



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

SCIENCE @ DIRECT®

Computer Aided Geometric Design 22 (2005) 327–347

COMPUTER  
AIDED  
GEOMETRIC  
DESIGN

[www.elsevier.com/locate/cagd](http://www.elsevier.com/locate/cagd)

# Simple local interpolation of surfaces using normal vectors

Takashi Nagata<sup>1</sup>

*Integrated Volume-CAD System Program, The Institute of Physical and Chemical Research (RIKEN),  
2-1 Hirosawa, Wako, Saitama 351-0198, Japan*

Received 5 March 2004; received in revised form 20 October 2004; accepted 30 January 2005

Available online 14 March 2005

---

## Abstract

A simple algorithm for surface interpolation is proposed. Its central idea is quadratic interpolation of a curved segment from the position and normal vectors at the end points, with the aid of generalized inverses. It is then used to recover the curvature of triangular or quadrilateral patches. The methodology has the following distinctive features: (i) The algorithm is efficient and completely local, requiring only the position vectors and normals given at the nodes of a patch, and hence it is suitable for parallel processing. (ii) The  $C^0$  continuity is always attained, and errors in the normals diminish rapidly with the increase in the number of nodes. (iii) Since the approach can account for discontinuity (multiplicity) of normals, sharp edges and singular points, as well as non-manifolds, can be treated quite easily. (iv) Because of the low degree of the interpolation, it is rather robust and amenable to numerical analyses in comparison with the traditional cubic and more elaborate approximations. Validity and effectiveness of the formulation are checked through several examples.

© 2005 Elsevier B.V. All rights reserved.

*Keywords:* Local interpolation; Normal vector; Generalized inverse; Quadratic polynomial patch; Sharp edge; Singular point; Non-manifold; Parallel processing

---

## 1. Introduction

Sophisticated surface interpolations through NURBS and Bézier parameterizations, etc. are powerful tools for CAD, where high level of continuity using a small number of patches is preferred. However, their

---

*E-mail address:* [tnagata@postman.riken.go.jp](mailto:tnagata@postman.riken.go.jp) (T. Nagata).

<sup>1</sup> Tel.: +81-048-462-4892, fax: +81-048-467-8705.

application to numerical simulation represented by the finite element method (FEM) is rather limited, mainly due to the following reasons.

- (A) Discretized models are already divided into a large number of small patches, for which elaborate interpolation is, in general, not necessary. In many engineering and applied physics, it is therefore more crucial to lower the complexity of interpolation.
- (B) Complex surface expressions discourage rapid evaluation of quantities necessary for physical analyses, such as distance, cross-sectional area, volume and surface integrals. Hence many numerical simulation programs accept only *linear or quadratic* surface interpolators. For instance, modeling of contact mechanics involves solution for intersection of a patch and a line. The process is analytic for the quadratic parametric representation, since it leads to a quartic equation. However, closed solutions cannot be obtained if the interpolant is cubic or higher, and hence sophisticated geometric descriptions are generally impractical. Similar situations are very common in real-world problems; they usually impose severe constraints on the mathematical models.
- (C) In contrast to the industrial design aiming at creation of fine shapes, simulation models are assumed to be already provided as CAD data, etc. Therefore, fidelity (convergence) to the original geometry is more important than visual quality, e.g., high level of smoothness. Here it should be stressed that *smoothness does not mean accuracy* of the algorithm. For example, Nielson's minimum norm network gives  $C^1$  interpolation, but its degree of algebraic precision is only one (Nielson, 1983).
- (D) Smooth local interpolators involve free parameters or assumptions on the derivatives. Such redundancy and a priori choice may be good for generating varied surfaces with pleasing shapes, but they are drawbacks to scientific applications where the accuracy needs to be respected. When a surface is already given and it is to be approximated by the traditional methods, the free parameters assuring the convergence are required to be evaluated. This is usually impossible, because the analytic properties of the original surface are generally unknown.
- (E) Highly continuous interpolation results in equations coupled tightly across the patches. This can be a critical hurdle against fast computation through distributed processing, etc.
- (F) Singular points, sharp edges and non-manifolds are often encountered in realistic applications. Most interpolation schemes presently available cannot deal with such intricate features in an efficient and robust way.

Because of the above restrictions, most researchers on FEM, etc., still resort to polyhedral models, but they oversimplify the original system neglecting the curvature, yielding significant errors in the analyses. A natural solution to this problem is to account for the normals of the original surfaces evaluated at the mesh vertices. However, existing interpolators summarized below are not suited for this type of quantitative simulation purposes.

There is a vast body of literature on smooth local interpolations. Since the approaches are quite common in geometric design, they are reviewed first. Piper (1987) proved that the tangent plane continuity of adjacent Bézier patches is always attained if and only if their degrees are quartic or higher. The author proposed an interpolation scheme for the quartic case, but it does not uniquely determine a surface only from the given data since the algorithm involves free parameters to be provided a priori. Peters (1991) suggested the singular parameterization, which assumes that the first and mixed derivatives of patches enclosing a vertex vanish at the parametric origin. Although the technique enables  $C^1$  interpolation, it may cause 'bulgy' surfaces demanding minimization of the variation of the cross-boundary derivatives. The

triangular interpolators published up to 1991, which avoid the vertex consistency problem experienced by the Coons–Gordon patches (Gordon, 1971), were surveyed and compared by Mann et al. (1992). They include the convex combination schemes by Nielson, Gregory, et al., and several split domain schemes. The paper concluded that none of the methods available at that time were satisfactory, and the choice of the boundary curves as well as the free parameters was an open question greatly affecting the quality of the interpolants.

After that, Loop (1994) proposed a sextic triangular Bézier patch to define a  $G^1$  spline surface. The scheme has free parameters which can be used to enforce the surface to interpolate given mesh vertices, but this often gives rise to undulations of the result. The degenerate polynomial patches by Neamtu and Pfluger (1994) attain completely local smooth interpolation from a triangular mesh with normals given at its vertices. The algorithm involves free parameters also. Hamann et al. (1997) constructed a  $C^0$  continuous surface using triangular rational-quadratic Bézier patches, but it was intended for approximating contours generated by the marching cubes method, and hence is not applicable directly to the problems addressed here. The triangular  $G^1$  interpolation suggested by Hahmann and Bonneau (2000) is valid for meshes of arbitrary topological type. Their algorithm was recently modified to allow completely free tangent directions of the mesh boundary curves (Hahmann and Bonneau, 2003). When combined with the minimum energy constraints (Hahmann et al., 2000), it yields plausible surfaces. However, it is unknown whether the assumption contributes to the convergence.

The point normal interpolation (Walton and Yeung, 1993; Walton and Meek, 1995) uses parametric cubic curve segments for the boundary of a surface patch, and each segment is determined so that its principal normals at the nodes are aligned with given directions. Although the algorithm is local and relatively simple, its physical meaning is unclear since normals of a surface generally have nothing to do with the principal normals of its boundary. The high degree of interpolation requires special care to avoid unrealistic shapes. The local Hermite diffuse interpolation by Rassineux et al. (2000) can generate continuous surfaces reproducing the original nodal positions and normals, through introduction of a singular weight function to the standard moving least squares approximation. The interpolation was successfully applied to remeshing of triangulated surfaces, but it still needs the information of neighboring triangles.

It should be pointed out that none of the above methodologies solve all the above problems (A)–(F) completely. Studies on (C) and (D) are especially scarce. Surprisingly enough, no paper cited above quantitatively discussed the accuracy of their interpolators viewed as function approximation schemes, except (Gordon, 1971; Nielson, 1983; Rassineux et al., 2000) mentioning the topic briefly. To fill this technical gap between CAD and numerical simulation, a simple algorithm for surface interpolation is proposed in this paper. Its central idea is quadratic interpolation of a curved segment from the position and normal vectors at the end points, with the aid of generalized inverses. It is then used to recover the curvature of triangular or quadrilateral patches. Addressing all the six difficulties, the algorithm has been designed to have the following distinctive features.

- (A) It has the minimum degree two of interpolation necessary for representation of the curvature.
- (B) The approach is quite simple, computationally inexpensive, and hence amenable to various physical evaluations. The low degree is desirable especially for contact algorithms involving implicitization and inversion, since closed-form solutions may be obtained.
- (C) Since the formulation accounts for discontinuity (multiplicity) of normals, sharp edges and singular points, as well as non-manifolds, can be treated quite easily.

- (D) The interpolation assures the  $C^0$  continuity, and converges to the original surface rapidly with the increase in the number of nodes, even in the existence of the singular features. Hence error in the normals can be sufficiently small using rather few patches. This implies asymptotic smoothness.
- (E) The algorithm is completely local requiring only the position vectors and normals given at the vertices of each patch. Therefore, it is suitable for parallel processing.

This paper is organized as follows. The next section formulates the problem and derives the interpolation algorithm. Section 3 assesses the convergence and accuracy of the interpolation through numerical examples, followed by Section 4 summarizing the results with a scope for future work.

## 2. Formulation

Assume that position vectors and normals are provided at the vertices of a polyhedral mesh. The objective here is to recover the curvature of the faceted model through interpolating each patch independently. The process consists of two steps:

- (A) To replace each edge of a patch with a curve orthogonal to the normals given at the end points.
- (B) To fill in the interior of the patch with a parametric polynomial surface reproducing the modified boundary.

They will be described in the following sections in order.

