Subdivision surfaces: a new paradigm for thin-shell finite-element analysis

Fehmi Cirak[1], Michael Ortiz[1,*,†] and Peter Schröder[2]

[1] *Graduate Aeronautical Laboratories, California Institute of Technology, Pasadena, CA 91125, U.S.A.*
[2] *Department of Computer Science, California Institute of Technology, Pasadena, CA 91125, U.S.A.*

SUMMARY

We develop a new paradigm for thin-shell finite-element analysis based on the use of subdivision surfaces for (i) describing the geometry of the shell in its undeformed configuration, and (ii) generating smooth interpolated displacement fields possessing bounded energy within the strict framework of the Kirchhoff–Love theory of thin shells. The particular subdivision strategy adopted here is Loop's scheme, with extensions such as required to account for creases and displacement boundary conditions. The displacement fields obtained by subdivision are H^2 and, consequently, have a finite Kirchhoff–Love energy. The resulting finite elements contain three nodes and element integrals are computed by a one-point quadrature. The displacement field of the shell is interpolated from nodal displacements only. In particular, no nodal rotations are used in the interpolation. The interpolation scheme induced by subdivision is non-local, i.e. the displacement field over one element depend on the nodal displacements of the element nodes and all nodes of immediately neighbouring elements. However, the use of subdivision surfaces ensures that all the local displacement fields thus constructed combine conformingly to define one single limit surface. Numerical tests, including the Belytschko *et al.* [10] obstacle course of benchmark problems, demonstrate the high accuracy and optimal convergence of the method. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: shells; finite elements; subdivision surfaces

1. INTRODUCTION

The Kirchhoff theory of thin plates and the Kirchhoff–Love theory of thin shells are characterized by energy functionals which depend on curvature; consequently they contain second-order derivatives of displacement (e.g. [1, 2]). The resulting Euler–Lagrange —or equilibrium—equations in turn take the form of fourth-order partial differential equations. It is well-known from approximation theory (e.g. Reference [3]) that in this context, the convergence of finite-element solutions

requires so-called C^1 interpolation. More precisely, in order to ensure that the bending energy is finite, the test functions have to be H^2, or square-integrable functions whose first- and second-order derivatives are themselves square-integrable. Unfortunately, for general unstructured meshes it is not possible to ensure C^1 continuity in the conventional sense of strict slope continuity across finite elements when the elements are endowed with purely local polynomial shape functions and the nodal degrees of freedom consist of displacements and slopes only [4]. Inclusion of higher-order derivatives among the nodal variables [5, 6] leads to well-known difficulties, e.g. the inability to account for stress and strain discontinuities in shells whose properties vary discontinuously across element boundaries [4], and, owing to the high order of the polynomial interpolation required, the presence of spurious oscillations in the solution.

The difficulties inherent in C^1 interpolation have motivated a number of alternative approaches, all of which endeavor to 'beat' the C^1 continuity requirement. Examples are: quasi-conforming elements obtained by relaxing the strict Kirchhoff constraint; the use of Reissner–Mindlin theories for thick plates and shells (which requires conventional C^0 interpolation only); reduced-integration penalty methods; mixed formulations; degenerate solid elements; and others. The different approaches proposed in the literature and relevant references thereof are too numerous to list here. Excellent reviews and insightful discussions may be found in References [4, 7, 8, 10–21]. C^0 elements often exhibit poor performance in the thin-shell limit—especially in the presence of severe element distortion. Such distortion may be due to a variety of pathologies such as shear and membrane locking. The proliferation of approaches and the rapid growth of the specialized literature attest to the inherent, perhaps insurmountable, difficulties in vanquishing the C^1 continuity requirement.

Simultaneously with the development of C^0 plate and shell elements, and for the most part unbeknownst to mechanicians, the field of computer-aided geometric design has taken considerable strides towards the efficient generation and representation of smooth surfaces. In particular, the use of *subdivision surfaces* [22–28] provides a powerful tool for generating smooth surfaces which either interpolate or approximate an arbitrary collection of points or 'nodes'. Here, smoothness is understood in the sense of H^2 surfaces, i.e. surfaces whose curvature tensor is L_2, or square summable. Subdivision surfaces follow as the limit of a recursive iteration based on a triangulation of the nodal point set, e.g. by recourse to the classical Loop scheme [24]. Within this framework, the treatment of complex geometries with intersections or curved boundaries is straightforward. Subdivision surfaces obtained by the Loop scheme are guaranteed to be H^2, i.e. to have finite bending energy, and are therefore ideally suited as *test functions* for plate and shell analyses. The smoothness of the limit surface may also be suitably relaxed in the presence of thickness or material discontinuities. The method of subdivision surfaces thus effectively solves the elusive and long-standing C^1-interpolation problem which has traditionally plagued plate and shell finite-element analyses. The ready availability of smooth approximating surfaces for arbitrary geometries and triangulations enables a return to the 'basic' finite-element method, e.g. the Rayleigh–Ritz method, with the attendant guarantee of optimal convergence (e.g. Reference [3]).

We propose the subdivision-surface concept as a new paradigm for plate and shell C^1 finite-element analyses. We rely on subdivision to generate smooth deformed surfaces from a triangulation of an arbitrary nodal point set. This nodal point set is displaced from that which defines the reference configuration of the shell according to an array of nodal displacements. The nodal displacements are determined as follows. The energy of the deformed plate or shell is given by a direct evaluation of the Kirchhoff or Kirchhoff–Love energy functional. The requisite boundedness of the bending energy is ensured by the H^2 property of the test deformed geometries. The

equilibrium displacements then follow simply by recourse to energy minimization. Within the framework of linear theories, this process of energy minimization leads to a symmetric and banded system of linear equations for the nodal displacements.

The triangles in the triangulation of the nodal point set may be regarded as three-node *finite elements*. In particular, the total energy of the shell is the sum of the local energies of the elements. These local energies in turn follow by integration over the domain of the element. However, the interpolation scheme to which the subdivision paradigm leads differs from conventional finite-element interpolation in a crucial respect: *the displacement field within an element depends not only on the displacements of the nodes attached to the element but also on the displacements of all the immediately adjacent nodes in the triangulation.* Thus, the displacement field within an element is determined by the nodal displacements of a 'link' or 'patch' of adjacent elements. (However, special rules are required for elements which abut on an edge of the shell.) Our approach shares some aspects in common with the finite-volume approach recently proposed by Rojek *et al.* [29]. For instance, the patches corresponding to neighbouring elements may overlap. However, when the displacement field is constructed by subdivision (as in our approach), the displacement representations within the intersection of two patches coincide exactly. Thus, subdivision leads to a unique and well-defined surface over the complete finite-element triangulation, as opposed to a collection of non-conforming local interpolations.

An additional advantage afforded by the present approach is that the geometrical modelling and the finite-element analysis are based on an *identical* representational paradigm, that is, both the undeformed and the deformed geometries of the plates and shells are described by recourse to subdivision. The necessity of a unique framework for geometric design and mechanical analysis has been addressed by various authors [30, 31]. The unification of the geometrical and finite-element representations offers a robust environment which effectively sidesteps many of the difficulties inherent in the currently available software tools, which suffer from heterogeneous and, therefore, error-prone interfaces.

The outline of this paper is as follows. In Section 2 we begin by summarizing the relevant equations of the Kirchhoff–Love theory of shells, of which the Kirchhoff theory of plates is a special case. Throughout the present work we confine our attention to the linear theory of shells under static loading. In Section 3 we briefly summarize the relevant aspects of the standard finite-element discretization of the Kirchhoff–Love thin-shell theory. In Section 4 we turn to the central problem of formulating strictly C^1 finite-element interpolation schemes using subdivision surfaces. We begin with a brief summary of the relevant constructions and results pertaining to subdivision surfaces. The application of subdivision surfaces to the finite-element analysis of thin shells is taken up next. Finally, the excellent performance of the method is established in Section 5 with the aid of selected convergence tests, including the obstacle course of benchmark tests proposed by Belytschko *et al.* [10].

2. THIN-SHELL BOUNDARY VALUE PROBLEM

In this section we summarize the field equations for the classical stress-resultant shell model. A detailed presentation of the classical shell theories can be found in Reference [32]. Here we follow Simo *et al.* [12, 13] elegant formulation of the Reissner–Mindlin theory, which we specialize to Kirchhoff–Love theory by explicitly constraining the shell director to remain normal to the deformed middle surface of the shell.
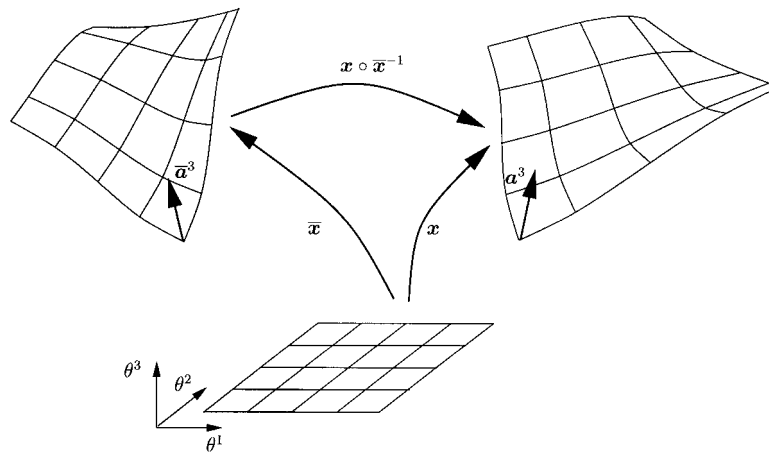
Figure 1. Shell geometry in the reference and the deformed configurations.

2.1. Kinematics of deformation

We begin by considering a shell whose underformed geometry is characterized by a middle surface of domain $\bar{\Omega}$ and boundary $\bar{\Gamma} = \partial\bar{\Omega}$. For simplicity we assume throughout that the thickness h of the shell is uniform. The shell deforms under the action of applied loads and adopts a deformed configuration characterized by a middle surface of domain Ω and boundary $\Gamma = \partial\Omega$.

The position vectors $\bar{\mathbf{r}}$ and $\mathbf{r}$ of a material point in the reference and deformed configurations of the shell may be parametrized in terms of a system $\{\theta^1, \theta^2, \theta^3\}$ of curvilinear coordinates as:

$$\bar{\mathbf{r}}(\theta^1, \theta^2, \theta^3) = \bar{\mathbf{x}}(\theta^1, \theta^2) + \theta^3\bar{\mathbf{a}}_3(\theta^1, \theta^2), \quad -\frac{h}{2} \leqslant \theta^3 \leqslant \frac{h}{2} \tag{1}$$

$$\mathbf{r}(\theta^1, \theta^2, \theta^3) = \mathbf{x}(\theta^1, \theta^2) + \theta^3\mathbf{a}_3(\theta^1, \theta^2), \quad -\frac{h}{2} \leqslant \theta^3 \leqslant \frac{h}{2} \tag{2}$$

The functions $\bar{\mathbf{x}}(\theta^1, \theta^2)$ and $\mathbf{x}(\theta^1, \theta^2)$ furnish a parametric representation of the middle surface of the shell in the reference and deformed configurations, respectively (Figure 1). The corresponding surface basis vectors are

$$\bar{\mathbf{a}}_\alpha = \bar{\mathbf{x}},_\alpha, \quad \mathbf{a}_\alpha = \mathbf{x},_\alpha \tag{3}$$

where here and henceforth Greek indices take the values 1 and 2, and a comma is used to denote partial differentiation. The covariant components of the surface metric tensors in turn follow as

$$\bar{a}_{\alpha\beta} = \bar{\mathbf{a}}_\alpha \cdot \bar{\mathbf{a}}_\beta, \quad a_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta \tag{4}$$

For later reference we also introduce the contravariant components of the undeformed and deformed surface metric tensors, $\bar{a}^{\alpha\beta}$ and $a^{\alpha\beta}$, respectively. The defining property of these components is

$$\bar{a}^{\alpha\gamma}\bar{a}_{\gamma\beta} = \delta^\alpha_\beta, \quad a^{\alpha\gamma}a_{\gamma\beta} = \delta^\alpha_\beta \tag{5}$$

We also note that the element of area over $\bar{\Omega}$ follows as

$$\mathrm{d}\bar{\Omega} = \sqrt{\bar{a}}\,\mathrm{d}\theta^1\mathrm{d}\theta^2 \tag{6}$$

where

$$\sqrt{\bar{a}} = |\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_2| \tag{7}$$

is the Jacobian of the surface co-ordinates $\{\theta^1, \theta^2\}$.

The shell director $\bar{\mathbf{a}}_3$ in the reference configuration coincides with the normal to the undeformed middle surface of the shell and hence has the properties

$$\bar{\mathbf{a}}_\alpha \cdot \bar{\mathbf{a}}_3 = 0, \quad |\bar{\mathbf{a}}_3| = 1 \tag{8}$$

which give $\bar{\mathbf{a}}_3$ explicitly in the form

$$\bar{\mathbf{a}}_3 = \frac{\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_2}{|\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_2|} \tag{9}$$

For the moment we allow the director $\mathbf{a}_3$ on the deformed configuration of the shell to be an arbitrary vector field.

The covariant base vectors in the reference and the current configurations follow simply as

$$\bar{\mathbf{g}}_\alpha = \frac{\partial \bar{\mathbf{r}}}{\partial \theta^\alpha} = \bar{\mathbf{a}}_\alpha + \theta^3 \bar{\mathbf{a}}_{3,\alpha}, \quad \bar{\mathbf{g}}_3 = \frac{\partial \bar{\mathbf{r}}}{\partial \theta^3} = \bar{\mathbf{a}}_3 \tag{10}$$

$$\mathbf{g}_\alpha = \frac{\partial \mathbf{r}}{\partial \theta^\alpha} = \mathbf{a}_\alpha + \theta^3 \mathbf{a}_{3,\alpha}, \quad \mathbf{g}_3 = \frac{\partial \mathbf{r}}{\partial \theta^3} = \mathbf{a}_3 \tag{11}$$

The corresponding covariant components of the metric tensors in both configurations are

$$\bar{g}_{ij} = \bar{\mathbf{g}}_i \cdot \bar{\mathbf{g}}_j, \quad g_{ij} = \mathbf{g}_i \cdot \mathbf{g}_j \tag{12}$$

where here and henceforth lowercase Latin indices take the values 1, 2 and 3.

