Surfaces for interactive graphical design

K. J. MacCallum*

Centre for Computing and Automation, Imperial College, London SW7

Based on the work of Coons, several attempts have been made to develop programs for designing surfaces using a computer driven interactive display. However, it is considered that the problems of man-design interaction have not previously been solved successfully. In this paper techniques are developed which allow manipulation of a surface without reference to its mathematical definition and which allow modification to influence only local areas of design. Using an extension of the Coons definition a bi-quartic patch is chosen as more suitable than the bi-cubic for design.

(Received December 1969)

1. Introduction

The description of surfaces has always played an important part in most branches of engineering design. A graphical description is normally provided by a set of points on the surface or by a set of plane curves taken through the surface. However, with the advantages of speed, accuracy and flexibility that a computer, especially when linked to a graphical display, can offer to aid the design process, a need arises to provide mathematical descriptions of surfaces suitable for the design of engineering objects. Methods are also required to interact with the surfaces in a natural way.

The most satisfactory way of manipulating a mathematical description of a surface while maintaining a visual control over it is to use an interactive display terminal and some previous workers have used this approach (e.g. Armit, 1969; Flanagan and Hefner, 1967). Although Armit's Multipatch system is a powerful and very flexible tool it appears to assume that the designer has some knowledge of the effect of altering surface coefficients. In its present form it also assumes that the designer has some fore-knowledge of the object he is designing. On the other hand, the 'surface moulding' techniques developed by Flanagan and Hefner allow a fairly natural control over the surface without involving the designer in mathematics but unfortunately they deal with a class of surface suitable only for designing aircraft fuselages and other similar objects.

This paper describes a class of surfaces based on the work of Coons (1964 and 1967) and introduces some new methods to make surface design on the display possible. The views expressed in the paper may be biased by the application for which the techniques have been developed. In particular the investigation has assumed that the surface being designed needs continuity no greater than the first derivative. Nonetheless, the author has attempted to present the methods out of context of the application since he feels they have something to offer in other fields of surface design.

2. Coons surfaces

It was in 1964 that Coons first proposed a class of surfaces for computer-aided design of space figures. These surfaces are claimed to have advantages for this purpose because they are well suited to display work and can conveniently be 'matched' or made continuous with adjoining surfaces of any type. More particularly their advantages stem firstly from the fact that they are parametric and secondly from the fact that they are bounded.

The use of parameters allows the easy definition of infinite real slopes as the ratio of two finite and computable numbers. This is a particularly important point in engineering surfaces since areas of rapidly changing slope often cause trouble in a mathematical definition. A parametrically defined surface also allows the definition of the three coordinate directions independently; any surface modification can thus be made independent of the orientation of the axis.

A bounded surface definition relates very closely to the process of designing engineering surfaces in that the boundaries form an integral part of the object. For example, a blending surface joining two fillets of differing radii is almost completely constrained by its boundaries. Even the definition of a ship's hull relies to some extent on the shape of certain boundaries or design curves. However, the requirements of these surfaces do not fully define the influence the boundaries have inside the surface. In the Coons surfaces mathematical influence or *blending* functions are chosen and constrained such that continuity with other surfaces across the common boundary can be defined up to any degree.

Although in Coons's earlier paper the emphasis was on a particular member of the class in which the boundaries *and* the blending functions between the boundaries were parametric cubics, the class is much more general and boundaries can take any definition whatsoever. None the less, almost all implementations and usages of the surfaces have been of this type. The

* *The author is on the staff of the British Ship Research Association and is seconded to the Centre for Computing and Automation, Imperial College. The work described in the paper forms part of a thesis for the degree of Ph.D. presented in London University.*

choice of a cubic gives the lowest order polynomial which can describe a space curve and can contain a point of inflexion, and it would therefore appear to be eminently suitable for design where continuity up to the first degree is all that is required. Typical examples of attempts to use the Coons bicubic patch are in ship design (Hamilton and Weiss, 1965), car body design (Hogue, 1966) and aircraft design (Eshleman and Meriwether, 1967, and Sabin, 1968).

3. Extension of Coons surfaces

One of the advantages of restricting all the boundaries of the surfaces and their blending functions to be of the same form is that the surface equation can be put in the very convenient tensor form

$$P = uMBM^tv^t$$

where for the cubic $u = [\,u^3 \quad u^2 \quad u \quad 1\,]$

$$M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} (00) & (01) & (00)_v & (01)_v \\ (10) & (11) & (10)_v & (11)_v \\ (00)_u & (01)_u & (00)_{uv} & (01)_{uv} \\ (10)_u & (11)_u & (10)_{uv} & (11)_{uv} \end{bmatrix}$$

and the superscript t indicates the transpose function. Typically in this notation a term such as $(01)_u$ represents the derivative of the surface taken with respect to u at the parameter values $u = 0$; $v = 1$. Thus this equation represents a bi-cubic surface defined in terms of its four corner points (parameter range is 0 to 1), the slopes along the boundaries at these corner points and the cross derivative terms (often called twists) at these corners. Techniques are available for fast evaluation of this form of equation. The development of this tensor form from the basic equations depends only on the fact that the boundaries can be defined as a blending of the boundary conditions necessary for continuity. However, a slightly different surface form can be produced by initially assuming the tensor form of the surface, instead of the basic form.