### 2.1. Interpolation of an edge using normals

#### 2.1.1. For smooth surfaces

Discontinuity in normals will be treated in Section 2.1.2. Here surfaces are assumed to be smooth yielding a unique normal everywhere. Consider a curve on such a surface depicted in Fig. 1. At its end points  $P_0$  and  $P_1$ , position vectors  $\mathbf{x}_0$ ,  $\mathbf{x}_1$  and unit normals  $\mathbf{n}_0$ ,  $\mathbf{n}_1$  of the surface, respectively, are given as input data. The first step of the interpolation approximates the curve using a quadratic polynomial of a parameter  $\xi \in [0, 1]$  as<sup>2</sup>

$$\mathbf{x}(\xi) \equiv \mathbf{x}_0 + (\mathbf{d} - \mathbf{c})\xi + \mathbf{c}\xi^2, \quad (1)$$

where  $\mathbf{x}$  denotes the position vector to a point on the curve, and

$$\mathbf{d} \equiv \mathbf{x}_1 - \mathbf{x}_0 \quad (2)$$

is a vector joining the end points. The coefficient  $\mathbf{c}$  is an unknown adding curvature to the segment. For an arbitrary value of this coefficient, Eq. (1) automatically satisfies the following boundary conditions.

$$\mathbf{x}_0 \equiv \mathbf{x}(0), \quad \mathbf{x}_1 \equiv \mathbf{x}(1). \quad (3)$$

Note that the vectors  $\mathbf{n}_0$ ,  $\mathbf{n}_1$  and  $\mathbf{d}$  need not be coplanar. The derivative of the position vector in Eq. (1)

$$\frac{d\mathbf{x}}{d\xi} = \mathbf{d} + (2\xi - 1)\mathbf{c} \quad (4)$$

<sup>2</sup> In this paper, a bold typeface is used for vectors (column matrices), and bold symbols in square brackets denote matrices.

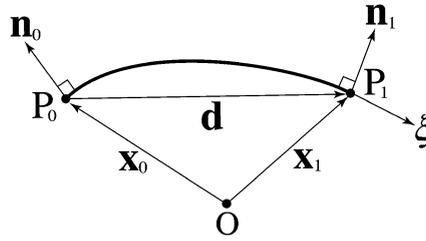


Fig. 1. Interpolation of a segment on a surface using the normals.

should be orthogonal to the normals  $\mathbf{n}_0$  and  $\mathbf{n}_1$  at  $\xi = 0, 1$ . Therefore

$$0 = \mathbf{n}_0^T \frac{d\mathbf{x}}{d\xi} \Big|_{\xi=0} = \mathbf{n}_0^T (\mathbf{d} - \mathbf{c}), \tag{5a}$$

$$0 = \mathbf{n}_1^T \frac{d\mathbf{x}}{d\xi} \Big|_{\xi=1} = \mathbf{n}_1^T (\mathbf{d} + \mathbf{c}) \tag{5b}$$

which can be collected in a matrix form as

$$\begin{bmatrix} \mathbf{n}_0^T \\ \mathbf{n}_1^T \end{bmatrix} \mathbf{c} = \begin{Bmatrix} \mathbf{n}_0^T \mathbf{d} \\ -\mathbf{n}_1^T \mathbf{d} \end{Bmatrix}. \tag{6}$$

Here  $^T$  indicates the transpose. Although the unknown vector  $\mathbf{c}$  has three components, Eqs. (5a), (5b) provide only two conditions, which are insufficient to solve the problem uniquely. However, recalling the fact that the segment was linearly approximated in the polyhedral mesh, the magnitude of the parameter  $\mathbf{c}$  adding curvature should naturally be minimized. This can be attained by

$$\mathbf{c} = \begin{bmatrix} \mathbf{n}_0^T \\ \mathbf{n}_1^T \end{bmatrix}^+ \begin{Bmatrix} \mathbf{n}_0^T \mathbf{d} \\ -\mathbf{n}_1^T \mathbf{d} \end{Bmatrix}, \tag{7}$$

where  $^+$  denotes the generalized inverse. For its definition, the readers are referred to e.g. (Stoer and Bulirsch, 1992, Section 4.8.5). Generalized inverse is usually calculated by means of singular value decomposition (Stoer and Bulirsch, 1992, Sections 6.4 and 6.7). It is, however, fairly complex and not suitable for small applications as is shown above. Instead, direct analytical evaluation using the following formula is recommended.

$$[\mathbf{A}]^+ = \lim_{\alpha \rightarrow +0} ([\mathbf{A}]^* [\mathbf{A}] + \alpha [\mathbf{E}])^{-1} [\mathbf{A}]^*, \tag{8}$$

where  $[\mathbf{A}]$  and  $[\mathbf{A}]^*$  are an arbitrary matrix and its transposed conjugate, respectively, and  $[\mathbf{E}]$  is the unit matrix with consistent dimension. A proof of the above formula is given in Appendix A.

Recalling the fact that the normals  $\mathbf{n}_0$  and  $\mathbf{n}_1$  are unit vectors, the generalized inverse in Eq. (7) is obtained through the formula (8) as

$$\begin{aligned} \begin{bmatrix} \mathbf{n}_0^T \\ \mathbf{n}_1^T \end{bmatrix}^+ &= ([\mathbf{n}_0, \mathbf{n}_1]^+)^T = \lim_{\alpha \rightarrow +0} [\mathbf{n}_0, \mathbf{n}_1] \frac{1}{(1 + \alpha)^2 - c^2} \begin{bmatrix} 1 + \alpha & -c \\ -c & 1 + \alpha \end{bmatrix} \\ &= \begin{cases} \frac{1}{1 - c^2} [\mathbf{n}_0, \mathbf{n}_1] \begin{bmatrix} 1 & -c \\ -c & 1 \end{bmatrix} & (c \neq \pm 1), \\ \frac{1}{2} [\mathbf{n}_0, \pm \mathbf{n}_0] & (c = \pm 1), \end{cases} \end{aligned} \tag{9}$$

where

$$c \equiv \mathbf{n}_0^T \mathbf{n}_1 \tag{10}$$

is the cosine of the angle formed by the normals. Substitution of the above into Eq. (7) yields the curvature parameter as

$$\mathbf{c}(\mathbf{d}, \mathbf{n}_0, \mathbf{n}_1) = \begin{cases} \frac{1}{1-c^2} [\mathbf{n}_0, \mathbf{n}_1] \begin{bmatrix} 1 & -c \\ -c & 1 \end{bmatrix} \begin{Bmatrix} \mathbf{n}_0^T \mathbf{d} \\ -\mathbf{n}_1^T \mathbf{d} \end{Bmatrix} & (c \neq \pm 1), \\ \frac{1}{2} [\mathbf{n}_0, \pm \mathbf{n}_0] \begin{Bmatrix} \mathbf{n}_0^T \mathbf{d} \\ \mp \mathbf{n}_0^T \mathbf{d} \end{Bmatrix} = \mathbf{0} & (c = \pm 1), \end{cases} \tag{11}$$

where  $\mathbf{0}$  denotes the zero vector. The above solution rigorously satisfies the boundary conditions of Eq. (6) for  $c \neq \pm 1$ . Linear edges of a planar patch are treated as a singular case ( $c = \pm 1$ ), remaining unchanged and exact after the interpolation. For the other singular cases, the interpolated curve may not be perpendicular to the normals  $\mathbf{n}_0$  and  $\mathbf{n}_1$  specified at the end points, unless  $\mathbf{n}_0 = \mathbf{n}_1 \perp \mathbf{d}$  happens to hold. This trouble occurs when the curvature undergoes intricate variation beyond the capability of quadratic polynomial representation defined by Eq. (1). It implies improperly coarse sampling, and hence should be avoided through subdivision.

Eq. (11) can be simplified as follows. The unit normals  $\mathbf{n}_0$  and  $\mathbf{n}_1$  are split into the average  $\mathbf{v}$  and deviation  $\Delta \mathbf{v}$  as

$$\mathbf{v} \equiv (\mathbf{n}_0 + \mathbf{n}_1)/2, \tag{12a}$$

$$\Delta \mathbf{v} \equiv (\mathbf{n}_0 - \mathbf{n}_1)/2 \tag{12b}$$

whose inner products with  $\mathbf{d}$  are denoted by

$$d \equiv \mathbf{d}^T \mathbf{v}, \tag{13a}$$

$$\Delta d \equiv \mathbf{d}^T \Delta \mathbf{v}. \tag{13b}$$

Application of Eq. (12b) to the definition (10) gives

$$c = \mathbf{n}_0^T (\mathbf{n}_0 - 2\Delta \mathbf{v}) = 1 - 2\Delta c, \tag{14a}$$

$$\Delta c \equiv \mathbf{n}_0^T \Delta \mathbf{v}. \tag{14b}$$

Then Eqs. (12a)–(14b) lead to another expression of Eq. (11) as

$$\mathbf{c}(\mathbf{d}, \mathbf{n}_0, \mathbf{n}_1) = \begin{cases} \frac{\Delta d}{1 - \Delta c} \mathbf{v} + \frac{d}{\Delta c} \Delta \mathbf{v} & (c \neq \pm 1), \\ \mathbf{0} & (c = \pm 1). \end{cases} \tag{15}$$

Eq. (15) may be advantageous for programming as well as for the following error analysis. Assume that the coefficients for  $\mathbf{v}$  and  $\Delta \mathbf{v}$  on the right-hand side of Eq. (15) contain numerical errors  $\delta_1$  and  $\delta_2$ , respectively, which change the parameter  $\mathbf{c}$  by

$$\Delta \mathbf{c} = [\mathbf{v}, \Delta \mathbf{v}] \begin{Bmatrix} \delta_1 \\ \delta_2 \end{Bmatrix}. \tag{16}$$

In view of Eqs. (12a), (12b), (14a) and (14b), this causes the error in Eq. (6) for the boundary conditions as

$$\Delta \equiv \begin{Bmatrix} \Delta_1 \\ \Delta_2 \end{Bmatrix} = \begin{bmatrix} \mathbf{n}_0^T \\ \mathbf{n}_1^T \end{bmatrix} \Delta \mathbf{c} = \begin{bmatrix} 1 - \Delta c & \Delta c \\ 1 - \Delta c & -\Delta c \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \end{Bmatrix} \tag{17}$$

whose squared magnitude is

$$|\Delta|^2 = \Delta_1^2 + \Delta_2^2 = (1 - \Delta c)^2 \delta_1^2 + \Delta c^2 \delta_2^2. \tag{18}$$

Note that  $\delta_1$  and  $\delta_2$  are decoupled in the above equation, i.e., errors of the coefficients for  $\mathbf{v}$  and  $\Delta \mathbf{v}$  on the right-hand side of Eq. (15) independently affect the accuracy of the orthogonality condition by Eq. (6).

