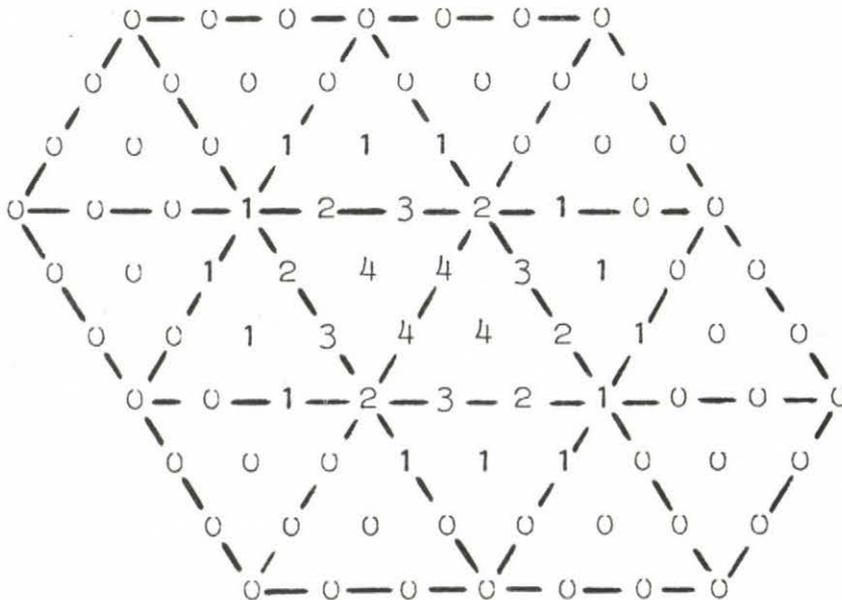
Starting from



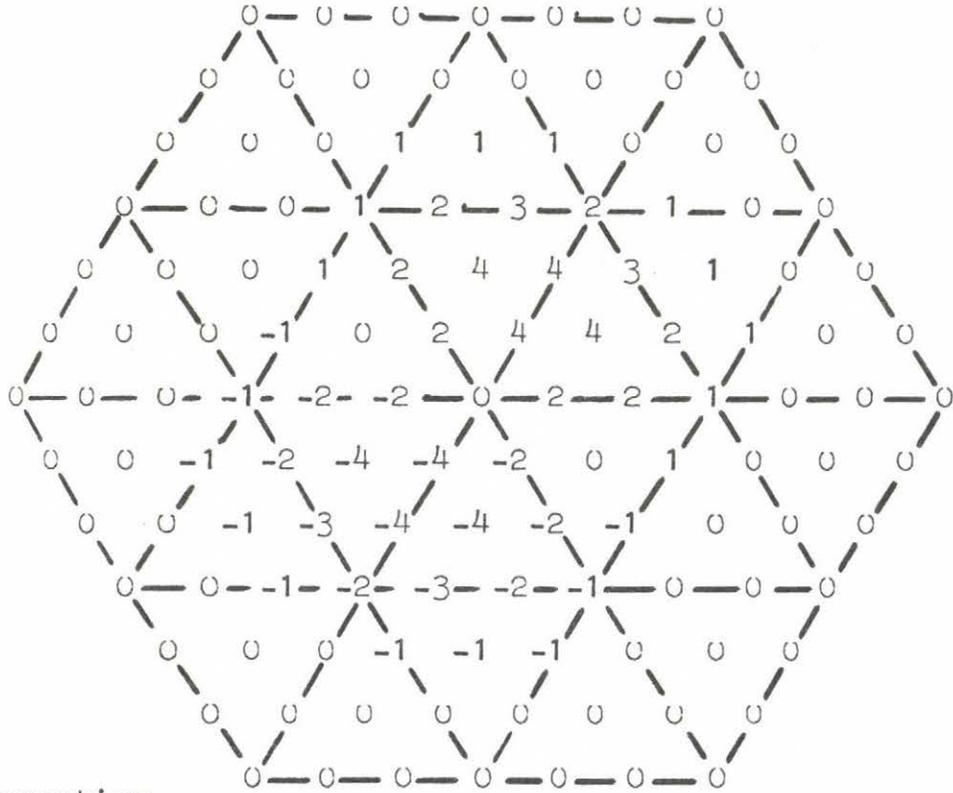
the shift and subtract towards the lower right gives



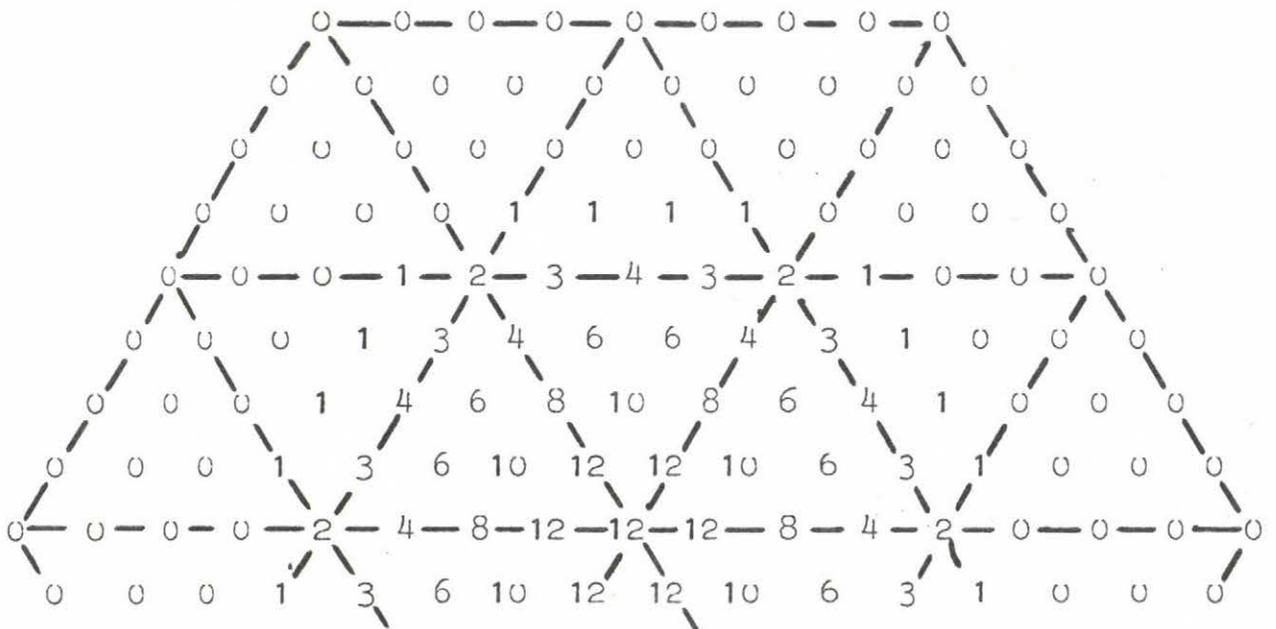
and the integration in the same direction



This has continuity of slope over all edges, but has continuity of curvature over those edges across which two integrations have taken place. Applying the shift and subtract in the third direction gives

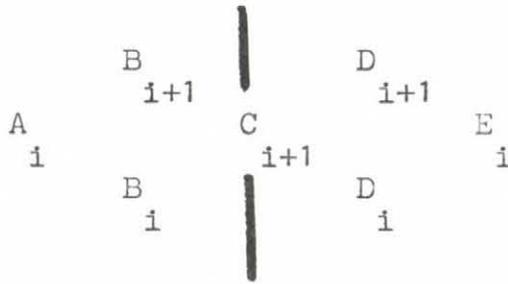


and integration



etc. the lower half being a reflection of the upper half. This shape is a quartic with continuity of slope and curvature across all edges, since each edge has been crossed by two out of the three integrations.

For the regular lattice the condition for continuity of second derivative is

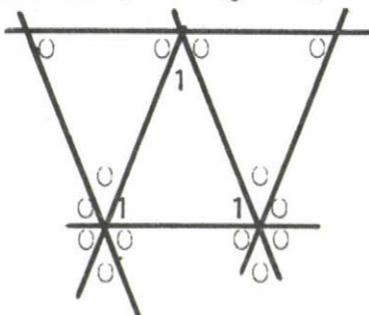
$$E_i - A_i = D_i - B_i + D_{i+1} - B_{i+1}$$


Odd order basis functions.

