Chapter One

Introduction
Introduction

The engineering environment has been revolutionized by the advent of Computer technology. Computer Graphics is placed at the forefront of this revolution and plays a fundamental role in engineering design, architecture and entertainment. Computer Graphics provides a set of tools to create pictures and to interact with them in a natural way. These tools comprise both hardware and software and together they permit programmers to fashion programs with a strong graphics capability. Data are presented visually through shapes, colours and textures rather than by tables of numbers. Words and numbers are replaced whenever possible by pictures, because the eye-brain system is very fast in recognizing and interpreting visual representations.

1.1 Interactive Graphics

With interactive graphics, a person instructs the computer using natural hand movements such as pointing and drawing. Interactive graphics involves two-way communication between the computer and the user. Upon receiving signals from the input device, the computer can modify the displayed picture appropriately. To the user it appears that the picture is changing instantaneously in response to commands. One can give a series of such commands each generating a graphical response from the computer. In this way, the user maintains an I/O dialogue with the computer.

The place where the user works with an interactive graphics application is known as the Graphics Workstation. It may be just a personal computer or a graphics terminal connected to the remote host. A "high end" graphics workstation has a local processor, mass storage, a high-resolution graphics display and various specialized input devices. In addition to a graphics display some workstations also have a dialogue display. A workstation that contains both types of terminals can separate pure drawing of pictures from any dialogue between the user and the application, which leaves the entire graphics display surface free for drawing. Furthermore, because the dialogue display is large, the user can be given more complete prompts for the next choice of action. A menu of options can remain fixed on the dialogue display at all times without interfering with the graphics. When a dialogue display is not available much the same effect can be obtained using windowing facilities. Among other handy features, windows permit portions of the display surface to be covered momentarily with message areas or menus. Although these dialogue
areas obscure parts of the display while they are being used, the obscured underlying parts are restored when the window is no longer needed.

In the last few years, the performance of graphical workstations had considerable growth, and as a result, solid rendering has become possible in the interactive design of 3D geometrical structures. Historically, the evolution has been towards more and more design intuitive / interactive input systems. A good example is the dynamic motion of the B-spline polygonal net or rubber sheeting [Loney, G.C. et.al. 1987]. With cheaper hardware, economic barriers to further development were removed and during the early 1970s, software systems for 2D, 2½D and 3D shapes and surface modelling began to provide the engineer with the ability to design and handle complete information concerning products and plants. Parallel developments in databases and graphic software for visual displays allowed the concept of interactive workstation to emerge.

Computer Aided Design acted as a focal point for a ferment of new ideas that provided the driving force for the fundamental changes in the way computers were used. The early CAD pioneers were motivated by the need to interact with the computer in a new kind of partnership. Much more was heard of man-machine interface and need for developing creative ideas using a graphic display for visual communications; since translating ideas into reality means describing and defining solid 3D objects, much more effort was devoted to 3D modelling and visualization. At the same time, since object definition involves a richness of product and object definition in many cases is the generation of vast volumes of data, much attention was paid to the nature and structure of databases to hold such information. Hence, the trend towards the description of curves and surfaces started. The field of computer aided curve and surface design started emerging from France and the United States [Farin, G. 1989]. In France, P.de Casteljau started the work on a curve and surface design system in 1959 (for Ciferon) and a short time later, P.Bezier started the work on his Unisurf system (for Renault). Also in the very early sixties J.Ferguson at Boeing worked on the implementation of cubic splines into a design system and S.Coons developed this surface schemes at MIT. The subject was normally named "Computer Aided Geometric Design"
1.2 F-Patch

One of the oldest techniques for defining curves and surfaces for CAD is due to Ferguson. [Piegl,L. 1989, Farin,G 1989]. Generally, free form curves and surfaces do not possess simple analytic forms. Instead, they are defined in piece wise segments and joined together with some specified continuity. With the advent of computer, composite computational methods were devised defining the composite surfaces as an assembly of curvilinear quadrilateral patches. Ferguson described the patch system in terms of parametric rather than Cartesian coordinates. By using parametric forms independence is gained from a particular system of coordinates. Using hermite interpolation Ferguson defines a basic curve as follows.