The Green–Lagrange strain tensor is defined as the difference between the metric tensors on the deformed and undeformed configurations of the shell, i.e.

$$E_{ij} = \tfrac{1}{2}(g_{ij} - \bar{g}_{ij}) \tag{13}$$

Using Equations (10)–(12), the Green–Lagrange strain of the shell is found to be of the form

$$E_{ij} = \alpha_{ij} + \theta^3 \beta_{ij} \tag{14}$$

to first order in the shell thickness h (e.g. [33, pp. 107–112]). The non-zero components of the tensors α_{ij} and β_{ij} are in turn related to the deformation of the shell as follows:

$$\alpha_{ij} = \tfrac{1}{2}(\mathbf{a}_i \cdot \mathbf{a}_j - \bar{\mathbf{a}}_i \cdot \bar{\mathbf{a}}_j) \tag{15}$$

$$\beta_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_{3,\beta} - \bar{\mathbf{a}}_\alpha \cdot \bar{\mathbf{a}}_{3,\beta} \tag{16}$$

In particular, the in-plane components $\alpha_{\alpha\beta}$, or membrane strains, measure the straining of the surface; the components $\alpha_{\alpha 3}$ measure the shearing of the director $\bar{\mathbf{a}}_3$; the component α_{33} measures the stretching of the director; and the components $\beta_{\alpha\beta}$, or bending strains, measure the bending or change in curvature of the shell, respectively.

The above kinematic relations allow for finite deformations as well as for shearing and stretching of the shell director. In the remainder of this paper we restrict our attention to the Kirchhoff–Love

theory of thin shells and, accordingly, we constrain the deformed director $\mathbf{a}_3$ to coincide with the unit normal to the deformed middle surface of the shell, i.e.

$$\mathbf{a}_\alpha \cdot \mathbf{a}_3 = 0, \quad |\mathbf{a}_3| = 1 \tag{17}$$

which yields

$$\mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{|\mathbf{a}_1 \times \mathbf{a}_2|} \tag{18}$$

Owing to these constraints, the shear strains $\alpha_{\alpha 3}$ vanish identically and the bending strains simplify to

$$\beta_{\alpha\beta} = \bar{\mathbf{a}}_{\alpha,\beta} \cdot \bar{\mathbf{a}}_3 - \mathbf{a}_{\alpha,\beta} \cdot \mathbf{a}_3 \tag{19}$$

It follows from these relations that, by virtue of the assumed Kirchhoff–Love kinematics, all the strain measures of interest may be deduced from the deformation of the middle surface of the shell.

For simplicity, we restrict the scope of subsequent discussions to linearized kinematics. To this end, we begin by writing

$$\mathbf{x}(\theta^1, \theta^2) = \bar{\mathbf{x}}(\theta^1, \theta^2) + \mathbf{u}(\theta^1, \theta^2) \tag{20}$$

where $\mathbf{u}(\theta^1, \theta^2)$ is the displacement field of the middle surface of the shell. To first order in $\mathbf{u}$ the membrane and bending strains then follow as

$$\alpha_{\alpha\beta} = \tfrac{1}{2}(\bar{\mathbf{a}}_\alpha \cdot \mathbf{u}_{,\beta} + \mathbf{u}_{,\alpha} \cdot \bar{\mathbf{a}}_\beta) \tag{21}$$

$$\beta_{\alpha\beta} = -\mathbf{u}_{,\alpha\beta} \cdot \bar{\mathbf{a}}_3 + \frac{1}{\sqrt{\bar{a}}}[\mathbf{u}_{,1} \cdot (\bar{\mathbf{a}}_{\alpha,\beta} \times \bar{\mathbf{a}}_2) + \mathbf{u}_{,2} \cdot (\bar{\mathbf{a}}_1 \times \bar{\mathbf{a}}_{\alpha,\beta})]$$

$$+ \frac{\bar{\mathbf{a}}_3 \cdot \bar{\mathbf{a}}_{\alpha,\beta}}{\sqrt{\bar{a}}}[\mathbf{u}_{,1} \cdot (\bar{\mathbf{a}}_2 \times \bar{\mathbf{a}}_3) + \mathbf{u}_{,2} \cdot (\bar{\mathbf{a}}_3 \times \bar{\mathbf{a}}_1)] \tag{22}$$

It is clear from these expressions that the displacement field $\mathbf{u}$ of the middle surface furnishes a complete description of the deformation of the shell and may therefore be regarded as the primary unknown of the analysis. It also follows that the deformed and undeformed domains Ω and $\bar{\Omega}$ are indistinguishable to within the order of approximation of the linearized theory and, in consequence, we drop the distinction between the two domains throughout the remainder of the paper.

2.2. Equilibrium deformations of elastic shells

Next, we seek to characterize the equilibrium configurations of the shell by recourse to energy principles. For simplicity, we shall assume throughout that the shell is linear elastic with a strain energy density per unit area of the form

$$W(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \frac{Eh}{1 - v^2} H^{\alpha\beta\gamma\delta} \alpha_{\alpha\beta} \alpha_{\gamma\delta} + \frac{1}{2} \frac{Eh^3}{12(1 - v^2)} H^{\alpha\beta\gamma\delta} \beta_{\alpha\beta} \beta_{\gamma\delta} \tag{23}$$

where E is Young's modulus, v is Poisson's ratio, and

$$H^{\alpha\beta\gamma\delta} = v\bar{a}^{\alpha\beta}\bar{a}^{\gamma\delta} + \tfrac{1}{2}(1 - v)(\bar{a}^{\alpha\gamma}\bar{a}^{\beta\delta} + \bar{a}^{\alpha\delta}\bar{a}^{\beta\gamma}) \tag{24}$$

In (23), the first term is the membrane strain energy density and the second term is the bending strain energy density. The membrane and bending stresses follow from (23) by work conjugacy, with the result

$$n^{\alpha\beta} = \frac{\partial W}{\partial \alpha_{\alpha\beta}} = \frac{Eh}{1-v^2} H^{\alpha\beta\gamma\delta} \alpha_{\gamma\delta} \tag{25}$$

$$m^{\alpha\beta} = \frac{\partial W}{\partial \beta_{\alpha\beta}} = \frac{Eh^3}{12(1-v^2)} H^{\alpha\beta\gamma\delta} \beta_{\gamma\delta} \tag{26}$$

The membrane and bending stress tensors $n^{\alpha\beta}$ and $m^{\alpha\beta}$ may be given a direct mechanistic interpretation as force and moment resultants [2, 12].

The shell is subject to a system of external dead loads consisting of distributed loads $\mathbf{q}$ per unit area of Ω and axial forces $\mathbf{N}$ per unit length of Γ. Under these conditions the potential energy of the shell takes the form

$$\Phi[\mathbf{u}] = \Phi^{\text{int}}[\mathbf{u}] + \Phi^{\text{ext}}[\mathbf{u}] \tag{27}$$

where

$$\Phi^{\text{int}}[\mathbf{u}] = \int_\Omega W(\boldsymbol{\alpha}, \boldsymbol{\beta}) \, d\Omega \tag{28}$$

is the elastic potential energy and

$$\Phi^{\text{ext}}[\mathbf{u}] = -\int_\Omega \mathbf{q} \cdot \mathbf{u} \, d\Omega - \int_\Gamma \mathbf{N} \cdot \mathbf{u} \, ds \tag{29}$$

is the potential energy of the applied loads.

The stable equilibrium configurations of the shell now follow from the principle of minimum potential energy:

$$\Phi[\mathbf{u}] = \inf_{\mathbf{v} \in V} \Phi[\mathbf{v}] \tag{30}$$

where V is the space of solutions consisting of all trial displacement fields $\mathbf{v}$ with finite energy $\Phi[\mathbf{v}]$. It is clear from the form of the elastic energy of the shell that such trial displacement fields must necessarily have square integrable first and second derivatives. Within the context of the linear theory, and under suitable technical restrictions on the domain Ω and the applied loads, it therefore follows that V may be identified with the Sobolev space of functions $H^2(\Omega, R^3)$. In particular, an acceptable finite-element interpolation method must guarantee that all trial finite element interpolants belong to this space.

The Euler–Lagrange equations corresponding to the minimum principle (30) may be expressed in weak form as

$$\langle D\Phi[\mathbf{u}], \delta\mathbf{u}\rangle = \langle D\Phi^{\text{int}}[\mathbf{u}], \delta\mathbf{u}\rangle + \langle D\Phi^{\text{ext}}[\mathbf{u}], \delta\mathbf{u}\rangle = 0 \tag{31}$$

which is a statement of the principle of virtual work. Here $\langle D\Phi[\mathbf{u}], \delta\mathbf{u}\rangle$ denotes the first variation of Φ at $\mathbf{u}$ in the direction of the virtual displacements $\delta\mathbf{u}$,

$$\langle D\Phi^{\text{int}}[\mathbf{u}], \delta\mathbf{u}\rangle = \int_\Omega [n^{\alpha\beta} \delta\alpha_{\alpha\beta} + m^{\alpha\beta} \delta\beta_{\alpha\beta}] \, d\Omega \tag{32}$$

is the internal virtual work and

$$\langle D\Phi^{\text{ext}}[\mathbf{u}], \delta\mathbf{u}\rangle = -\int_{\Omega} \mathbf{q} \cdot \delta\mathbf{u}\, d\Omega - \int_{\Gamma} \mathbf{N} \cdot \delta\mathbf{u}\, ds \tag{33}$$

is the external virtual work. The minimum principle (30) or, equivalently, the virtual work principle (31) are subsequently taken as a basis for formulating finite-element approximations to the equilibrium configuration of the shell.

3. FINITE-ELEMENT DISCRETIZATION

We now turn to the finite-element discretization of the potential energy of the shell (27) or, equivalently, its first variation (31). To this end, it proves convenient to adopt Voigt's notation and map symmetric second-order tensors into arrays by recourse to the conventions:

$$\mathbf{n} = \begin{pmatrix} n^{11} \\ n^{22} \\ n^{12} \end{pmatrix}, \quad \mathbf{m} = \begin{pmatrix} m^{11} \\ m^{22} \\ m^{12} \end{pmatrix}, \quad \boldsymbol{\alpha} = \begin{pmatrix} \alpha_{11} \\ \alpha_{22} \\ 2\alpha_{12} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_{11} \\ \beta_{22} \\ 2\beta_{12} \end{pmatrix} \tag{34}$$

The constitutive relations (Equations (25) and (26)) may similarly be written in the form:

$$\mathbf{n} = \frac{Eh}{1-v^2}\mathbf{H}\boldsymbol{\alpha} \tag{35}$$

$$\mathbf{m} = \frac{Eh^3}{12(1-v^2)}\mathbf{H}\boldsymbol{\beta} \tag{36}$$

where we write

$$\mathbf{H} = \begin{pmatrix} (\bar{a}^{11})^2 & v\bar{a}^{11}\bar{a}^{22} + (1-v)(\bar{a}^{12})^2 & \bar{a}^{11}\bar{a}^{12} \\ & (\bar{a}^{22})^2 & \bar{a}^{22}\bar{a}^{12} \\ \text{sym.} & & \frac{1}{2}[(1-v)\bar{a}^{11}\bar{a}^{22} + (1+v)(\bar{a}^{12})^2] \end{pmatrix} \tag{37}$$

which replaces (24) within the Voigt formalism. Using these conventions, the internal virtual work (32) may be recast in the convenient form

$$\langle \Phi^{\text{int}}[\mathbf{u}], \delta\mathbf{u}\rangle = \int_{\Omega} \left[\frac{Eh}{1-v^2}\delta\boldsymbol{\alpha}^{\text{T}}\mathbf{H}\boldsymbol{\alpha} + \frac{Eh^3}{12(1-v^2)}\delta\boldsymbol{\beta}^{\text{T}}\mathbf{H}\boldsymbol{\beta} \right] d\Omega \tag{38}$$

Next we proceed to partition the domain Ω of the shell into a finite-element mesh, the precise nature of which will remain unspecified for now. The collection of element domains in the mesh is $\{\Omega_K,\ K = 1,\ldots,\text{NEL}\}$, where Ω_K denotes the domain of element K and NEL is the total number of elements in the mesh. The finite-element mesh may be taken as a basis for introducing a displacement interpolation of the general form

$$\mathbf{u}_h(\theta^1, \theta^2) = \sum_{I=1}^{\text{NP}} N^I(\theta^1, \theta^2)\mathbf{u}_I \tag{39}$$

where $\{N^I,\ I = 1,\ldots,\text{NP}\}$ are the shape functions, $\{\mathbf{u}_I,\ I = 1,\ldots,\text{NP}\}$ are the corresponding nodal displacements, and NP is the number of nodes in the mesh. Owing to the linearity of the

dependence of the finite-element interpolant $\mathbf{u}_h$ on the nodal displacements, an application of (21) and (22) to (39) gives the finite-element membrane and bending strains in the form

$$\boldsymbol{\alpha}_h(\theta^1, \theta^2) = \sum_{I=1}^{\text{NP}} \mathbf{M}^I(\theta^1, \theta^2) \mathbf{u}_I \tag{40}$$

$$\boldsymbol{\beta}_h(\theta^1, \theta^2) = \sum_{I=1}^{\text{NP}} \mathbf{B}^I(\theta^1, \theta^2) \mathbf{u}_I \tag{41}$$

for some matrices $\mathbf{M}^I$ and $\mathbf{B}^I$. The precise form of this matrices is given in Appendix A.3. Finally, the introduction of (39)–(41) into the principle of virtual work (31) yields the equations of equilibrium for the nodal displacements:

$$\mathbf{K}_h \mathbf{u}_h = \mathbf{f}_h \tag{42}$$

where, by a slight abuse of notation, here we take $\mathbf{u}_h$ to signify the array of nodal displacements,

$$\mathbf{K}_h^{IJ} = \sum_{K=1}^{\text{NEL}} \int_{\Omega_K} \left[\frac{Eh}{1-v^2} (\mathbf{M}^I)^{\text{T}} \mathbf{H} \mathbf{M}^J + \frac{Eh^3}{12(1-v^2)} (\mathbf{B}^I)^{\text{T}} \mathbf{H} \mathbf{B}^J \right] \mathrm{d}\Omega \equiv \sum_{K=1}^{\text{NEL}} \mathbf{K}_K^{IJ} \tag{43}$$

is the stiffness matrix, and

$$\mathbf{f}_h^I = \sum_{K=1}^{\text{NEL}} \left\{ \int_{\Omega_K} \mathbf{q} N^I \, \mathrm{d}\Omega + \int_{\Gamma_K \cap \Gamma} \mathbf{N} N^I \, \mathrm{d}s \right\} \equiv \sum_{K=1}^{\text{NEL}} \mathbf{f}_K^I \tag{44}$$

is the nodal force array. It should be carefully noted that, as expected, the global stiffness and force arrays just defined follow by the assembly of low-dimensionality element stiffness and force arrays $\mathbf{K}_K^{IJ}$ and $\mathbf{f}_K^I$, respectively, as is standard in the finite-element method.