This observation suggests the following practical handling of singular cases in Eq. (15). Eq. (14a) implies that the singularity ( $c = \pm 1$ ) occurs when  $\Delta c$  is 1 or 0, making the denominators of the first or second term, respectively, vanish on the right-hand side of Eq. (15). Accordingly, the first or second term is neglected when  $1 - \Delta c < \varepsilon$  or  $\Delta c < \varepsilon$  holds, respectively, where  $\varepsilon$  is a small positive value. This suppresses generation of unrealistic curves at the singularity, still utilizing the information on normals and giving a better approximation than simply applying  $\mathbf{c} = \mathbf{0}$ , due to the decoupled nature of Eq. (18). The threshold  $\varepsilon$  can be a constant regardless of the geometric scale, because  $\Delta c$  in Eq. (14b) is dimensionless.

2.1.2. For surfaces with discontinuous normals

In the previous section, the discussion was limited to smooth surfaces. However, treatment of singular points with discontinuous (multiple) normals should be established for the methodology to stand advanced usage. Pathological surfaces with two-dimensional distribution of singular points are deemed unimportant for engineering, and hence omitted here. Then the dimension of the singular region is zero or one. The former and the latter correspond to an isolated singular point (Fig. 2(a)) and a sharp edge (Fig. 2(b)), respectively.

Isolated singular points do not cause much difficulty. As depicted in Fig. 2(a), the neighborhood of such a point  $P_0$  can be divided by lines emanating from  $P_0$ . Let  $P_0P_1$  be one such line, then the normal  $\mathbf{n}_0$  at the singular point  $P_0$  can be given as a limit approaching  $P_0$  along the line. The normal can be used for the interpolation of Section 2.1.1 to recover the curvature of  $P_0P_1$ , retaining the continuity of the adjacent patches sharing the segment.

In contrast, sharp edges demand a special care. The sharp edge  $P_0P_1$  shown in Fig. 2(b) has unit normals  $\mathbf{n}_0$  and  $\mathbf{n}_1$  at the end points as a part of the boundary of the surface  $S$ , and the normals yield an interpolation of the edge through Eqs. (15) and (1). Similar process using the normals  $\mathbf{n}_{0'}$  and  $\mathbf{n}_{1'}$  of the surface  $S'$  results in another interpolation for the same edge. Since the two curves are generally not

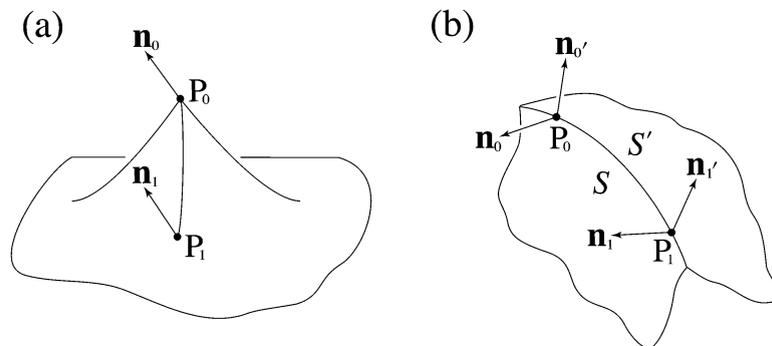


Fig. 2. Discontinuities of normals on surfaces: (a) isolated singular point; (b) sharp edge.

identical, direct application of the algorithm in Section 2.1.1 may violate the continuity. One way to solve the problem is to simultaneously take into account all four normals, thus giving a unique interpolation of the edge  $P_0P_1$ . This is accomplished through imposing the boundary conditions of Eq. (6) for both the surfaces  $S$  and  $S'$  in Fig. 2(b) and determining the coefficient  $\mathbf{c}$  from

$$[\mathbf{n}]\mathbf{c} = \{\mathbf{d}\}, \tag{19a}$$

$$[\mathbf{n}] \equiv \begin{bmatrix} \mathbf{n}_0^T \\ \mathbf{n}_1^T \\ \mathbf{n}_{0'}^T \\ \mathbf{n}_{1'}^T \end{bmatrix}, \quad \{\mathbf{d}\} \equiv \begin{Bmatrix} \mathbf{n}_0^T \mathbf{d} \\ -\mathbf{n}_1^T \mathbf{d} \\ \mathbf{n}_0^T \mathbf{d} \\ -\mathbf{n}_{1'}^T \mathbf{d} \end{Bmatrix}. \tag{19b}$$

Since  $[\mathbf{n}]$  in Eq. (19a) is a  $4 \times 3$  matrix, the vector  $\mathbf{c}$  completely meeting the boundary conditions is usually nonexistent. Nevertheless, the best compromise can always be gained with the aid of the generalized inverse just like Eq. (7), i.e.

$$\mathbf{c} = [\mathbf{n}]^+ \{\mathbf{d}\}. \tag{20}$$

Here ‘best’ means that the above coefficient  $\mathbf{c}$  has the minimum norm  $|\mathbf{c}|$  among all least-square solutions. Recalling the formula (8), the generalized inverse can be written as

$$\begin{aligned} [\mathbf{n}]^+ &\equiv \lim_{\alpha \rightarrow +0} ([\mathbf{n}]^T [\mathbf{n}] + \alpha [\mathbf{E}])^{-1} [\mathbf{n}]^T \\ &= \lim_{\alpha \rightarrow +0} [\mathbf{n}]_{\alpha}^{-1} [\mathbf{n}]^T. \end{aligned} \tag{21}$$

Here the following matrix function of  $\alpha$  is introduced

$$\begin{aligned} [\mathbf{n}]_{\alpha} &\equiv [\mathbf{n}]^T [\mathbf{n}] + \alpha [\mathbf{E}] \\ &\equiv [\mathbf{s}] + \alpha [\mathbf{E}] \\ &\equiv [\mathbf{s}_1 + \alpha \mathbf{e}_1, \mathbf{s}_2 + \alpha \mathbf{e}_2, \mathbf{s}_3 + \alpha \mathbf{e}_3], \end{aligned} \tag{22}$$

where

$$\begin{aligned} [\mathbf{s}] &\equiv [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3] \equiv [\mathbf{n}]^T [\mathbf{n}] \\ &= \mathbf{n}_0 \mathbf{n}_0^T + \mathbf{n}_1 \mathbf{n}_1^T + \mathbf{n}_{0'} \mathbf{n}_{0'}^T + \mathbf{n}_{1'} \mathbf{n}_{1'}^T \quad (\because \text{Eq. (19b)}) \end{aligned} \tag{23}$$

is a  $3 \times 3$  symmetric matrix, and the unit matrix  $[\mathbf{E}]$  of the same dimension is divided into column vectors as

$$[\mathbf{E}] \equiv [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]. \tag{24}$$

The limit operation in Eq. (21) can be performed analytically. First, the inverse of Eq. (22) is written as

$$[\mathbf{n}]_{\alpha}^{-1} = \frac{[\mathbf{a}]}{D}, \quad [\mathbf{a}] \equiv \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \end{bmatrix}, \tag{25a}$$

where  $D$  and  $[\mathbf{a}]$  are the determinant and adjoint of the matrix  $[\mathbf{n}]_{\alpha}$ , respectively, defined by

$$\mathbf{a}_i \equiv (\mathbf{s}_j + \alpha \mathbf{e}_j) \times (\mathbf{s}_k + \alpha \mathbf{e}_k), \tag{25b}$$

$$D \equiv \det[\mathbf{n}]_{\alpha} = \mathbf{a}_i^T (\mathbf{s}_i + \alpha \mathbf{e}_i) \quad (\{i, j, k\} = \{1, 2, 3\}, \{2, 3, 1\}, \{3, 1, 2\}). \tag{25c}$$

Next, Eq. (25b) is expanded with respect to the parameter  $\alpha$  as

$$\mathbf{a}_i = \mathbf{e}_i \alpha^2 + \mathbf{a}_i^{(1)} \alpha + \mathbf{a}_i^{(0)}, \tag{26a}$$

where

$$\mathbf{a}_i^{(1)} \equiv \mathbf{s}_j \times \mathbf{e}_k + \mathbf{e}_j \times \mathbf{s}_k, \tag{26b}$$

$$\mathbf{a}_i^{(0)} \equiv \mathbf{s}_j \times \mathbf{s}_k \quad (\{i, j, k\} = \{1, 2, 3\}, \{2, 3, 1\}, \{3, 1, 2\}). \tag{26c}$$

This yields the adjoint  $[\mathbf{a}]$  of Eq. (25a) as a quadratic polynomial of  $\alpha$ , i.e.,

$$[\mathbf{a}] = [\mathbf{E}] \alpha^2 + [\mathbf{a}]^{(1)} \alpha + [\mathbf{a}]^{(0)}, \tag{27a}$$

where Eq. (24) was used for manipulating the first term, and the following  $3 \times 3$  matrices are introduced.