\[ P(u) = ( A_0, A_1, T_0^u, T_1^u) \cdot h(u), \ u \in (1,0) \] \[ Eq.1.1 \]

where \( h(t) = \begin{cases} (2t+1)(t-1)^2, & \text{for } 0 < t < 1 \\ (3-2t), & \text{for } 0 < t < 1 \\ t(t-1)^2, & \text{for } 0 < t < 1 \end{cases} \]

\( A_0 \) and \( A_1 \) are the end points and \( T_0^u \) and \( T_1^u \) are the tangent vectors at the end points. \( t \) is the interpolation parameter in the direction of \( u \).

This interpolation generalizes to surfaces as follows

\[ P(u,v) = [ P_0(u), P_1(u), ( T_0^v, T_1^v) \cdot k(u), ( T_0^v, T_1^v) \cdot k(u) ] \cdot h(v), \ u,v \in (0,1) \] \[ Eq.1.2 \]

where \( k(t) = (2t+1)(t-1)^2, t^2(3-2t) \)

and the surface described by this equation, often called the F- patch is used to interpolate through a set of rectangularly arranged 3D data points. There are two basic ways to use Ferguson's method: input tangents at every data point in \( u \)- and in \( v \)-directions and allowing the system to compute automatically. The first of these two methods although theoretically possible, it is practically totally infeasible. [Piegl,L. 1989]. To get the tangents automatically using only positional information, Ferguson suggested using cubic splines to interpolate each row and column yielding a network of curves of continuity class \( C^2 \). The surface interpolant will, in general be \( C^1 \) only since isoparametric lines are simple cubic hermite curves.
1.3 The Coons Patch

A general theory of surface patches was described by Coons which showed how four boundary curves can be blended into a smooth patch using general blending functions allowing any of continuity between patches. Whereas the F-patch interpolates to positional data i.e. to points, the Coons patch interpolates to wire-frame data i.e. to the boundary curves of a rectangular patch [Forrest, A.R. 1972, Coons, S.A. 1974, Piegl, L. 1988].

It has the form

\[
P(u,v) = \sum_{i=0}^{1} \sum_{r=0}^{1} P_{r0}(i,v) h_{r}^{i}(u) + \sum_{j=0}^{1} \sum_{s=0}^{1} P_{0s}(u,j) h_{s}^{j}(v) - \sum_{i=0}^{1} \sum_{r=0}^{1} \sum_{j=0}^{1} \sum_{s=0}^{1} P_{rs}(i,j) h_{r}^{i}(u) h_{s}^{j}(v) \quad \cdots (Eq. 1.3)
\]

where \( h_{r}^{i}(u) \) and \( h_{s}^{j}(v) \) are blending functions.

Coons work has been very significant in that it has stimulated the development of other surface representations. A rigorous mathematical foundation of Coons and Ferguson type surface definitions as well as methods to interpolate network of curves were given by Gordon.

1.4 Bezier Curves and Surfaces

One of the most popular and widely used methods of CAD is the Bezier method. [Gordon, W.J. et.al. 1974, Piegl, L. 1984, Piegl, L. 1989]. The introduction of the control polygon by Bezier helped bring designers the intuition needed in the efficient control of surface geometry. Modifying the polygon modifies the curve in a predictable way. Surfaces are defined in a similar way by the input of open polygon.

From the mathematical point of view, any curve in 3D can be expressed as follows

\[
P(u) = a_{0} + \sum_{i=1}^{3} a_{i} f(u) \quad \cdots Eq. 1.4
\]

where \( a_{0} \) is a translation vector and \( a_{1}, a_{2}, \) and \( a_{3} \) form a basis on 3D. Curves can be defined in \( 'n' \) dimensions in the same way and subsequently the result can be projected on to 2D or 3D. The resulting curve has the form

\[
P(u) = a_{0} + \sum_{i=1}^{n} a_{i} f_{in}(u) \quad \cdots Eq. 1.5
\]
where $a_i$ are the projections of the $n$ dimensional vector and

$$f_{in}(u) = \frac{(-u)^{i-1}}{(i-1)!} \frac{d^{i-1}}{du^{i-1}} \left[ (1-u)^n - 1 \right]$$

From a pure mathematical point of view, the Bezier curve is the Bernstein approximation to the control polygon i.e. it is not a good approximation especially in case of high degree. However from a design point of view this is not a big disadvantage because the control polygon is not considered as a primitive function to be approximated. It is used instead as a design tool to generate free form curves.