The computation of element arrays requires the evaluation of integrals extended to the domain of each element, cf. Equations (43) and (44). These integrals may efficiently be evaluated by recourse to numerical quadrature without compromising the order of convergence. For instance, the application of a quadrature rule to the calculation of the element stiffness matrices leads to the expression

$$\mathbf{K}_K^{IJ} = \sum_{G=1}^{\text{NQ}} \left[\frac{Eh}{1-v^2} (\mathbf{M}^I)^{\text{T}} \mathbf{H} \mathbf{M}^J + \frac{Eh^3}{12(1-v^2)} (\mathbf{B}^I)^{\text{T}} \mathbf{H} \mathbf{B}^J \right]_{(\theta_G^1, \theta_G^2)} \sqrt{a(\theta_G^1, \theta_G^2)} w_G \tag{45}$$

where (θ_G^1, θ_G^2) are the quadrature points, w_G are the corresponding quadrature weights, and NQ is the number of quadrature points in the rule. The element force arrays $\mathbf{f}_K^I$ may be computed likewise.

Sufficient conditions for the quadrature rule to preserve the order of convergence of the finite-element method may be found in Reference [34]. In general, these conditions demand that certain shape function derivatives be computed exactly and, consequently, place a lower bound on the order NQ of the quadrature rule. These theoretical considerations and our numerical tests show that a one-point quadrature rule is sufficient to compute all element arrays of interest.

4. SUBDIVISION SURFACES AND FINITE-ELEMENT INTERPOLATION

It is clear from the preceding developments that the central problem in thin-shell finite-element analysis is the formulation of shape functions N^I which are H^2 (or 'C^1' in the usual finite-element terminology), as this property ensures the finiteness of the energy of the trial displacement fields. In this section we develop one such interpolation scheme based on the notion of subdivision surface.

4.1. Subdivision surfaces

We begin by reviewing the essential ideas behind subdivision surfaces using one-dimensional examples first, then moving to the two-dimensional manifold (with boundary) setting relevant to shells. Here we limit ourselves to reviewing various elementary properties of subdivision. The interested reader is referred to [26, 27, 35, 36] for more details and further pointers to the literature.

Throughout this discussion the term *vertex* will be used to refer to nodes in the mesh. For example, a vertex has associated with it a particular nodal position. A vertex has a topological neighbourhood defined by the structure of the mesh. For example, its 1-*ring* neighbours are all those vertices which share an edge with it (and recursively for its k-ring). This distinction between vertices and nodal positions is typically not needed when dealing with finite elements, but it is important to keep these distinctions in mind when dealing with smooth subdivision.

At the highest level of description we may say that subdivision schemes construct smooth surfaces through a limiting procedure of repeated refinement starting from an initial mesh. This initial mesh will also be referred to as the *control mesh* of the surface. Generally, subdivision schemes consist of two steps. First the mesh is refined, e.g., all faces are quadrisected, followed by the computation of new nodal positions. These positions are simple, linear functions of the nodal positions of the coarser mesh. For the schemes of interest these computations are local, i.e. they involve only nodal positions of the coarser mesh within a small, finite topological neighbourhood, leading to very efficient implementations. Using a suitable choice of weights, such subdivision schemes can be designed to produce a smooth surface in the limit. Subdivision methods which result in limit surfaces whose curvature tensor is square integrable are especially appealing for geometrical modelling applications and for the purpose of thin-shell analysis.

The first such schemes were proposed by Catmull and Clark [22] and Doo and Sabin [23]. Since then, many other schemes have been proposed and studied extensively in the mathematical geometric modelling literature. The methods can be separated into two groups:

 (i) *Interpolating schemes*: The nodal positions of the coarser mesh are fixed, while only the nodal positions of new vertices are computed when going from a coarser to a finer mesh. Consequently, the nodal positions of the initial mesh, as well as any nodes produced during subdivision, interpolate the limit surface. Interpolating schemes for quadrilateral meshes have been introduced by Kobbelt *et al.* [31], while Dyn *et al.* [25] and Zorin *et al.* [37] described interpolating schemes for triangular meshes. In both cases the limit surfaces are C^1 but their curvatures do not exist. Therefore, these schemes are not suitable for the special application of thin-shell analysis.

(ii) *Approximating schemes*: These schemes compute both new nodal positions for the newly created vertices, as well as for the vertices inherited from the coarser mesh, i.e., those which
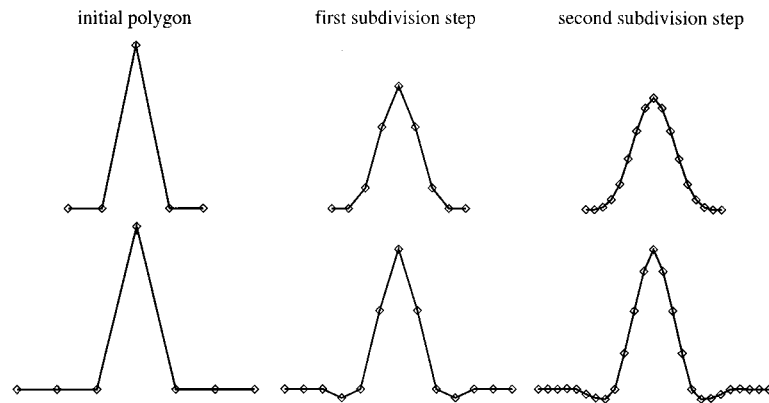
Figure 2. Two subdivision methods: approximating (first row) and interpolating (second row).

already carried nodal positions. Consequently, the nodal positions of the initial mesh are not samples of the final surface. The schemes of Catmull–Clark [22] and Doo–Sabin [23] fall into this class and operate on quadrilateral meshes. An approximating scheme for triangular meshes introduced by Loop [24] is used in the present work. This scheme produces limit surfaces which are globally C^2 except at a number of isolated points where they are only C^1. However, their principal curvatures are square integrable [38], making them good candidates for thin-shell analysis.

A simple one-dimensional example of an approximating as well as an interpolating scheme is shown in Figure 2. We assume that the polygon with the nodes $\mathbf{x}_I^k$ is the result of subdivision step k. In the subsequent refinement $k + 1$ for the approximating scheme, a new vertex gets a nodal position $\mathbf{x}_{2I+1}^{k+1}$ which is the average of its two neighbouring nodes $\mathbf{x}_I^k$ and $\mathbf{x}_{I+1}^k$:

$$\mathbf{x}_{2I+1}^{k+1} = \tfrac{1}{2}(\mathbf{x}_I^k + \mathbf{x}_{I+1}^k) \tag{46}$$

The nodal positions of the existing vertices are recomputed as

$$\mathbf{x}_{2I}^{k+1} = \tfrac{1}{8}(\mathbf{x}_{I-1}^k + 6\mathbf{x}_I^k + \mathbf{x}_{I+1}^k) \tag{47}$$

Since throughout the subdivision process all nodal positions are recomputed as weighted averages of nearby vertices, the resulting scheme does not interpolate the nodal positions of the control mesh, but rather approximates them.

In contrast, the interpolating scheme (see Figure 2) does not modify the co-ordinates of nodes existing at the previous refinement level:

$$\mathbf{x}_{2I}^{k+1} = \mathbf{x}_I^k \tag{48}$$

Only the newly generated vertices receive new nodal co-ordinates:

$$\mathbf{x}_{2I+1}^{k+1} = \tfrac{1}{16}(-\mathbf{x}_{I-1}^{k} + 9\mathbf{x}_{I}^{k} + 9\mathbf{x}_{I+1}^{k} - \mathbf{x}_{I+2}^{k}) \qquad (49)$$

Repeating this process *ad infinitum* leads to smooth curves. In the case of the approximating scheme above, these curves are actually cubic splines which are C^2 continuous. The interpolating scheme on the other hand is known as the 4pt scheme, and it can be shown that the resulting curves are $C^{2-\varepsilon}$ continuous [39]. Since the entire subdivision process is linear, the resulting limit curves (or surfaces) are linear combinations of basis functions (sometimes referred to as 'fundamental solutions' of the subdivision process). The compact support of the subdivision rules ensures that the basis functions are compactly supported as well. For the approximating scheme above, this support extends two vertices to the left and two vertices to the right (and hence is a 2-ring), while the 4pt scheme has a support covering three vertices to each the left and right (hence a 3-ring). Both schemes belong to the class of regular subdivision schemes since the weights are the same for every vertex and every level. In the two-dimensional case, we will see that the weights will depend on the valence, i.e. the number of edges attached to a vertex, but are otherwise the same from level to level. These rules are also referred to as semi-regular. Similarly one can adapt the rules to non-smooth features such as boundaries or creases [40], as well as other boundary conditions [41].

In the one-dimensional case, subdivision schemes do not offer an important advantage since curves of the desired smoothness are easy to construct with traditional approaches such as Hermitian interpolation. In the two-dimensional, arbitrary-topology manifold setting, by contrast, subdivision methods offer significant advantages over other methods of smooth-surface construction. In fact, their original invention was motivated by the difficulties of constructing smooth-surface models of arbitrary topology. For example, it is well known that C^2 arbitrary-topology surfaces built with traditional patches require up to sixth-order polynomials, leading to cumbersome computations and difficult-to-manage cross-patch continuity conditions. Additionally, the resulting degrees of freedom often lack physical meaning. In the subdivision setting however, the only degrees of freedom are the nodal positions and the resulting surfaces are guaranteed to be smooth without the need to enforce cross-patch continuity conditions.

In the following sections we introduce the refinement rules used in Loop's subdivision scheme for surfaces, and, based on the concept of the *subdivision matrix*, briefly discuss the basic ideas behind the scheme's smoothness analysis. This discussion will serve to prepare the way for the parameterization of subdivision surfaces needed for the evaluation of positions, tangents, and curvatures. In particular, it is possible to evaluate these quantities exactly through the use of eigenanalysis without going to the limit. While we focus on Loop's scheme, we hasten to point out that the basic ideas and machinery apply equally well to other subdivision schemes. In particular, the Catmull–Clark scheme, with quadrilateral elements, is a promising alternative for finite-element computations, since quadrilateral elements generally perform relatively better for very coarse meshes.

4.2. Refinement rules

In Loop's subdivision scheme, the control mesh and all refined meshes consist of triangles only. These are refined by quadrisection (Figure 4). After the refinement step, the nodal positions of the refined mesh are computed as weighted averages of the nodal positions of the unrefined mesh.

Valence of a vertex	:	The number of element edges attached to a vertex
Regular vertex	:	Vertex with valence six
Irregular vertex	:	Vertex with valence unequal to six
One ring of a vertex	:	Set of vertices incident to the vertex
One ring of an element	:	Set of elements incident to the element

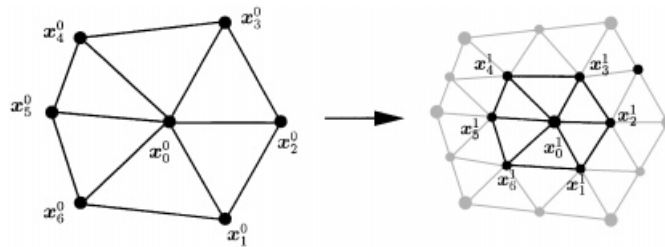Figure 3. Summary of frequently used terms in the case of triangle meshes.

Figure 4. Refinement of a triangular mesh by quadrisection.

We distinguish two cases: new vertices associated with the edges of the coarser mesh, and old vertices of the coarse mesh.

