Interproximation II: Interpolation and Approximation Using Cubic B-Spline Curves*

Fuhua Cheng †

Dept. of Computer Science, Univ. of Kentucky, Lexington, KY 40506

Brian A. Barsky

Computer Science Division-EECS Dept., Univ. of California, Berkeley, CA 94720

ABSTRACT

An algorithm for the construction of a non-uniform cubic B-spline interproximating curve is presented. The curve has minimum energy on each of its components. It interpolates the exact data points and approximates the uncertain ones by passing through the regions that specify the range of the uncertain data points. The new algorithm improves our previous work in several aspects, including parametrization technique, end conditions, numerical stability and shape editing capability.

Keywords and phrases. B-splines, interpolation, approximation, uncertain data, interproximation

1. Introduction

A typical interpolation-based curve/surface design system has a rigid template for its input, i.e., all the input data must be 2D/3D points. For applications such as reconstruction of natural phenomena or digitized images, motion detection, and 2D/3D shape design, this is simply not flexible enough since in these applications the data sometimes are uncertain (i.e., one only has a range of a point but not the exact location). Data fitting methods such as least-squares approximation may be used to approximate uncertain data points (or, interpolate perturbed data points, by backward error analysis). However, these methods do not guarantee that the resulting curve or surface would pass through specific data points, nor would the curve or surface pass through specific regions. What one needs in these applications is a method that can interpolate exact data points and approximate the uncertain ones by passing through the regions that specify the range of the uncertain data points. We call such a process

---

* A preliminary version was presented at the IFIP TC 5/WG 5.10 Conference "Modeling in Computer Graphics: Methods and Applications", Genoa, Italy, June 28 - July 1, 1993.

† Currently visiting the Shape Modeling Lab, Univ. of Aizu, Japan.

"interproximation" in our previous work [3] due to the fact that it is a cross between interpolation and approximation. Basically, interproximation is a process to choose a curve/surface, among the many curves/surfaces that satisfy the requirement, that meets some constraint so the resulting curve/surface is relatively smooth.

A solution to a more general problem in this direction, the generalized Hermite-Birkhoff interpolation, has been given by Ritter [16]. His solution allows both the function and its derivatives to interproximate the given data. For most of the applications in computer-aided geometric design and related areas, however, fitting derivatives to given data usually is not required.

A more specific algorithm for the construction of an interproximating cubic spline curve was presented in our previous work [3]. The curve is generated based on minimizing the *energy* on both the x- and y-components of the possible interproximating curves. Geometric smoothness of the curve is achieved through this energy-minimizing process. This technique allows a user to design a curve with more flexibility and less efforts (trial-and-error iterations). It can also be used to remove undesired oscillations generated on an ordinary interpolating curve.

This work, however, has several disadvantages. First, the geometric meaning is not clear by using the classical concept of *reproducing kernel* in the construction of function spaces. Second, the user has to specify the tangent at the start point of the curve and, due to the way the function space is constructed, it is not even possible to consider any other alternatives. Further, since uniform spaced knots are used in our previous approach, one needs to choose data points in a way so that oscillations of the resulting curve would to accommodated, a burden that only experienced user can handle properly.

In this paper, we improve these problems by presenting an algorithm for the construction of a non-uniform cubic B-spline curve that interproximates a given data set. The new technique improves our previous work in several aspects: (1) better parameter spacing technique, (2) more efficient and numerically stable computation process, (3) requiring no input on boundary conditions, and (4) requiring less efforts in selecting the input data and allowing more flexibility in shape modification. The new technique is also easier to be integrated with other modeling systems. A paper with some similarity but using probabilistic point constraint has also been presented recently [15].

Details of the new work will be given in the subsequent sections. We will start our work with a formal definition of the problem in Section 2. The solution and the algorithm will be presented in Section 3. Implementation issues and concluding remarks are given in Section 4. Possible extensions of this work will be discussed in Section 5.

2. Background

We describe the background and formulate the problem in this section. The approach is different from the previous work. The differences include using non-uniformly spaced knots in the parameter space, different end-point conditions, and different formulation of the function spaces.