$$[\mathbf{a}]^{(1)} \equiv \begin{bmatrix} (\mathbf{a}_1^{(1)})^T \\ (\mathbf{a}_2^{(1)})^T \\ (\mathbf{a}_3^{(1)})^T \end{bmatrix}, \quad [\mathbf{a}]^{(0)} \equiv \begin{bmatrix} (\mathbf{a}_1^{(0)})^T \\ (\mathbf{a}_2^{(0)})^T \\ (\mathbf{a}_3^{(0)})^T \end{bmatrix}. \tag{27b}$$

Substitution of Eq. (26a) into Eq. (25c) tells that the determinant is cubic, i.e.

$$\begin{aligned} D &= (\mathbf{e}_i \alpha^2 + \mathbf{a}_i^{(1)} \alpha + \mathbf{a}_i^{(0)})^T (\mathbf{s}_i + \alpha \mathbf{e}_i) \\ &= \alpha^3 + D^{(2)} \alpha^2 + D^{(1)} \alpha + D^{(0)} \end{aligned} \tag{28}$$

whose coefficients are given by

$$D^{(2)} \equiv \mathbf{e}_i^T (\mathbf{s}_i + \mathbf{a}_i^{(1)}), \tag{29a}$$

$$D^{(1)} \equiv \mathbf{s}_i^T \mathbf{a}_i^{(1)} + \mathbf{e}_i^T \mathbf{a}_i^{(0)}, \tag{29b}$$

$$D^{(0)} \equiv \mathbf{s}_i^T \mathbf{a}_i^{(0)} \quad (i = 1, 2, 3). \tag{29c}$$

Substituting Eqs. (27a) and (28) into Eq. (21) through Eq. (25a) leads to the following expression.

$$[\mathbf{n}]^+ = \lim_{\alpha \rightarrow +0} \frac{[\mathbf{n}]^T \alpha^2 + [\mathbf{a}]^{(1)} [\mathbf{n}]^T \alpha + [\mathbf{a}]^{(0)} [\mathbf{n}]^T}{\alpha^3 + D^{(2)} \alpha^2 + D^{(1)} \alpha + D^{(0)}}. \tag{30}$$

It is assured that the left-hand side of the above equation always exists and is unique. Consequently, if the terms of degree less than  $k$  ( $\leq 3$ ) vanish in the denominator, so do the numerator, and the limit becomes the ratio of the  $k$ th degree coefficients. Evaluation of the ratios for  $k = 0, 1, 2, 3$  finally yields the generalized inverse in question as

$$[\mathbf{n}]^+ = \begin{cases} \frac{1}{D^{(0)}} [\mathbf{a}]^{(0)} [\mathbf{n}]^T & (D^{(0)} \neq 0), \\ \frac{1}{D^{(1)}} [\mathbf{a}]^{(1)} [\mathbf{n}]^T & (D^{(0)} = 0 \text{ and } D^{(1)} \neq 0), \\ \frac{1}{D^{(2)}} [\mathbf{n}]^T & (D^{(0)} = 0 \text{ and } D^{(1)} = 0 \text{ and } D^{(2)} \neq 0), \\ [\mathbf{O}] & (\text{otherwise}), \end{cases} \tag{31}$$

where  $[\mathbf{O}]$  is the zero matrix. The last case corresponds to  $[\mathbf{n}] = [\mathbf{O}]$  and never occurs for  $[\mathbf{n}]$  defined by Eq. (19b), since its rows are all unit vectors. Note also that the coefficients  $D^{(k)}$  should be treated as

zeros in floating-point computation, if their absolute values reach below a small positive threshold  $\varepsilon$ . The threshold can be a fixed number irrespective of the geometric scale, because the coefficients defined by Eqs. (29a)–(29c) are all dimensionless. For more safety, similar constraint can be imposed also on the magnitude of the solution  $\mathbf{c}$  in Eq. (20).

Substitution of the above generalized inverse into Eq. (20) yields the curvature parameter  $\mathbf{c}$  which determines the interpolating curve of Eq. (1). Using the curve as a common edge for both the surfaces  $S$  and  $S'$  in Fig. 2(b) retains the continuity. As mentioned before, the edge may not always be perpendicular to the four given normals. However, if the number of independent conditions in Eq. (19a) is reduced to three or less, the boundary conditions are completely met. This occurs, for example, when one of the surfaces  $S$  and  $S'$  in Fig. 2(b) is planar. If both the surfaces are planar, the interpolated segment becomes linear and exact. Since any infinitesimal fraction of a smooth piece in a sharp edge has this situation, the present interpolation is expected to converge to the original surface through subdivision.

The methodology is applicable also to open surfaces. Assume that the surface  $S'$  in Fig. 2(b) is removed leaving its normals  $\mathbf{n}_{0'}$  and  $\mathbf{n}_{1'}$  at the boundary. Then, the edge  $P_0P_1$  of the resultant open surface  $S$  can be interpolated identically to the above procedure. This improves the quality of the curve rather than using only the two normals  $\mathbf{n}_0$  and  $\mathbf{n}_1$ . Although there are an infinite number of candidates for the supplementary normals  $\mathbf{n}_{0'}$  and  $\mathbf{n}_{1'}$ , the most natural choice is the principal normals of the edge.

For non-manifolds, an edge can be shared by more than two patches. Take a general setting where the common edge  $P_0P_1$  of the surfaces  $S$  and  $S'$  in Fig. 2(b) is connected also to other surfaces  $S_1, \dots, S_p$  whose normals are  $\mathbf{n}_{0(1)}, \dots, \mathbf{n}_{0(p)}$  at  $P_0$  and  $\mathbf{n}_{1(1)}, \dots, \mathbf{n}_{1(p)}$  at  $P_1$ , respectively. The orthogonality conditions for these normals need to be respected together with Eq. (19a). This can be done through appending the corresponding rows in the coefficient matrix and vector in Eq. (19b) as

$$[\mathbf{n}] \equiv \begin{bmatrix} \mathbf{n}_0^T \\ \mathbf{n}_1^T \\ \mathbf{n}_{0'}^T \\ \mathbf{n}_{1'}^T \\ \mathbf{n}_{0(1)}^T \\ \mathbf{n}_{1(1)}^T \\ \vdots \\ \mathbf{n}_{0(p)}^T \\ \mathbf{n}_{1(p)}^T \end{bmatrix}, \quad \{\mathbf{d}\} \equiv \begin{bmatrix} \mathbf{n}_0^T \mathbf{d} \\ -\mathbf{n}_1^T \mathbf{d} \\ \mathbf{n}_{0'}^T \mathbf{d} \\ -\mathbf{n}_{1'}^T \mathbf{d} \\ \mathbf{n}_{0(1)}^T \mathbf{d} \\ -\mathbf{n}_{1(1)}^T \mathbf{d} \\ \vdots \\ \mathbf{n}_{0(p)}^T \mathbf{d} \\ -\mathbf{n}_{1(p)}^T \mathbf{d} \end{bmatrix} \quad (32)$$

which requires modification of the matrix  $[\mathbf{s}]$  in Eq. (23) as

$$[\mathbf{s}] \equiv [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3] \equiv [\mathbf{n}]^T [\mathbf{n}] \\ = \mathbf{n}_0 \mathbf{n}_0^T + \mathbf{n}_1 \mathbf{n}_1^T + \mathbf{n}_{0'} \mathbf{n}_{0'}^T + \mathbf{n}_{1'} \mathbf{n}_{1'}^T + \sum_{i=1}^p (\mathbf{n}_{0(i)} \mathbf{n}_{0(i)}^T + \mathbf{n}_{1(i)} \mathbf{n}_{1(i)}^T) \quad (33)$$

while all the other equations remain unchanged, and thus Eq. (20) gives the best curvature parameter  $\mathbf{c}$ . It should be noted that Eqs. (6), (19a) and (19b) for two and four normals, respectively, are special cases of the general situation represented by Eq. (32), and hence the associated algorithms can be merged into a single program.

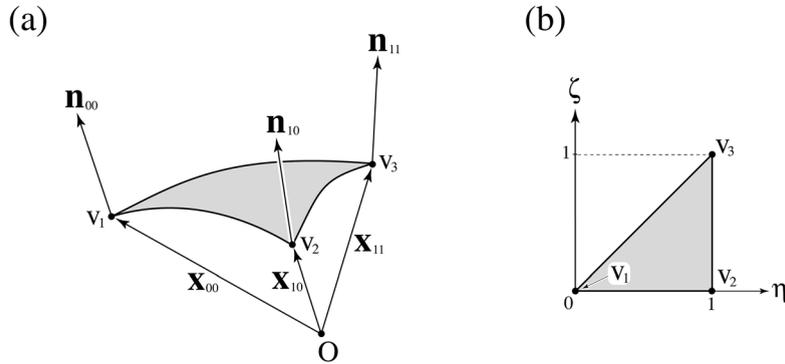


Fig. 3. Interpolation of a triangular patch using the normals.

### 2.2. Interpolation of a patch using normals

Polygonal patches can be interpolated through recovering the curvature of its boundary based on the algorithm of Section 2.1, and then filling in its interior. Here the process is described for triangular and quadrilateral patches as the simplest and most important examples.