The coordinates of the newly generated nodes $\mathbf{x}_1^1, \mathbf{x}_2^1, \mathbf{x}_3^1, \ldots$ on the edges of the previous mesh are computed as

$$\mathbf{x}_I^{k+1} = \frac{3\mathbf{x}_0^k + \mathbf{x}_{I-1}^k + 3\mathbf{x}_I^k + \mathbf{x}_{I+1}^k}{8}, \quad I = 1, \ldots, N \tag{50}$$

whereby index I is to be understood in modulo arithmetic. The old vertices get new nodal positions according to

$$\mathbf{x}_0^{k+1} = (1 - Nw)\mathbf{x}_0^k + w\mathbf{x}_1^k + \cdots + w\mathbf{x}_N^k \tag{51}$$

where $\mathbf{x}^k$ are the nodal positions of the mesh at level k and $\mathbf{x}^{k+1}$ are the respective positions for the mesh $k + 1$. The valence of the vertex, i.e. the number of edges incident on it, is denoted by N. Note that all newly generated vertices have valence 6 while only the vertices of the original mesh may have valence other than 6. We will refer to the former case (valence $= 6$) as *regular* and to the latter case as *irregular*. Equations (50) and (51) are visualized in symbolic form in Figure 5 with the so-called *subdivision mask*. At this stage it is not obvious how to choose the parameter w to get C^1 continuous surfaces. In the original scheme, Loop [24] proposed

$$w = \frac{1}{N} \left[\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{N} \right)^2 \right] \tag{52}$$

As it turns out other values for w also give smooth surfaces. For example, Warren's [42] choice for w is simpler to evaluate than that of Equation (52):

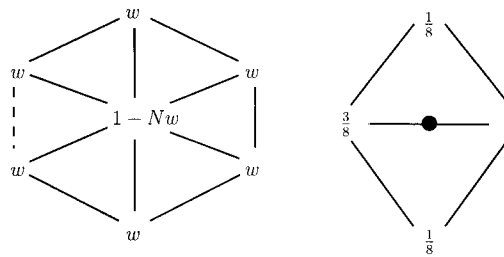$$w = \frac{3}{8N} \quad \text{for } N > 3 \quad w = \frac{3}{16} \quad \text{for } N = 3 \tag{53}$$

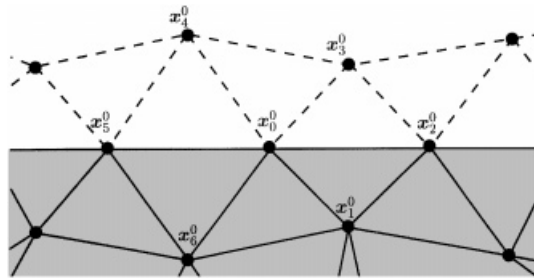Figure 5. Refinement mask for Loop's subdivision scheme.

Figure 6. Smooth boundary approximation with artificial nodes.

Although the choice for the weights appears somewhat arbitrary, the motivation for this choice will be discussed in the next section. In any case, the weights used by the subdivision scheme depend only on the connectivity of the mesh and are independent of the nodal positions.

So far we have ignored the boundary of the mesh, where the subdivision rules need to be modified. Two choices are possible here. The first method considers the boundary as a one-dimensional curve and applies the rules of Equations (46) and (47) to any vertices on the boundary (see [40]). Another method for the treatment of boundaries, which we employ in our implementation, was proposed by Schweitzer [35]. For each boundary edge, one temporary vertex is defined, after which the ordinary rules are applied. The nodal positions of the temporary vertices are set to (see Figure 6):

$$\mathbf{x}_3^0 = \mathbf{x}_0^0 + \mathbf{x}_2^0 - \mathbf{x}_1^0 \quad \text{and} \quad \mathbf{x}_4^0 = \mathbf{x}_0^0 + \mathbf{x}_5^0 - \mathbf{x}_6^0 \tag{54}$$

This particular choice of temporary vertices effectively reproduces the one-dimensional subdivision rules.

This simplified boundary treatment with temporary vertices leads to approximation errors for corners and boundary vertices with valence other than 4. However, these errors do not inhibit the convergence of the finite-element method. For modelling surfaces with creases or shell intersections, the edge rules can also be applied within the mesh on appropriately tagged edges. Figure 7 shows an example of a geometrically more complex shape, a distributor cap, using such crease rules, demonstrating the ability of subdivision surfaces to model intricate shapes in a straightforward fashion. The entire shape is a single, piecewise smooth surface. In this example, the original mesh is chosen to be least-squares optimal in terms of geometric approximation fidelity. For
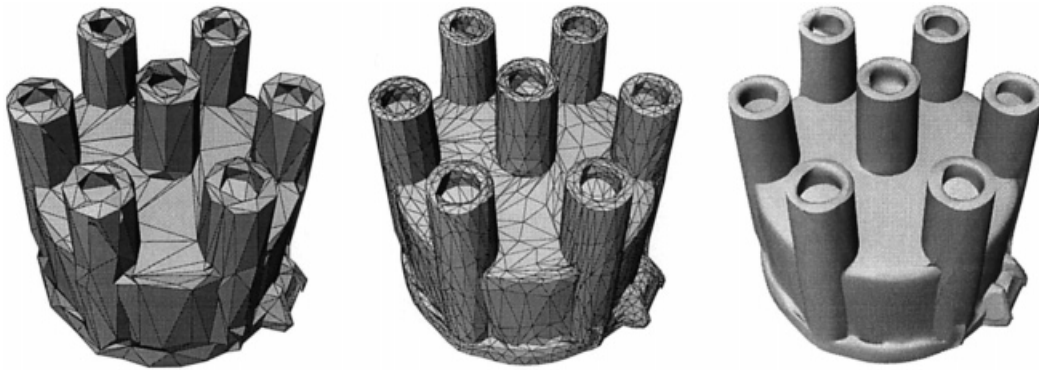
Figure 7. Distributor cap: control mesh, first subdivided mesh, and the resulting limit surface (original data courtesy Hugues Hoppe).

finite-element simulations, this triangulation first needs to be improved to ensure a uniform aspect-ratio bound. Algorithms for performing this task have been described by Kobbelt *et al.* [43]. In instances where the original geometry is given in some other form, a remeshing step can be applied to produce high-quality subdivision meshes, as demonstrated by Lee *et al.* [44].

4.3. Convergence and smoothness

Subdivision methods for arbitrary-topology surfaces were introduced more than 20 years ago, but not widely used in applications until the early 1990s when theoretical breakthroughs put their convergence and smoothness analysis on a solid mathematical foundation [26, 27, 35]. If all vertices are regular (valence 6 for triangles or valence 4 for rectangular schemes), powerful Fourier methods can be used to establish smoothness properties. These techniques do not apply in the arbitrary-topology manifold setting, in which irregular vertices cannot be avoided. The critical question in this setting concerns the smoothness properties around irregular vertices. The key tool in this analysis is the local subdivision matrix and its structure. The fact that the analysis of these schemes can be reduced to the analysis of a local matrix is due to the local support of the basis functions. In other words, the behaviour of the surface in a neighbourhood of a node depends only on those basis functions whose support overlaps a neighbourhood of the node.

For Loop's scheme, the relevant neighbourhood around a vertex is a 2-ring, i.e. all vertices which can be reached by traversing no more than two edges. The subdivision matrix expresses the linear relationship between the nodal positions in a 2-ring around a given vertex at level k and the nodal positions around the same vertex in the 2-ring at level $k + 1$. This 2-ring analysis is required to *establish* smoothness. Details of this approach, including necessary and sufficient conditions, can be found in Reif [26], Zorin [27], and Schweitzer [35]. Once the analytic properties have been established, quantities at a point, such as position or tangents, can be computed using an even smaller subdivision matrix which relates 1-rings at level k to those at level $k + 1$. In the following we will discuss only this simpler setting since it is the one which is needed for actual computations.

Let $\mathbf{X}^k$ be the vector of nodal positions of a vertex with valence N and its 1-ring neighbors at level k, $\mathbf{X}^k = (\mathbf{x}_0^k, \mathbf{x}_1^k, \ldots, \mathbf{x}_N^k)$. Note that in the surface case we treat this vector as an $N + 1$

vector of three-dimensional vectors, while in the functional setting it would be an $N + 1$ vector of scalars. We can now express the linear relationship between the nodal positions at level k and $k + 1$ with an $(N + 1) \times (N + 1)$ matrix $\mathbf{S}$:

$$\mathbf{X}^{k+1} = \mathbf{S}\mathbf{X}^k \tag{55}$$

The entries of the matrix $\mathbf{S}$ are given by the subdivision rules (Equations (51) and (50)). The study of the limit surface then amounts to examining

$$\mathbf{X}^\infty = \lim_{k \to \infty} \mathbf{S}^k \mathbf{X}^0 \tag{56}$$

From this it immediately follows that the limit surface cannot exist if $\mathbf{S}$ has an eigenvalue of modulus larger than 1. Furthermore, it can be shown that it must have a single eigenvalue 1 and all other eigenvalues must have modulus strictly smaller than 1. This property also implies that subdivision schemes are affinely invariant, i.e. an affine transformation applied to the nodal positions of the control mesh results in a limit surface having undergone the same affine transformation, which is a desirable property in practical applications. The associated right eigenvector is easily seen to be the vector of all 1's.

In the following we summarize the main results as needed for our finite-element setting. Assume that the subdivision matrix has a complete set of eigenvectors (this property holds for all schemes of practical interest, although it is not necessary for the analysis). Since the subdivision matrix $\mathbf{S}$ is not self-adjoint, let $\mathbf{R}_I$ and $\mathbf{L}_J$ be the right and left eigenvectors of $\mathbf{S}$, respectively

$$\mathbf{L}_J \cdot \mathbf{R}_I = \delta_{IJ} \tag{57}$$

and let λ_I be the associated eigenvalues in non-increasing magnitude order, with $\lambda_0 = 1$. Using the eigenvalue decomposition, the nodal vector $\mathbf{X}^0$ can be written as

$$\mathbf{X}^0 = \sum_{I=0}^N \mathbf{a}_I^0 \mathbf{R}_I \tag{58}$$

with $\mathbf{a}_I^0 = \mathbf{L}_I \cdot \mathbf{X}^0$. Choosing this basis, Equation (56) takes the simple form

$$\mathbf{X}^\infty = \lim_{k \to \infty} \mathbf{S}^k \sum_{I=0}^N \mathbf{a}_I^0 \mathbf{R}_I = \lim_{k \to \infty} \sum_{I=0}^N \lambda_I^k \mathbf{a}_I^0 \mathbf{R}_I \tag{59}$$

From this representation and the facts that $\lambda_0 = 1$ and $|\lambda_I| < 1$ for $I = 1, \ldots, N$, it follows immediately that the centre nodal position as well as all nodal positions in the 1-ring converge to

$$\mathbf{X}^\infty = \mathbf{a}_0^0 \mathbf{R}_0 \tag{60}$$

Since $\mathbf{R}_0$ has the form $\mathbf{R}_0 = (1, 1, \ldots, 1)$ all the nodal positions in the 1-ring neighbourhood actually converge to a single position:

$$\mathbf{a}_0^0 = \mathbf{L}_0 \cdot \mathbf{X}^0 \tag{61}$$

For the original Loop subdivision scheme, $\mathbf{L}_0$ is given as

$$\mathbf{L}_0 = (1 - Nl, l, \ldots, l) \quad \text{with} \quad l = \frac{1}{3/(8w) + N} \tag{62}$$

In practical terms this means that we can evaluate the limit position of the surface given at any finite subdivision level by simply applying the mask shown in Figure 8. We can continue this
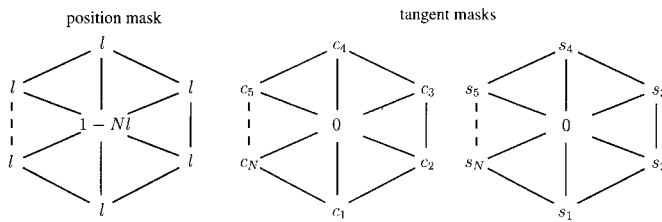
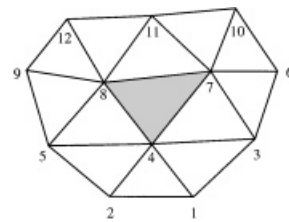Figure 8. Limit masks for Loop's subdivision scheme.

Figure 9. A regular box-spline patch with 12 control points.

analysis and compute tangent vectors in a similar way. Assume that the control nodal positions have been translated by $-\mathbf{a}_0$ so that the limit position for our selected vertex is the origin. The leading term of Equation (59) is now controlled by the subdominant eigenvalues. Here we have $\lambda_1 = \lambda_2 > \lambda_3$, which results in

$$\lim_{k \to \infty} \frac{\mathbf{X}^k}{\lambda_1^k} = \mathbf{a}_1^0 \mathbf{R}_1 + \mathbf{a}_2^0 \mathbf{R}_2 \tag{63}$$

indicating that the nodal positions in the 1-ring all converge to a common plane spanned by the vectors $\mathbf{a}_1^0$ and $\mathbf{a}_2^0$. While these vectors are not necessarily orthogonal, they do span a plane for almost all initial configurations. An explicit formula for two tangent vectors to the limit surface is

$$\mathbf{t}_1 = \mathbf{L}_1 \cdot \mathbf{X}^0, \quad \mathbf{t}_2 = \mathbf{L}_2 \cdot \mathbf{X}^0 \tag{64}$$

whence the shell director follows as

$$\mathbf{a}_3 = \frac{\mathbf{t}_1 \times \mathbf{t}_2}{|\mathbf{t}_1 \times \mathbf{t}_2|} \tag{65}$$

In certain settings we may have $\lambda_1 \geqslant \lambda_2 > \lambda_3$, as pointed out by Schweitzer [35]. The above equations for tangent vectors continue to hold, requiring an argument only slightly more involved than the one above. For Loop's scheme, $\mathbf{L}_1, \mathbf{L}_2$ are given by

$$\mathbf{L}_1 = (0, c_1, c_2, \ldots, c_N), \quad \mathbf{L}_2 = (0, s_1, s_2, \ldots, s_N) \tag{66}$$

with

$$c_I = \cos \frac{2\pi(I-1)}{N}, \quad s_I = \sin \frac{2\pi(I-1)}{N}$$

Again we find that a very simple computation (see the masks in Figure 8) allows us to compute the limit surface normal given the subdivision mesh at any level k. So far we have only discussed the evaluation of position and tangents of the limit surface at parametric locations corresponding to control vertices. For numerical evaluation of the Kirchhoff–Love energy functional we need to evaluate these quantities—and curvature quantities—at quadrature points. This is particularly simple in the case of Loop's scheme (and similarly so for Catmull–Clark's scheme) since Loop's scheme actually generalizes quartic box splines (Catmull–Clark's scheme generalizes bi-cubic splines). If the valencies of the vertices of a given triangle are all equal to 6, the resulting piece of limit surface is exactly described by a single quartic box-spline patch, for which very efficient evaluation

schemes exist at arbitrary parameter locations. We call such a patch *regular* (see Figure 9). Regular patches are controlled by 12 basis, or shape, functions (see Appendix A.1), since only their support overlaps the given patch. If a triangle is irregular, i.e. one of its vertices has valence other than 6 the resulting patch is not a quartic box spline. Arbitrary parameter locations may nevertheless be treated simply by the method described in the next section.