Let $\{ \mathbf{D}_i \mid 1 \le i \le n+m \}$ be a set of 2D data where $\mathbf{D}_{i_j}$, $1 \le j \le n$, are 2D points and $\mathbf{D}_{i_k}$, $1 \le k \le m$, are 2D regions with $\mathbf{D}_{i_j} = \mathbf{P}_j = (x_j, y_j)$ and $\mathbf{D}_{i_k} = \mathbf{A}_k \times \mathbf{B}_k = [a_k, b_k] \times [c_k, d_k]$. The objective of this work is to construct a cubic B-spline curve that interpolates $\mathbf{D}_{i_j}$ and passes through $\mathbf{D}_{i_k}$ with the smoothest shape. This is accomplished in part by minimizing the energy of the x- and y- components of an interproximating B-spline curve. The representation of such a curve will follow the traditional approach, i.e., a piecewise curve of $n+m-1$ cubic B-spline segments with the endpoints of the segments interproximating the given data.

Let H be the set of all cubic B-spline curves defined as follows:

$$\mathbf{S}(t) = \sum_{i=-2}^{n+m-1} \mathbf{C}_{i+2} B_{i,3}(t)$$

where $B_{i,3}(t)$ are B-spline basis functions of degree 3 defined by the knot vector $\{\tau_i \mid -2 \le i \le n+m+3 \}$ and $\mathbf{C}_i$ are 2D control points. The knot vector implies a parameter space defined by a range of τ_1 to τ_{n+m} for t. We shall assume that the knots satisfy the following conditions: $\tau_{-2} = \tau_{-1} = \tau_0 = \tau_1 = 0$, $\tau_{n+m} = \tau_{n+m+1} = \tau_{n+m+2} = \tau_{n+m+3} = 1$ and for $2 \le i \le n+m-1$

$$\tau_i - \tau_{i-1} = |\mathbf{Q}_i - \mathbf{Q}_{i-1}|^{1/2} \Big/ \sum_{j=2}^{n+m} |\mathbf{Q}_j - \mathbf{Q}_{j-1}|^{1/2} \tag{2.1}$$

where $\mathbf{Q}_i$ equals $\mathbf{P}_j$ if $i = i_j$ and $\mathbf{Q}_i$ equals the center of $\mathbf{A}_k \times \mathbf{B}_k$ if $i = i_k$. Equation (2.1) is based on the *centripetal model* developed by E.T.Y. Lee [12]. An interpolating curve with such a knot parametrization usually is "fairer" (closer to the data polygon) than those obtained with the uniform or chord length parametrization, see Figure 1.

Figure 1. Cubic B-spline interpolation curves obtained with (a) uniform, (b) chord length, and (c) centripetal parametrization.

For each B-spline curve $\mathbf{S}(t)$ contained in H, we define the (bending) *energy* of $\mathbf{S}(t)$ to be

$$|\mathbf{S}| \equiv \int_0^1 \left[\left(\frac{d^2 \mathbf{S}_x(u)}{dt^2}\right)^2 + \left(\frac{d^2 \mathbf{S}_y(u)}{dt^2}\right)^2\right] du \tag{2.2}$$

where $\mathbf{S}_x$ and $\mathbf{S}_y$ are the x- and y-components of $\mathbf{S}$. Definition (2.2) follows that of Kjellander [10]. More discussions on the definition of energy can be found in E.T.Y. Lee [11]. The value, $|\mathbf{S}|$, represents the (overall) smoothness of the curve; the smaller the energy, the smoother the curve. Hence, our problem is to find $\hat{\mathbf{S}} \in H$ such that

$$\begin{aligned} \hat{\mathbf{S}}(\tau_{i_j}) &= \mathbf{P}_j, \qquad 1 \le j \le n \\ \hat{\mathbf{S}}(\tau_{i_k}) &\in \mathbf{A}_k \times \mathbf{B}_k, \qquad 1 \le k \le m \end{aligned} \tag{2.3}$$

and

$$|\hat{\mathbf{S}}| = \min \left\{ |\mathbf{S}| \;\middle|\; \mathbf{S} \in H, \mathbf{S} \textit{ satisfies } (2.3) \right\}.$$

