

RATIONAL CUBICS AND CONICS REPRESENTATION: A PRACTICAL APPROACH

M. Sarfraz

*Department of Information & Computer Science, King Fahd University of Petroleum and Minerals, KFUPM #
1510, Dhahran 31261, Saudi Arabia.
sarfraz@ccse.kfupm.edu.sa*

Z. Habib

*National College of Textile Engineering, Faisalabad, Pakistan.
zulfiqar_habib@hotmail.com*

Abstract: A rational cubic spline, with one family of shape parameters, has been discussed with the view to its application in Computer Graphics. It incorporates both conic sections and parametric cubic curves as special cases. The parameters (weights), in the description of the spline curve can be used to modify the shape of the curve, locally and globally, at the knot intervals. The rational cubic spline attains parametric C^2 smoothness whereas the stitching of the conic segments preserves visually reasonable smoothness at the neighboring knots. The curve scheme is interpolatory and can plot parabolic, hyperbolic, elliptic, and circular splines independently as well as bits and pieces of a rational cubic spline.

Key Words: Computer Graphics, Interpolation, Spline, Conic, Rational Cubic,

1. INTRODUCTION

Piecewise rational cubic spline functions provide a powerful tool for designing of curves, surfaces and some analytic primitives such as conic sections that are widely used in engineering design and various Computer Graphics applications. These applications may be representing some font outline, round corner in an object, or it may be a smooth fit to a given data. Several segments of curves, to compose a desired curve outline, can have different mathematical descriptions. For example, a font "S" when designed, appears to have straight lines, conics, and cubics as essential parts of its outline. Single mathematical formulation for the precise definition of various types of geometry shapes is one of the major advantages of the rational cubic spline functions.

This research describes the parametric C^2 rational cubic spline representation possessing a family of shape control parameters. This family of shape parameters has been utilized to produce straight line segments, conics, and cubics. The features of maintaining some reasonable amount of continuity between conic and cubic arcs, estimated end derivatives, conic (circular, elliptical,

parabolic, and hyperbolic) splines, and circular arcs for given radius or center, are further achievements in this research.

There are many schemes in the literature for shape control using cubic interpolants. For brevity, the reader is referred to work done by various authors^[1-14]. The presented curve scheme encompasses and extends the results of Sarfraz et al.^[10]. In Sarfraz et al.^[10], a C^1 rational cubic spline with approximated derivatives at control points, was used but continuity between conic and cubic arcs was not discussed. Intermediate point interpolation scheme and circular arcs, presented in Jamaludin^[7], are not practical as the space curves and exact circular arcs are not possible. In Gregory et al.^[4-5], end derivatives are based on the assumption of the user, which is not convenient. Moreover, the conics were not discussed at all. We have estimated most suitable end derivatives for more pleasing results. In Hoschek^[4], rational quadratic spline is used for circular spline. We are using very simple technique using rational cubic spline for the same circular spline. In addition, the scheme has the following properties, which may lead to a more useful approach to curve and surface design in CAGD:

- The curve has C^2 continuity between the rational cubic arcs and G^1 continuity between cubic and conic arcs.
- Most suitable end derivatives are estimated.
- The scheme is local, i.e. shape control parameters will not significantly affect the adjacent parts of the design curve.
- Any part of the rational cubic spline can be made conic (with exact circle and ellipse) or straight line using the same interpolant.
- The method is suitable for space curves and hence can also be generalized to surfaces.

The paper has been organized in such a way that a C^2 parametric rational cubic spline scheme, together with determination of tangents at the knot points, is considered in Section 2. Analysis of the designing curve has been made in Section 3. Conditions for conics and straight line segments are given in Section 4. The Section 5 mentions about circular arcs. This section also

covers all types of circular arcs in space. In Section 6, we have presented a scheme to calculate end derivatives (tangents). For practical purposes, an algorithm has been designed in Section 7 where as the demonstration has been made in Section 8. The Section 9 concludes the paper.