4.4. Function evaluation for arbitrary parameter values

In this section we discuss the evaluation of function values and derivatives for Loop subdivision surfaces at arbitrary parameter locations. These function evaluations arise during the computation of element stiffness and force arrays by numerical quadrature, Equation (45). Despite early attempts [22], the proper parameterization of subdivision surfaces has been until recently an unsolved problem in the vicinity of irregular vertices. Stam [28, 45] presented an elegant solution to this problem based on the eigendecomposition of the subdivision matrix.

A convenient local parameterization of the limit surface may be obtained as follows. For each triangle in the control mesh we choose (θ^1, θ^2) as two of its barycentric co-ordinates within their natural range

$$T = \{(\theta^1, \theta^2), \ \text{s.t.} \ \theta^\alpha \in [0,1], \ \theta^1 + \theta^2 \leqslant 1\} \tag{67}$$

The triangle T in the (θ^1, θ^2)-plane may be regarded as a master or standard element domain. It should be emphasized that this parameterization is defined locally for each element in the mesh. The entire discussion of parameterization and function evaluation may therefore be couched in local terms.

Consequently, we proceed to consider a generic element in the mesh and introduce a local numbering of the nodes lying in its immediate 1-neighbourhood. For regular patches such as depicted in Figure 9, Loop's scheme leads to classical quartic box splines. Therefore, the local parameterization of the limit surface may be expressed in terms of box-spline shape functions, with the result

$$\mathbf{x}(\theta^1, \theta^2) = \sum_{I=1}^{12} N^I(\theta^1, \theta^2) \mathbf{x}_I \tag{68}$$

where now the label I refers to the local numbering of the nodes. The precise form of the shape functions $N^I(\theta^1, \theta^2)$ is given in Appendix A.1. The surface within the shaded triangle in Figure 9 is controlled by the 12 local control vertices. In contrast to Hermitian interpolation, the surface is solely controlled by the position of these control vertices, and first- or second-order derivatives at the nodes are not utilized. The image of the standard domain T under the embedding (68) constitutes the geometric domain of the element under consideration within the limit surface. The embedding (68) may thus be regarded as a conventional isoparametric mapping from the standard domain T onto Ω, with (θ^1, θ^2) playing the role of natural co-ordinates.

The parameterization in Equation (68) may also be used for the displacement field. Locally one then has

$$\mathbf{u}_h(\theta^1, \theta^2) = \sum_{I=1}^{12} N^I(\theta^1, \theta^2) \mathbf{u}_I \tag{69}$$

For function evaluation on irregular patches the mesh has to be subdivided until the parameter value of interest is interior to a regular patch. At that point the regular box-spline parameterization
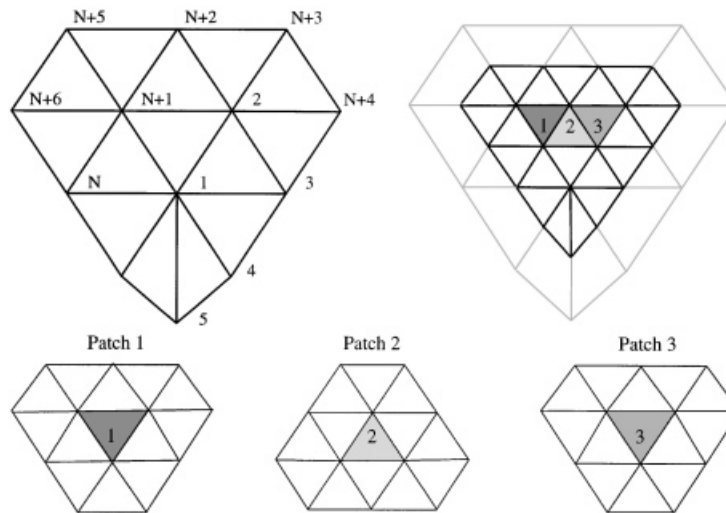
Figure 10. Refinement near an extraordinary vertex.

applies once again. It should be noted that the refinement is performed for parameter evaluation only. For simplicity we assume that irregular patches have one irregular vertex only. This restriction can always be met for arbitrary initial meshes through one step of subdivision, which has the effect of separating all irregular vertices.

Figure 10 illustrates the basic idea for arbitrary parameter evaluation. It shows an irregular patch (center) with a single vertex of valence 7. After one level of subdivision this patch is divided into four patches, three of which are regular. If the desired parameter value lies within those patches we may immediately evaluate the surface by using the canonical regular-patch evaluation routine. If our desired parameter location lies within the fourth sub-patch, we must repeatedly subdivide until it falls within a regular patch. This can be achieved for any parameter value away from the irregular vertex. As shown in Figure 10, after one subdivision step the triangles marked one, two, and three are regular patches. The action of the subdivision operator for this entire neighbourhood can again be described by a matrix

$$\mathbf{X}^1 = \mathbf{A}\mathbf{X}^0 \tag{70}$$

The matrix $\mathbf{A}$ now has dimension $(N+12, N+6)$ and its entries can be derived from the subdivision rules as presented in Section 4.2. For the proposed shell element with one-point quadrature at the barycentre of the element, a single subdivision step is sufficient, since the sampling point (centre of the initial patch) lies in sub-patch 2. We define 12 selection vectors $\mathbf{P}_I$, $I = 1, \ldots, 12$, of dimension $(N + 12)$ which extract the 12 box-spline control points for sub-patch 2 from the $N + 12$ points of the refined mesh. The entries of $\mathbf{P}_I$ are zeros and one depending on the indices of the initial and refined meshes. To evaluate the function values in the three triangles with the box-spline shape functions N^I, a co-ordinate transformation must be performed. The relation between the co-ordinates (θ^1, θ^2) of the original triangles and the co-ordinates $(\tilde{\theta}^1, \tilde{\theta}^2)$ of the refined triangles
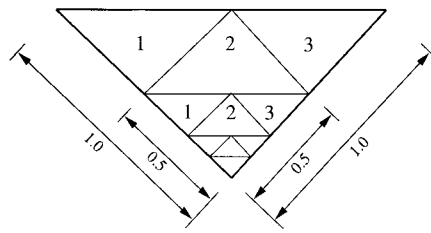
Figure 11. Refinement in the parameter space.

can be established from the refinement pattern in Figure 11. For the centre of sub-patch 2 we have the following relation:

$$\text{Triangle 2:} \quad \tilde{\theta}^1 = 1 - 2\theta^1, \quad \tilde{\theta}^2 = 1 - 2\theta^2 \tag{71}$$

The function values and derivatives for sub-patch 2 can now be evaluated using the interpolation rule

$$\mathbf{x}(\theta^1,\theta^2) = \sum_{I=1}^{12} N^I(\tilde{\theta}^1,\tilde{\theta}^2)\mathbf{P}_I\mathbf{A}\mathbf{X}^0 \tag{72}$$

Differentiation of the above equation yields the partial derivatives of $\mathbf{x}$ at the desired location. In these calculations, the Jacobian matrix of the simple mapping (72) between the co-ordinates (θ^1,θ^2) and $(\tilde{\theta}^1, \tilde{\theta}^2)$ has to be included. The result is

$$\mathbf{x},_\alpha(\theta^1,\theta^2) = -2\sum_{I=1}^{12} N^I,_\alpha(\tilde{\theta}^1,\tilde{\theta}^2)\mathbf{P}_I\mathbf{A}\mathbf{X}^0 \equiv \mathbf{a}_\alpha(\theta^1,\theta^2) \tag{73}$$

The Jacobian of the embedding required in the quadrature rule (45) can in turn be computed directly from $\mathbf{a}_\alpha$ through (7). Second-order derivatives such as required for the computation of bending strains follow simply by direct differentiation of the interpolation rule (73).

The evaluation procedure just described is a simplified version of the scheme presented by Stam [45] for function evaluation at arbitrary parameter values. Since we are only interested in a one-point quadrature rule, a single subdivision step is sufficient. For other rules one may need to subdivide further. Stam describes the general case in which the eigendecomposition of $\mathbf{A}$ is exploited to perform the implied subdivision steps through suitable powers of $\mathbf{A}$ in the eigenbasis, a simple and efficient procedure.

4.5. Dirichlet boundary conditions

Due to the local support of the subdivision scheme, the boundary displacements are influenced only by the nodal positions in the 1-neighbourhood of the boundary, i.e. the first layer of vertices inside the domain as well as a collection of artificial or 'ghost' vertices just outside the domain. For example, for built-in boundary conditions the displacements of the nodes inside, outside, and on the boundary must be zero (see Figure 12). The deformation of the boundary computed with the limit formula as described in Section 4.3 shows that this results in zero displacements and in fixed tangents. Other boundary conditions can be accommodated in a similar way (see Figure 12). The enforcement of prescribed boundary displacements is equally straightforward. General displacement
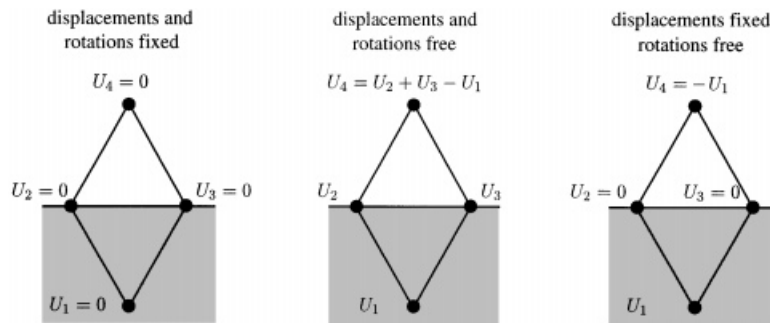
Figure 12. Boundary conditions.

boundary conditions may be formulated with the aid of a local reference frame defined by the surface normal and the tangent to the boundary of the shell. The prescribed displacement boundary conditions may be regarded as linear constraints on the displacement field and treated accordingly during the solution process. Such linear constraints may be enforced by a variety of methods [9]. In all the calculations reported here, the displacement boundary conditions are enforced by the penalty method, with a penalty stiffness equal to 100 times the maximum diagonal component of the stiffness matrix.

4.6. Implementation of subdivision-based shell elements

We close this section with selected remarks on matters of implementation. The implementation of the element requires a few data structures in addition to those typical of the conventional finite element method. For example, tables of element adjacencies are needed for the computation of element arrays. In our implementation, we use a `Triangle` C-structure which stores the following information for each element of the mesh:

```
typedef struct _T {
  /* pointer to the element vertices */
  /* pointer to the neighbor triangles */
  /* array of pointers to the vertices in the 1-neighborhood */
  /* number of nodes in the 1-neighborhood */
  /* information about the boundary conditions of the edges */
} Triangle;
```

The solution process consists of the following steps:

  (i) Subdivide the initial mesh once in order to separate the irregular vertices (the initial mesh could include triangles with more than one irregular vertex—in order to limit the number of possible mesh patterns during the parameter evaluation, we refine the mesh once so that each triangle has at most one irregular vertex).
 (ii) Generate artificial nodes and elements at the boundaries (see Section 4.2).
(iii) Find the 1-neighbourhood of each element and gather the local nodes in accordance with the local numbering convention.
 (iv) Create local co-ordinate systems at the nodes if necessary. For the creation of the local co-ordinate systems, we employ the limit formula of Equation (65) for the shell normal.

  (v) Compute and assemble the element stiffness matrices and force arrays.
 (vi) Introduce displacement boundary condition constraints (see Section 4.5).
(vii) Solve the system of equations.
(viii) Compute the limit positions of the nodes, Equation (60).

The elements under consideration here have three nodes and three displacement degrees of freedom per node. We use one-point quadrature with the sole quadrature point located at the barycentre of the element, i.e. at $\theta_G^1 = 1/3$ and $\theta_G^2 = 1/3$. The corresponding weight is $w = 1/2$, which, evidently, is the area of the standard triangle T.

   In sum, the preceding developments lead to the definition of a class of fully conforming 'C^1' triangular elements containing three nodes and one quadrature point. This combination of attributes, namely, the low order of interpolation and quadrature required, render the element particularly attractive from the standpoint of computational efficiency. As mentioned earlier, subdivision surfaces may also be used to define four-node square shell elements, but this extension of the method will not be pursued here.

5. EXAMPLES

We proceed to establish the convergence characteristics of the method by running the obstacle course of test cases proposed by Belytschko *et al.* [10]. The shells in these tests cases take simple spherical or cylindrical shapes which can be readily described analytically. A preliminary step in the calculations is to approximate the exact surface of the shell by a subdivision surface. Several methods of approximation are possible, including:

 (i) Least-squares approximation of the exact surface by the limit surface.
(ii) Placement of the control-mesh nodes on the exact surface.

and others. However, theoretical considerations and our own numerical tests show that the error incurred in the approximation of the shell geometry is of higher order than the finite-element error, and consequently both methods of approximation result in the same convergence rates asymptotically. It should also be noted that the question of geometrical approximation is rendered moot within an integrated computer-aided geometrical design (CAGD)—finite-element analysis framework. In this environment, the subdivision surface generated by the CAGD module becomes the true shell surface to be analysed by the finite-element analysis module.