#### 2.2.1. For triangular patches

Consider the triangular patch depicted in Fig. 3(a). Its spatial geometry can be described using parameters  $\eta, \zeta$  defined in the region of Fig. 3(b) as

$$\{\mathbf{x}(\eta, \zeta) \mid 0 \leq \zeta \leq \eta \leq 1\} \tag{34}$$

where  $\mathbf{x}$  denotes the position vector to any point on the patch. Assume that the positions of the vertices  $v_1, v_2$  and  $v_3$  in Fig. 3(a)

$$\mathbf{x}_{00} \equiv \mathbf{x}(0, 0), \tag{35a}$$

$$\mathbf{x}_{10} \equiv \mathbf{x}(1, 0), \tag{35b}$$

$$\mathbf{x}_{11} \equiv \mathbf{x}(1, 1) \tag{35c}$$

and the associated unit normals

$$\mathbf{n}_{00} \equiv \mathbf{n}(0, 0), \tag{36a}$$

$$\mathbf{n}_{10} \equiv \mathbf{n}(1, 0), \tag{36b}$$

$$\mathbf{n}_{11} \equiv \mathbf{n}(1, 1) \tag{36c}$$

are given as input data. Here the function of Eq. (34) is approximated by the following quadratic polynomial.<sup>3</sup>

$$\mathbf{x}(\eta, \zeta) = \mathbf{c}_{00} + \mathbf{c}_{10}\eta + \mathbf{c}_{01}\zeta + \mathbf{c}_{11}\eta\zeta + \mathbf{c}_{20}\eta^2 + \mathbf{c}_{02}\zeta^2. \tag{37}$$

<sup>3</sup> Eqs. (1) and (37) can be rewritten equivalently in terms of the Bernstein bases. However, the advantage of the Bézier representations is not so eminent for the simple quadratic curve and patch. For instance, their bounding boxes can be obtained directly from Eqs. (1) and (37), without using the convex hull property. Other geometric computation of curve length, surface area, intersection, etc., may be performed more efficiently through the expanded forms.

Based on the theoretical results presented in Section 2.1, the three edges of the patch are interpolated in the form of Eq. (1) as

$$\mathbf{x}(\eta, 0) = \mathbf{x}_{00} + (\mathbf{d}_1 - \mathbf{c}_1)\eta + \mathbf{c}_1\eta^2, \quad (38a)$$

$$\mathbf{x}(1, \zeta) = \mathbf{x}_{10} + (\mathbf{d}_2 - \mathbf{c}_2)\zeta + \mathbf{c}_2\zeta^2, \quad (38b)$$

$$\mathbf{x}(\eta, \eta) = \mathbf{x}_{00} + (\mathbf{d}_3 - \mathbf{c}_3)\eta + \mathbf{c}_3\eta^2. \quad (38c)$$

Comparing Fig. 1 with the edges in Fig. 3(a) and remembering Eq. (2) yields the coefficients in the above equations as

$$\mathbf{d}_1 \equiv \mathbf{x}_{10} - \mathbf{x}_{00}, \quad \mathbf{c}_1 \equiv \mathbf{c}(\mathbf{d}_1, \mathbf{n}_{00}, \mathbf{n}_{10}), \quad (39a)$$

$$\mathbf{d}_2 \equiv \mathbf{x}_{11} - \mathbf{x}_{10}, \quad \mathbf{c}_2 \equiv \mathbf{c}(\mathbf{d}_2, \mathbf{n}_{10}, \mathbf{n}_{11}), \quad (39b)$$

$$\mathbf{d}_3 \equiv \mathbf{x}_{11} - \mathbf{x}_{00}, \quad \mathbf{c}_3 \equiv \mathbf{c}(\mathbf{d}_3, \mathbf{n}_{00}, \mathbf{n}_{11}), \quad (39c)$$

where  $\mathbf{c}$  is the function defined by Eq. (15), or Eq. (20) through Eqs. (31) and (32) for multiple normals. Note that the vectors  $\mathbf{d}_1$ ,  $\mathbf{d}_2$  and  $\mathbf{d}_3$  satisfy

$$\mathbf{d}_1 + \mathbf{d}_2 = \mathbf{d}_3, \quad (40)$$

and hence are linearly dependent. The curves of Eqs. (38a)–(38c) need to coincide with the following boundary of the surface given by Eq. (37)

$$\mathbf{x}(\eta, 0) = \mathbf{c}_{00} + \mathbf{c}_{10}\eta + \mathbf{c}_{20}\eta^2, \quad (41a)$$

$$\mathbf{x}(1, \zeta) = (\mathbf{c}_{00} + \mathbf{c}_{10} + \mathbf{c}_{20}) + (\mathbf{c}_{01} + \mathbf{c}_{11})\zeta + \mathbf{c}_{02}\zeta^2, \quad (41b)$$

$$\mathbf{x}(\eta, \eta) = \mathbf{c}_{00} + (\mathbf{c}_{10} + \mathbf{c}_{01})\eta + (\mathbf{c}_{11} + \mathbf{c}_{20} + \mathbf{c}_{02})\eta^2, \quad (41c)$$

so the corresponding coefficients must be identical, i.e.,

$$\mathbf{c}_{00} = \mathbf{x}_{00}, \quad (42a)$$

$$\mathbf{c}_{10} = \mathbf{d}_1 - \mathbf{c}_1, \quad (42b)$$

$$\mathbf{c}_{20} = \mathbf{c}_1, \quad (42c)$$

$$\mathbf{c}_{00} + \mathbf{c}_{10} + \mathbf{c}_{20} = \mathbf{x}_{10}, \quad (42d)$$

$$\mathbf{c}_{01} + \mathbf{c}_{11} = \mathbf{d}_2 - \mathbf{c}_2, \quad (42e)$$

$$\mathbf{c}_{02} = \mathbf{c}_2, \quad (42f)$$

$$\mathbf{c}_{10} + \mathbf{c}_{01} = \mathbf{d}_3 - \mathbf{c}_3, \quad (42g)$$

$$\mathbf{c}_{11} + \mathbf{c}_{20} + \mathbf{c}_{02} = \mathbf{c}_3, \quad (42h)$$

which can be solved uniquely for the surface coefficients as

$$\mathbf{c}_{00} = \mathbf{x}_{00}, \quad (43a)$$

$$\mathbf{c}_{10} = \mathbf{d}_1 - \mathbf{c}_1, \quad (43b)$$

$$\mathbf{c}_{01} = \mathbf{d}_2 + \mathbf{c}_1 - \mathbf{c}_3, \quad (43c)$$

$$\mathbf{c}_{11} = \mathbf{c}_3 - \mathbf{c}_1 - \mathbf{c}_2, \quad (43d)$$

$$\mathbf{c}_{20} = \mathbf{c}_1, \quad (43e)$$

$$\mathbf{c}_{02} = \mathbf{c}_2. \quad (43f)$$

Once the boundary is interpolated, therefore, the parametric representation of the patch is readily obtained. Substituting Eqs. (39a)–(39c) and (43a)–(43f) into Eq. (37) and manipulating the result gives another description of the patch

$$\mathbf{x}(\eta, \zeta) = \mathbf{x}_{00}(1 - \eta) + \mathbf{x}_{10}(\eta - \zeta) + \mathbf{x}_{11}\zeta - \mathbf{c}_1(1 - \eta)(\eta - \zeta) - \mathbf{c}_2(\eta - \zeta)\zeta - \mathbf{c}_3(1 - \eta)\zeta \tag{44}$$

which might be useful for coding, since it involves only the vertex and edge data, thus dispensing with calculation of the surface coefficients  $\mathbf{c}_{ij}$ , etc. In the above equation, scalar multipliers consist of three factors  $1 - \eta$ ,  $\eta - \zeta$  and  $\zeta$ , whose values are 1 at the vertices  $v_1, v_2$  and  $v_3$  in Fig. 3, respectively, and vanish at the other two. This is valid even for floating-point arithmetic, so Eq. (44) precisely reproduce the position vectors to the given vertices. It should also be reminded that replacement of  $\mathbf{c}_1, \mathbf{c}_2$  and  $\mathbf{c}_3$  in Eq. (44) with zero vectors leads to linear interpolation.

Partial differentiation of Eq. (44) and application of Eqs. (39a)–(39c) yields

$$\mathbf{x}_\eta \equiv \frac{\partial \mathbf{x}}{\partial \eta} = \mathbf{d}_1 + \mathbf{c}_1((\eta - \zeta) - (1 - \eta)) + (\mathbf{c}_3 - \mathbf{c}_2)\zeta, \tag{45a}$$

$$\mathbf{x}_\zeta \equiv \frac{\partial \mathbf{x}}{\partial \zeta} = \mathbf{d}_2 + \mathbf{c}_2(\zeta - (\eta - \zeta)) + (\mathbf{c}_1 - \mathbf{c}_3)(1 - \eta), \tag{45b}$$

which are required for evaluation of normals at an arbitrary location on the patch. This also indicates that the vectors  $\mathbf{d}_1$  and  $\mathbf{d}_2$  are nothing but the partial derivatives  $\mathbf{x}_\eta$  and  $\mathbf{x}_\zeta$ , respectively, for the linear interpolation.