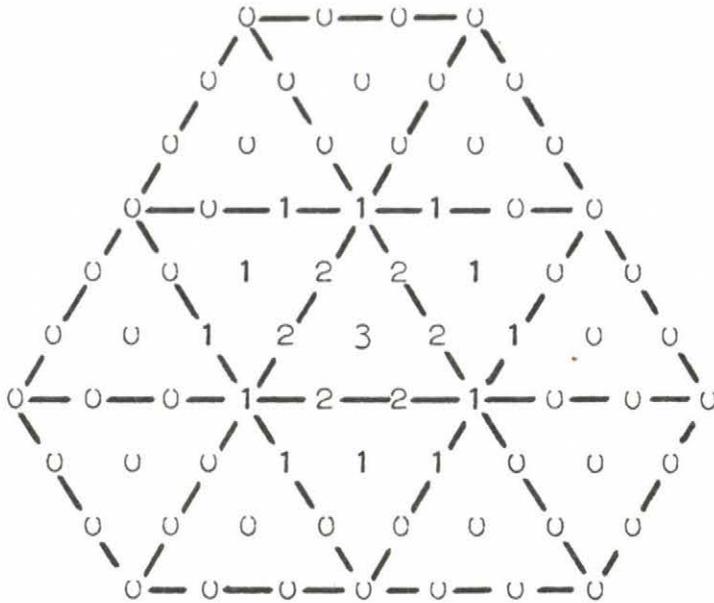
This process of applying three convolution operators in the three directions of symmetry of the lattice has thus increased the order of continuity by two, while increasing the order of the individual pieces by three and the side of the planform by one. A complete family of sets of basis functions with mode centres at the lattice points, and with continuity of successive even derivatives, may be generated by repeating the whole process. The unit impulse at each lattice point may be regarded as the simplest member of this family; the next two are those we have examined already, and further members have exactly the properties expected of them by considering the family as a whole.

Even order basis functions.

There is also another family, analogous to the even order univariate B-splines, which may be generated by applying the same process to different starting points. In particular, we may start with the triangular plateau, which does not even have position continuity with its neighbours.



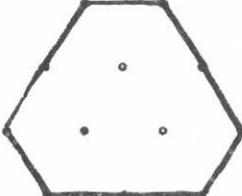
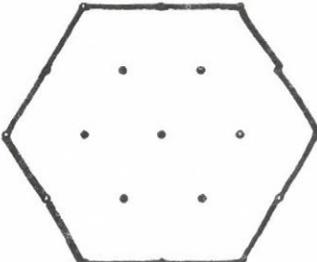
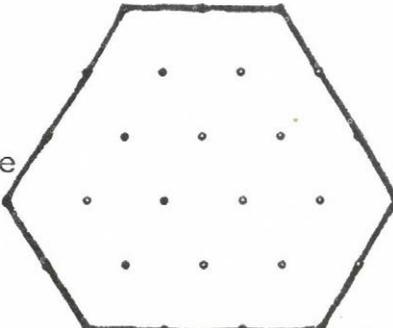
and the third



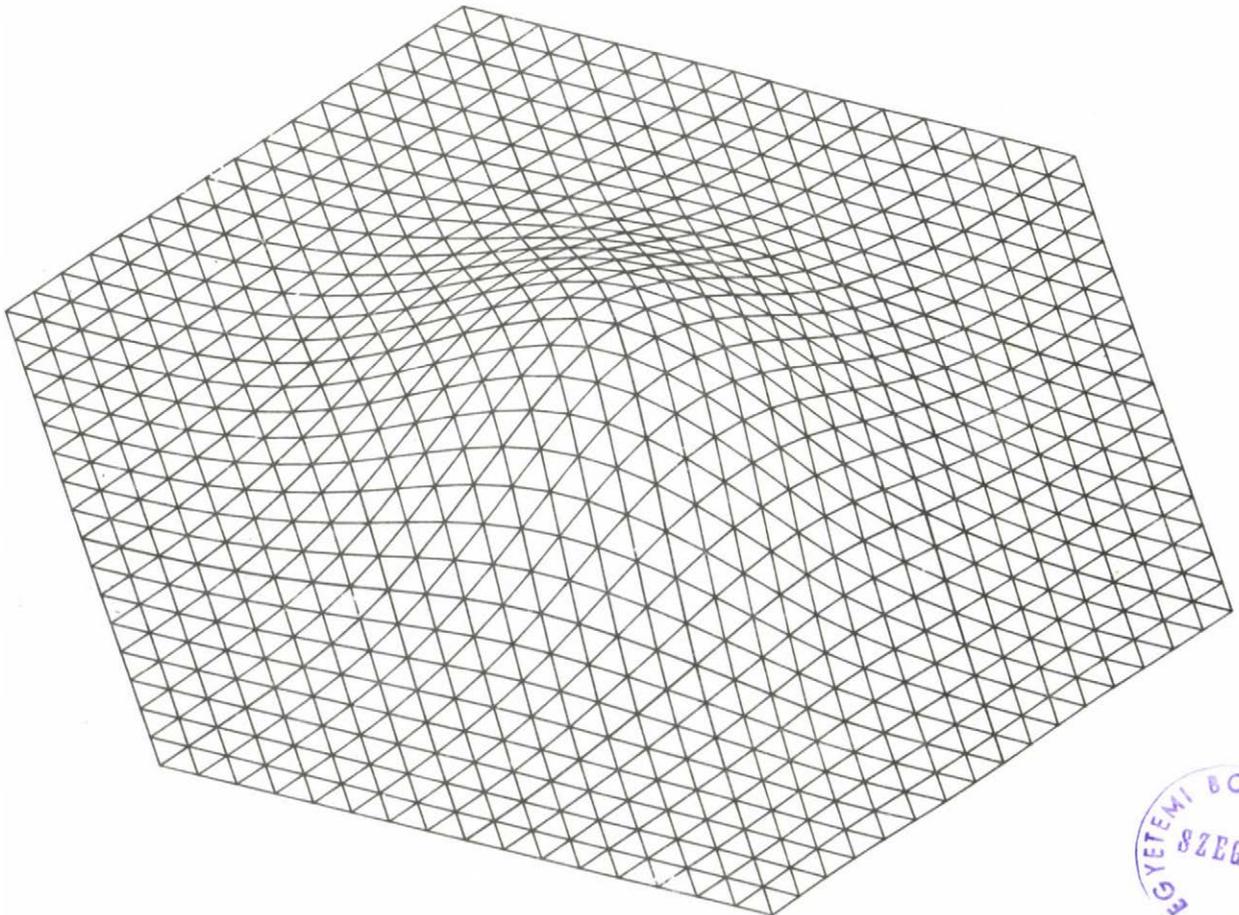
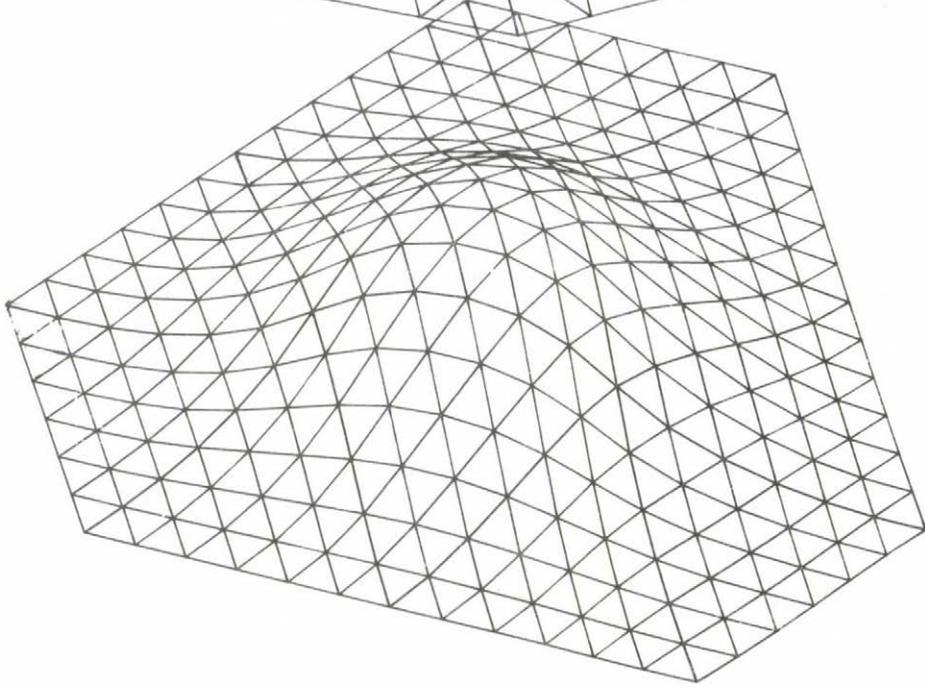
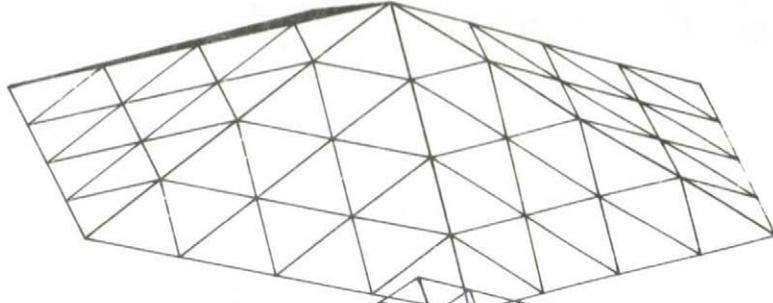
which has slope continuity across all boundaries. The pieces are all cubics.