Consider first a general curve equation, $f(u)$, which could serve as a boundary to a surface, defined by m different coefficients where the general form of a coefficient is

$$K_i = \left.\frac{df^{p_i}(u)}{du^{p_i}}\right|_{u=u_i}$$

for $i = 1, 2, \ldots, m$, the superscript p_i being the degree of differentiation applied to the function. Note that the u_i's need not all be different. The equation of the curve, $f(u)$, can then be written as a linear combination of these m coefficients multiplied by a corresponding blending function. A blending function must be a function of a set of m linearly independent functions of the parameter u. Hence the equation of the curve may be written as

$$f(u) = [H_1uH_2u \; - - - - \; H_mu] \begin{bmatrix} K_1 \\ K_2 \\ - - \\ K_m \end{bmatrix} = \bar{H}\bar{K}$$

The H functions are the functions of m linearly independent functions of the parameter u and are subject to the following constraints:

$$\left.\frac{d^{p_i}(H_ju)}{du^{p_i}}\right|_{u=u_i} = \begin{cases} 1 \text{ if } j = i \\ 0 \text{ otherwise} \end{cases}$$

for $i = 1, 2, \ldots, m$

The vector $\bar{H}$ can always be written in terms of its basic vector (i.e. the m linearly independent functions of (u)) and an $m \times m$ constant matrix M. Thus

$$f(u) = \bar{H}\bar{K} = \bar{Q}M\bar{K}$$

Extending the argument, we can write the equation of a surface as

$$f(uv) = \bar{H}_u\bar{K}\bar{H}_v{}^t$$

where
$$H_u = [H_1uH_2u \; - - - - \; H_mu]$$
$$H_v = [H_1vH_2v \; - - - - \; H_nv]$$
$$K = \begin{bmatrix} K_{11}K_{12} \; - - - - & K_{1n} \\ K_{21} \; - - & \\ - - & \\ K_{m1} \; - - - - - & K_{mn} \end{bmatrix}$$

with the coefficients subject to the conditions

$$K_{ik} = \left.\frac{d^{p_i+q_k}f(uv)}{du^{p_i}dv^{q_k}}\right|_{\substack{u=u_i \\ v=v_k}}$$

and the H functions subject to the constraints

$$\left.\frac{d^{p_i}(H_ju)}{du^{p_i}}\right|_{u=u_i}, \left.\frac{d^{q_i}(H_jv)}{dv^{q_i}}\right|_{v=v_i} = \begin{cases} 1 \text{ if } j = i \\ 0 \text{ otherwise} \end{cases}$$

Now in the same way as the H vector in the curve equation can be put in terms of the basis vector, $\bar{H}_u$ can be put in the form uM_u where M_u is an $m \times m$ square constant matrix and $\bar{H}_v$ can be expressed in the form M_vv^t where M_v is an $n \times n$ square constant matrix. It is usual in design work to let $m = n$ and the two basis vectors be the same in which case

$$M_u = M_v^t = M.$$

Thus $$f(u, v) = uMBM^tv^t$$

It can be seen that this surface form can include information internal to the surface as well as the boundary conditions. The next Section shows how this property can be used to advantage in surface design.

4. Dragging

The most important requirement when providing a mathematical definition of surfaces for design is to allow the designer to 'converse' with the surfaces in a natural manner. Thus it is essential that the equations involved should not be apparent to the user in any design operations. An interface has to be provided between the mathematics and the operations to which a designer is accustomed; for example he should deal in terms of points, slopes and lines on the surface. In previous attempts to interact with Coons' surfaces the user had been compelled to alter the coefficients describing the surface equations, regardless of whether these have any meaning to him. It is felt that with a computer-driven interactive display the most satisfactory way of providing the correct interface is to use a light pen or other pointing