For the construction of a C^2 rational cubic spline in Subsection 2.1, we need to manipulate second derivative of (2.1), which is as follows:

$$P_i^{(2)}(t) = \frac{2 \{A_i q^3 + B_i q^2 (1 - q) + C_i q (1 - q)^2 + E_i (1 - q)^3\}}{h_i \{1 + (g_i - 2)q(1 - q)\}^3} \quad (2.3)$$

Where

$$\begin{aligned} A_i &= (g_i + 1)(D_{i+1} - D_i) - D_{i+1} + D_i, \\ B_i &= 3(D_{i+1} - D_i), \\ C_i &= 3(D_i - D_i), \\ E_i &= (g_i + 1)(D_i - D_i) - D_{i+1} + D_i, \end{aligned} \quad (2.4)$$

And

$$D_i = (F_{i+1} - F_i) / h_i. \quad (2.5)$$

Fig. 1 Spline curves with various end conditions: (a) with distance based derivatives, (b)-(c) with exact derivatives.

2. THE RATIONAL CUBIC SPLINE

Let $F_i \in \mathbb{R}^m, i = 0, 1, \dots, n$, be a given set of data points at the distinct knots $t_i \in \mathbb{R}$. Also let $D_i \in \mathbb{R}^m$, denote derivative values at the knots. Define a parametric C^0 piecewise rational cubic $P : \mathbb{R} \rightarrow \mathbb{R}^m$ as follows:

$$P(t) = P_i(t) = \frac{N_i(t)}{M_i(t)}, \quad i = 0, 1, \dots, n - 1, \quad (2.1)$$

Where

$$N_i(t) = (1 - q)^3 F_i + q(1 - q)^2 (g_i + 1)W_i + q^2 (1 - q)(g_i + 1)W_i + q^3 F_{i+1}$$

$$M_i(t) = (1 - q)^2 + q(1 - q)g_i + q^2,$$

$$q = (t - t_i) / h_i, \quad h_i = t_{i+1} - t_i.$$

The following choice of control vertices

$$\begin{aligned} V_i &= F_i + \frac{1}{1 + g_i} D_i, \\ W_i &= F_{i+1} + \frac{1}{1 + g_i} D_{i+1}, \end{aligned} \quad (2.2)$$

leads (2.1) to a C^1 piecewise rational cubic Hermite spline. The choice of parameters $g_i > -1$ ensures a strictly positive denominator in the rational cubic. Thus from Bernstein Bezier theory, the curve lies in the convex hull of the control points $\{F_i, V_i, W_i, F_{i+1}\}$ and is variation diminishing.

2.1. Estimation of Tangent Vectors

There are different choices of the tangent vectors D_i at F_i , which can be opted for practical implementation for the computation of a curve with specific amount of smoothness. For C^1 curve methods, some reasonable tangent approximation method can be used. The distance-based approximations are found reasonably good as far as pleasing smoothness is concerned. We, now, define the tangent vectors D_i at F_i . For open curves, the end conditions are defined as:

$$\begin{aligned} D_0 &= 2(F_1 - F_0) - (F_2 - F_0) / 2, \\ D_n &= 2(F_n - F_{n-1}) - (F_n - F_{n-2}) / 2. \end{aligned} \quad (2.6)$$

This choice will control the direction of the curve properly at the end segments. The tangents at the interior knots, for $i = 1, 2, \dots, n-1$, are given by:

$$D_i = a_i (F_i - F_{i-1}) + (1 - a_i)(F_{i+1} - F_i) \quad (2.7)$$

Where

$$a_i = \frac{|F_{i+1} - F_i|}{|F_{i+1} - F_i| + |F_i - F_{i-1}|}, \quad i = 0, \dots, n.$$

For closed curves, the end conditions are defined as:

$$F_{-1} = F_{n-1}, \quad F_{n+1} = F_1,$$

And the tangents at the interior knots are same as in (2.7) but $i = 0, 1, \dots, n$. The experiments have shown that the use of the distance-based approximated derivatives, corresponding to any control polygon (open or closed), provides visually pleasing output. Figure 1(a) is the display of this derivative scheme for an "S" shaped data. For further details, the reader is referred to Sarfraz et al.^[10]

3. DESIGN CURVE ANALYSIS

The parameters g_i are mainly meant to be used freely to control the shape of the curve. At the same time, for the convenient of the designer, it is also required that the ideal geometric properties of the curve are not lost. The geometric properties, like variation diminishing, convex hull, and positivity, are the ones which need to be presented in the description of the design curve.