Again, complete cycles of three convolutions may be applied repeatedly, generating another complete family of sets of basis functions. This family has the mode centres at the centres of the triangles. For design using the mode amplitudes as design variables it is possible to use modes centred on all the triangles. If the mode amplitudes are being computed to interpolate data at the mode centres, however, it is only possible to use the similarly oriented triangles, because either set of modes forms a complete set, and so both together are not linearly independent.

Summary of regular basis functions.

| order of pieces | order of continuity | shape of planform | |
|-----------------|---------------------|--|-----------|
| - | - | . | (impulse) |
| 0 | - |  | (plateau) |
| 1 | position |  | (pyramid) |
| 3 | slope |  | |
| 4 | curvature |  | |
| 6 | third derivative |  | |

etc.



Irregular basis functions.

To these two complete families we may add the analogue of the rectangular modes of mixed order, in which the orders in the two directions are not the same.

Let the number of convolutions in each direction, starting with the hexagonal pyramid, be a , b and c , respectively. Then the orders of continuity achieved over the corresponding directions will be $b+c$, $c+a$ and $a+b$, counting position continuity as order zero.

If the desired orders of continuity are A , B and C then the equations

$$A = b + c$$

$$B = c + a$$

and $C = a + b$

may be solved to give

$$a = (B + C - A) / 2$$

$$b = (C + A - B) / 2$$

$$c = (A + B - C) / 2$$

Clearly, a , b and c are all integral or nonintegral together. If all are integral and non-negative we have a solution. If all are non-integral we must start instead from the triangular plateau, for which the corresponding equations are

$$a = (B + C - A + 1) / 2$$

$$b = (C + A - B + 1) / 2$$

$$c = (A + B - C + 1) / 2$$

If the integral solution has negative values for any of a , b or c then it is not possible to synthesize basis functions with the specified properties, and higher than specified continuity will have to be accepted for feasibility.

Edge conditions.

These are the analogue for bivariate basis functions of the end conditions of univariate functions. There are two cases, in fact, the edge conditions and the corner conditions, but the same approaches apply to both.

Boundaries of any kind are an irregularity of the infinite lattice considered so far in this section, and it is a weakness of the convolution approach to generation of basis functions that it will tolerate no irregularity whatever. We therefore revert to rather more ad-hoc methods for dealing with this very practical necessity.

There are two methods for modifying the mode shapes near the edge of a region of interest to give more desirable properties. One is reflection, the other subtraction.

Reflection uses the concept of a wave reaching the fixed end of its medium. A reflected wave is generated by the end of just the right magnitude and phase to allow the end point itself to remain stationary under the combined effects of the original and generated waves. If we consider the set of basis functions of a given order as being a time sequence approaching the end, then we can combine each function with the appropriate reflection to give the properties we want. Now the reflection which gives zero resultant at the end is in fact the reflection in both ordinate and abscissa of the original wave. If the wave is $y = f(x)$, approaching the end $x = 0$. then the reflected wave is $y = -f(-x)$. The modified wave is just $y = f(x) - f(-x)$

It is valuable for all basis functions except that centred on the edge to be modified in this way, and for the remaining one to be modified so that $\sum f = 1$ remains valid.

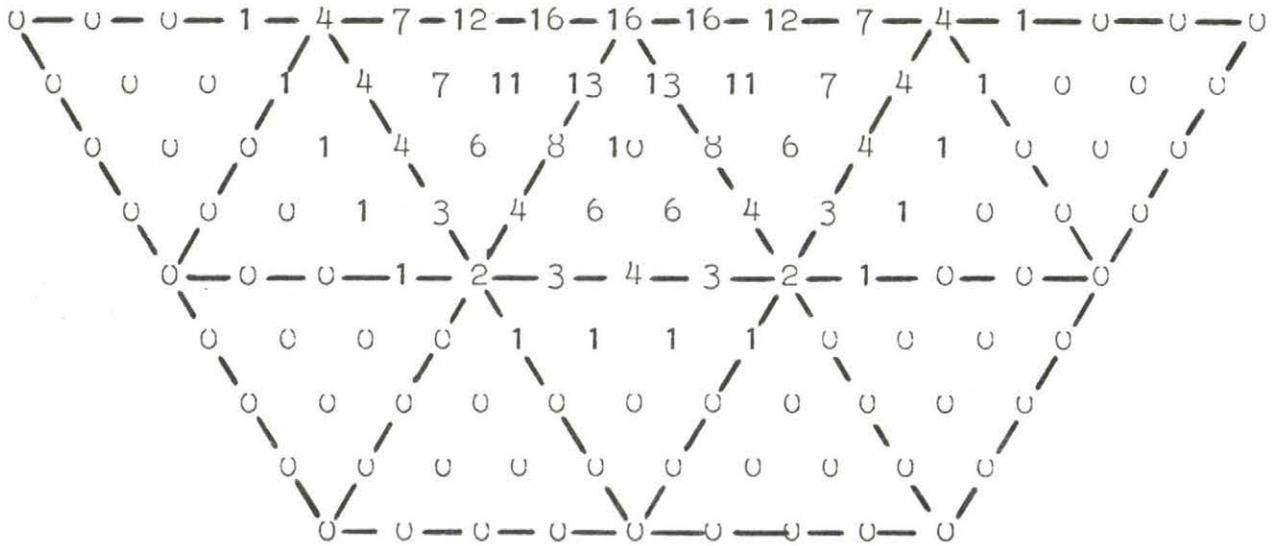
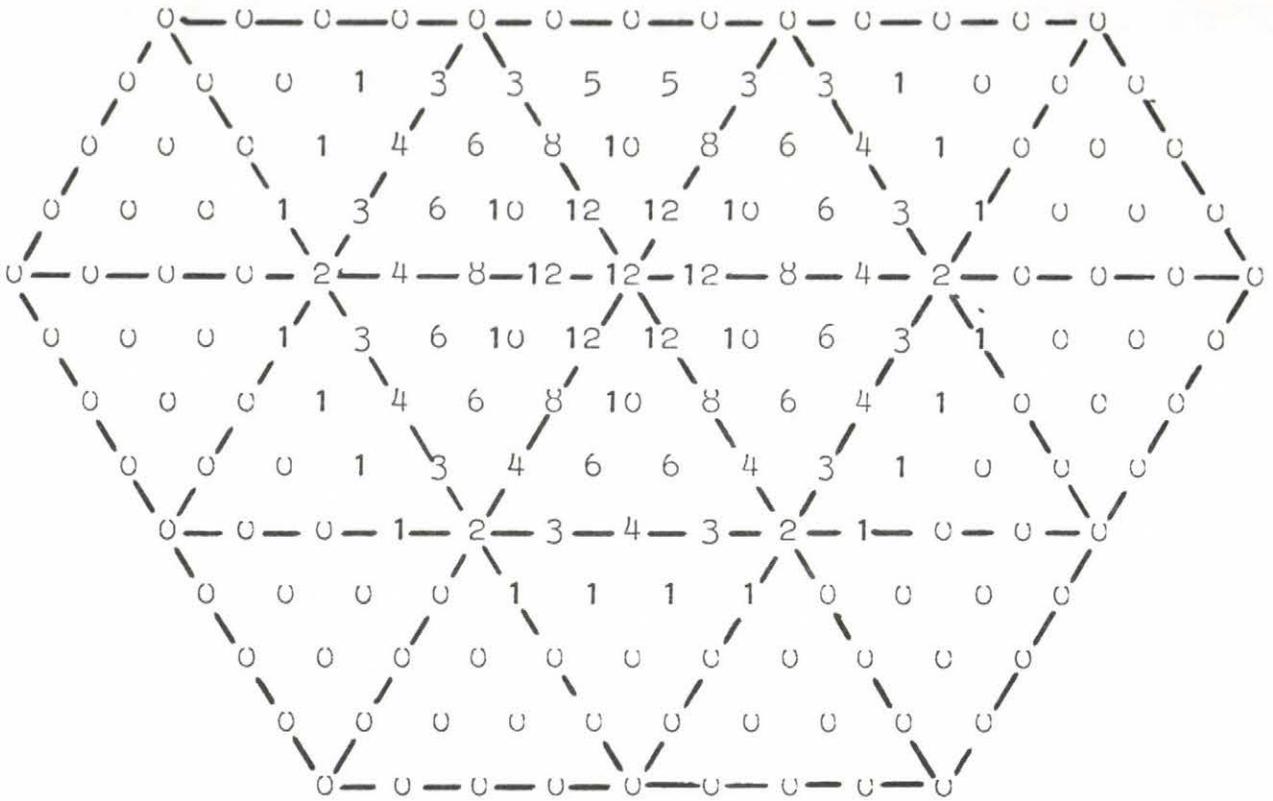
This is because that basis function will then have the edge value as its coefficient, a point of considerable importance in practice. The boundaries of a region to be represented are usually known, and being able to use them directly in the shape description is essential for convenience.