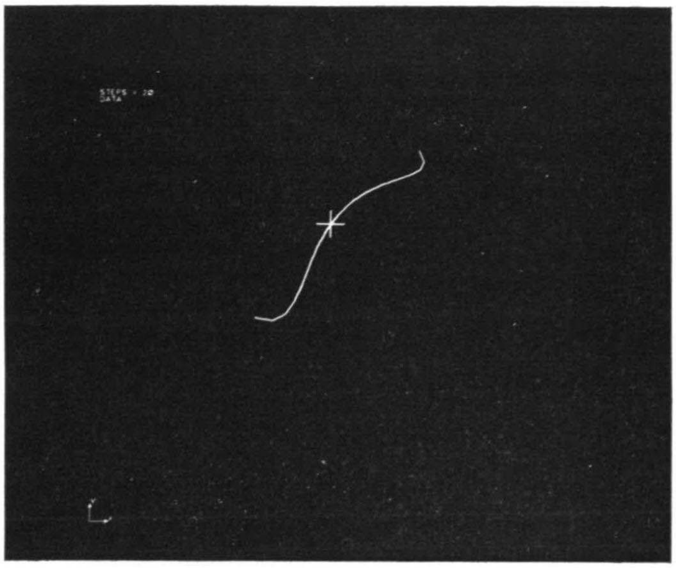

Fig. 1(a). Dragging a quartic curve—position a

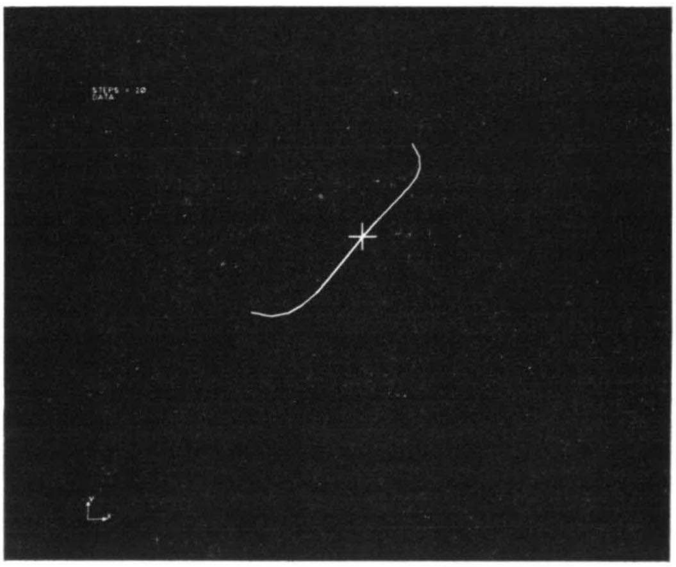

Fig. 1(b). Dragging a quartic curve—position b

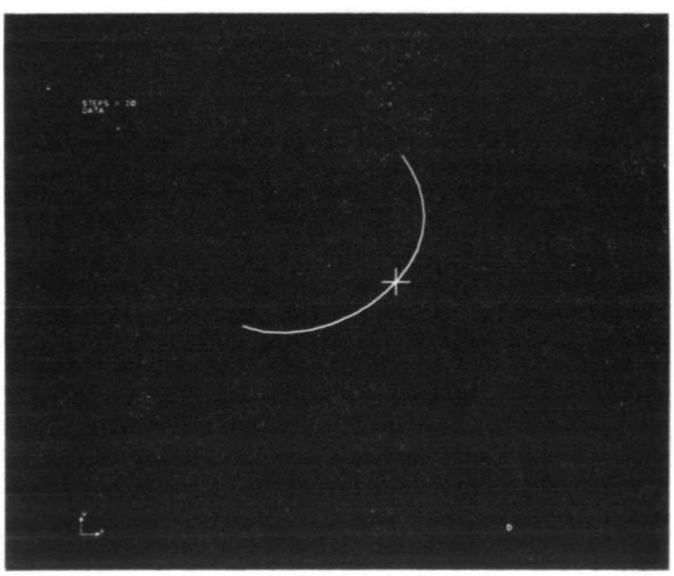

Fig. 1(c). Dragging a quartic curve—position c

device to indicate points to be altered. The designer can then proceed to move these points while allowing the program to relate alterations in points to alterations in surface coefficients. This moulding operation is called 'dragging'. With the class of surfaces under consideration a number of patches will be joined together at common boundaries with continuity across these boundaries. To give the designer control over the extent of the change caused by a dragging operation, the operation should only alter one surface. However, if the constraints on dragging conflict with the continuity constraints, then the effect will be transmitted further. At most, however, the effect should carry into the neighbouring surface. The range of the change will be termed the 'effective segment'.

The constraints which must be made on dragging operations are few but important. The first is that modifications to the surface should only occur in the plane of the display and no display depth coordinates should be affected. This virtually means that one is designing in two dimensions although if dealing with a rotated view of an object all three real dimensions are altering. The reasoning behind this constraint is fairly obviously that with two-dimensional displays only crude control can be provided over changes in depth. This constraint is, in fact, easily satisfied owing to the choice of surface description. Since the equations are defined parametrically the three coordinates can be modified independently of each other and the change in depth coordinate can always be held at zero.