- For the constraints, $g_i^3 - 1, \quad "i$, it is very obvious that the rational cubic is characterized as of Bernstein Bezier form.
- Thus following the Bernstein Bezier theory, the piece of curve $P_i(t)$ lies in the convex hull of

$$\{F_i, V_i, W_i, F_{i+1}\}$$

It also follows the variation diminishing property within the convex hull. That is any straight line crossing the control polygon of $\{F_i, V_i, W_i, F_{i+1}\}$ does not cross the curve more than its control polygon.

For the practical implementation, we choose $g_i^3 - 1$. The *interval tension* properties are apparent for the rational Hermite form and are explained in the following subsections.

3.1. Interval Tension

The interval shape property is obvious from the following limit behaviour. That is, the increase in the shape parameter in any interval tightens the curve towards the line segment joined by the control points.

$$\lim_{g_i \rightarrow 0} P(t) = (1 - q)F_i + qF_{i+1}.$$

3.2. Global Tension

Applying the interval property above successively, the design curve converges to the control polygon as the derivatives, either being distance-based or computed from the system of equations, are bounded.

4. CONIC AND LINEAR SEGMENTS

Conic and straight line are the most important parts in designing which can be achieved through rational cubic interpolant, so that we can use the same interpolant for all types of curves. The procedure is as follows:

Let U_i be taken as the point of intersection of tangents at F_i and F_{i+1} (in case the tangents are parallel, U_i can be taken as the point where the arc is desired to be splitted, for example, it may be the inflection or the middle point, etc.) Then, we have:

$$\begin{aligned} V_i &= \frac{F_i + g_i U_i}{1 + g_i}, \\ W_i &= \frac{F_{i+1} + g_i U_i}{1 + g_i} \end{aligned} \quad (4.1)$$

It can be noted that the rational cubic Eq. (2.1) is the generalization of the following rational quadratic:

Fig. 2 Curvature plots of Spline curves with exact derivatives: (a) with distance based end derivatives, (b) with conic compatible end derivatives.

For a higher continuity than C^1 , more complicated constraints are required to be fit. For example, for rational cubic spline, the constraints lead to a tridiagonal linear system of Eq.s. This system is diagonally dominant and hence provides a unique solution. This system can be solved using some tridiagonal linear system solver like *LU* decomposition method. Their details are as follows:

C^1 constraints

$$P^{(1)}(t_i^+) = P^{(1)}(t_i^-), \quad i = 1, \dots, n$$

give

$$D_i = g_{i-1}(F_i - F_{i-1}) - D_{i-1},$$

and C^2 constraints

$$P^{(2)}(t_i^+) = P^{(2)}(t_i^-), \quad i = 1, \dots, n - 1$$

lead to the following system of equations:

$$\begin{aligned} h_i D_i + (h_i (g_{i-1} - 1) + h_{i-1} (g_i - 1)) + h_{i-1} D_{i+1} \\ = g_{i-1} D_i + g_i D_{i-1}, \quad i = 1, \dots, n - 1 \end{aligned} \quad (2.8)$$

For the need of graphical results, exact derivatives may be computed from (2.8) together with the end conditions in (2.6). Figure 1(b) is the demonstration for this derivative scheme. The end conditions used here may not be appropriate for the objectives of this paper. Therefore, a reasonable choice has been made in Section 5, which demonstrates the "S" shaped data in Fig. 1(c). The difference can be seen in Fig. 2 demonstrating curvature plots of Figures 1(b) and 1(c) in Figures 2(a) and 2(b) respectively.

$$Q(\alpha) \circ Q_i(\alpha) = \frac{(1-q)^2 F_i + q(1-q)g_i Y_i + q^2 F_{i+1}}{(1-q)^2 + q(1-q)g_i + q^2}, \quad i = 0, 1, \dots, n-1, \quad (4.2)$$