Since our work will be performed on the basis of individual components and the technique involved for each component is the same, it is sufficient to consider this problem for the first component only, i.e., cubic spline functions. Let F be the set of all cubic B-spline functions defined on [0,1] with respect to the knot vector $\{ \tau_i \mid -2 \le i \le n+m+3 \}$

$$f(t) = \sum_{i=-2}^{n+m-1} e_{i+2} B_{i,3}(t)$$

where e_i are real number coefficients. For each $f \in F$, we define the energy of f to be

$$|f| \equiv \int_0^1 [f^{(2)}(t)]^2 dt$$

where $f^{(2)}$ is the second derivative of f with respect to t. Then our problem is to find $\hat{f} \in F$ such that

$$\begin{aligned} \hat{f}(\tau_{i_j}) &= x_j, \qquad 1 \le j \le n \\ \hat{f}(\tau_{i_k}) &\in A_k, \qquad 1 \le k \le m \end{aligned} \tag{2.4}$$

and

$$|\hat{f}| = \min \left\{ |f| \;\middle|\; f \in F, f \textit{ satisfies } (2.4) \right\}$$

We need two extra conditions to solve this problem since the dimension of F is $n+m+2$ and there are only $n+m$ fitting conditions. We adopt the "natural" conditions, i.e., the second derivatives of f at the endpoints of the parameter range, τ_1 and τ_{n+m}, are set to zero.

$$f^{(2)}(\tau_1) = 0, \qquad f^{(2)}(\tau_{n+m}) = 0 \tag{2.5}$$

Through simple algebra, it is easy to check that

$$f^{(2)}(\tau_1) = e_0 B^{(2)}_{-2,3}(\tau_1) + e_1 B^{(2)}_{-1,3}(\tau_1) + e_2 B^{(2)}_{0,3}(\tau_1)$$
$$= \frac{6}{(\tau_2-\tau_1)^2}(e_0 - \frac{\tau_3+\tau_2-2\tau_1}{\tau_3-\tau_1} e_1 + \frac{\tau_2-\tau_1}{\tau_3-\tau_1} e_2)$$

Hence, the first condition in (2.5) implies that

$$e_0 = (\frac{\tau_3 + \tau_2 - 2\tau_1}{\tau_3 - \tau_1})\, e_1 - (\frac{\tau_2 - \tau_1}{\tau_3 - \tau_1})\, e_2$$

Similarly, the second condition in (2.5) implies that

$$e_{n+m+1} = (\frac{\tau_{n+m-2}+\tau_{n+m-1}-2\tau_{n+m}}{\tau_{n+m-2}-\tau_{n+m}})\, e_{n+m} - (\frac{\tau_{n+m-1}-\tau_{n+m}}{\tau_{n+m-2}-\tau_{n+m}})\, e_{n+m-1}$$

Consequently, cubic B-spline functions contained in F which satisfy the natural conditions can be expressed as

$$f(t) = \sum_{i=1}^{n+m} e_i\, w_i(t) \tag{2.6}$$

where

$$w_1(t) = \frac{\tau_3+\tau_2-2\tau_1}{\tau_3-\tau_1} B_{-2,3}(t) + B_{-1,3}(t) \tag{2.7}$$

$$w_2(t) = -\frac{\tau_2-\tau_1}{\tau_3-\tau_1} B_{-2,3}(t) + B_{0,3}(t) \tag{2.8}$$

$$w_i(t) = B_{i-2,3}(u), \quad 3 \le i \le n+m-2 \tag{2.9}$$

$$w_{n+m-1}(t) = -\frac{\tau_{n+m-1}-\tau_{n+m}}{\tau_{n+m-2}-\tau_{n+m}} B_{n+m-1,3}(t) + B_{n+m-3,3}(t) \tag{2.10}$$

$$w_{n+m}(t) = \frac{\tau_{n+m-2}+\tau_{n+m-1}-2\tau_{n+m}}{\tau_{n+m-2}-\tau_{n+m}} B_{n+m-1,3}(t) + B_{n+m-2,3}(t) \tag{2.11}$$

By defining $\hat{F}$ as the set of all cubic B-spline functions defined by (2.6) then the problem that we are in a position to solve is to find $\hat{f} \in \hat{F}$ such that (2.4) is satisfied and

$$|\hat{f}| = \min\left\{ |f| \;\middle|\; f \in \hat{F}, f \;\; satisfies \;\; (2.4) \right\}$$