There are number of ways to define surfaces using Bezier technique. Probably the widely used surfaces in industry are those of tensor product surfaces using the Bernstein basis functions and they can be written in the following form.

$$P(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} p_{ij} B_i^m(u) B_j^n(v) \quad Eq\ 1.6$$

where $p_{ij}$ form a control net.

The important property of Bezier technique is that it provides a nice geometry with which curves and surfaces can be easily computed and modified. Having sketched the control polygon, the basis function can be forgotten since polygon conveys all information necessary to deal with the curve. Because of its simplicity and geometrical background Bezier technique is popular.

### 1.5 B-Splines

B-Splines are proper generalizations of Bernstein polynomials and therefore Bezier curves and surfaces from a subspace of B-splines curves and surfaces. B-splines were first introduced by Schoenberg. [Piegl,L. et.al.1987, Piegl,L.1989, Farin,G. 1989]. The definition in the divided difference form is as follows.

Let $t_i \leq t_{i+1}$ be real numbers for
\[ i = \infty \to + \infty \text{ and } g_i(s+t) = (s-t)^k - 1 = \begin{cases} (s-t)^k & s \geq t \\ 0 & s < t \end{cases} \]

then \( M_i, p(t) = g_k(t_i, \ldots, t_{i+p+1}, t) \)

In Computer Aided Design (CAD), the normalized B-spline is

\[ N_i, p(t) = (t_i + p + 1 - t_i) M_i, p(t) \quad \text{...Eq 1.7} \]

B-Splines applied to curve and surface definitions by Reisenfeld [Barnhill,R..et.al.1974, Gordon,W.J. et.al. 1974] were found to be useful for making local modification possible on curves and surfaces. Gordon and Reisenfeld made use of the control polygon in their work with B-splines, exploiting the non-global behaviour of the B-spline basis. Multiple coincident vertices are used to pull the curve closer to the control polygon while increasing the order of the spline polynomial tends to pull the curve away from the polygon. Local control is added by changing vertex location. This concept is extended to B-spline surfaces and proves to be useful as an interactive tool for surface definition.

Curve and surface schemes discussed above are used to represent free form curves and surfaces in engineering design. However, standard analytical shapes such as lines, circles, conics, and quadric surfaces are as important as free forms. Rational curves and surface schemes are capable of incorporating both the types of elements. The notion of using rational polynomials for CAD applications arises from the discipline of projective geometry [Riesenfeld,R.F.1981]. The concept is to use homogeneous coordinates. [Roger,D.F. et.al.1985,Arokiasamy,A.1989, Newman,W.M.et.al. 1979]

Most of the techniques discussed above can be used to fit curves and surfaces to data. Cubic splines, B-spline based techniques, bicubic interpolation or lofting are examples of widely used methods. Now, developments in triangular patches, contouring, four-dimensional surfaces and multistage methods make surface representation easier, given scattered data. In all the techniques one has to feed more amount of data for proper representation, using complicated mathematics. Moreover, control points are to be specified.
The aim of this thesis is to generate surfaces and objects with much easier methods without feeding more data or control points. Using standard analytic shapes such as circles, squares, epicycloids and a combination of the above, many different classes of surfaces and objects are generated. During interpolation the curvature of the profile is easily changed and hence the shape changes accordingly as different contours are used to generate the surface or the object. Moreover, by proper choice of coordinate system, the Z coordinate of the surface (in the XY plane) is changed and hence the local control of the point generated is achieved.

The main advantage of this work is that when two contours are interpolated, a desired cavity is generated in the surface (Fig. 83 to Fig. 90). This is achieved when the Z coordinate is made zero. When the Z coordinate is allowed to change as a function of the curve generation parameter or as a function of the interpolating parameter many different varieties of surfaces and objects are generated.