By raising the degree of (4.2), by multiplying the numerator and denominator by $(1-q)+q$, one can obtain the following rational cubic form

$$P(\alpha) \circ P_i(\alpha) = \frac{(1-q)^3 F_i + q(1-q)^2(g_i+1)V_i + q^2(1-q)(g_i+1)W_i + q^3 F_{i+1}}{(1-q)^3 + q(1-q)^2(g_i+1) + q^2(1-q)(g_i+1) + q^3}, \quad i = 0, 1, \dots, n-1, \quad (4.3)$$

This form is similar to (2.1) as

$$(1-q)^3 + q(1-q)^2(g_i+1) + q^2(1-q)(g_i+1) + q^3 = (1-q)^2 + q(1-q)g_i + q^2$$

In a similar way, by degree raise, one can also observe that the rational quadratic (4.2) is the generalization of the following linear interpolant:

$$L(\alpha) \circ L_i(\alpha) = (1-q)F_i + qF_{i+1}, \quad i = 0, 1, \dots, n-1 \quad (4.4)$$

This linear interpolant will be used for the drawing of straight line segments later on.) Hence, for conic section properties and choice of shape parameters, various conics are recovered depending upon the nature of weights (see [12]). That is, the i^{th} arc will be:

- parabolic if $g_i > 2$
- hyperbolic if $g_i > 2$
- elliptic if $g_i > 2$
- straight line: $g_i = 0$ (can be considered as a second method for straight line segment)
- circular if

$$\gamma_i = 2 \cos \phi \quad (4.5)$$

where ϕ is the angle between $F_{i+1} - F_i$ and $U_i - F_i$ and

$$U_i = F_i + m_i T_i \quad (4.6)$$

where T_i is the unit vector along D_i and

$$m_i = \frac{(F_{i+1} - F_i)^2}{2(F_{i+1} - F_i)T_i} \quad (4.7)$$

is determined by the condition

$$|U_i - F_i| = |U_i - F_{i+1}|$$

Now we are imposing continuity conditions so that both segments can share same tangent direction at knot to ensure G^1 continuity. For this we are changing the direction of exact derivatives with same magnitude. Thus, to preserve continuity at F_i , we take

$$D_i = \frac{|D_i|}{|U_i - F_i|} (U_i - F_i) \quad (4.8)$$

$$W_{i-1} = F_i - \frac{1}{1+g_{i-1}} D_i$$

Fig. 3 Spline curves: (a) Circular spline, (b) Elliptic spline, (c) Parabolic spline, (d) Hyperbolic spline.

To achieve a straight line, one needs to replace U_i with F_{i+1} . Similarly, for continuity at F_{i+1} , we take

$$D_{i+1} = \frac{|D_{i+1}|}{|F_{i+1} - U_i|} (F_{i+1} - U_i), \quad (4.9)$$

$$V_{i+1} = F_{i+1} + \frac{1}{1 + g_{i+1}} D_{i+1}$$

and replacement of U_i with F_i leads to a straight line.

5. CIRCULAR ARCS

This section is devoted for the construction of circular arc. The cases, for a given radius and given center, are independently discussed in Subsections 5.1 and 5.2 respectively.

5.1. Circular Arc For Given Radius

Let r be the given radius of the circular arc such that

$$r > \frac{|F_{i+1} - F_i|}{2} \quad (4.10)$$

Fig. 4 Curvature plots of Fig. 3: (a) Circular spline, (b) Elliptic spline, (c) Parabolic spline, (d) Hyperbolic spline.

Then, the center M can lie anywhere on the circle centered at

$$N = \frac{(F_i + F_{i+1})}{2}, \quad (4.11)$$

and having radius b as follows:

$$b = \sqrt{r^2 - \frac{|F_{i+1} - F_i|^2}{4}}. \quad (4.12)$$

It will be preferred that it should lie on the plane passing through F_i, F_{i+1} and U_i' , where U_i' is the intersection of F_i, F_{i+1} and D_{i+1} . Therefore circular arc should lie on the side of U_i' .