The indexing of the fitting conditions (2.4) would lead to sparse matrices which are not numerically stable and efficient when performing Gaussian elimination in the computation process. Fortunately, this problem can be resolved by changing indices of the knots and the associated B-spline basis functions. We will rename τ to be $\{u_1, u_2, \cdots, u_n, v_1, v_2, \cdots, v_m\}$ where $u_j = \tau_{i_j}$, $1 \le j \le n$, and $v_k = \tau_{i_k}$, $1 \le k \le m$. We also rename $\{w_i(t)\}$ defined in (2.7) - (2.11) to be $\{g_1(t), \cdots, g_n(t), \; h_1(t), \cdots, h_m(t)\}$ with $g_j(t) = w_{i_j}(t)$, $1 \le j \le n$, and $h_k(t) = w_{i_k}(t)$, $1 \le k \le m$. If we rewrite (2.6) as

$$f(t) = \sum_{j=1}^{n} \alpha_j \; g_j(t) + \sum_{k=1}^{m} \beta_k \; h_k(t) \tag{2.12}$$

and let $\hat{F}$ be the set of cubic B-spline functions defined by (2.12) then our work is to find α_j and β_k for $\hat{f} \in \hat{F}$ such that

$$\hat{f}(u_j) = x_j \,, \qquad 1 \le j \le n \tag{2.13}$$

$$\hat{f}(v_k) \in A_k \,, \qquad 1 \le k \le m \tag{2.14}$$

and

$$|\hat{f}| = \min\left\{ |f| \;\middle|\; f \in \hat{F}, f \;\; satisfies \;\; (2.13) \;\; and \;\; (2.14) \right\}. \tag{2.15}$$

In our previous approach, one needs to construct $g_j(t)$ and $h_k(t)$ as the *reproducing kernel* of $\tilde{F}$ (Lemma 3.1 [3]) and then express $\hat{f}$ as a linear combination of g_j and h_k. using the above approach, this step is not necessary since the optimal solution, as a B-spline function, is automatically a linear combination of the B-spline basis functions at the knots.

3. Solution and Algorithm

The construction of $\hat{f}$ which satisfies (2.13), (2.14) and (2.15) is based on a two-stage process: first compute the coefficients β_k, then compute the coefficients α_j. The process of computing β_k is a *quadratic programming* problem. We first convert the fitting conditions in (2.13) and (2.14) into matrix form.

Define

$$N_1 \equiv \left[a_{i,j} \right]_{n \times n} , \quad a_{i,j} = g_j(u_i) \quad 1 \le i, j \le n \tag{3.1}$$

$$M_1 \equiv \left[b_{i,j} \right]_{n \times m} , \quad b_{i,j} = h_j(u_i) \quad 1 \le i \le n, \; 1 \le j \le m \tag{3.2}$$

$$M_2 \equiv \left[c_{i,j} \right]_{m \times n}, \quad c_{i,j} = g_j(v_i) \quad 1 \le i \le m,\ 1 \le j \le n \tag{3.3}$$

$$N_2 \equiv \left[d_{i,j} \right]_{m \times m}, \quad d_{i,j} = h_j(v_i) \quad 1 \le i, j \le m \tag{3.4}$$

The fitting conditions in (2.13) imply that

$$N_1 \boldsymbol{\alpha} + M_1 \boldsymbol{\beta} = \mathbf{x}$$

or

$$\boldsymbol{\alpha} = N_1^{-1} \mathbf{x} - N_1^{-1} M_1 \boldsymbol{\beta} \tag{3.5}$$

where $\boldsymbol{\alpha} \equiv (\alpha_1, \alpha_2, .., \alpha_n)^T$, $\boldsymbol{\beta} \equiv (\beta_1, \beta_2, .., \beta_m)^T$ and $\mathbf{x} \equiv (x_1, x_2, .., x_n)^T$. Similarly, the fitting conditions in (2.14) imply that

$$M_2 \boldsymbol{\alpha} + N_2 \boldsymbol{\beta} \in \mathbf{A} \tag{3.6}$$

where $A \equiv \prod_{k=1}^{m} A_k$. By substituting (3.5) into (3.6) we have

$$(N_2 - M_2 N_1^{-1} M_1) \boldsymbol{\beta} \in \mathbf{A} - M_2 N_1^{-1} \mathbf{x} \tag{3.7}$$

Therefore, the essential work is to find $\boldsymbol{\beta}$ such that $|\,\hat{f}\,|$ is minimum subject to constraint (3.7).