Equally, it is useful for all functions except the last two to have zero derivative at the end, so that the edge derivatives may be related to at most two coefficients. This is achieved in the reflection method by using a reflected wave with two components, so that the modified waveform becomes

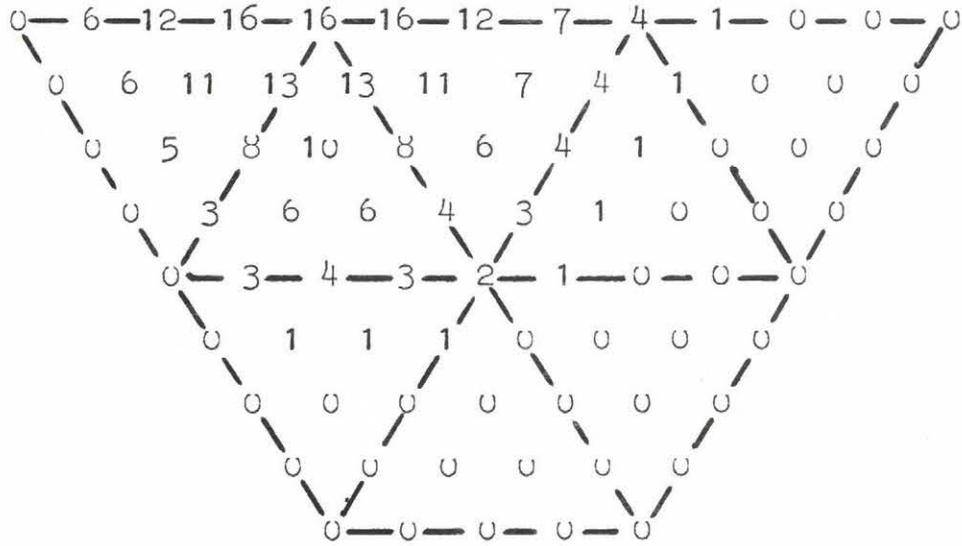
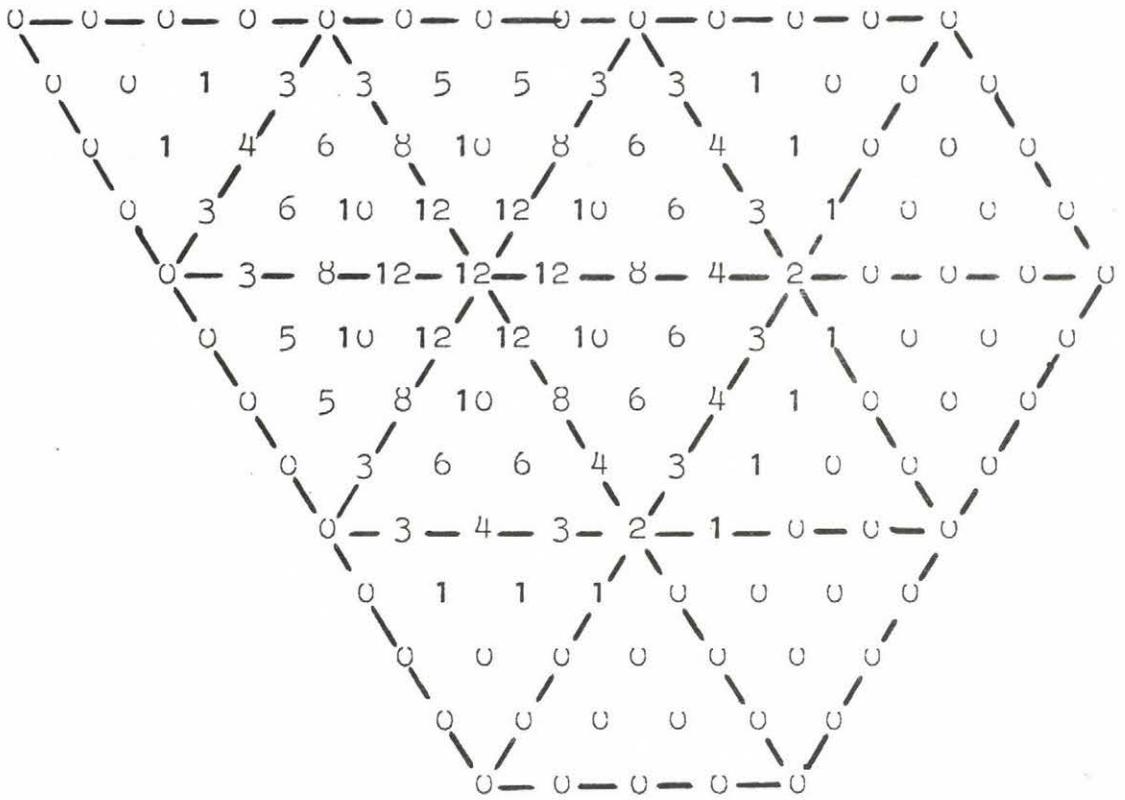
$$y = f(x) - a*f(-x) - b*f(1-x)$$

a and b being chosen so that both y and dy/dx are zero at x = 0. Obviously this depends on f being of high enough order.