The second constraint on dragging is that the effective segment should, if possible, retain its character. Although this constraint is a subjective quality it can by experience be defined more precisely by the following conditions:

(*a*) For a given dragging operation, the maximum distance of the curve from its first position to its final position should be small in relation to the length of the curve. This is a necessary condition for a curve to retain its character. It is important to note that intermediate positions of the curve should not affect the final position.

(*b*) The effective segment should retain its end points and end slopes for continuity of the first derivative and by definition nothing outside the effective segment should alter.

(*c*) The deviation from the original curve should be a maximum at the dragged point. This condition is not so stringent as the others although large variations from it cannot be tolerated.

(*d*) The deviation from the original curve should be of one sign, or in other words, the new curve should lie totally on the same side of the original curve.

Since the choice of boundaries for Coons' patches has been almost exclusively cubics it is important to consider their capabilities in dragging. The single cubic is completely defined by its two end points and two end parametric slopes. By condition (*b*), for a single cubic to be an effective segment, the end points and end real slopes must remain the same. It would appear that this would allow a scale factor to be applied to the cubic's parametric end slope to satisfy the dragging constraints. However, it can be shown that, when considering con-

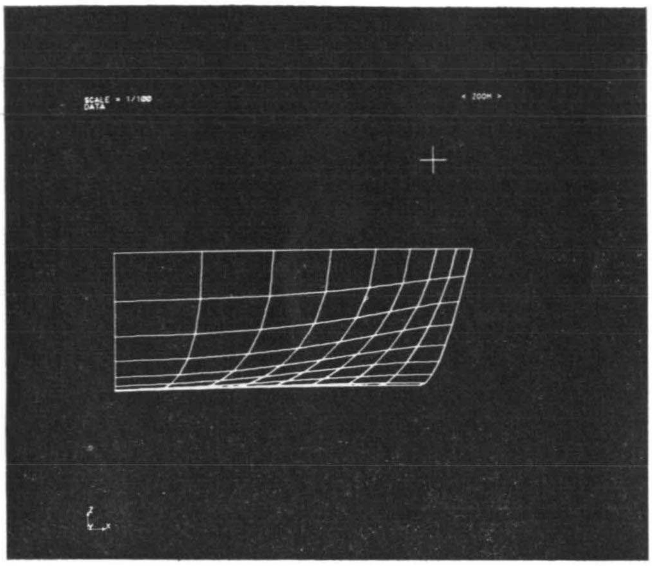

Fig. 2. Single surface patch before splitting

tinuity across boundaries of surface patches, this scale factor would have to be applied across the patch. Thus dragging a boundary curve in this way would affect the opposite boundary. It follows that if the condition is enforced that a single curve can be the effective segment, the cubic curve is unsatisfactory.

To allow the dragging of a single curve segment an extra degree of freedom is required over the cubic. The quartic is the most satisfactory curve type which provides this. The extra degree of freedom can be used to satisfy the dragging constraints while maintaining the other four degrees of freedom in order to satisfy the continuity conditions. In practice it is found that all the dragging conditions can be satisfied except condition (*c*): still more degrees of freedom are required to ensure that the maximum deviation occurs at the dragged point. However, the higher the order of the curve chosen the larger the number of points of inflexion which can be introduced and the more unsuitable the curve is for design. Using the terminology of Section 3 the quartic can be defined in terms of its end points, end slopes, and point at parameter $= \frac{1}{2}$, i.e.

$$f(u) = [H_1u H_2u H_3u H_4u H_5u] \begin{bmatrix} (0) \\ (1) \\ (0)_u \\ (1)_u \\ (h) \end{bmatrix}$$

where h indicated the parameter at value $= \frac{1}{2}$. By the method of differentials

$$\delta f(u) = \delta(0) \,.\, H_1u + \delta(1) \,.\, H_2u + \delta(0)_u \,.\, H_3u + \delta(1)_u \,.\, H_4u + \delta(h) \,.\, H_5u + f'(u) \,.\, \delta u$$