   We note for further reference that a strictly C^1 finite-element method for Kirchhoff–Love shell theory containing the complete set of third-order polynomials within its interpolation satisfies the error bound [3]:

$$\|\mathbf{u}_h - \mathbf{u}\|_{2,\Omega} \leqslant \frac{C}{\mathrm{NP}} |\mathbf{u}|_{3,\Omega} \tag{74}$$

where $\mathbf{u}$ is the exact solution, $\mathbf{u}_h$ is the finite-element solution, C is a constant, NP is the number of nodes in the mesh, $\|\cdot\|_{2,\Omega}$ is the standard norm over $H^2(\Omega; R^3)$, or 'energy norm', and $|\cdot|_{3,\Omega}$ is the standard semi-norm over $H^3(\Omega; R^3)$.

   The central question to be ascertained now is whether the method developed in the foregoing exhibits the optimal convergence rate implied by the bound (74). All the calculations described subsequently are carried out with one-point quadrature. The successive mesh refinements considered in convergence studies are obtained by regular refinement.
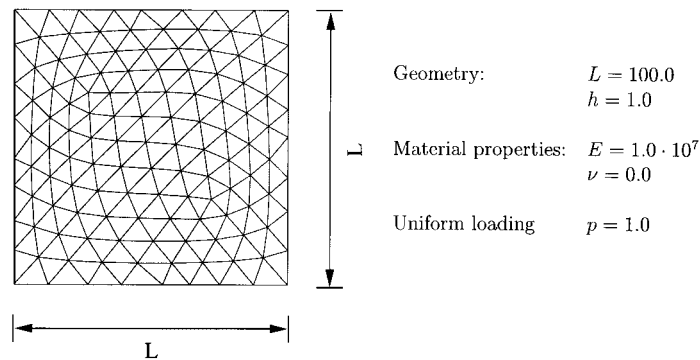
Figure 13. Definition of the plate test problem and a typical mesh used in the calculations.
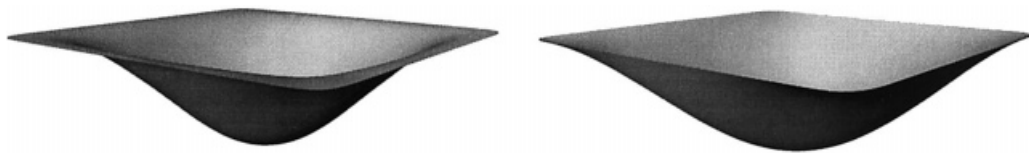

Figure 14. Calculated deformed shapes for the clamped and simply supported plates (deflections scaled differently in both cases).

We additionally compare the performance of the proposed approach with that of two other shell elements:

ASM: The assumed-strain four-noded shell element of Simo *et al.* [13].

DKT-CST: A flat three-node element with no membrane/bending coupling [46]. In this element the discrete Kirchhoff triangle (DKT) formulation of Batoz [47] is utilized for the bending response, and standard constant strain interpolation is used for the membrane response. The resulting element has six degrees of freedom per node.

5.1. Rectangular plate

As a first example, we consider the simple case of a square plate under uniform loading $p = 1$ (see Figure 13). The length of the plate is $L = 100$ and the thickness is $h = 1$. These dimensions place the plate well within the scope of Kirchhoff's theory. The Young's modulus is $E = 10^7$ and the Poisson's ratio is $v = 0$. In order to test the treatment of displacement boundary conditions described in Section 4.5, we analyse both a simply supported and a clamped plate. The entire plate is discretized into finite elements with no account taken of the symmetry of the plate. A typical mesh used in the calculations is shown in Figure 13. The artificial or ghost nodes used to enforce the boundary conditions are not shown in the figure.

The computed limit surfaces of the simply supported and clamped plates following deformation are shown in Figure 14. The high degree of smoothness of these deflected shapes is noteworthy. An appealing feature of the problem under consideration is that it is amenable to an exact analytical solution. For instance, the maximum displacement at the centre of the plate is found to be: $u_{max} \approx 0.487$ for the simply supported plate; and $u_{max} \approx 0.151$ for the clamped plate [1]. It
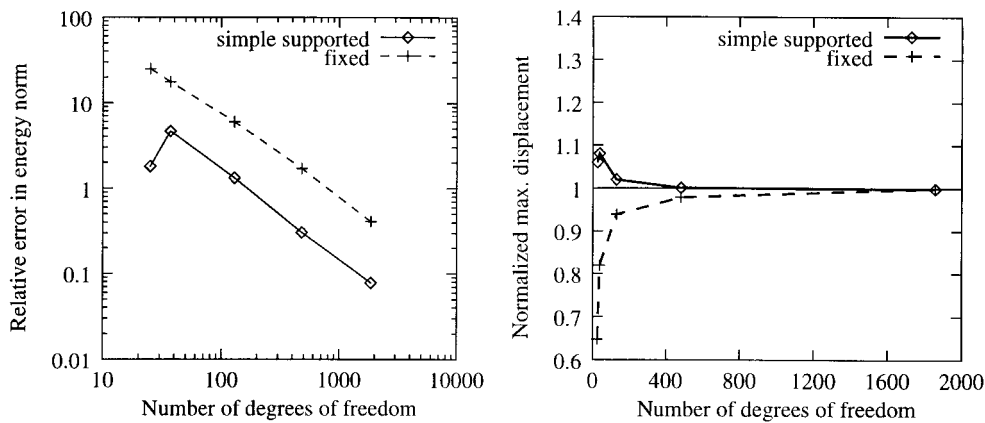
Figure 15. Convergence of the energy and maximum-displacement errors for the simply supported and clamped plate problems.
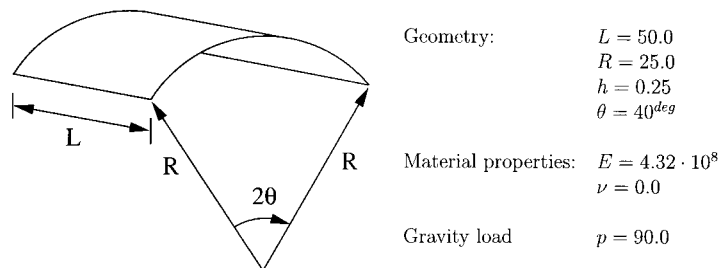

Figure 16. Definition of Scordelis-Lo problem.

therefore follows that the solution error can be computed exactly for this problem. The computed maximum-displacement and energy-norm errors are shown in Figure 15 as a function of the number of degrees of freedom. In all cases, the optimal convergence rate $O(NP^{-1})$ is attained in the energy norm, which attests to the good convergence properties of the method. These results, and those presented subsequently, also demonstrate the sufficiency of the one-point quadrature rule.

5.2. Scordelis-Lo Roof

The Scordelis-Lo Roof is a membrane-stress dominated problem and, as such, it provides a useful test of the ability of the finite-element element method to represent complex states of membrane strain. The problem concerns an open cylindrical shell loaded by gravity forces, (see Figure 16). In our calculations, the length of the cylinder is $L = 50$; its radius is $R = 25$; the angle subtended by the roof is $\phi = 80°$; the thickness is $h = 0.25$; the Young's modulus is $E = 4.32 \times 10^8$; and the Poisson's ratio is $v = 0$.

The undeformed control mesh and the deformed limit surface are shown in Figure 17. A convergence plot for the maximum displacement is shown in Figure 18. The displacements are normalized by the value 0.3024 as given in Reference [10]. The excellent convergence characteristics of the
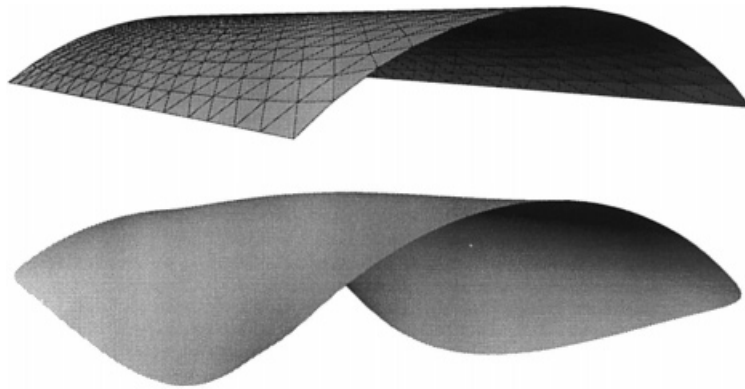
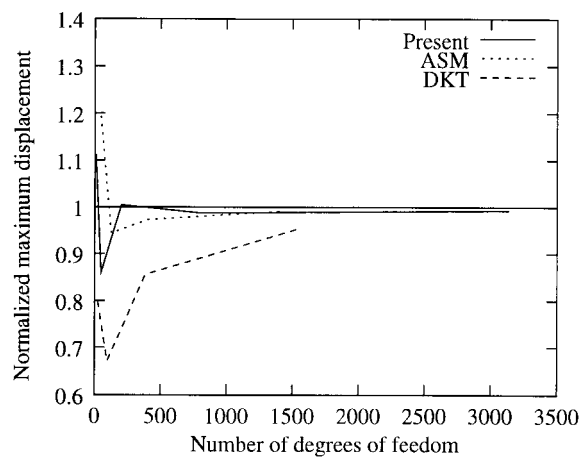Figure 17. Scordelis-Lo Roof: typical control mesh; deformed limit surface.

Figure 18. Displacement-convergence plot for Scordelis-Lo Roof.

method are evident from the figure. In particular, the subdivision element outperforms both the assumed-strain and the DKT elements.

5.3. Pinched cylinder

The pinched-cylinder problem tests the method's ability to deal with inextensional bending modes and complex membrane states. The problem concerns a cylindrical shell pinched under the action of two diametrically opposite unit loads located within the middle section of the shell. We consider two cases: free-end boundary conditions; and ends constrained by two rigid diaphragms. The length of the cylinder is $L = 600$; the radius is $R = 300$; the thickness is $h = 3$; the Young's modulus is $E = 3 \times 10^6$; and the Poisson's ratio is $v = 0.3$. A control-mesh node is placed at the point of application of the loads. It is interesting to note, however, that, owing to the non-local character of the shape functions, the point load is spread over several nodes. This is in contrast to other
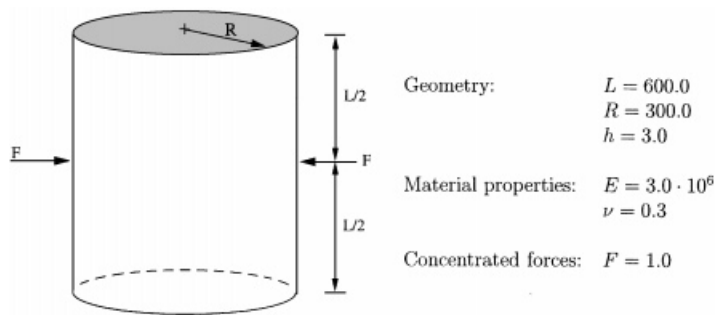
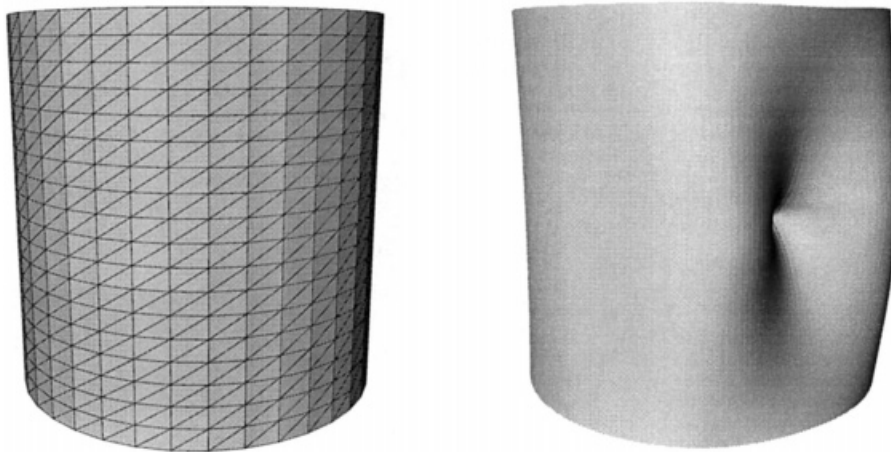Figure 19. Definition of the pinched-cylinder problem.

Figure 20. Pinched-cylinder problem: typical control mesh; deformed limit surface.

methods, e.g. those based on Hermitian interpolation. It should be carefully noted that the shape functions developed here possess the requisite partition-of-unity property and, in consequence, the resultant of all nodal forces exactly matches the applied load.

The computed deformed limit surface is shown in Figure 20. Here again, the high degree of smoothness of the deformed surface, which attests to the conforming nature of the method, is noteworthy. Displacement-convergence results are shown in Figure 21. In the diaphragm case, we monitor the displacements under the loads, which are normalized by the analytical solution of 1.82488×10^{-5}. The excellent convergence properties of the method are evident from the figure. As may be seen, the subdivision element converges faster than both the assumed strain and the DKT elements. The analytical solution for the free-end case has been given by Timoshenko [1]. Timoshenko's solution gives a value of 4.520×10^{-4} for the displacements under the loads, and a value of 4.156×10^{-4} for the change in diameter at the free ends. The convergence of this latter quantity is shown in Figure 21. Here again, the robust convergence characteristics of the numerical solution is noteworthy.
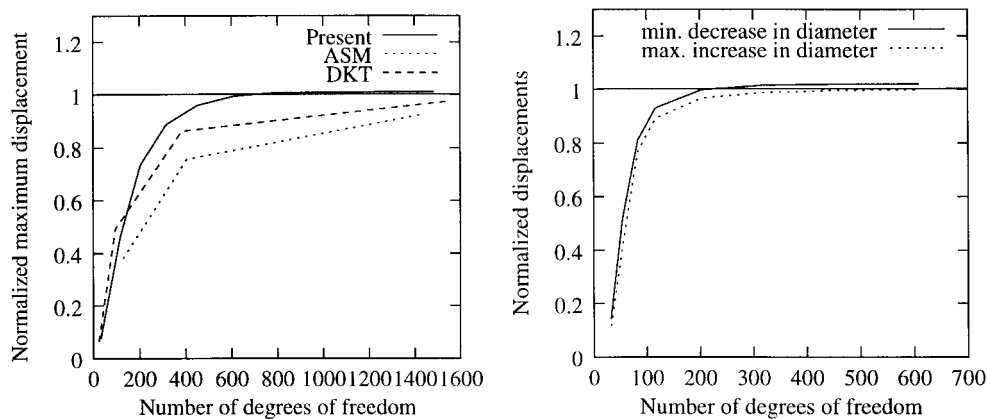
Figure 21. Displacement convergence plots for pinched-cylinder problem: rigid diaphragm case; free-end case.