Let e_1 be the rotation of F_{i+1} around N by an angle θ on the plane passing through F_i, F_{i+1} and U_i' . Where $q = 90^\circ$ for anti-clockwise rotation and $q = -90^\circ$ for clockwise rotation of circular arc. Now

$$e = \frac{e_1 - N}{|e_1 - N|}$$

is a unit vector passing through N and perpendicular to $F_{i+1} - F_i$. Then $M = N + be$ will be the center of the circular arc. Let ϕ be the angle between $N - M$ and $F_i - M$, then

$$g_i = 2 \cos f$$

Take $f = -\phi$ for anti clockwise rotation of circular arc. Let T_ϕ be the rotation of F_{i+1} around F_i through angle ϕ on the plane passing through F_i, F_{i+1} and U_i' . Then

$$T_i = \frac{T_\phi - F_i}{|T_\phi - F_i|}$$

is a unit tangent vector at F_i .

Now use Eq. (4.6) to find U_i , Eq. (4.1) to find control points V_i and W_i , Eq. (4.8) for continuity at F_i , Eq. (4.9) for continuity at F_{i+1} and finally use rational cubic interpolant in Eq. (2.1) for required circular arc. In this scheme, the radius r can be used as a shape control parameter.

5.2. Circular Arc For Given Center

let M be the given center of the circular arc such that

$$|M - F_i| = |F_{i+1} - M|$$

Let M_ϕ be the rotation of M by F_i through angle q on the plane passing through F_i, F_{i+1} and M . Where $q = 90^\circ$ for clockwise rotation of circular arc and $q = -90^\circ$ for anti-clockwise rotation. Then

$$T_i = \frac{M_\phi - F_i}{|M_\phi - F_i|}$$

is a unit tangent vector at F_i . Let ϕ be the angle between $F_{i+1} - F_i$ and T_i , therefore

$$g_i = 2 \cos f$$

Now use Eq. (4.6) to find U_i , Eq. (4.1) to find control points V_i and W_i Eq. (4.8) for continuity at F_i , Eq. (4.9) for continuity at F_{i+1} . Finally, we use rational cubic interpolant, in Eq. (2.1), for required circular arc.

6. END CONDITIONS

A compatible choice, which is quite appropriate for the curve scheme of this paper, is presented here. For tangent at first point, let q_0 be the angle between $F_1 - F_0$ and $F_2 - F_0$. Take $q_0 = -q_0$ for anti-clockwise rotation of points F_0, F_1 and F_2 . Let T_0 be the rotation of F_1 around F_0 by an angle q_0 on the plane passing through F_0, F_1 and F_2 . Then

$$\begin{aligned} m_0 &= \frac{(F_1 - F_0)^2}{2(F_1 - F_0)T_0}, U_0 = F_0 + m_0 T_0 \\ V_0 &= \frac{F_0 + 2U_0}{3}, D_0 = 3(V_0 - F_0) \end{aligned} \quad (6.1)$$

where m_0 is determined by the condition

$$|U_0 - F_0| = |U_0 - F_1|$$

and V_0 & D_0 are taken from (4.1) & (2.3) respectively.

Fig. 5 (a) Rational Cubic Spline with second last interval as circular arc piece, (b) the corresponding curvature plot.

Similarly for tangent at last point, let q_n be the angle between $F_{n-1} - F_n$ and $F_{n-2} - F_n$. Take $q_n = -q_n$ for anti-clockwise rotation of points F_n, F_{n-1} and F_{n-2} . Let T_n be the rotation of F_{n-1} around F_n by an angle q_0 on the plane passing through F_n, F_{n-1} and F_{n-2} . Then

$$\begin{aligned} m_{n-1} &= \frac{(F_{n-1} - F_n)^2}{2(F_{n-1} - F_n)T_n}, U_{n-1} = F_n + m_{n-1} T_n \\ W_{n-1} &= \frac{F_n + 2U_{n-1}}{3}, D_n = 3(F_n - W_{n-1}) \end{aligned} \quad (6.2)$$

where μ_{n-1} is determined by the condition

$$|U_{n-1} - F_n| = |U_{n-1} - F_{n-1}|$$

and W_{n-1} & D_n are taken from (4.1) & (2.3) respectively.

Fig. 6 (a) A rational cubic spline with 2nd interval as circular arc for radius $r = 16$, (b) the corresponding curvature plot.

Fig. 7 (a) A cubic spline with its 2nd interval as circular arc for center $M = (0, 10, 2.5)$, (b) the corresponding curvature plot.