To keep the end points and end slopes constant

$$\delta(0) = \delta(1) = \delta(0)_u = \delta(1)_u = 0$$

It can also be assumed that $\delta u = 0$

Hence $\delta f(u) = \delta(h) \,.\, H_5u$

Thus the algorithm for dragging the central portion of a quartic curve is given by

$$\delta(h) = \delta f(u)/H_5u$$

It is found that when the point being dragged is central to the curve the variation from condition (*c*) is almost unnoticeable. When the point being dragged is close to the end of a single curve segment the deviation from condition (*c*) becomes unacceptable and some other algorithm must be used. It is, however, not enough for the end slopes to be changed without changing the end points since any other curve to which it is adjacent will change in the opposite direction. This will obviously contradict condition (*d*). To satisfy constraint (*d*) the common end point must be changed together with the slope at that end point, and hence the effective segment becomes two curve segments. Since for dragging purposes the two equations will act as a single segment the required alteration to the end point and end slope can be found by adding a single quartic curve over both segments. If the two curves have parametric scale factors of r_1 and r_2 (normally both 1 for independent curves but see Section 5 on surface splitting) then the effective parameter value at the join is

$$u_p = \frac{r_1}{r_1 + r_2}$$

If the same H_5 blending function is used as for quartic dragging then its value at the end point (parameter u_p) is H_p and at the dragged point is H_d. The change in the end point is then given by

$$\delta f(u)_p = \delta f(u)_d \,.\, \frac{H_p}{H_d}$$

In a similar way the change in the end slope is given by

$$\delta f'(u)_p = \delta f(u)_d \,.\, \frac{H_p'}{H_d}$$

The equation of the quartic surface is given by

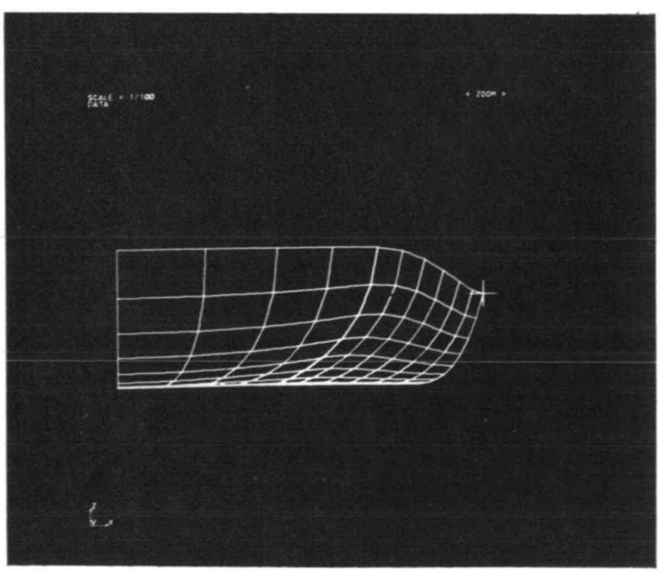

Fig. 3. Surface after splitting showing the effect of dragging a corner point

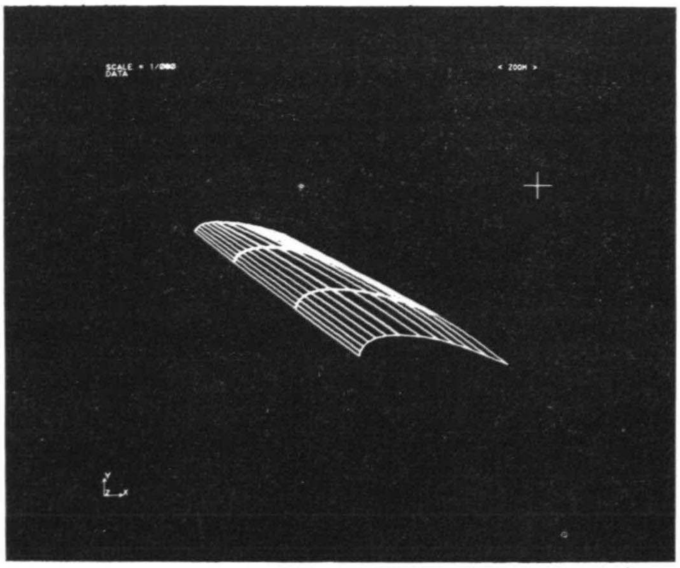

Fig. 4. Six patch surface (patch boundaries are displayed brighter)

$$f(u, v) = uMBM^tv^t$$

$$\text{where } M = \begin{bmatrix} -8 & -8 & -2 & 2 & 16 \\ 18 & 14 & 5 & -3 & -32 \\ -11 & -5 & -4 & 1 & 16 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{and } B = \begin{bmatrix} (00) & (01) & (00)_v & (01)_v & (0h) \\ (10) & (11) & (10)_v & (11)_v & (1h) \\ (00)_u & (01)_u & (00)_{uv} & (01)_{uv} & (0h)_u \\ (10)_u & (11)_u & (10)_{uv} & (11)_{uv} & (1h)_u \\ (h0) & (h1) & (h0)_v & (h1)_v & (hh) \end{bmatrix}$$

where h indicates the parameter at value $= \frac{1}{2}$. It is interesting to note that by using quartic boundaries not only do the boundaries have extra degrees of freedom for dragging but the surface has a coefficient (hh) which can be altered without affecting any neighbouring surfaces. For dragging a surface the techniques used for dragging curves can be simply extended.