2.2.2. For quadrilateral patches

The quadrilateral patch illustrated in Fig. 4(a) can also be interpolated in a similar way. The vertices  $v_1, v_2, v_3$  and  $v_4$  need not be coplanar. The position vectors and normals at the vertices are named according to the rule used in Eqs. (35a)–(35c) and (36a)–(36c). To account for the position and normal given at the new vertex  $v_4$ , two terms  $\eta^2\zeta$  and  $\eta\zeta^2$  are added to the surface equation (37) as

$$\mathbf{x}(\eta, \zeta) = \mathbf{c}_{00} + \mathbf{c}_{10}\eta + \mathbf{c}_{01}\zeta + \mathbf{c}_{11}\eta\zeta + \mathbf{c}_{20}\eta^2 + \mathbf{c}_{02}\zeta^2 + \mathbf{c}_{21}\eta^2\zeta + \mathbf{c}_{12}\eta\zeta^2, \tag{46}$$

where the domain of the parameters  $\eta$  and  $\zeta$  is defined as Fig. 4(b). Similarly to Eqs. (38a)–(38c) and (39a)–(39c), the four edges of the patch are interpolated as

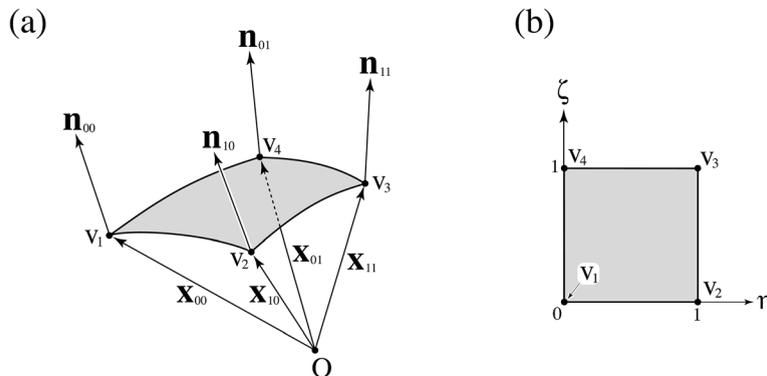


Fig. 4. Interpolation of a quadrilateral patch using the normals.

$$\mathbf{x}(\eta, 0) = \mathbf{x}_{00} + (\mathbf{d}_1 - \mathbf{c}_1)\eta + \mathbf{c}_1\eta^2, \quad (47a)$$

$$\mathbf{x}(1, \zeta) = \mathbf{x}_{10} + (\mathbf{d}_2 - \mathbf{c}_2)\zeta + \mathbf{c}_2\zeta^2, \quad (47b)$$

$$\mathbf{x}(\eta, 1) = \mathbf{x}_{01} + (\mathbf{d}_3 - \mathbf{c}_3)\eta + \mathbf{c}_3\eta^2, \quad (47c)$$

$$\mathbf{x}(0, \zeta) = \mathbf{x}_{00} + (\mathbf{d}_4 - \mathbf{c}_4)\zeta + \mathbf{c}_4\zeta^2 \quad (47d)$$

whose coefficients are given by<sup>4</sup>

$$\mathbf{d}_1 \equiv \mathbf{x}_{10} - \mathbf{x}_{00}, \quad \mathbf{c}_1 \equiv \mathbf{c}(\mathbf{d}_1, \mathbf{n}_{00}, \mathbf{n}_{10}), \quad (48a)$$

$$\mathbf{d}_2 \equiv \mathbf{x}_{11} - \mathbf{x}_{10}, \quad \mathbf{c}_2 \equiv \mathbf{c}(\mathbf{d}_2, \mathbf{n}_{10}, \mathbf{n}_{11}), \quad (48b)$$

$$\mathbf{d}_3 \equiv \mathbf{x}_{11} - \mathbf{x}_{01}, \quad \mathbf{c}_3 \equiv \mathbf{c}(\mathbf{d}_3, \mathbf{n}_{01}, \mathbf{n}_{11}), \quad (48c)$$

$$\mathbf{d}_4 \equiv \mathbf{x}_{01} - \mathbf{x}_{00}, \quad \mathbf{c}_4 \equiv \mathbf{c}(\mathbf{d}_4, \mathbf{n}_{00}, \mathbf{n}_{01}). \quad (48d)$$

The vectors  $\mathbf{d}_1$ ,  $\mathbf{d}_2$ ,  $\mathbf{d}_3$  and  $\mathbf{d}_4$  are again constrained by the relationship

$$\mathbf{d}_1 + \mathbf{d}_2 = \mathbf{d}_3 + \mathbf{d}_4. \quad (49)$$

The interpolated curves of Eqs. (47a)–(47d) must agree with the boundary of the patch obtained from Eq. (46) as

$$\mathbf{x}(\eta, 0) = \mathbf{c}_{00} + \mathbf{c}_{10}\eta + \mathbf{c}_{20}\eta^2, \quad (50a)$$

$$\mathbf{x}(1, \zeta) = (\mathbf{c}_{00} + \mathbf{c}_{10} + \mathbf{c}_{20}) + (\mathbf{c}_{01} + \mathbf{c}_{11} + \mathbf{c}_{21})\zeta + (\mathbf{c}_{02} + \mathbf{c}_{12})\zeta^2, \quad (50b)$$

$$\mathbf{x}(\eta, 1) = (\mathbf{c}_{00} + \mathbf{c}_{01} + \mathbf{c}_{02}) + (\mathbf{c}_{10} + \mathbf{c}_{11} + \mathbf{c}_{12})\eta + (\mathbf{c}_{20} + \mathbf{c}_{21})\eta^2, \quad (50c)$$

$$\mathbf{x}(0, \zeta) = \mathbf{c}_{00} + \mathbf{c}_{01}\zeta + \mathbf{c}_{02}\zeta^2 \quad (50d)$$

which uniquely determines the coefficients for the surface as

$$\mathbf{c}_{00} = \mathbf{x}_{00}, \quad (51a)$$

$$\mathbf{c}_{10} = \mathbf{d}_1 - \mathbf{c}_1, \quad (51b)$$

$$\mathbf{c}_{01} = \mathbf{d}_4 - \mathbf{c}_4, \quad (51c)$$

$$\mathbf{c}_{11} = \mathbf{d}_2 - \mathbf{d}_4 + \mathbf{c}_1 - \mathbf{c}_2 - \mathbf{c}_3 + \mathbf{c}_4, \quad (51d)$$

$$\mathbf{c}_{20} = \mathbf{c}_1, \quad (51e)$$

$$\mathbf{c}_{02} = \mathbf{c}_4, \quad (51f)$$

$$\mathbf{c}_{12} = \mathbf{c}_2 - \mathbf{c}_4, \quad (51g)$$

$$\mathbf{c}_{21} = \mathbf{c}_3 - \mathbf{c}_1. \quad (51h)$$

The above formulation can be extended to general polygonal patches. However, the triangular patch with quadratic description of Eq. (37) is regarded as the best choice to enjoy the advantage of the simple interpolation.

### 3. Numerical examples

Validity and efficiency of the interpolation are assessed here. Throughout this section, triangular meshes are used and an arbitrary unit for length is assumed. First, the algorithm is applied to a unit

<sup>4</sup> Note the difference in the definition of the vectors  $\mathbf{d}_3$  and  $\mathbf{c}_3$  from that of Eq. (39c) for the triangular patch.

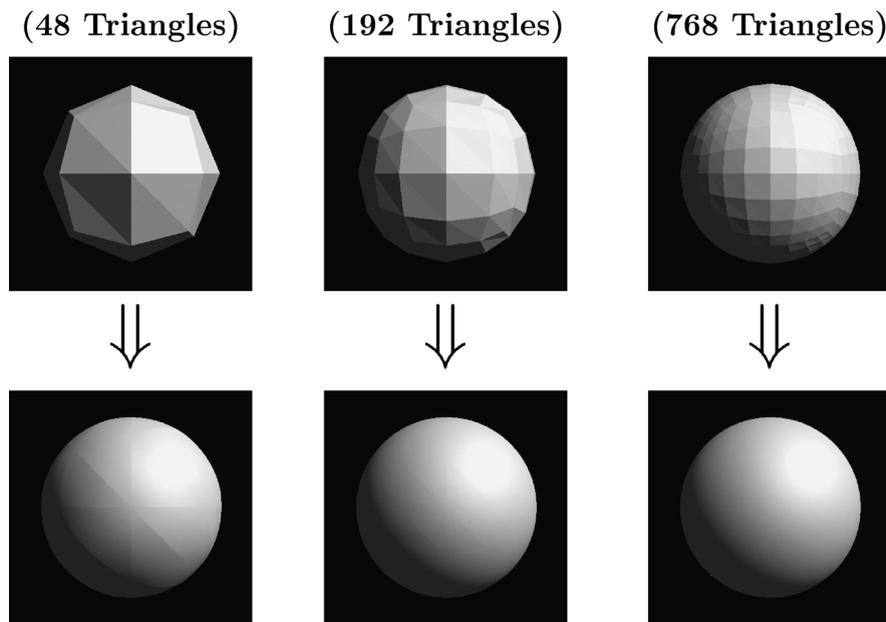


Fig. 5. Effect of the interpolation on triangulated models of a unit sphere.

sphere as a typical smooth surface (Fig. 5). The upper row in the figure contains three polyhedral models with different numbers of triangles. Although all vertices lie exactly on the sphere, the faceted approximation gives poor accuracy. If the normals on the vertices are provided, however, the curvature of the same meshes can be recovered through the proposed interpolation, as depicted in the lower row of Fig. 5. Note the high reproduction quality of the approach even for very few triangular patches. Since the proposed technique does not respect the  $C^1$  or  $G^1$  condition, slight discrepancy of normals at the patch boundaries is observed for the coarsest mesh. Nevertheless, as clearly illustrated by the figure, the error vanishes quite rapidly with the increase in the number of patches. The present model thus converges to the original surface, as will be shown quantitatively below.

The same test is performed for a right circular cone and a right cylinder, as depicted in Figs. 6 and 7, respectively. The models are closed surfaces including the discal bases, where the base radius  $R$  and the height  $H$  are both set to unity. They are typical examples with discontinuous normals discussed in Section 2.1.2. No explicit information concerning the existence and location of the singular point (apex) and the sharp edges is given to the program; they are automatically detected during the rank test in Eq. (31). Figs. 6 and 7 indicate that the interpolation reproduces the apex of the cone and the sharp edges on the boundaries of the bases quite well.