7. ALGORITHM

In this section a high-level description of an algorithm for generating an interpolating curve, which modifies its shape interactively according to the proposals described in above sections, is given. The algorithm is presented in C-like pseudo code.

```
//----- Construction of exact derivatives
ExactDerivatives
{
    Calculate D[0] using (6.1);
    Calculate D[n] using (6.2);
    Find exact derivatives D[i], i=1..n-1, using (2.5).
}
//----- To preserve continuity at F[i] & F[i+1]
PreserveContinuity
```

```

{
    if (i > 0) // to preserve continuity at F[i]
        use Eq. (4.8);
    if (i < n) // to preserve continuity at F[i+1]
        use Eq. (4.9);
}
//----- Control points V[i], W[i]
ControlPoints
{
    if (Spline=='Line')
        use Eq. (4.4);
    else if (Spline=='Conic' || Spline=='Circular')
    {
        use Eq. (4.3);
    }
    else // for cubic spline
        use Eq. (2.2);
}
//----- Circular, Elliptic, Parabolic and Hyperbolic spline
ConicSpline
{
    Calculate D[0] using (6.1);
    T[0] = (D[0]-F[0])/AbsVec(D[0]-F[0]);
    for (i=0; i<n; i++)
    {
        Mu = SqrVec(F[i+1]-F[i]) / (2.0*DotProd(F[i+1]-F[i],T[i]));
        U = F[i] + Mu*T[i];
        D[i+1] = F[i+1] - U;
        T[i+1] = D[i+1]/AbsVec(D[i+1]);
        if (Spline== 'Circular')
            Gama[i] = 2.0*DotProd(F[i+1]-F[i],U-F[i]) /
                (AbsVec(F[i+1]-F[i])*AbsVec(U-F[i]));
            ControlPoints;
        }
    }
}
//----- Circular arc for ith segment
CircularArc
{
    Spline = 'Circular';
    T[i] = D[i]/AbsVec(D[i]);
    Mu = SqrVec(F[i+1]-F[i]) / (2.0*DotProd(F[i+1]-
F[i],T[i]));
    U = F[i] + Mu*T[i];
    Gama[i] = 2.0*DotProd(F[i+1]-F[i],U-F[i]) /
        (AbsVec(F[i+1]-F[i])*AbsVec(U-F[i]));
    ControlPoints;
    PreserveContinuity;
}
//----- Circular arc for ith segment when radius is given
CircularArcForRadius
{
    Spline = 'Circular';
    N = (F[i]+F[i+1])/2.0;
    b = sqrt( sqrt(Radius) - sqrt(AbsVec(F[i]-F[i+1])/2.0) );
    if (ClockWise)
        Angle = -Pi/2.0; // for clockwise rotation of points
    else
        Angle = Pi/2.0; // for anti clockwise rotation of points
    Ud = PointOfIntersection(F[i],D[i],F[i+1],D[i+1]);
    e = Rotation(F[i+1],N,Angle,Ud); // Rotation of F[i+1]
about N through Angle on the
// plane of F[i+1], N & Ud.
    e = (e-N)/AbsVec(e-N);
    M = N + b*e;
    Phi = acos(DotProd(N-M,F[i]-M)/(AbsVec(N-
M)*AbsVec(F[i]-M)));
    Gama[i] = 2.0*cos(Phi);
    if (!ClockWise)
        Phi = -Phi; // for anti clockwise rotation of points
        T = Rotation(F[i+1],F[i],Phi,M); // Rotation of F[i+1]
about F[i] through Phi on the
// plane of F[i+1], F[i] & M.
        T = (T-F[i])/AbsVec(T-F[i]);
        Mu = SqrVec(F[i+1]-F[i]) / (2.0*DotProd(F[i+1]-F[i],T));
        U = F[i] + Mu*T;
        ControlPoints;
        PreserveContinuity;
    }
}
//----- Circular arc for ith segment when center is given
CircularArcForCenter
{
    Spline = 'Circular';
    if (ClockWise)
        Angle = Pi/2.0; // for clockwise rotation of points
    else
        Angle = -Pi/2.0; // for anti clockwise rotation of
points
        T = Rotation(M,F[i],Angle,F[i+1]); // Rotation of M about
F[i] through Angle on the
// plane of M, F[i] and F[i+1].
        T = (T-F[i])/AbsVec(T-F[i]);
        Phi = acos(DotProd(F[i+1]-F[i],T)/AbsVec(F[i+1]-F[i]));
        Gama[i] = 2.0*cos(Phi);
        Mu = SqrVec(F[i+1]-F[i]) / (2.0*DotProd(F[i+1]-F[i],T));
        U = F[i] + Mu*T;
        ControlPoints;
        PreserveContinuity;
    }
}
//----- Main body of the algorithm
main()
{
    read input data F[i] and shape control parameters
    Gamma[i],
        i = 0..n;
    ExactDerivatives; // Exact derivatives D[i], i = 0..n
    Spline = 'Cubic';
    for(i=0; i<n; i++)
        ControlPoints; // Control points V[i], W[i] for
cubic spline

    //----- Conic Spline (Circular, elliptic, parabolic,
hyperbolic)
    read nature of Spline (Circular, Conic);
    if (Spline != 'Circular')
        read Gama[i] for i = 0..n-1;
    ConicSpline; // Control points V[i], W[i] for
conic spline, i = 0..n-1.
    //----- Circular arc for ith segment
    read position i;
    CircularArc; // Control points V[i], W[i] for circular
arc.
    //----- Circular arc for ith segment when radius is given
    read position i and Radius;
    if (Radius <= AbsVec(F[i]-F[i+1])/2.0)
        cout << "Invalid Radius" << "\n";
    CircularArcForRadius; // Control points V[i],
W[i] for circular arc.
    //----- Circular arc for ith segment when center is given
    read position i and Center M;
    if (Abs(AbsVec(F[i]-M)-AbsVec(F[i+1]-M)) > 0.01)
        error("Give correct center...");
    CircularArcForCenter; // Control points V[i], W[i] for
circular arc.
    //----- Straight line segment
    Spline = 'Line';
    ControlPoints; // Control points V[i], W[i]
    //----- Rational cubic spline
    for (i=0; i<n; i++)
        for(j=0; j<=20; j++)