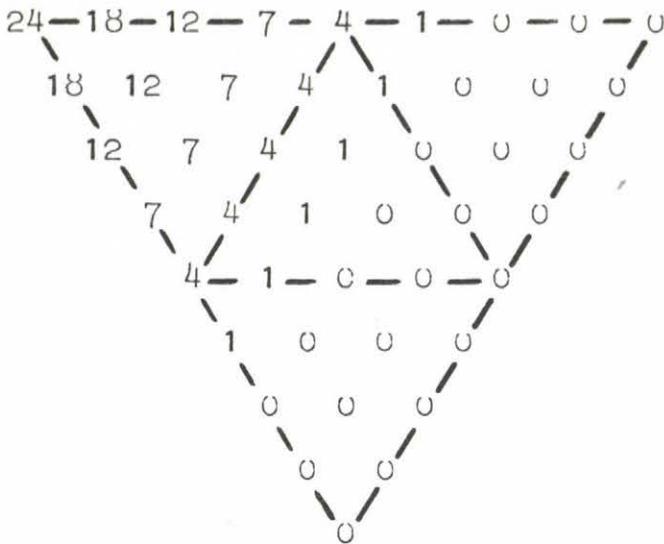
This method can be applied to the odd order basis functions on a triangular lattice, to give the appropriate edge conditions, and the corner conditions for acute corners, but only as far as edge position control. The result of this is shewn for the curvature-continuous basis set on the next few pages. The basic mode is overlapped over the edge by varying amounts, and the appropriate correction added to each. The modified shapes are then overlapped over a second edge in an exactly similar way to give the corner conditions. It is interesting to notice that, although the surface equations are quartic, the boundary curves are all cubic splines, and are, in fact, the cubic B-splines defined by the coefficients of the modes centred on the boundary. Where they meet the corner they take the form of the natural spline, whose second derivative is zero at the end. No explanation is offered for these facts, although analogous behaviour is shewn by bases of other orders.



Edge-modified basis functions.



Corner-modified basis functions



Corner-modified basis function cntd.

The subtraction method is a discipline for generating basis functions with special properties. For example, continuity may need to be sacrificed to some extent to handle a region with blunt corners.

The principle of the method is that $\sum f$ is a constant. As each basis function is designed it is subtracted from an array of values initially all equal to that constant, leaving the residue still to be divided into the functions yet to be designed. The form of the residue can then guide the design of the next function. When no further splitting is possible, the residue is the final function.

During this process it is helpful to realise that a basis function of any specific order of continuity is bound to be a linear combination of the functions generated by convolution as above. Thus both $\sum f = 1$ and the desired continuity can be maintained almost automatically during the design process.

3.4 Regular triangular partitionings of the sphere.

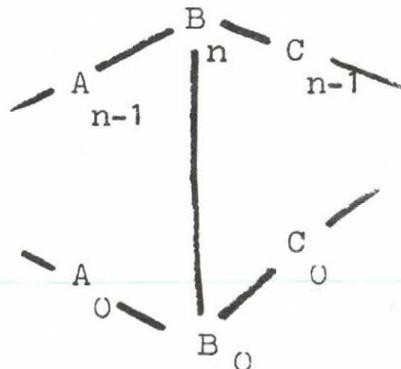
While providing a marginally smoother surface than the rectangular bases, the regular triangular lattice offers considerably less flexibility, because there is no analogue of the unequal interval univariate bases, or of the tartan rectangular bivariate basis. Either the lattice is regular or it is irregular.

B-Spline bases over an irregular plane triangulation would be a very powerful and useful tool, but would still not quite be sufficient for industrial 3D shape design. The ideal would be bases defined over any partitioning into triangles and rectangles of any parameter manifold.

This section extends the basis functions over a regular plane lattice to some regular non-planar lattices, the triangular partitionings of the sphere into the tetrahedron, the octahedron and the icosahedron. These results are not expected to be directly useful in their own right, but provide some results on which more general methods can be built by further research.

The position continuous bases are trivial, being simply pyramids with vertices at each of the polyhedron vertices. We examine here the shapes of slope-continuous bases.

As was discussed in section 3.2 above, the condition for slope continuity across the boundary between two triangular patches is



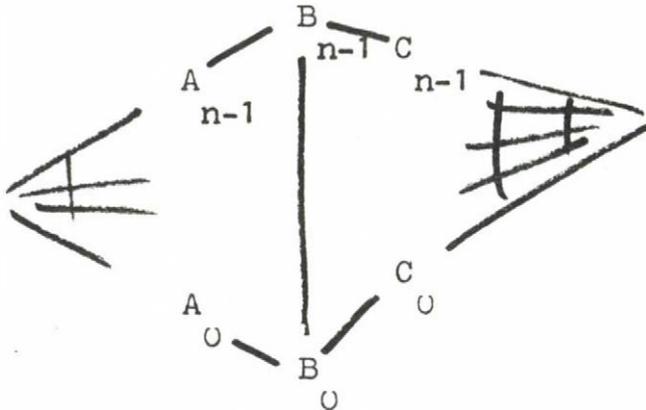
$$C_i = A_i + a(B_i - A_i) + b(B_{i+1} - A_i)$$

where $a+b = 2$ gives symmetry between the two patches and
 $a = b$ gives symmetry along the boundary.

For a regular partitioning with both symmetries, therefore, we get $a = b = 1$, which generates the regular triangular lattice.

Whatever the values of a and b it is not possible in general to have both $A_i B_i C_i$ collinear and also $A_{n-1} B_n C_{n-1}$ collinear.

If, however, we have the special case where the boundary controlled by the B_i is only of order $n-1$, then the two triangles can be transformed by singularities at the far corners into rectangles of order $n-1$ by n .

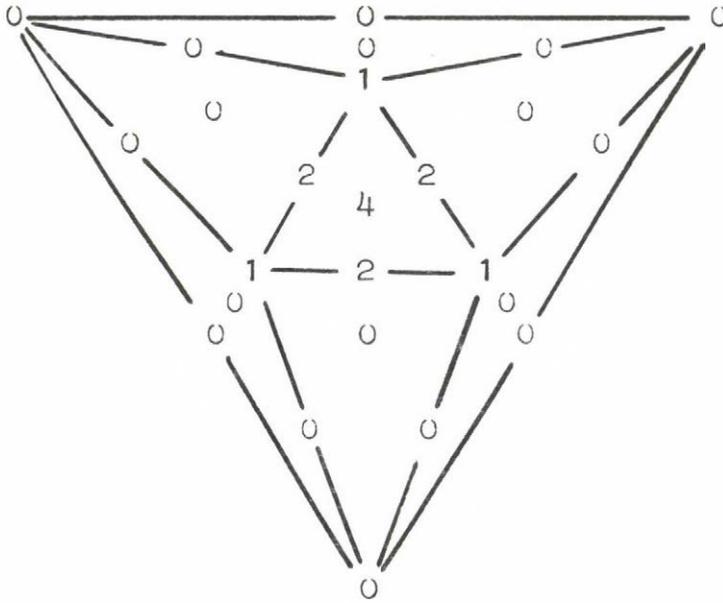


The looser relationship

$$C_i = B_i + a(B_i - A_i) + \frac{1}{n-1} b(B_i - B_{i-1}) + \frac{(n-1-i)c}{n-1} (B_{i+1} - B_i)$$

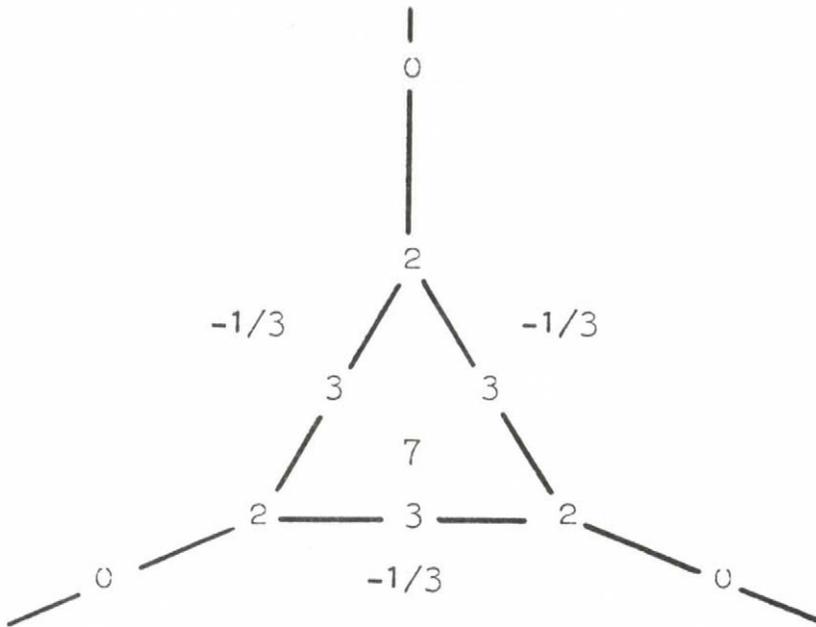
then corresponds to the condition of slope continuity.