1.6 Interpolation Techniques with Single Contours

While interpolating single contours, generally, the surface element is generated in the XY plane with Z=0. The boundary of the surface element is decided by the representation of the contour. Only the parametric representation of the space curve is chosen and hence it is possible to combine different contours in generating a surface. For example, when a surface is drawn, one half of the surface may have a circular boundary and the other half may be the sides of a square. Any portion of the boundary can be changed by changing from one contour to the other (Fig. 40 to Fig. 44). It amounts to changing the curvature of the profile. In this approach the connectivity of the contour is automatically taken care of.

While generating any surface, it is difficult to introduce a desired cavity in the surface. If the interpolation parameter is initialized to 0.5 instead of 0.0 then a cavity with the boundary as that of the outer boundary will be automatically introduced. This is termed as "self cavity surface" in this work (Fig. 49 to Fig. 61). This technique is useful in designing washers which have the same inner and outer boundaries. The only problem with this approach is that self intersecting curves cannot be used. It is well known that use of self intersecting curves leads to an undesirable effect in interpolation (Fig. 82 and Fig 127).
Surface normal can be used to indicate the shape and orientation of a surface patch. The direction of the normal can be changed by the proper choice of the Z-coordinate. Instead of specifying control points the Z-coordinate can be allowed to change as a function of the interpolating parameter (Fig. 63 to Fig. 82) or as a function of the curve generating parameter. In order to make things simpler and to avoid inputting data the Z-coordinate will be made as a value of a trigonometric or exponential function of the interpolating parameter. As one knows the variation of these simple functions, it is easy to predict the change and hence the nature of the surface. But when the Z-coordinate is made as a function of the curve generation parameter, sometimes it will introduce torsion in the surface and it becomes difficult to predict the nature of the surface.

Finally, with single contours, curvature control and normal control can be combined to produce a completely new class of surfaces and objects. With all these cavities can also be introduced (Fig. 69 to Fig. 71).

1.7 Interpolation Techniques with Two Contours

While interpolating single contours self cavity surfaces are generated where the inner and the outer boundaries are same. But when two contours are interpolated any desired cavity not the same as that of the outer can be introduced. If two contours are to be interpolated one has to start with two contours which are space curves in 3D. One contour starts with the interpolating parameter 0.0 and the other with 1.0. When the Z-coordinate is made zero a flat surface with the desired cavity is generated (Fig. 83 to Fig. 90). When the Z-coordinate is allowed to vary as a function of the interpolating parameter(Normal control) any type of surface with a cavity can be generated. Also either the outer boundary or the inner boundary can be controlled as it is done with the single contour interpolating techniques. Both normal control and boundary control can be coupled depending upon the requirement.

1.8 Non-Linear Interpolation with Two Contours

The process of object generation involves selecting one point from each of the two contours and connecting them through the interpolating functions which are non-linear functions of the interpolating parameter (Fig. 98 to Fig. 102). This curve defines the shape of the object in the Z-direction at these particular points. It should be noted that the shape is not the same, in general, as one moves from one point to the next on the given contours.
A wide choice of interpolating functions is possible in the case of non-linear interpolation and hence a rich class of objects can be generated. With the proper choice of the contours and using also by mixing different interpolating functions it is possible to generate any type of object as shown in Fig. 102.

1.9 Applications and Conclusion

This work shows that the interpolation technique is a versatile technique by generating a variety of surfaces and objects. With this technique, a cylinder of any cross section (Fig. 92 and Fig. 93), a cone of any base (Fig. 65 to Fig. 68), baskets (Fig. 104 and Fig. 105), bottles (Fig. 106 and Fig. 107), flower vases etc., are generated. Washers with any desired cavity and nuts are designed. As a major application vibrations of rectangular and circular membranes are simulated. Many objects generated with this technique cannot be easily generated by other techniques such as surface of revolution (Fig. 100 and Fig. 102), sweeping, etc. Although complex surfaces and objects are generated, the technique involves simple mathematics. The main advantage is that there is no need for inputting enormous data.

1.10 Implementation

The whole work has been implemented using Turbo Pascal (Version-7). The details regarding the parametric equations, interpolating functions and the range of values used for generating each surface and object are provided in the appendix.