```

```

{
  t = j/20;
  calculate P(t) using (2.1);
  plot graph;
}
} // end of main algorithm.

```

8. DEMONSTRATION

The demonstration in this section has used derivative end conditions of Section 5. Figure 3 contains a block of conic splines where Fig. 3(a) is Circular spline, Fig. 3(b) is Elliptic spline, Fig. 3(c) is Parabolic spline, and Fig. 3(d) is Hyperbolic spline. Fig. 4 represents the corresponding curvature plots of the curves in Fig. 3. The demonstration of a rational cubic spline when it contains conic segments is made in Fig. 5. Figure 5(a) is a rational cubic spline with second last curve as circular spline, the corresponding curvature plot in Fig. 5(b) verifies the result. The demonstration of the spline, where some circular arcs with specific radius are to be plotted, is given in Fig. 6. The Fig. 6(a) is a rational cubic spline with 2nd interval as circular arc for radius $r = 16$ where as the Fig. 6(b) is its corresponding curvature plot. Figure 7 demonstrates the circular arcs at some specific center. The Fig. 7(a) is a cubic spline with its 2nd interval as circular arc for center $M = (0,10,2.5)$ where as the Fig. 7(b) is its corresponding curvature plot.

9. CONCLUSION AND FUTURE WORK

We have described an interval controlled rational cubic interpolation scheme. The scheme offers a number of possible ways in which the shape of the corresponding curves may be altered by the users. It is therefore felt that such a scheme could be a useful addition to an interactive design package, with the user having enough control over the curve segments. The provision of the shape parameters, in the description of the piecewise rational functions, provides freedom to modify the shape in desirous regions in a stable manner.

The rational spline scheme is meant for parametric curves and is capable of designing plane as well as space curves. It is an interpolatory rational spline scheme enjoying all the ideal geometric properties. It has features to produce all types of conic curves in such a way that the whole design curve may be produced as a circular, elliptic, parabolic, or a hyperbolic spline curve. In addition, the desired conic pieces may also be fitted within the rational cubic spline.