The symmetry conditions then require that $a = 1$ and $b + c = 0$, and so we have a degree of freedom, which is quite sufficient. For the value $b = 2/3$ the triangular plane lattice is obtained; for the value $b = 0$ a partitioning of the sphere into an octahedral pattern as obtained. The simplest slope continuous basis functions over this partitioning are a set of cubic triangular patches with quadratic boundaries which can be represented graphically as



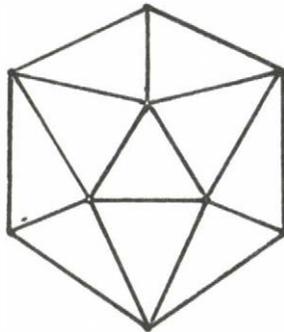
Because the mode centres are in the centres of the faces the control points associated with these basis functions are connected by the edges of a cube, the dual of an octahedron.

The tetrahedral partitioning is given by $b = -1$ and the simplest slope-continuous basis is

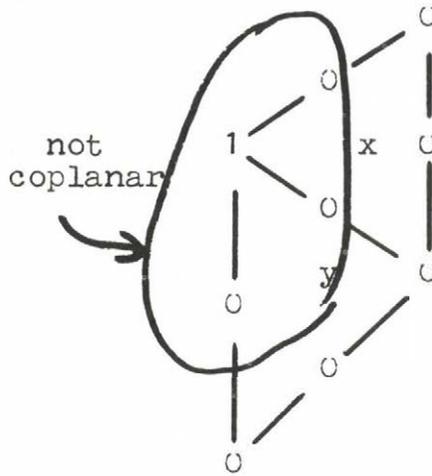


The associated control points are at the vertices of the dual tetrahedron.

When we try to find the basis for the icosahedron, however, the pattern has to be changed slightly. If we had cubic triangles and the obvious planform

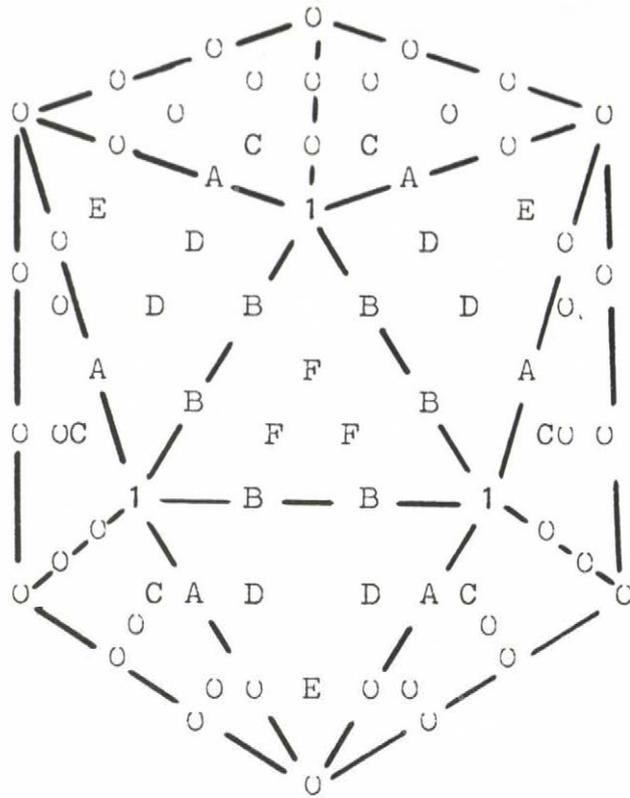


then slope continuity could not be maintained across all boundaries, because where the planform outline has a convexity the quadratic edges would give three zeros adjacent to a lattice point. This would force the value and first derivatives at that point to be zero, and thence the entire function.



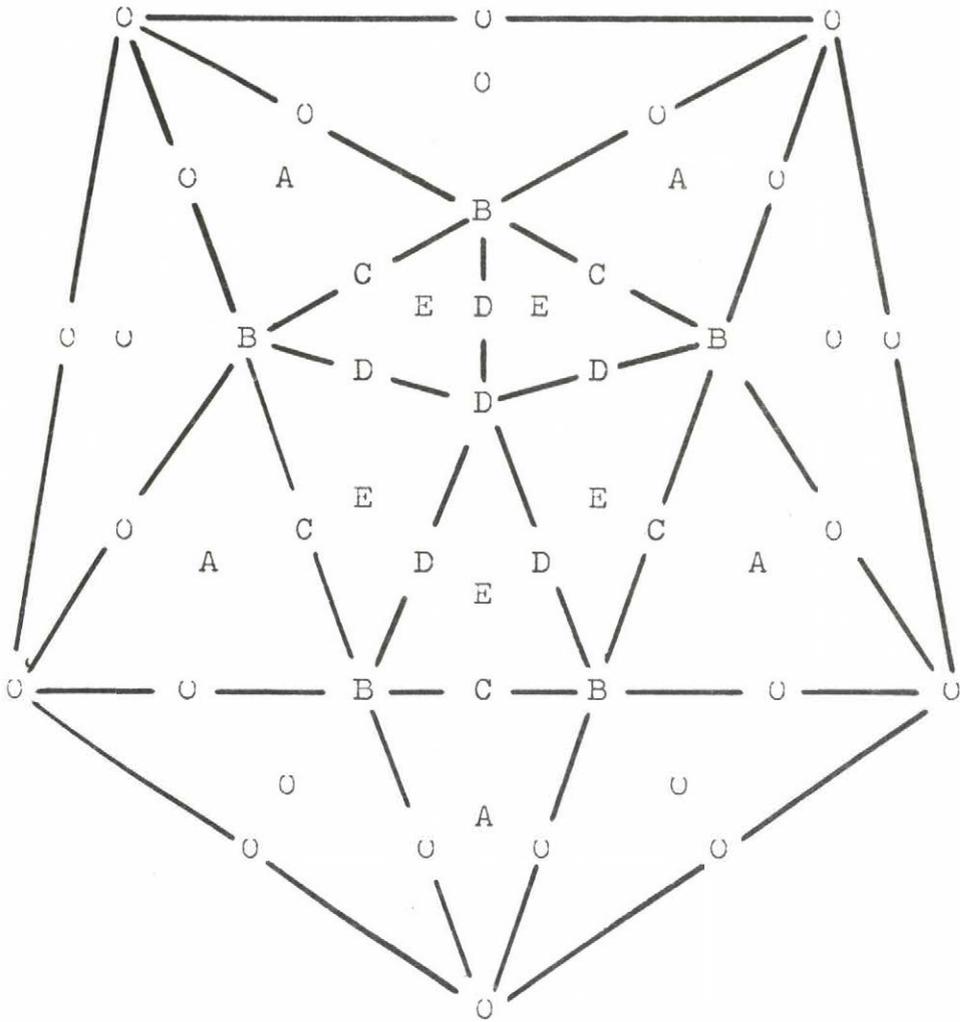
It is therefore necessary either to increase the planform so that the outline always contains at least three triangles at each vertex, or for the surface to have cubic patch boundaries which in turn means that the patches themselves must be quartic. If we take the latter course the basis functions are described by numbers related to the value of b which is equal to $(\sqrt{5} - 1)/2 = 2 \sin 18 = 0.61804$

| | | | | | | |
|---|---|------|---|-----|---|---------|
| A | = | 1 | - | b/2 | = | .69098 |
| B | = | 3/2 | + | b/2 | = | 1.80902 |
| C | = | b/8 | | | = | .07726 |
| D | = | 19/8 | - | 2b | = | 1.13892 |
| E | = | 3b/4 | - | 1/4 | = | .21353 |
| F | = | 1/2 | + | 3b | = | 2.35412 |

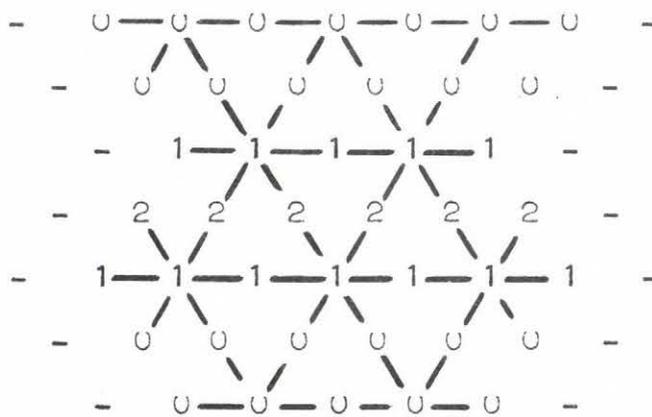


The plane lattice may be considered in this light as needing cubic boundaries for the same reason as the icosahedron, but its patches degenerate into cubics because the lattice is plane.