Define

$$G = \left[g_{i,j} \right]_{n \times n}, \quad g_{i,j} = \int_0^1 g_i^{(2)}(u)\, g_j^{(2)}(u) du, \quad 1 \le i, j \le n \tag{3.8}$$

$$H = \left[h_{i,j} \right]_{m \times m}, \quad h_{i,j} = \int_0^1 h_i^{(2)}(u)\, h_j^{(2)}(u) du, \quad 1 \le i, j \le m \tag{3.9}$$

$$Q = \left[q_{i,j} \right]_{n \times m}, \quad q_{i,j} = \int_0^1 g_i^{(2)}(u)\, h_j^{(2)}(u) du, \quad 1 \le i \le n,\ 1 \le j \le m \tag{3.10}$$

G and H are symmetric matrices. From (2.12) we have

$$\begin{aligned} |\,\hat{f}\,| &= \int_0^1 \Big(\sum_{i=1}^{n} \alpha_i g_i^{(2)}(u) + \sum_{i=1}^{m} \beta_i h_i^{(2)}(u) \Big)^2 du \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j g_{i,j} + \sum_{i=1}^{m} \sum_{j=1}^{m} \beta_i \beta_j h_{i,j} + 2 \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j q_{i,j} \end{aligned}$$

$$= \boldsymbol{\alpha}^T G \boldsymbol{\alpha} + \boldsymbol{\beta}^T H \boldsymbol{\beta} + 2\boldsymbol{\alpha}^T Q \boldsymbol{\beta}$$

Therefore, from (3.5), we have

$$|\hat{f}| = (N_1^{-1}\mathbf{x} - D\boldsymbol{\beta})^T G (N_1^{-1}\mathbf{x} - D\boldsymbol{\beta}) + \boldsymbol{\beta}^T H \boldsymbol{\beta} + 2(N_1^{-1}\mathbf{x} - D\boldsymbol{\beta})^T Q \boldsymbol{\beta}$$

where $D = N_1^{-1} M_1$. Through simple algebra it is easy to check that

$$|\hat{f}| = C + \boldsymbol{\beta}^T (H - 2Q^T D + D^T G D)\boldsymbol{\beta} + 2(N_1^{-1}\mathbf{x})^T (Q - GD)\boldsymbol{\beta}$$

where $C = (N_1^{-1}\mathbf{x})^T G (N_1^{-1}\mathbf{x})$ is a constant. The fact that G being symmetric has been used in the above derivation. Therefore, to minimize $|\hat{f}|$ it is sufficient to minimize

$$\Gamma(\boldsymbol{\beta}) = \boldsymbol{\beta}^T W \boldsymbol{\beta} + 2(N_1^{-1}\mathbf{x})^T Z \boldsymbol{\beta} \tag{3.11}$$

where

$$W = H - 2Q^T D + D^T G D \tag{3.12}$$

$$Z = Q - GD \tag{3.13}$$

with constraint (3.7). This is a well-known quadratic programming problem in nonlinear programming [1]. Standard optimization routine such as NAG [13] can be used to solve this problem.

The matrix $D = N_1^{-1} M_1$ can be computed as follows. Let

$$N \equiv \begin{bmatrix} N_1 & M_1 \\ M_2 & N_2 \end{bmatrix}_{(n+m)\times(n+m)} .$$

Perform block Gaussian elimination on

$$N^T = \begin{bmatrix} N_1^T & M_2^T \\ M_1^T & N_2^T \end{bmatrix}$$

to get

$$\begin{bmatrix} N_1^T & M_2^T \\ 0 & U^T \end{bmatrix} .$$

This corresponds to

$$\begin{bmatrix} I & 0 \\ L & I \end{bmatrix} N^T = \begin{bmatrix} N_1^T & M_2^T \\ 0 & U^T \end{bmatrix} \tag{3.14}$$

with L containing the multipliers. Since $LN_1^T + M_1^T = 0$, it follows that

$$N_1^{-1} M_1 = -L^T .$$

The block Gaussian elimination process also gives us the value of $N_2 - M_2 N_1^{-1} M_1$ in (3.7) as it is easy to see now that its value is equal to the submatrix U contained in (3.14).