Overall smoothness of the rational cubic spline is C^2 whereas the conics are stitched with G^1 continuity. The curve scheme is extendable to surfaces and authors are looking to publish it in a subsequent paper.

Some extra shape control parameters are also expected to be included to have some more degrees of freedom for designers. These shape parameters may be like point, interval, or biased shape control parameters used by various authors including Nielson^[8], Foley^[3], Barsky^[1], Sarfraz^[10-14], Gregory et al^[4-5].

ACKNOWLEDGEMENT

The authors are thankful to the anonymous referees for their valuable comments and suggestions in the improvement of the manuscript.

REFERENCES

- [1] B. A. Barsky, Computer Graphics and Geometric Modeling using Beta-Splines, Springer-verlag, 1986.
- [2] G. Farin, Curves and Surfaces for Computer Aided Geometric Design, Academic Press, 1988.
- [3] T. A. Foley, "Surface Interpolation with Tension Controls using Cardinal Bases", Computer Aided Geometric Design, Vol. 6, pp. 97-109, 1989.
- [4] J. A. Gregory, M. Sarfraz, and P. K. Yuen, "Interactive Curve Design using C^2 Rational Splines", Computers and Graphics, Vol. 18(2), 153-159, 1994.
- [5] J. A. Gregory and M. Sarfraz, "A Rational Spline with Tension", Computer Aided Geometric Design, Vol. 7, 1-13, 1990.
- [6] J. Hoschek, "Circular Splines", Computer-Aided Design, Vol. 24, pp. 611-618, 1992.
- [7] M. A. Jamaludin, H. B. Said, and A. A. Majid, "Shape Control of Parametric cubic curves", Proceedings of CAD/Graphics' 95, China, SPIE Proceedings Series Vol. 2644, pp. 128-133, 1995.
- [8] G. M. Nielson, "Rectangular v-splines", IEEE Computer Graphics and Applics. 6, 35-40, 1986.
- [9] L. Piegl, and W. Tiller, "The NURBS book", Springer, 1995.
- [10] M. Sarfraz, Z. Habib and M. Hussain, "Piecewise Interpolation for Designing of Parametric Curves", Proceedings of IEEE Conference on Information Visualization, IV' 98, London, pp. 307-313, 1998.
- [11] M. Sarfraz, M. Hussain, and Z. Habib, "Local Convexity Preserving Rational Cubic Spline Curves", Proceedings IEEE Conference on Information Visualization, IV' 97, London, pp. 211-218, 1997.
- [12] M. Sarfraz, "Curves and Surfaces for CAD using C^2 Rational Cubic Splines", Engineering with Computers, Vol. 11(2), pp. 94-102, 1995.
- [13] M. Sarfraz, "Designing of Curves and Surfaces using Rational Cubics", Computers and Graphics, Vol. 17(5), 529-538, 1993.
- [14] M. Sarfraz, "A Rational Cubic Spline with Biased, Point and Interval Tension", Computers and Graphics, Vol. 16, pp. 427-430, 1992.

BIOGRAPHY

Muhammad Sarfraz received a Ph.D. degree from Brunei University in 1990. He is currently an associate professor and chairman of research committee in the department of information & computer Science at King Fahd University of Petroleum and Minerals. He has published around 90 journal and conference papers, edited seven proceedings of International conferences, given various keynotes and invited talks. His areas of interest are computer Graphics, Computer Aided Geometric Design, Font Outline Capture and representation, Document & Image Processing, Digitization of Geometric Models, Scientific Visualization, Geometric Modeling, Surface

Reconstruction, Geometric Algorithms, Error Analysis, Spline Theory, Computational Geometry, Spline Wavelets, and Genetic Algorithms. He is member of IEEE, ACM, AMS IVS, PMS, ISOSS and is organizer of various international conferences. He has achieved various medals and awards in education and research.

Zulfiqar Habib received his master degree from Punjab University Lahore, Pakistan in 1990. He is currently a Ph.D. student in the Department of Mathematics and Computer Science at Kagoshima University, Japan. His areas of interest are Computer Graphics, Computer Geometric Design, Surface Reconstruction, Geometric Algorithms, Error Analysis, Spline Theory, and Computational Geometry.