5.4. Hemispherical Shell

The case of a spherical shell provides an example of a surface which cannot be triangulated without the inclusion of irregular nodes, or nodes of valence different from 6. Under these conditions, a naive implementation of a box-spline-based finite-element method necessarily breaks down, which illustrates the need for the more general treatment developed here. We consider a shell of hemispherical shape deforming under the action of four point loads acting on its edge. The radius of the hemisphere is $R = 10$; the thickness is $h = 0.04$; Young's modulus is $E = 6.825 \times 10^7$; and Poisson's ratio is $v = 0.3$. The edge of the shell is free. The applied loads have a magnitude $F = 2$ and define two pairs of diametrically opposite loads alternating in sign at $90°$ (see Figure 22). A typical control mesh and the corresponding deformed limit surface are shown in Figure 23. The presence of irregular nodes in the control mesh should be carefully noted. The convergence of the radial displacement under the applied loads is shown in Figure 24. The displacements are normalized by the exact solution 0.0924 as given by Reference [10]. The hemispherical shell is a standard test for assessing an element's ability to represent inextensional deformations. Critical to the performance of the element is its ability to bend without developing parasitic membrane strains. Accordingly, the flat DKT element with uncoupled membrane and bending stiffness performs particularly well in the pinched-hemisphere shell problem. However, a careful examination of Figure 24 reveals that the subdivision element attains convergence faster than even the DKT element. The excellent convergence characteristics of the method are again evident from the figure.

6. SUMMARY AND CONCLUSIONS

We have presented a new paradigm for thin-shell finite-element analysis based on the use of subdivision surfaces for (i) describing the geometry of the shell in its undeformed configuration, and (ii) generating smooth interpolated displacement fields possessing bounded energy within the strict framework of the Kirchhoff–Love theory of thin shells. Several salient attributes of the

Geometry: $\qquad$ $R = 10.0$
$\qquad\qquad\qquad\qquad\quad h = 0.04$

Material properties: $\quad E = 6.825 \cdot 10^{7}$
$\qquad\qquad\qquad\qquad\quad \nu = 0.3$

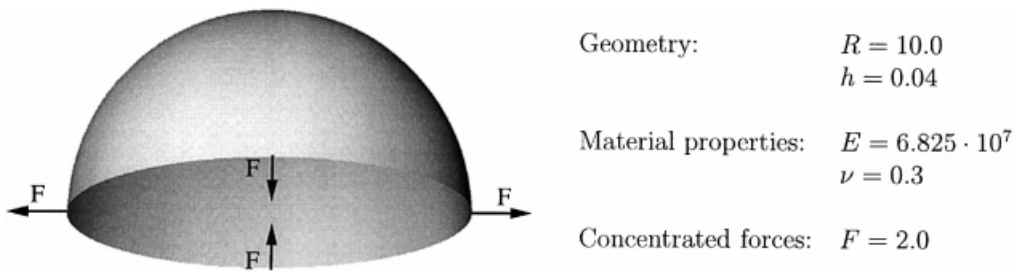Concentrated forces: $\quad F = 2.0$

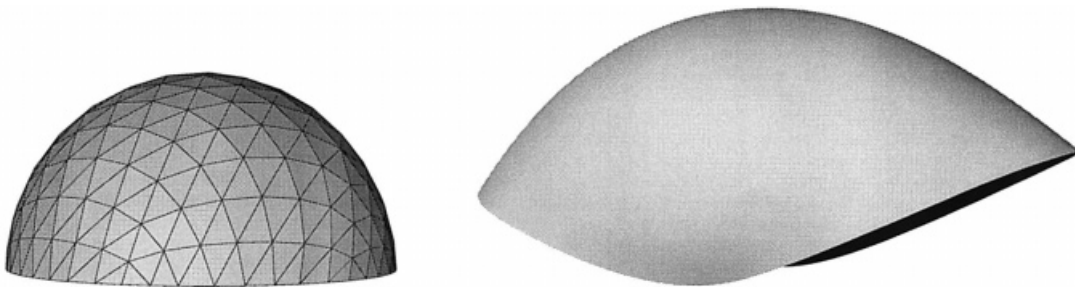Figure 22. Definition of the pinched-hemisphere problem.

Figure 23. Pinched-hemisphere problem: control mesh; deformed limit surface.

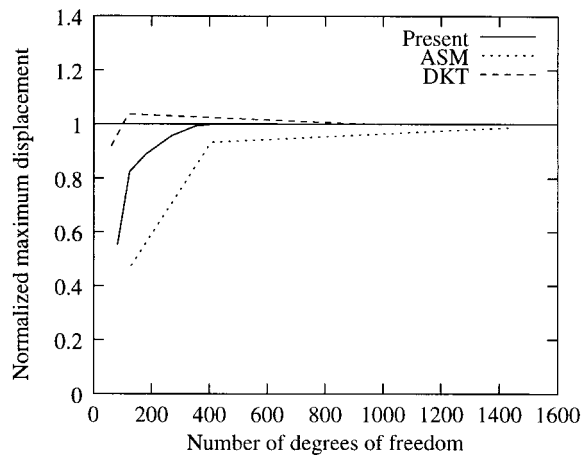Figure 24. Pinched-hemisphere problem: convergence of the normalized displacements.

proposed interpolation scheme bear emphasis:

 (i) The undeformed and deformed surfaces of the shell, or equivalently the displacement field
    thereof, follow by the recursive application of local subdivision rules to nodal data defined
    on a *control triangulation* of the surface. The particular subdivision strategy adopted here is

Loop's scheme. Whether directly at regular nodes of valency 6, or following an appropriate number of subdivision steps in the case of irregular nodes, the limit surface can be described locally by quartic box splines. Subdivision rules may also be defined for square meshes. The subdivision rules may also be generalized to account for creases and displacement boundary conditions.

(ii) The limit surfaces obtained by subdivision, and the displacement fields they support, are C^2 except at isolated extraordinary points. The displacement field is guaranteed to be H^2, i.e. to be square-summable and to have first and second square-summable derivatives. In consequence, the interpolated displacement fields have a finite Kirchhoff–Love energy. In the parlance of the finite-element method, the proposed interpolation scheme is strictly C^1 and therefore meets the convergence requirements of the displacement finite-element method.

(iii) The triangles in the control mesh induce subregions on the limit surface which may be regarded as *bona fide* finite elements. In particular, the energy—and all other extensive properties—of the shell may be computed as a sum of integrals extended to the domains of the elements. These element integrals may conveniently be evaluated by numerical quadrature without compromising the order of convergence of the method. A one-point quadrature rule is sufficient for the computation of element integrals resulting from Loop's subdivision scheme.

(iv) The displacement field of the shell is interpolated from nodal displacements only. In particular, no nodal rotations are used in the interpolation. However, the interpolation scheme developed here differs from conventional C^0 finite-element interpolation in a crucial respect: the displacement field over an element depends on the displacements at the three nodes of the element *and* on the displacements of the 1-ring of nodes connected to the element. In this sense, the interpolation rule is *nonlocal*. However, the use of subdivision surfaces ensures that all the local displacement fields defined over overlapping patches combine conformingly to define one single limit surface.

In sum, subdivision surfaces enable the finite-element analysis of thin shells within the strict confines of Kirchhoff–Love theory while meeting all the convergence requirements of the displacement finite-element method. In particular the ability to couch the analysis within the framework of Kirchhoff–Love theory entirely sidesteps the difficulties associated with the use of C^0 methods in the limit of very thin shells. In a particularly pleasing way, subdivision surfaces enable the return to the most basic—and fundamental—of finite element approaches, namely, constrained energy minimization over a suitable subspace of interpolated displacement fields, or Rayleigh–Ritz approximation. Finite-element methods formulated in accordance with this prescription satisfy the orthogonality property, i.e. the error function is orthogonal to the space of finite-element interpolants; and possess the best-approximation property, i.e., the energy norm of the error is minimized by the finite-element solution. These properties render the basic finite-element method exceedingly robust and account for much of its success. Our numerical experiments show that the approach proposed here does indeed lead to the optimal convergence rate predicted by finite-element theory.

Another key advantage afforded by the approach developed here is that subdivision surfaces provide a common representational paradigm for both solid modeling and shell analysis, with the attendant unification of traditionally heterogeneous software tools. By virtue of this unification, surface geometries generated by a computer-aided geometry design (CAGD) module can be directly utilized by the shell-analysis module without the need for any intervening geometrical manipulation. As a consequence, high-level algorithms developed in the field of computer aided geometric design can be integrated simply into the shell analysis software.

In closing, a number of possible extensions of the theory are worth mentioning. Firstly, recursive subdivision provides an effective basis for mesh adaption. By retaining the hierarchy of finite-element representations generated by subdivision, the application of multiresolution methods and related techniques, such as wavelets, becomes straightforward. Indeed, the application of wavelet methods to the description of complex and intricate geometries has already been extensively pursued within the field of computer graphics [48]. Finally, the extension of the proposed approach to the non-linear range appears straightforward. In this particular context, the sole use of nodal displacements in the interpolation is expected to simplify the solution procedure by eliminating the need for introducing complex schemes for the nonsingular parametrization of the shell director.

APPENDIX

A.1. Regular patches

For regular patches the shape functions are given by the 12 box-spline basis functions [28]. The local numbering of the nodes adopted here is as in Figure 9.

$$
\begin{aligned}
N_1 &= \tfrac{1}{12}(u^4 + 2u^3 v) \\
N_2 &= \tfrac{1}{12}(u^4 + 2u^3 w) \\
N_3 &= \tfrac{1}{12}(u^4 + 2u^3 w + 6u^3 v + 6u^2 vw + 12u^2 v^2 + 6uv^2 w + 6uv^3 + 2v^3 w + v^4) \\
N_4 &= \tfrac{1}{12}(6u^4 + 24u^3 w + 24u^2 w^2 + 8uw^3 + w^4 + 24u^3 v + 60u^2 vw + 36uvw^2 \\
&\quad + 6vw^3 + 24u^2 v^2 + 36uv^2 w + 12v^2 w^2 + 8uv^3 + 6v^3 w + v^4) \\
N_5 &= \tfrac{1}{12}(u^4 + 6u^3 w + 12u^2 w^2 + 6uw^3 + w^4 + 2u^3 v + 6u^2 vw + 6uvw^2 + 2vw^3) \\
N_6 &= \tfrac{1}{12}(2uv^3 + v^4) \\
N_7 &= \tfrac{1}{12}(u^4 + 6u^3 w + 12u^2 w^2 + 6uw^3 + w^4 + 8u^3 v + 36u^2 vw + 36uvw^2 + 8vw^3 \\
&\quad + 24u^2 v^2 + 60uv^2 w + 24v^2 w^2 + 24uv^3 + 24v^3 w + 6v^4) \\
N_8 &= \tfrac{1}{12}(u^4 + 8u^3 w + 24u^2 w^2 + 24uw^3 + 6w^4 + 6u^3 v + 36u^2 vw + 60uvw^2 \\
&\quad + 24vw^3 + 12u^2 v^2 + 36uv^2 w + 24v^2 w^2 + 6uv^3 + 8v^3 w + v^4) \\
N_9 &= \tfrac{1}{12}(2uw^3 + w^4) \\
N_{10} &= \tfrac{1}{12}(2v^3 w + v^4) \\
N_{11} &= \tfrac{1}{12}(2uw^3 + w^4 + 6uvw^2 + 6vw^3 + 6uv^2 w + 12v^2 w^2 + 2uv^3 + 6v^3 w + v^4) \\
N_{12} &= \tfrac{1}{12}(w^4 + 2vw^3)
\end{aligned}
\tag{A1}
$$

where the barycentric co-ordinates (u, v, w) are subject to the constraint

$$u + v + w = 1 \tag{A2}$$

The local curvilinear co-ordinates (θ^1, θ^2) for the element may be identified with the barycentric co-ordinates (v, w).