A single quartic curve with a central point dragged to a number of positions is shown in **Figs. 1(a)**, **(b)** and **(c)**.

5. Splitting

The previous section has described a method of designing with curves and surfaces while controlling the extent of the alteration. In practice, as a design progresses the range of an alteration will become smaller and more local. The corresponding technique which can be used in designing surfaces with patches is that initially the complete design surface is described by a single patch and hence any dragging will affect the complete surface. As the design develops a requirement for more flexibility in the surface and a more local area of design is felt. To satisfy this need the surface is 'split' along some line lying on the surface: the result is two patches, which are constrained to be continuous across their common boundary, instead of the original single patch. The two patches now provide the extra degrees of freedom and the local area of influence required for further dragging operations. The designer can progress in this manner successively splitting patches and sub-patches and dragging more and more local regions.

The only constraint on splitting is a simple one: identity. In other words, the result of splitting a patch into two parts should be that the two sub-patches should initially be identical to the single patch. If this is not the case the splitting process will not be a convergent one. Since the line about which a patch is to be split must therefore become a new boundary curve and the two new patches should be of the same type as the original, this line must be an iso-parameter line on the original patch.

If the split is made along a line of constant parameter, u, at value $u = t$, and the new sub-patch between $u = 0$ and $u = t$ represented by

$$\bar{f}(s, v)$$

then $$u = st$$

and

$$\begin{aligned} \bar{f}(10) &= f(t0), & \bar{f}(11) &= f(t1) \\ \bar{f}_s(00) &= t\,.f_u(00), & \bar{f}_s(01) &= t\,.f_u(01) \\ \bar{f}_s(10) &= t\,.f_u(t0), & \bar{f}_s(11) &= t\,.f_u(t1) \\ \bar{f}_{sv}(00) &= t\,.f_{uv}(00), & \bar{f}_{sv}(01) &= t\,.f_{uv}(01) \\ \bar{f}_{sv}(10) &= t\,.f_{uv}(t0), & \bar{f}_{sv}(11) &= t\,.f_{uv}(t1) \end{aligned}$$

If all the other values remain the same as on the global patch it can be shown that for the tensor form of the surface

$$\bar{f}(s, v) \equiv f(u, v)$$

i.e. the two surfaces are identical in their mathematical form. It is useful to consider that the new patch is the same as the original patch with a scale factor t applied in one parametric direction. The residual patch is also identical with a scale factor of $(1 - t)$. **Fig. 2** shows a single surface patch before splitting. In **Fig. 3** the