Consequently, the coefficients α_j and β_k of the optimal solution $\hat{f}$ which satisfies (2.13), (2.14) and (2.15) can be computed as follows.

1. Compute N_1, M_1, M_2, N_2, G, H and Q defined by (3.1), (3.2), (3.3), (3.4), (3.8), (3.9) and (3.10), respectively.
2. Solve $N_1 \boldsymbol{\gamma} = \mathbf{x}$ to get $\boldsymbol{\gamma} = N_1^{-1} \mathbf{x}$.
3. Perform block Gaussian elimination on

$$N^T = \begin{bmatrix} N_1^T & M_2^T \\ M_1^T & N_2^T \end{bmatrix}$$

to determine $D = N_1^{-1} M_1$ and $U = N_2 - M_2 N_1^{-1} M_1$.

4. Compute W and Z defined by (3.12) and (3.13), respectively.
5. Minimize

$$\Gamma(\boldsymbol{\beta}) = \boldsymbol{\beta}^T W \boldsymbol{\beta} + 2\boldsymbol{\gamma}^T Z \boldsymbol{\beta}$$

subject to

$$U \boldsymbol{\beta} \in \mathbf{A} - M_2 \boldsymbol{\gamma}$$

6. Compute $\boldsymbol{\alpha}$ defined by (3.5).

The construction of the matrices N_1, M_1, N_2 and M_2 may be performed by using the corresponding expressions of w_i in (2.7) - (2.11) to find the values of g_j or h_j at the given knots. These matrices are all banded matrices with a band width of at most 3. The construction of the matrices G, H and Q may be performed by first applying the technique of *integration by parts* to the expressions in (3.8) - (3.10) to find the following expressions:

$$g_{i,j} = g_i^{(2)}(1) g_j^{(1)}(1) - g_i^{(3)}(1) g_j(1) - g_i^{(2)}(0) g_j^{(1)}(0) - g_i^{(3)}(0) g_j(0)$$
$$h_{i,j} = h_i^{(2)}(1) h_j^{(1)}(1) - h_i^{(3)}(1) h_j(1) - h_i^{(2)}(0) h_j^{(1)}(0) - h_i^{(3)}(0) h_j(0)$$
$$q_{i,j} = g_i^{(2)}(1) h_j^{(1)}(1) - g_i^{(3)}(1) h_j(1) - g_i^{(2)}(0) h_j^{(1)}(0) - g_i^{(3)}(0) h_j(0)$$

and then computing their values at the given knots. G and H are also banded matrices. Since the values of g_i, h_j and their first, second and third derivatives at the knots will be used several times in the construction of these matrices, one should build a look-up table for these values and reference appropriate entries of the table to construct these matrices.

The block Gaussian elimination process can be performed in a very efficient and stable manner since N_1 and N_2 are both banded square matrices and M_1 and M_2 are sparse matrices with the non-zero entries concentrated on a banded neighborhood of the diagonal line; the number of non-zero entries within each row of these matrices is at most 3.

The Gaussian elimination process required in Steps 2 and 3 should be performed by exploiting the fact that N_1, M_1, N_2 and M_2 are banded matrices. This fact allows the Gaussian elimination process to be carried out in linear time.

The minimization problem of the quadratic form (3.11) with constraint (3.7) can be carried out by calling the NAG FORTRAN Library Routine, E04NAF [13]. This routine requires 29 parameters as input. In the above case, however, only 8 of them are variables; the remaining ones are constants.

4. Conclusions

This paper presents an algorithm for the construction of a non-uniform interproximating cubic B-spline curve with minimum energy on each of its components. The curve interpolates the exact data points and approximates the uncertain ones by passing through the rectangular regions that specify the range of the uncertain data points. This approach allows a user to construct a desired curve with more flexibility and fewer trial-and-error iterations than conventional approach. Two examples of non-uniform cubic interproximating B-spline curves constructed with the algorithm presented here are shown in Figures 2, and 3.