The alternative, of increasing the size of the planform until there are no convex corners, appears to give a structure as illustrated on the next page, but this does not in fact work because there are too many continuity equations for the degrees of freedom available. Such an alternative should in any case be regarded as a special case, because it has a parallel in the plane lattice, the ridge function. We may regard the tetrahedral and octahedral bases described above as being of anomalously low order, because they share with this extended icosahedral function the property of reaching more than halfway round the sphere. On these grounds we expect that any slope-continuous basis over an irregular triangulation will have pieces of order at least four.



Extended icosahedral basis.



Ridge function.

Conclusions

Bezier-like polynomials can be described for triangular regions. These have the advantage over the familiar rectangular forms of having the same order curve for all straight lines in the parameter plane, thus avoiding the reduction in wavelength for diagonal features which can occur with rectangular forms. The conditions for continuity of position and first and second derivatives are all usably transparent.

B-spline like functions can be described for regular triangular partitionings of the plane and of the sphere, the key to the latter being the use of pieces whose equation is one order higher than that of the boundary curves of the pieces. Such triangular lattices are marginally less usable than the current state of the art for rectangular lattices, because no equivalent of the tartan form (unequal intervals) as yet exists.

Recommendations for further research

The work of 3.2 above is reasonably complete. Further research in this direction would be to devise methods for pieces of surface with five or even n sides.

The methods of 3.3 are restricted to absolutely regular lattices. There are other approaches to generating the univariate B-spline basis functions, and analogues of these (in particular of the Cox algorithm) for triangular partitionings could be sought, using the results obtained here for comparison and evaluation. The unequal interval versions of these other methods may then give techniques for defining functions over irregular triangulations.

Further work in the topic of 3.4 could well be to work out higher order functions, and by comparison of these with the first and second order functions, to develop systematic generation techniques. Convolution methods probably exist, with the shift replaced by rotation about one of the axes of symmetry of the polyhedron, and the integration operator modified appropriately.

REFERENCES AND BIBLIOGRAPHY

- Adams, J.A. Cubic spline fitting with controlled
end conditions.
Computer Aided Design vol 6 no 1 pp2-9 1974
- Adams, J.A. Intrinsic method for curve definition.
Computer Aided Design
vol 7 no 4 pp 243-249 Oct 1975
- Ahlberg, J.H., Nilson, E.N. and Walsh, J.L. The theory of splines
and their applications.
Academic Press. 1967
- Alexander, J.Y. Private communications 1966
- Altham, D.W., Beck, D.A. and Wall, J.F. Review of research on
multivariable least squares spline fitting.
report Ae 306 British Aircraft Corporation
Preston Division 1970
- Anthony, J. BIARC curves in NELAPT.
NEL Memo M1/274, National Engineering Laboratory
East Kilbride, Glasgow. 1973
- Barnhill, R.E. and Riesenfeld, R.F. Computer Aided Geometric Design.
Academic Press. 1974
- Barnhill, R.E. Smooth Interpolation over triangles.
pp 45-70 in Barnhill and Riesenfeld above. 1974
- Bengtsson, B.E. and Nordbeck, S. Construction of isarithms
and isarithmic maps by computers.
BIT vol 4 pp 87-105 1964
- Bezier, P. Example of an existing system in the motor industry.
pp207-218 in Hawthorne and Edwards below. 1971

- Bezier, P. Numerical Control - Mathematics and Applications.
John Wiley 1972 (translated by Forrest)
- Birkhoff, G. and Garabedian, H.L. Smooth surface interpolation.
J. Maths and Physics vol 39 1960
- Bolton, K.M. Biarc curves.
Computer Aided Design vol 7 no 2 pp89-92 April 1975
- Braid, I.C. Designing with volumes.
Computer aided design group, University of Cambridge
Computer Laboratory, Cambridge, England. 1973
- Braid, I.C. Six systems for shape design and representation
- a review.
Computer aided design group document 87
Computer aided design group, Cambridge University
Computer Laboratory, Cambridge, England.
- Butterfield, K.R. Orientation properties of splines
related to cubics.
Dept. of Mech. Engineering,
Southampton University 1969
- Clenshaw and Hayes, J.G. Curve and surface fitting.
J. Inst. Maths and Applics. vol 1 pp164-183 1965
- Cline, A.K. Scalar- and Planar-valued curve fitting
using splines under tension.
CACM vol 17 no 4 pp218-220 1974
- Comba, P.G. A language for three-dimensional geometric
processing - written form.
IBM Corp, New York. 1965
- Comba, P.G. A procedure for detecting intersections of
three dimensional objects.
IBM Corp, New York. 1967

- Coolidge, J.L. A history of geometrical methods.
Oxford University Press 1940
- Coolidge, J.L. A history of the conic sections and quadric surfaces.
Oxford University Press. 1945
- Coons, S.A. Surfaces for computer aided design of space forms.
Project MAC report MAC-TR-41
Massachusetts Institute of Technology 1967
- Davies, K.J. GNC - a graphical NC processor.
Proc. Prolamat 1973
North Holland. 1973
- Davies, P.J. The surface fitting techniques used in the
APTLFT-FMILL surface milling program.
Tech memo Aero 1409. Royal Aircraft Establishment
Farnborough, U.K. 1972
- Davies, R.S. Computer aid in wing design.
Proc. CAD74. IPC Science and Technology Press. 1974
- den Hartog, C. Computer Aided Shape Technique.
paper without date or affiliation.
Probably written late 1960s or early 1970s
- Duncan, J.P. Numerical control machining of surfaces with
tabular specifications.
pp 337-338 in Proc. 3rd. Can. Cong. App. Mech.
Calgary 1971
- Duncan, J.P. and Mair, S.G. The method of highest point in die
design and machining.
Proc. 11th Intl. Conf. of NCSociety, Toronto, 1974
- Earnshaw, J.L. and Yuille, I.M. A method of fitting equations
for curves and surfaces to sets of points
defining them approximately.
Computer Aided Design Winter 1971 pp 19-22

- Engeli, M. EUKLID - Eine Einfuhrung.
Fides Rechenzentrum, Zurich. 1974
- Ferguson, J. Multivariable curve interpolation.
JACM vol 11 no 2 pp221-228 April 1964
- Fletcher, R., Grant, J. A. and Hebden, M. D. The calculation of
linear best Lp approximations.
Computer Journal vol 14 no 3 pp 276-279 1971
- Forrest, A. R. Curves and Surfaces for Computer Aided Design.
Computer Aided Design Group, University of Cambridge
Computer Laboratory, Cambridge, England 1968
- Forrest, A. R. Interactive interpolation and approximation by
Bezier polynomials.
Computer Journal vol 15 no 1 pp 71-79 1972
- Forrest, A. R. A new curve for computer aided design.
C.A.D. Group Document 66
Computer Laboratory, Cambridge University 1972
- Forrest, A. R. Notes of Chaikins algorithm.
CGM-74-1. University of East Anglia
School of Computing Studies. 1974
- Fririon, J. Etude de l influence du deplacement d un sommet
d un polygone caracteristique.
Courbes Unisurf report 0800 JF/MG Oct 1971
- Fririon, J. Etude de l influence du deplacement d un sommet
d un reseau caracteristique.
Courbes Unisurf report 0800 JF/MG Oct 1971
- Gordon, W. J. Blending function methods of bivariate and
multivariate interpolation and approximation.
report GMR-834-B
General Motors Research Laboratory. 1970