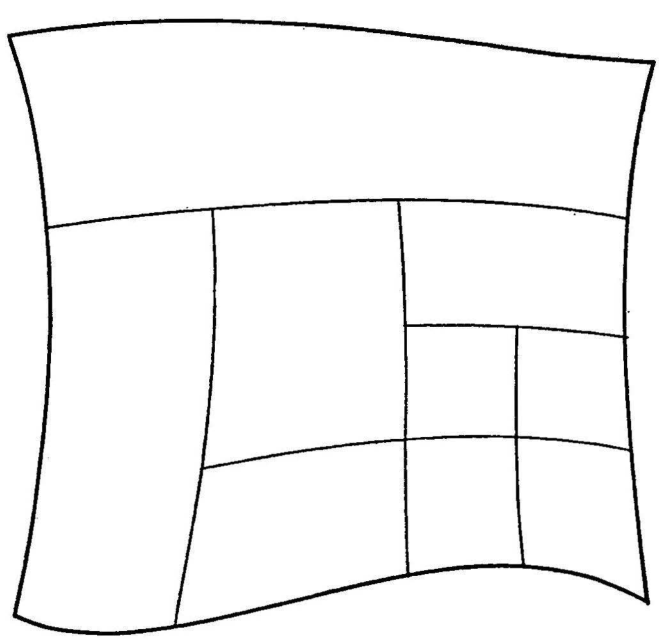

Fig. 5. Example of irregular splitting of surface

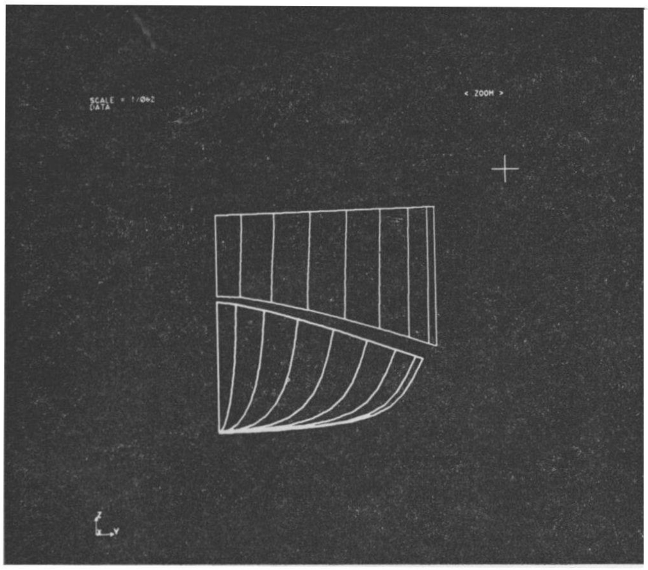

Fig. 6. Two patches before joining

surface has been split and it can be seen that the effect of dragging a corner point is localised to a single patch. **Fig. 4** shows a surface which has been split into six patches. The implications of the proof of splitting can be quite significant. A very important condition of this proof is that all the terms in the tensor must be present. In published work on the development of bi-cubic surfaces what Coons refers to as the first canonical form and the first correction surface have been treated separately and then added together to form one surface equation. In some work the importance of including twist terms in the correction surface has been neglected by making them equal to zero (Ferguson, 1963). However, for splitting a surface, since the twist terms of the sub-patch at the end of the operation are non-zero, the full tensor form must be used. Thus the need for splitting has shown that Coons' canonical and first correction surface should always be treated as one combined surface.

Since the constraint of identity of splitting is satisfied there is no need to have a rectangular mesh of patches describing a surface. It is perfectly possible to split sub-patches into several smaller units and so produce an irregular surface such as is shown in **Fig. 5**. This capability is extremely important since it truly allows the designer to deal with local areas of the design. In a region where more detail is required it is possible to have more patches and because of the dragging constraints modifications will be local to that area. This does not, however, stop other regions being treated in a less detailed manner.

6. Joining

Having illustrated the requirement for splitting patches in order to converge on a design, the requirement for joining patches should not be neglected. It was stated in the last section that the approach to be used in design is first to approximate the complete surface with a single patch and then to converge towards design satisfaction from that point. There are situations though, when it is more useful to be able to join surfaces before the convergence begins.

The first of these situations is when the complete surface has a discontinuity in slope such as would occur at a knuckle or chine on a ship. The line of discontinuity is a design constraint in the same way as the overall boundaries, and it will always be configured as a patch boundary. It is useful in this case, therefore, since the influence of one side of the discontinuity on the other is minimal, to design both parts separately and join them together at a later stage.

Another situation in which it is more useful to join surfaces is when the continuous surface is composed of two distinct objects joined by blending surfaces. An example of this would be an aircraft fuselage and wing with the wing fillet as the blending surface. Although this forms one continuous surface it is important that the designer should consider the wing separately from the fuselage and should leave the blending problem until later.

The problem of joining surfaces is more straightfoward than that of splitting and the only constraint which can be applied is that one of the patches (the principal patch) does not change. In splitting surfaces it was seen that a patch has associated with it a scale factor in each parameter direction relating the surface coefficients to a common parametric space. In joining surfaces the two parametric spaces of the surfaces are distinct and do not have a common factor. The most satisfactory solution is found by keeping the scale factor in both the principal patch and the subsidiary patch constant while altering the slopes in the subsidiary patch for first order continuity satisfaction. **Fig. 7** illustrates the result of joining the two surface patches shown in **Fig. 6**.

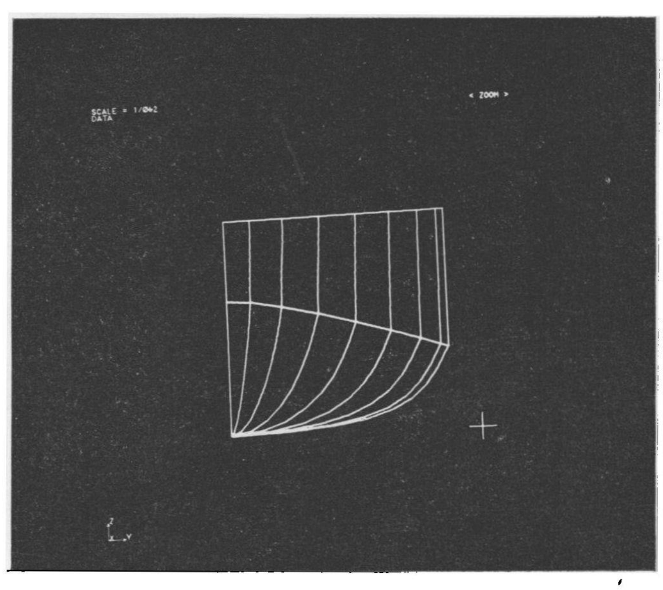

Fig. 7. Two patches after joining with slope continuity at one end of the common boundary

7. Conclusions

This paper has described a class of surfaces, based on those of Coons, suitable for interactive design work. Techniques have been developed which allow the designer to modify the surface in a natural way and to introduce in the design process an implicit definition of the range of any alteration. Thus the designer can start with a general concept of his design and as he progresses and his ideas develop he can converge on his requirements, introducing more flexibility and detail into local areas.