A.2. Irregular patches

As discussed in Section 4.4, a closed-form representation for the shape functions is not available at irregular vertices. The shell surface within each element is, however, completely described by the nodal positions of the element and its 1-ring. For the one-point quadrature rule used in calculations, the regular patch configuration is recovered after the application of one single subdivision step. In the following we give a more general version of the function evaluation scheme for arbitrary parameter values. The number of subdivision steps required and the attendant co-ordinate transformations can be computed by the following algorithm due to Stam [28]:

```
/* determine the necessary number of subdivisions */
u = v+w;
k = -log10(u)/log10(2.0);
k = ceil(k);
pow2 = pow(2.0, (double)(k-1));
/* determine in which domain (v,w) lies */
v *= pow2; w *= pow2;
if (v > 0.5) {
  v = 2.0*v-1.0;   w = 2.0*w;
  whpa = 1;
} else if (w > 0.5) {
  v = 2.0*v;   w = 2.0*w-1.0;
  whpa = 3;
} else {
  v = 1.0 - 2.0*v;   w = 1.0-2.0*w;
  whpa = 2;
}
```

The function value at any parameter value is given by

$$\mathbf{x}(\theta^1, \theta^2) = \sum_{I=1}^{NP} N^I(\tilde{\theta}^1, \tilde{\theta}^2) \mathbf{P}_I \mathbf{A}^k \mathbf{X}^0 \tag{A3}$$

which generalizes Equation (72). Here again, the vectors $\mathbf{P}_I$ extract the control nodes of the box-spline patch. The variable `whpa` in the above code gives the number of the subpatch, which contains the coordinates (v, w), and the variable `k` the power of the matrix $\mathbf{A}$. For Loop's scheme, the subdivision matrix $\mathbf{A}$ has the following form:

$$\mathbf{A} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{0} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{pmatrix} \tag{A4}$$

with

$$\mathbf{S}_{11} = \frac{1}{8} \begin{pmatrix} 8 - 8nw & 8w & 8w & 8w & \dots & 8w \\ 3 & 3 & 1 & 0 & \dots & 1 \\ 3 & 1 & 3 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 3 & 1 & 0 & \dots & 1 & 3 \end{pmatrix}$$

$$\mathbf{S}_{21} = \frac{1}{16} \begin{pmatrix} 2 & 6 & 0 & 0 & \cdots & 0 & 0 & 6 \\ 1 & 10 & 1 & 0 & \cdots & 0 & 0 & 1 \\ 2 & 6 & 6 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & \cdots & 0 & 1 & 10 \\ 2 & 0 & 0 & 0 & \cdots & 0 & 6 & 6 \\ 0 & 6 & 0 & 0 & \cdots & 0 & 0 & 2 \\ 0 & 6 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 6 & 2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & \cdots & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 2 & 6 \end{pmatrix}, \quad \mathbf{S}_{22} = \frac{1}{16} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 \\ 6 & 2 & 0 & 0 & 0 \\ 2 & 6 & 2 & 0 & 0 \\ 0 & 2 & 6 & 0 & 0 \\ 6 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 6 & 2 \\ 0 & 0 & 0 & 2 & 6 \end{pmatrix}$$

It should be noted that the dimensions of the matrices $\mathbf{S}_{11}$ and $\mathbf{S}_{21}$ depend on the valence of the irregular vertex.

A.3. Membrane and bending strain matrices

The membrane and bending-strain matrices take the form

$$\mathbf{M}^I = \begin{bmatrix} N^I,_1 \, \mathbf{a}_1 \cdot \mathbf{e}_1 & N^I,_1 \, \mathbf{a}_1 \cdot \mathbf{e}_2 & N^I,_1 \, \mathbf{a}_1 \cdot \mathbf{e}_3 \\ N^I,_2 \, \mathbf{a}_2 \cdot \mathbf{e}_1 & N^I,_2 \, \mathbf{a}_2 \cdot \mathbf{e}_2 & N^I,_2 \, \mathbf{a}_2 \cdot \mathbf{e}_3 \\ (N^I,_2 \, \mathbf{a}_1 + N^I,_1 \, \mathbf{a}_2) \cdot \mathbf{e}_1 & (N^I,_2 \, \mathbf{a}_1 + N^I,_1 \, \mathbf{a}_2) \cdot \mathbf{e}_2 & (N^I,_2 \, \mathbf{a}_1 + N^I,_1 \, \mathbf{a}_2) \cdot \mathbf{e}_3 \end{bmatrix} \quad \text{(A5)}$$

and

$$\mathbf{B}^I = \begin{bmatrix} \mathbf{B}_1^I \cdot \mathbf{e}_1 & \mathbf{B}_1^I \cdot \mathbf{e}_2 & \mathbf{B}_1^I \cdot \mathbf{e}_3 \\ \mathbf{B}_2^I \cdot \mathbf{e}_1 & \mathbf{B}_2^I \cdot \mathbf{e}_2 & \mathbf{B}_2^I \cdot \mathbf{e}_3 \\ \mathbf{B}_3^I \cdot \mathbf{e}_1 & \mathbf{B}_3^I \cdot \mathbf{e}_2 & \mathbf{B}_3^I \cdot \mathbf{e}_3 \end{bmatrix} \quad \text{(A6)}$$

respectively. In the above expressions, $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ are the basis vectors of an orthonormal-coordinate reference frame, and

$$\mathbf{B}_1^I = -N^I,_{11} \, \mathbf{a}_3 + \frac{1}{\sqrt{a}}[N^I,_1 \, \mathbf{a}_{1,1} \times \mathbf{a}_2 + N^I,_2 \, \mathbf{a}_1 \times \mathbf{a}_{1,1} + \mathbf{a}_3 \cdot \mathbf{a}_{1,1}(N^I,_1 \, \mathbf{a}_2 \times \mathbf{a}_3 + N^I,_2 \, \mathbf{a}_3 \times \mathbf{a}_3)]$$

$$\mathbf{B}_2^I = -N^I,_{22} \, \mathbf{a}_3 + \frac{1}{\sqrt{a}}[N^I,_1 \, \mathbf{a}_{2,2} \times \mathbf{a}_2 + N^I,_2 \, \mathbf{a}_1 \times \mathbf{a}_{2,2} + \mathbf{a}_3 \cdot \mathbf{a}_{2,2}(N^I,_1 \, \mathbf{a}_2 \times \mathbf{a}_3 + N^I,_2 \, \mathbf{a}_3 \times \mathbf{a}_3)]$$

$$\mathbf{B}_3^I = -N^I,_{12} \, \mathbf{a}_3 + \frac{1}{\sqrt{a}}[N^I,_1 \, \mathbf{a}_{1,2} \times \mathbf{a}_2 + N^I,_2 \, \mathbf{a}_1 \times \mathbf{a}_{1,2} + \mathbf{a}_3 \cdot \mathbf{a}_{1,2}(N^I,_1 \, \mathbf{a}_2 \times \mathbf{a}_3 + N^I,_2 \, \mathbf{a}_3 \times \mathbf{a}_3)]$$

REFERENCES

1. Timoshenko S. *Theory of Plates and Shells*. McGraw-Hill Book Company: New York, 1940.
2. Green AE, Zerna W. *Theoretical Elasticity* (2 edn). Oxford University Press: England, 1968.
3. Ciarlet PG. *The Finite Element Method for Elliptic Problems*. North-Holland: Amsterdam, 1978.
4. Zienkiewicz OC, Taylor RL. *The Finite Element Method*, vol. 2. McGraw-Hill Book Company: Berkshire, 1989.
5. Argyris JH, Scharpf DW. The sheba family of shell elements for the matrix displacement method. *Aeronautics Journal of Royal Aeronautical Society* 1968; **71**:873–883.
6. Bell K. A refined triangular plate bending finite element. *International Journal for Numerical Methods in Engineering* 1969; **1**:101–122.
7. Ashwell DG, Gallagher RH (eds), *Finite Elements for Thin Shells and Curved Members*. Wiley: London, 1976.
8. Hughes TJR, Liu WK. Nonlinear finite element analysis of shells: Part ii. Two dimensional shells. *Computer Methods in Applied Mechanics and Engineering* 1981; **27**:167–182.
9. Carey GF, Oden JT. *Finite Elements—Computational Aspects*. Prentice-Hall: Englewood Cliffs, NJ, 1984.
10. Belytschko T, Stolarski H, Liu WK, Carpenter N, Ong JSJ. Stress projection for membrane and shear locking in shell finite-elements. *Computer Methods in Applied Mechanics and Engineering* 1985; **51**:221–258.
11. Hughes TJR. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1987.
12. Simo JC, Fox DD. On a stress resultant geometrically exact shell model. Part i: formulation and optimal parameterization. *Computer Methods in Applied Mechanics and Engineering* 1989; **72**:267–304.
13. Simo JC, Fox DD, Rifai MS. On a stress resultant geometrically exact shell model. Part ii: the linear theory; computational aspects. *Computer Methods in Applied Mechanics and Engineering* 1989; **73**:53–92.
14. Simo JC, Fox DD, Rifai MS. On a stress resultant geometrically exact shell model. Part iii: computational aspects of the nonlinear theory. *Computer Methods in Applied Mechanics and Engineering* 1990; **79**:21–70.
15. Simo JC, Kennedy JG. On a stress resultant geometrically exact shell model. part v. nonlinear plasticity: formulation and integration algorithms. *Computer Methods in Applied Mechanics and Engineering* 1992; **96**:133–171.
16. Büchter N, Ramm E. Shell theory versus degeneration—a comparison in large rotation finite element analysis. *International Journal for Numerical Methods in Engineering* 1992; **34**:39–59.
17. Andelfinger U, Ramm E. Eas-elements for two dimensional, three dimensional, plate- and shell structures and their equivalence to hr-elements. *International Journal for Numerical Methods in Engineering* 1993; **36**:1311–1337.
18. Roehl D, Ramm E. Large elasto-plastic finite element analysis of solids and shells with the enhanced assumed strain concept. *International Journal of Solids and Structures* 1996; **33**:3215–3237.
19. Wriggers P, Eberlein R, Reese S. A comparison of 3-dimensional continuum and shell elements for finite plasticity. *International Journal of Solids and Structures* 1996; **33**:3309–3326.
20. Bucalem ML, Bathe KJ. Finite element analysis of shell structures. *Archives of Computer Methods in Engineering* 1997; **4**:3–61.
21. MacNeal RH. Perspective on finite elements for shell analysis. *Finite Element Analysis Design* 1998; **30**:175–186.
22. Catmull E, Clark J. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer Aided Design* 1978; **10**(6):350–355.
23. Doo D, Sabin M. Behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design* 1978; **10**(6):356–360.
24. Loop C. Smooth subdivision surfaces based on triangles. *Master's Thesis*, University of Utah, Department of Mathematics, 1987.
25. Dyn N, Levin D, Gregory JA. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions of the Graphics* 1990; **9**(2):160–169.
26. Reif U. A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design* 1995; **12**(2):153–174.
27. Zorin D. A method for analysis of c^1- continuity of subdivision surfaces. *SIAM Journal of Numerical Analysis* 2000, in print.
28. Stam J. Fast evaluation of loop triangular subdivision surfaces at arbitrary parameter values. In *Computer Graphics* (*SIGGRAPH '98 Proceedings, CD-ROM supplement*), 1998.
29. Rojek J, Onate E, Postek E. Application of explicit fe codes to simulation of sheet and bulk metal forming processes. *Journal of the Materials Processing Technology* 1998; **80**:620–627.
30. Kagan P, Fischer A, Bar-Yoseph PZ. New b-spline finite element approach for geometrical design and mechanical analysis. *International Journal for Numerical Methods in Engineering* 1998; **41**:435–458.
31. Kobbelt L, Hesse T, Prautzsch H, Schweizerhof K. Iterative mesh generation for fe-computations on free form surfaces. *Engineering Computation* 1997; **14**:806–820.
32. Naghdi PM. The theory of shells. *Handbuch der Physik, Mechanics of Solids II*, vol. VI a/2. Springer: Berlin, 1972.
33. Basar Y, Krätzig WB. *Mechanik der Flächentragwerke*. Friedr. Vieweg & Sohn: Braunschweig, 1985.
34. Strang G, Fix GJ. *An Analysis of the Finite Element Method*. Prentice-Hall: Englewood Cliffs, NJ, 1973.
35. Schweitzer JE. Analysis and Application of Subdivision Surfaces. *Ph.D. Dissertation*, Department of Computer Science and Engineering, University of Washington, Seattle, 1996.

Copyright © 2000 John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Engng. 2000; **47**:2039–2072

36. Zorin D, Schröder P (eds), *Subdivision for Modeling and Animation*, Computer Graphics (*SIGGRAPH '99 Course Notes*), 1999.
37. Zorin D, Schröder P, Sweldens W. Interpolating subdivision for meshes with arbitrary topology. In *Computer Graphics* (*SIGGRAPH '96 Proceedings*), 1996.
38. Reif U, Schröder P. Curvature smoothness of subdivision surfaces, submitted for publication.
39. Deslauriers G, Dubuc S. Interpolation dyadique. In *Fractals, Dimensions Non Entières et Applications*. Masson: Paris, 1987; 44–45.
40. Hoppe H, DeRose T, Duchamp T, Halstead M, Jin H, McDonald J, Schweitzer J, Stuetzle W. Piecewise smooth surface reconstruction. In *Computer Graphics* (*SIGGRAPH '94 Proceedings*), 1994.
41. Biermann H, Levin D, Zorin D. Smooth subdivision surfaces with normal control. *Preprint*, Courant Institute of Mathematical Sciences, New York, 1999.
42. Warren J. Subdivision methods for geometric design. Unpublished manuscript, Department of Computer Science, Rice University, November 1995.
43. Kobbelt L, Campagna S, Seidel HP. A general framework for mesh decimation. In *Graphics Interface 98*, Robins Southern Printing: Vancouver, BC, 1998; 43–50.
44. Lee A, Sweldens W, Schröder P, Cowsar L, Dobkin D. Maps: multiresolution adaptive parameterization of surfaces. In *Computer Graphics* (*SIGGRAPH '98 Proceedings*), 1998.
45. Stam J. Fast evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Computer Graphics* (*SIGGRAPH '98 Proceedings*), 1998.
46. Haußer C. Effiziente Dreieckselemente für Flächentragwerke. *Ph.D. Dissertation*, Institut for Structural Mechanics, University Stuttgart, Stuttgart, Germany, 1996.
47. Batoz JL. An explicit formulation for an efficient triangular plate-bending element. *International Journal for Numerical Methods in Engineering* 1982; **18**:1077–1089.
48. Zorin D, Schröder P, Sweldens W. Interactive multiresolution mesh editing. In *Computer Graphics* (*SIGGRAPH '97 Proceedings*), 1997.