Fig. 8 shows the result of the interpolation applied to a torus whose major and minor radii are  $R = 2$  and  $r = 1$ , respectively. Although the torus has no singular point, it is not convex and involves saddle points, implying different type of difficulty. As evident from Fig. 8, however, the algorithm succeeds in its interpolation. It is notable that the geometrical fidelity is advanced even for the extremely coarse mesh on the left-hand side in the figure. As will be demonstrated quantitatively below, one of the distinctive features of the proposed method is its robustness, avoiding instability due to insufficient geometrical information, which is a usual drawback of many elaborate approaches using high-degree interpolators.

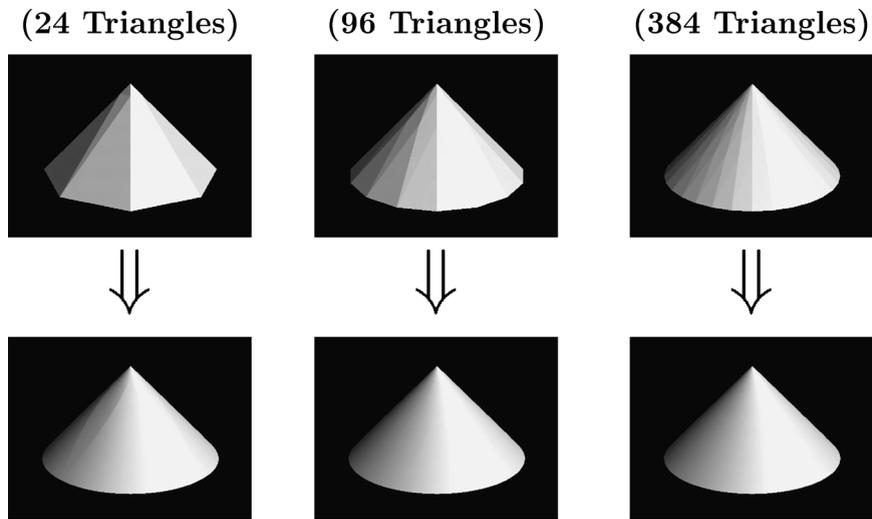


Fig. 6. Effect of the interpolation on triangulated models of a cone ( $R = H = 1$ ).

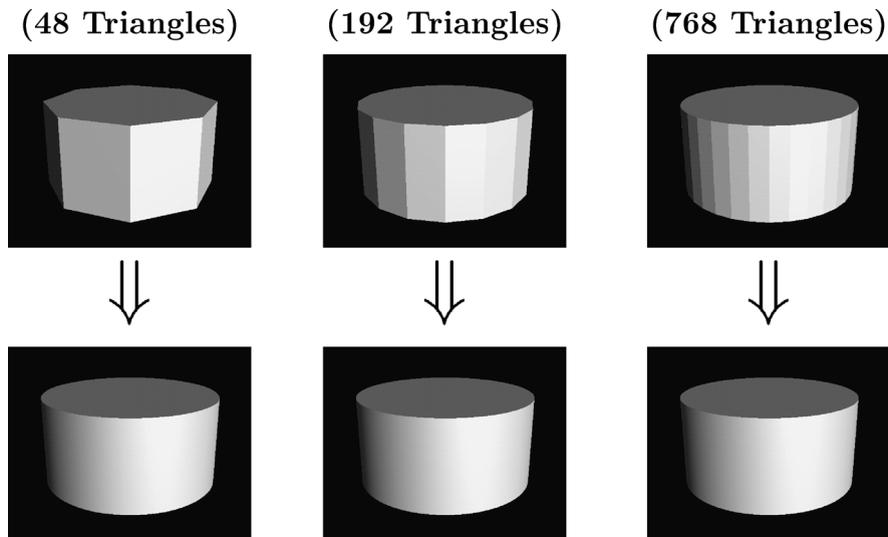


Fig. 7. Effect of the interpolation on triangulated models of a cylinder ( $R = H = 1$ ).

Points on a surface may be classified into elliptic, parabolic and hyperbolic points. Since the above four examples cover all these types, the present technique is expected to have similar effectiveness for a general geometry.

Visual quality is not sufficient for the methodology to be a reliable tool for numerical analyses. Next step is to validate the convergence and accuracy of the interpolation quantitatively. Fig. 9(a) plots the interpolation errors for different mesh resolutions including all the cases of Figs. 5–8. Both axes are in logarithmic scale. In the figure, the error  $\delta_{\max}$  is the maximum distance of the interpolated surface

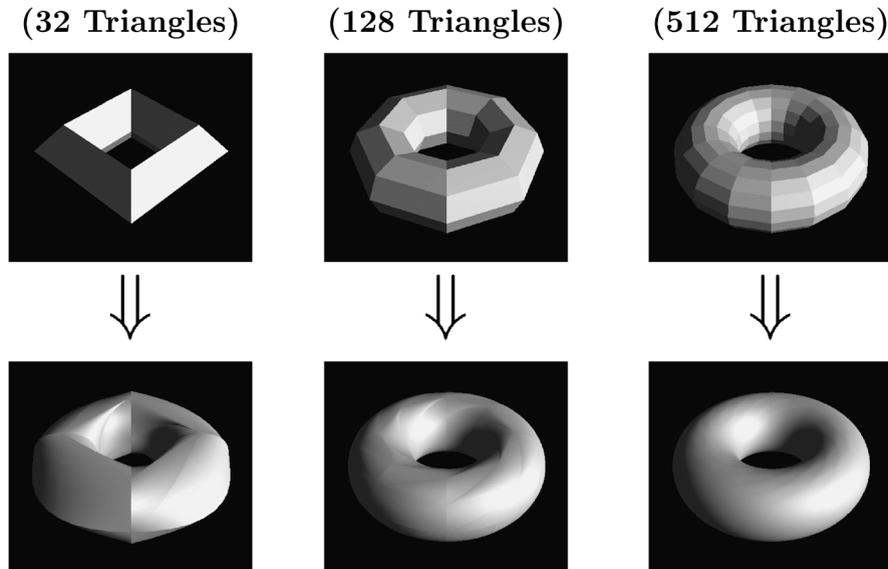


Fig. 8. Effect of the interpolation on triangulated models of a torus ( $R = 2, r = 1$ ).

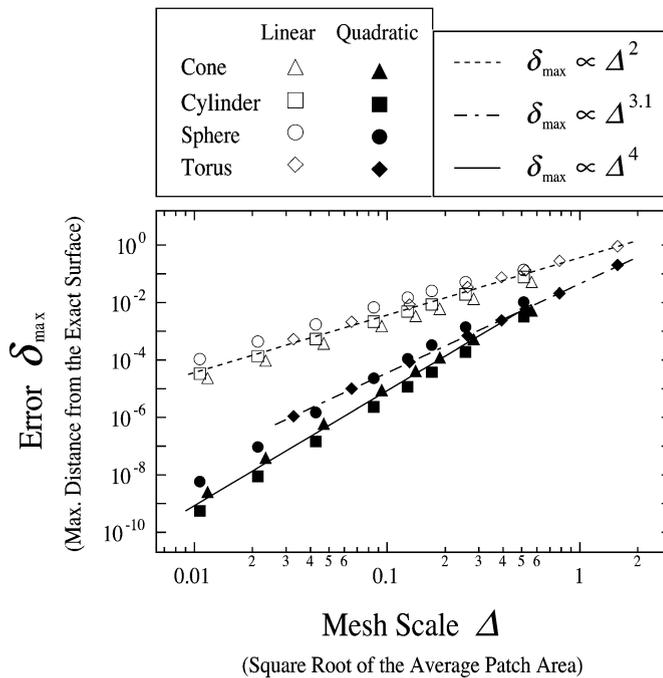
from the exact geometry, evaluated using a sufficient number of sample points.<sup>5</sup> The mesh scale  $\Delta$  is defined as the square root of the average patch area. The solid and hollow symbols represent the results by the present interpolation using quadratic polynomial patch and the linear approximation, respectively, both showing decrease in the error  $\delta_{\max}$  with the mesh scale  $\Delta$ , thus converging to the original surface. Inspection of the figure reveals that the convergence of the linear method is quadratic regardless of the geometry. The proposed algorithm always attains a higher order of convergence which, however, varies for different surfaces: quartic for the sphere, the cone and the cylinder; nearly cubic for the torus. This can be confirmed from Fig. 9(b) indicating the ratio of  $\delta_{\max}$  to  $\delta_{\max}^2$  of the present and linear interpolations, respectively, which approaches a constant value except for the torus. The difference may be attributed to the aforementioned character of the torus, but it is not obvious whether the situation can be improved through controlling the local mesh density and topology. In any case, all the examples demonstrate that the present algorithm realizes upgraded geometrical precision inaccessible by polyhedral models with a practical number of facets.

It is remarkable that the present approach provides better results for all the data given in Fig. 9(a), over several orders of magnitude of the mesh scale. This pictures the reliability of the proposed interpolation, which appears well-balanced between accuracy and robustness. The major factor of the success is considered to be the minimization of the parameter  $\mathbf{c}$  generating the curvature. Indeed, this condition chooses a patch as close to a plane as possible, thus circumventing unrealistic surfaces often produced by sophisticated interpolations forcing strict boundary conditions.

As a more realistic example, a Wavefront OBJ model (triangular mesh with normals) of the Utah teapot (Crow, 1987) is interpolated next. Fig. 10(a) is the polyhedral approximation of the mesh, whose curvature is recovered well by the proposed algorithm as shown in Fig. 10(b). The present methodology

<sup>5</sup> Regardless of the mesh resolution: 442 370 points for the cone; 884 738 points for the cylinder and the sphere; 589 824 points for the torus. They involve the singular points (the apex of the cone and the shape edges of the discal bases).