- Gordon,W.J. and Riesenfeld,R.F. Bernstein-Bezier methods
for the computer aided design of free-form
curves and surfaces.
JACM vol 21 no 2 pp 293-310 April 1974
- Gordon,W.J. and Riesenfeld,R.F. B-Spline curves and surfaces.
pp 95-126 in Barnhill and Riesenfeld above. 1974
- Greville,T.N.E. Theory and Applications of Spline Functions
Academic Press. 1969
- Haverlik,I. and Krcho,J. Automatizacia tvorby vrstevnicovych map
hl adiska primarnych a sekundarnych izociarovych
poli.
Geodeticky a Kartograficky obzor
vol 19/61 no 6 pp 151-158 1973
- Hawthorne,W. and Edwards,G.R. A discussion on computer aids
in mechanical engineering design and manufacture.
Proc. Roy. Soc. Lond. A 321 pp 143-248 1971
- Hayes,J.G. New shapes from bicubic splines.
Proc. CAD74. IPC Science and Technology Press 1974
- Heap,B.R. Algorithms for the production of contour maps
over an irregular triangular mesh.
NPL report DNAC 10. National Physical Laboratory
Teddington, Middx. England.
- Helmy,B. Some problems in numerical analysis
(spline functions).
M.Sc. Thesis. Inst. of Statistical Studies and
Research, Cairo University. 1973
- Hosaka,M.,Kimura,F. and Kakishita,N. A unified method
for processing polyhedra.
pp768-772 in Proc. IFIP Congress 1974

- Jeffreys,H. Cartesian Tensors.
Cambridge University Press 1969
- Kansy, von K. Erfassung der Geometrie von Strassen.
Nachrichten aus dem Karten- und Vermessungswesen
reihe 1 heft 65 pp115-122 1974
- Kansy, von K. Approximation of smooth curves by spline curves.
internal note, Gesellschaft fur Mathematik
und Datenverarbeitung Bonn. 1975
- Klein,J.L. A rational approach to propellor geometry.
paper 11 in Propellors 75 conference.
Soc.Naval Arch. and Marine Engineers 1975
- LaFata,P. and Rosen,J.B. An interactive display for
approximation by linear programming.
CACM vol 13 no 11 pp 651-659 Nov 1970
- Lidbro,N. Modern Aircraft Geometry.
Aircraft Engineering pp 388-394 Nov 1956
- Lidbro,N. Analytische Formbestimmung van Schiffen.
Schiffstechnik heft 42 band 8 pp91-96 1961
- Luh,J.Y.S. and Krolak,R.J. A mathematical model for
mechanical part description.
CACM vol 8 no 2 pp125-129 1965
- MacCallum,K.J. Surfaces for interactive graphical design.
Computer Journal vol 13 no 4 pp352-358 Nov 1970
- MacCallum,K.J. Mathematical design of hull surfaces.
J. of Roy. Inst. of Naval Architects.
vol 114 no 3 pp 359-373 1972

- Macurek, I. and Vencovsky, J. Programming the AGIECUT NC wire spark erosion machine using the CKDAPT system. Intl. Conference on Computer aided Manufacture and Numerical Control. University of Strathclyde 1974.
- Manning, J.R. Continuity conditions for spline curves. Computer Journal vol 17 no 2 pp181-186 1974
- Manning, J.R. Feature lines on curved surfaces. information publication IP 146 Shoe and Allied Trades Research Association. Kettering, Northants, 1974
- Maxwell, E.A. General Homogeneous Coordinates in space of three dimensions. Cambridge University Press. 1961
- Maxwell, E.A. The methods of Plane Projective Geometry based on the use of general homogeneous coordinates. Cambridge University Press. 1963
- McConalogue, D.J. A quasi-intrinsic scheme for passing a smooth curve through a discrete set of points. Computer Journal vol 13 no 4 pp392-396 Nov 1970
- McLain, D.H. Artificial intelligence techniques in a practical graphics problem. Proc. IFIP Congress 1974 pp 501-506 1974
- McWaters, J. Description of 2D bounded geometry in APT. CAM Congress. Hamilton, Ontario, 1974
- Mehlum, E. Curve and surface fitting based on variational criteria for smoothness. Central Institute for Industrial Research Oslo, Norway. 1969

- Mehlum, E. and Sorensen, P.F. Example of an existing system in the shipbuilding industry; the Autokon system. in Hawthorne and Edwards above. 1971
- Mehlum, E. Nonlinear splines. in Barnhill and Riesenfeld above. 1974
- Nielson, G.M. Some piecewise polynomial alternatives to splines under tension. pp 209-236 in Barnhill and Riesenfeld above 1974
- Nutbourne, A.W., McLellan, P.M. and Kensit, R.M.L. Curvature profiles for plane curves. Computer Aided Design vol 4 no 4 pp176-184 July 1972
- Okino, N. et al. TIPS-1; technical information processing systems for computer aided design, drawing and manufacture. Proc. Prolamat 1973 North Holland 1973
- Overas, A. private communication 1976
- Pochop, V. private communications 1974
- Powell, M.J.D. Piecewise quadratic surface fitting for contour plotting. pp 253-272 in Software for Numerical Mathematics ed Evans, D.J. Academic Press 1974.
- Rhynsburger, D. Analytic delineation of Thiessen polygons. Geographical Analysis vol 5 no 2 pp133-144 1973
- Riesenfeld, R.F. Applications of B-spline approximation to geometric problems of computer aided design. Ph.D. thesis, Syracuse University. 1973
- Ris, G. Raccordement a l'ordre n entre carreaux de surface definis par des polynomes biparametrique a coefficients vectoriels. Doctoral thesis. University of Nancy. 1975

- Robertson, R.G. Descriptive Geometry.
Pitman, 1966
- Sabin, M.A. Interrogations of Parametric Surfaces.
Proc. CG70 Intl. Conf. Plenum Press 1970
- Sabin, M.A. An existing system in the aircraft industry; the
British Aircraft Corporation Numerical Master
Geometry system.
pp 197-206 in Hawthorne and Edwards above. 1971
- Sabin, M.A. Cursive script output.
to appear in Software, Practise and Experience.
- Sandel, G. Geometry of compound curve.
Z. Angew. Math. Mech., vol 17 no 5 pp301-302 1937
- Schoenberg, I.J. Contribution to the problem of approximation
of equidistant data by analytic functions.
Quart. Appl. Math. vol 4 pp45-49 1946
- Schweikert, D.G. An interpolation curve using a spline in tension.
J. Math. and Physics vol 45 pp312-317 1966
- Shelley, J.H. The development of curved surfaces for aero-design.
Gloster Aircraft Company Ltd. 1946
- Shippey, G.A. Piecewise approximation using circular arc segments.
internal note Tech./IEG/67/3, Ferranti Ltd,
Ferry Road, Edinburgh. 1967
- Shippey, G.A. Interpolation through a set of data points using
circular arc segments.
internal note Tech/IEG/67/15, Ferranti Ltd,
Ferry Road, Edinburgh. 1967

- Smith,D.J.L. and Merryweather,H. The use of analytic surfaces
for the design of centrifugal impellers by
computer graphics.
Intl. Jour. for Num. Meth. in Engineering
vol 7 pp 137-154 1973
- Spath,H. Exponential spline computing.
Computing vol 4 no 3 pp225-233 1969
- Thielheim,F. and Starkweather,W. The fairing of ship lines on
a high speed computer.
David Taylor Model Basin report 1474 Jan 1961
- Thorne,J. private communications 1973
Stone Manganese Marine Ltd.
- Veron,M. Contribution a l etude des surfaces numeriques
unsurf - conditions de raccordement.
Doctoral thesis. University of Nancy. 1973
- Voelcker,H.B. et al. An introduction to PADL,
publication TM-22 Production Automation Project,
University of Rochester.
- Walter,H. Computer aided design in aircraft industry.
pp 355-378 in Computer aided design.
ed Vlietstra,J. and Wielinga,R.F.
North Holland. 1973
- Weiss,R.A. BE VISION. A package of IBM7090 fortran programs
to draw orthographic views of combinations of
planes and quadric surfaces.
JACM vol 13 no 2 pp194-204 April 1966