The type of surface chosen is a bi-quartic as opposed to the bi-cubic used by previous workers. The choice of this degree follows from the requirements for dragging curves without the effect propagating over a number of patches. Using the bi-quartic it is possible to change the inside of the surface without affecting any neighbouring patches.

The type of surface described is most useful in applications where free-form surfaces are present. Where the surface is entirely constrained by design considerations there is little creative work to be done. In many applications, however, such as ship hull design, car body styling, and blending surface design, the constraints indicate only the general shape of surface required. Creative design based on aesthetics, experience, or a number of empirical rules, is still required in these applications. It is here that the use of an interactive display can provide important advantages over traditional techniques.

8. Acknowledgement

The work described was done on a PDP–7 with model 340 display and is being carried out under Contract F/HP/09 placed by the Minister of Technology with Imperial College, for the study of the application of Interactive Computer Graphics in Ship Design and Production. The PDP–7 and 340 display were partly supported by Science Research Council Grant B/SR/2071 'Computer Processing of Three Dimensional Shapes'.

The author is grateful to the Minister of Technology for permission to publish the paper.

References

ARMIT, A. P. (1969). A Multipatch Design System for Coons' Patches, I.E.E. Conference Publication No. 51 I.E.E. International Conference on Computer Aided Design, Southampton, pp. 152–161.
COONS, S. A. (1964). Surfaces for Computer-Aided Design of Space Figures, M.I.T. MAC-M-253.
COONS, S. A. (1967). Surfaces for Computer-Aided Design of Space Forms, M.I.T. MAC-TR-41.
ESHLEMAN, A. L., and MERIWETHER, H. D. (1967). Graphic Applications to Aerospace Structural Design Problems, Douglas Aircraft.
FERGUSON, J. C. (1963). Multi-Variable Curve Interpolation, Boeing Company, D2-22504.
FLANAGAN, D. L., and HEFNER, O. V. (1967). Surface Molding, *Astronautics and Aeronautics*, p. 58.
HAMILTON, M. L., and WEISS, A. D. (1965). An Approach to Computer-Aided Preliminary Ship Design, M.I.T. ESL–TM–228.
HOGUE, W. M. (1966). Computer-Aided Design in Body Engineering, Society of Automotive Engineers Congress.
SABIN, M. A. (1968). Private Communications.

Book review

Sparse Matrix Proceedings, Edited by Ralph A. Willoughby, 1969; 184 pages. (*IBM*)

This book consists of papers (or more often extended abstracts thereof) presented at a symposium on sparse matrices and their applications organised by IBM. The papers may be divided into four groups, namely those concerned with

(*a*) Establishing the equations,
(*b*) Manipulative techniques,
(*c*) Computational techniques,
(*d*) Reviewing the field.

The first group contains papers showing how problems in structural engineering, electrical power system analysis, etc., give rise to large sets of linear and nonlinear algebraic equations involving sparse matrices, and papers in the second group describe methods of sorting and ordering the equations to reduce either the matrix bandwidth or the subsequent 'fill-in' of zero elements when performing an LU decomposition. Many papers in this group lean heavily upon graph theory although the most complete and self-contained of these (by Tewarson) only uses matrix algebra. The papers in the third group describe essentially programming devices to handle either the raw or processed matrices. List-processing and bit-map techniques are discussed and compared and several papers describe what amount to special-purpose compilers that determine the non-zero elements in the factors L and U and then generate a fast program (sometimes at machine-code level) to compute these and only these elements explicitly. The fourth group consists of three eminently readable papers by Wolfe, Orchard-Hays, and Danzig *et al.*, which survey the field of linear programming and the development of sparse matrix packages. The final item is the edited transcription of the panel discussion at which the point was made with some force that operating systems and manufacturers' software often prevent efficient implementation of the algorithms described by the various authors. The consistency and continuity of this final piece of reporting is indicative of the trouble taken by the editor, Ralph Willoughby, in carrying out his task. Copies of the proceedings may be obtained, while stocks last, from

Ralph A. Willoughby, Editor,
International Business Machines Corporation,
Thomas J. Watson Research Center,
Post Office Box 218,
Yorktown Heights,
New York 10598,
U. S. A.

and anyone concerned with sparse matrices should take advantage of this free offer.

C. G. BROYDEN (Colchester)