(a)



(b)

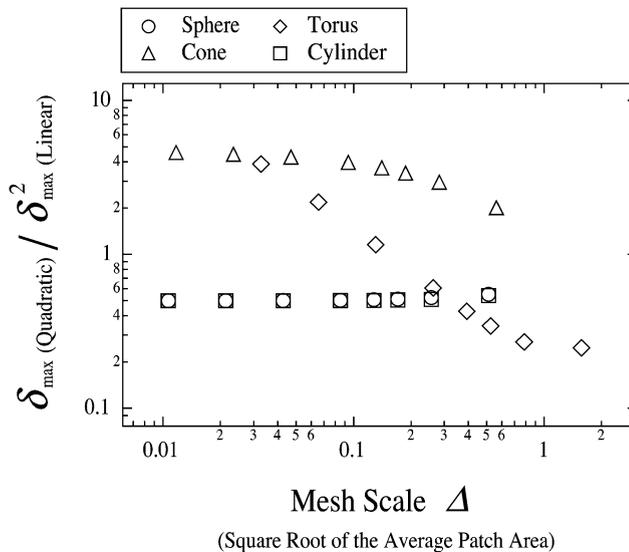


Fig. 9. Interpolation error as a function of the mesh scale.

(a) Linear Interpolation



(b) Present Method



(c) Present Method with Estimated Normals



Fig. 10. Effect of the local interpolation on a general geometry.

in conjunction with normal estimators enables interpolation using only positional data. Fig. 10(c) is an illustration of such cases, where only the coordinates of the mesh vertices are used and the normals are approximated through a variant of the quadratic fit method (Meek and Walton, 2000). Comparison of the result with Fig. 10(b) implies that the quadratic interpolator is robust against errors in normals.

#### 4. Concluding remarks

A simple interpolation algorithm has been proposed and validated through numerical examples. The methodology has the following distinctive features:

- (A) The algorithm is efficient and completely local requiring only the position vectors and normals given at the nodes of a patch, and hence it is suitable for parallel processing.
- (B) The  $C^0$  continuity is always attained, and errors in the normals diminish rapidly with the increase in the number of nodes.

- (C) Since the approach can account for discontinuity (multiplicity) of normals, sharp edges and singular points as well as non-manifolds can be treated quite easily.
- (D) Because of the low degree of the interpolation, it is rather robust and amenable to numerical analyses in comparison with the traditional cubic and more elaborate approximations.

As stated in Section 1, this study aims primarily at refinement of simulation models in engineering, thus yielding improved accuracy. In this context, the interpolation scheme has already been applied successfully to engineering problems. These include: (i) high-precision machining data generation for an aspherical lens; (ii) simulation of elastoplastic 3-D continuum dynamics. Both prohibited the usage of traditional sophisticated surface descriptions, due to severe tolerance as well as geometrical and physical complexity of the systems. In (i), the present method readily yielded positional accuracy of around  $10^{-13}$  m, which could not be attained by other interpolators showing poor or no convergence. The formulation of (ii) is based on a novel finite element with arbitrary curved geometry, which is described efficiently and precisely by the proposed quadratic patches. Application of the surface representation to contact problems is also being explored, because the low degree enables analytical solution of many equations involved in the modeling. Those results will be published in the near future.

**Appendix A. Proof of Eq. (8)**

The singular value decomposition

$$[\mathbf{A}] = [\mathbf{U}][\mathbf{D}][\mathbf{V}]^* \tag{A.1}$$

exists for any matrix  $[\mathbf{A}]$ , where  $[\mathbf{U}]$  and  $[\mathbf{V}]$  are unitary matrices satisfying

$$[\mathbf{U}]^{-1} = [\mathbf{U}]^*, \quad [\mathbf{V}]^{-1} = [\mathbf{V}]^*, \tag{A.2}$$

and all the nondiagonal elements of the matrix  $[\mathbf{D}]$  are zeros, i.e.,

$$[\mathbf{D}]_{[i,j]} = \begin{cases} \sigma_i & (i = j \leq r), \\ 0 & (\text{otherwise}), \end{cases} \tag{A.3}$$

where: the suffix  $[i, j]$  extracts the  $(i, j)$ -element;  $r$  and  $\sigma_i, i = 1, \dots, r$  are the rank and the positive singular values of  $[\mathbf{A}]$ , respectively. Substituting Eq. (A.1) into the right-hand side of Eq. (8) results in

$$\begin{aligned} \lim_{\alpha \rightarrow +0} ([\mathbf{A}]^*[\mathbf{A}] + \alpha[\mathbf{E}])^{-1}[\mathbf{A}]^* &= \lim_{\alpha \rightarrow +0} ([\mathbf{V}][\mathbf{D}]_{\alpha}^{-1}[\mathbf{V}]^*)([\mathbf{V}][\mathbf{D}]^T[\mathbf{U}]^*) \\ &= \lim_{\alpha \rightarrow +0} [\mathbf{V}][\mathbf{D}]_{\alpha}^{\sim}[\mathbf{U}]^*, \end{aligned} \tag{A.4}$$

where Eq. (A.2) was used, and the following matrices are defined:

$$([\mathbf{D}]_{\alpha})_{[i,j]} \equiv ([\mathbf{D}]^T[\mathbf{D}] + \alpha[\mathbf{E}])_{[i,j]} = \begin{cases} \sigma_i^2 + \alpha & (i = j \leq r), \\ \alpha & (i = j > r), \\ 0 & (\text{otherwise}), \end{cases} \tag{A.5}$$

$$([\mathbf{D}]_{\alpha}^{\sim})_{[i,j]} \equiv ([\mathbf{D}]_{\alpha}^{-1}[\mathbf{D}]^T)_{[i,j]} = \begin{cases} \sigma_i / (\sigma_i^2 + \alpha) & (i = j \leq r), \\ 0 & (\text{otherwise}). \end{cases} \tag{A.6}$$

Therefore, Eq. (A.4) converges to

$$[\mathbf{V}][\mathbf{D}]^+[\mathbf{U}]^*$$

which is nothing but the generalized inverse in the form of Eq. (6.4.13) in (Stoer and Bulirsch, 1992, p. 351). Here  $[\mathbf{D}]^+$  is the generalized inverse of  $[\mathbf{D}]$  given by

$$([\mathbf{D}]^+)_{[i,j]} = \begin{cases} 1/\sigma_i & (i = j \leq r), \\ 0 & (\text{otherwise}). \end{cases} \quad \square \quad (\text{A.7})$$

## References

- Crow, F., 1987. The origins of the teapot. *IEEE Computer Graph. Appl.* 7 (1), 8–19.
- Gordon, W.J., 1971. Blending-function methods of bivariate and multivariate interpolation and approximation. *SIAM J. Numer. Anal.* 8 (1), 158–177.
- Hahmann, S., Bonneau, G.-P., 2000. Triangular  $G^1$  interpolation by 4-splitting domain triangles. *Computer Aided Geometric Design* 17, 731–757.
- Hahmann, S., Bonneau, G.-P., 2003. Polynomial surfaces interpolating arbitrary triangulations. *IEEE Transactions on Visualization and Computer Graphics* 9 (1), 99–109.
- Hahmann, S., Bonneau, G.-P., Taleb, R., 2000. Smooth irregular mesh interpolation. In: Cohen, A., Rabut, C., Schumaker, L.L. (Eds.), *Curve and Surface Fitting: Saint-Malo 1999*. Vanderbilt University Press, Nashville, pp. 237–246.
- Hamann, B., Trotts, I.J., Farin, G.E., 1997. On approximating contours of the piecewise trilinear interpolant using triangular rational-quadratic Bézier patches. *IEEE Transactions on Visualization and Computer Graphics* 3 (3), 215–227.
- Loop, C., 1994. A  $G^1$  triangular spline surface of arbitrary topological type. *Computer Aided Geometric Design* 11, 303–330.
- Mann, S., Loop, C., Lounsbery, M., Meyers, D., Painter, J., DeRose, T., Sloan, K., 1992. A survey of parametric scattered data fitting using triangular interpolants. In: Hagen, H. (Ed.), *Curve and Surface Design*. SIAM, Philadelphia, pp. 145–172. Chapter 8.
- Meek, D.S., Walton, D.J., 2000. On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. *Computer Aided Geometric Design* 17, 521–543.
- Neamtu, M., Pfluger, P.R., 1994. Degenerate polynomial patches of degree 4 and 5 used for geometrically smooth interpolation in  $\mathbb{R}^3$ . *Computer Aided Geometric Design* 11, 451–474.
- Nielson, G.M., 1983. A method for interpolating scattered data based upon a minimum norm network. *Mathematics of Computation* 40 (161), 253–271.
- Peters, J., 1991. Parametrizing singularly to enclose data points by a smooth parametric surface. In: *Proceedings of Graphics Interface '91*, Calgary, Alberta.
- Piper, B.R., 1987. Visually smooth interpolation with triangular Bézier patches. In: Farin, G. (Ed.), *Geometric Modeling: Algorithms and New Trends*. SIAM, Philadelphia, pp. 221–233.
- Rassineux, A., Villon, P., Savignat, J.-M., Stab, O., 2000. Surface remeshing by local Hermite diffuse interpolation. *Internat. J. Numer. Methods Engrg.* 49, 31–49.
- Stoer, J., Bulirsch, R., 1992. *Introduction to Numerical Analysis*. Springer-Verlag, New York.
- Walton, D.J., Meek, D.S., 1995. Point normal interpolation for stereolithography modelling. *Comput. Graph.* 19 (3), 345–353.
- Walton, D.J., Yeung, M., 1993. Geometric modelling from CT scans for stereolithography apparatus. In: Tang, Z. (Ed.), *New Advances in CAD & Computer Graphics (Proceedings of CAD/Graphics '93)*. International Academic Publishers, Beijing, China, pp. 417–422.