The new algorithm improves our previous work [3] in several aspects, including (1) The parametric knots are parametrized using the centripetal model [12] instead of uniform parametrization. This approach gives fairer interproximating curve than those obtained with the uniform or the chord length parametrization. (2) B-splines, instead of reproducing kernels, are used in the computation process. The construction of the matrices and the Gaussian elimination process, hence, can be performed more efficiently and numerically stably, since all the involved matrices are banded. (3) The new approach does not require user input of boundary conditions. (4) The new approach allows the user to make local modification by adjusting the control points of the curve and can easily be integrated with any B-spline or NURB based modeling systems.

The algorithm has been implemented on a Sequent Balance 20000 multiprocessing machine using X Window System environment.

5. Future Work

Future work in this direction includes surface interproximation and interproximation for other curve representations. The curve and surface representations to be considered include B-spline surfaces, Beta-spline curves/surfaces [2], and NURB curves/surfaces.

Although E.T.Y. Lee's *centripetal model* [12] can also be used for other curve representations, we need to study appropriate parametrization scheme for surface interproximation since the centripetal model can not be used for the surface interproximation problem directly.

We will also study different approaches in defining the "fairness" of a curve, such as using curvature and other means [7-9,14]. It looks like that the "fairness" of an interproximating surface for gridded data can be achieved through a two-stage process: first, minimize the energy of the "interproximating curves" that interproximate the given data in u direction, then minimize the energy of the "interpolating curves" that interpolate the control vertices of the interproximating curves constructed in the first stage in the v direction. This is motivated by the observation that a tensor product spline surface fitting problem can be separated into a two-stage fitting problem [4-5]. However, further study is required for "fairing" surface which interproximates scattered data. Related work in this area can be found in [6,14].

Acknowledgements The authors wish to thank Weihua Li for implementing the algorithm presented in this paper. The work of the first author was supported in part by IBM.

References

1. M. Avriel, *NONLINEAR PROGRAMMING: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.
2. B.A. Barsky, The Beta-spline: A local representation based on shape parameters and fundamental geometric measures, *Ph.D. Thesis*, Dept. of Computer Science, University of Utah, 1981.
3. F. Cheng and B.A. Barsky, Interproximation: Interpolation and Approximation Using Cubic Spline Curves, *Computer Aided Design* 23,10 (December 1991), 700-706.
4. F. Cheng and A. Goshtasby, A parallel B-spline surface fitting algorithm, *ACM Trans. on Graphics* 8,1 (January 1989), 41-50.
5. C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York (1978).
6. R. Franke and G.M. Nielson, Scattered data interpolation and applications: A tutorial and survey, *Geometric Modeling: Methods and Applications*, H. Hagen and D. Roller, ed., Springer-Verlag (1991), 131-160.
7. B.K.P. Horn, The curve of least energy, *ACM Trans. Math. Software,* 9,4 (December 1983), 441-460.
8. E. Jou, Minimal Energy Splines, *Ph.D. Thesis*, Computer Science Dept., Univ. of Maryland, College Park, Maryland, 1989.
9. M. Kallay, A method to approximate the space curve of minimal energy and prescribed length, *CAD* 19,2 (March 1987), 73-76.
10. J.A.P. Kjellander, Smoothing of cubic parametric splines, *Computer-Aided Design*, 15, 5 (1983), 175-179.
11. E.T.Y. Lee, Energy, Fairness and a Counterexample, *Computer-Aided Design*, 22, 1 (January/February 1990), 37-40.
12. E.T.Y. Lee, On choosing nodes in parametric curve interpolation, *Computer Aided Design*, 21, 6 (July/August, 1989), 363-370.

13. NAGFLIB, *NAG FORTRAN Library Routine Document* (November 1983).

14. G.M. Nielson, Scattered Data Modeling, *IEEE Computer Graphics & Appl.* 13,1 (1993), 60-70.

15. A. Rappoport, Y. Hel-Or and M. Herman, Interactive design of smooth objects using probabilistic point constraints, *acceted for publication in ACM Trans. on Graphics.*

16. K. Ritter, Generalized Spline Interpolation and Nonlinear Programming, *Approximation with Special Emphasis on Spline Functions*, Ed. I.J. Schoenberg, Academic Press, London (1969), 75-118.