



Subdivision shells with exact boundary control and non-manifold geometry

Fehmi Cirak and Quan Long

Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, U.K.

SUMMARY

We introduce several new extensions to subdivision shells that provide an improved level of shape control over shell boundaries and facilitate the analysis of shells with non-smooth and non-manifold joints. To this end, modified subdivision schemes are used that enable to relax the continuity of the limit surface along prescribed crease edges and to create surfaces with prescribed limit positions and normals. Furthermore, shells with boundaries in form of conic sections, such as circles or parabolas, are represented with rational subdivision schemes which are defined in analogy to rational b-splines. In terms of implementation, the difference between the introduced and conventional subdivision schemes is restricted to the use of modified subdivision stencils close to the mentioned geometric features. Hence, the resulting subdivision surface is in most parts of the domain identical to standard smooth subdivision surfaces. The particular subdivision scheme used in this paper constitutes an improved version of the original Loop's scheme and is as such based on triangular meshes. As in the original subdivision shells, surfaces created with the modified scheme are used for interpolating the reference and deformed shell configurations. At the integration points, the subdivision surface is evaluated using a newly developed discrete parameterisation approach. In the resulting finite elements the only degrees of freedom are the mid-surface displacements of the nodes and additional Lagrange parameters for enforcing normal constraints. The versatility of the newly developed elements is demonstrated with a number of geometrically nonlinear shell examples.

KEY WORDS: subdivision interpolation, rational subdivision, non-manifold geometry, shells, isogeometric analysis

*Correspondence to: f.cirak@eng.cam.ac.uk

1. Introduction

The unification of shape parameterisation techniques used in Computer Aided Design (CAD) and the discretization techniques used in finite element analysis gained lately momentum with the introduction of the isogeometric analysis paradigm [26, 14]. The basic idea of isogeometric analysis is to use the same set of shape, or basis, functions for geometric modelling and finite element analysis. In practice, one uses for finite element analysis shape functions from CAD since the conventional finite element shape functions are not sufficiently versatile for geometric modelling. One of the promises of isogeometric analysis is that a unified representation may facilitate the integration of the presently disparate geometric modelling and analysis tools. In addition, in comparison to conventional Lagrangian finite element shape functions spline based CAD shape functions have a number of superior approximation properties, such as positivity, variation diminishing property, smoothness and refinability [40, 19, 41]. In the spirit of isogeometric analysis, in [8, 11] the subdivision surfaces were proposed for geometric modelling and analysis of thin shells. In this paper we introduce a number of extensions to [8, 11] which enable to exactly control the geometric shape of the boundary and to exactly prescribe displacement and rotational boundary conditions. The improved level of control over shell boundaries is used for the analysis of shells with non-smooth and non-manifold features, such as T-joints, and conic sections as boundaries.

Subdivision is a powerful geometric modelling technique for generating smooth curves and surfaces on meshes with arbitrary connectivity. From an algorithmic viewpoint, subdivision schemes create a smooth curve or surface starting from a coarse *control mesh* by repeated *refinement* and *averaging*. It can be shown that the subdivided control mesh converges to a smooth limit curve or surface if certain restrictions on the choice of the refinement and averaging operators are made. Although subdivision techniques are used in computer graphics since the seventies [18, 6], their origin goes back as early as to the work of de Rham [16] in the forties. For computer graphics purposes subdivision schemes are generally considered only from an algorithmic viewpoint and in applications usually a very fine faceted representation is used as a proxy for the smooth limit surface. Alternatively, subdivision schemes can also be considered as the generalisation of spline surfaces to arbitrary connectivity meshes [41, 38], in which case most of the available know-how on splines can be transferred to subdivision. In fact, the most widely used subdivision schemes are on regular meshes identical to tensor product b-spline surfaces. However, subdivision schemes lead to smooth limit surfaces even when the control mesh has arbitrary connectivity. This is a clear advantage over conventional shape representation techniques based on b-splines or non-uniform rational b-splines (NURBS).

The original subdivision surfaces, such as the Doo-Sabin [18], Catmull-Clark [6] and Loop [33] schemes, were designed to give smooth surfaces on arbitrary connectivity meshes for closed objects with no boundaries. Later, several ad-hoc extensions were introduced in order to deal with boundaries and non-smooth features, such as creases, [24, 45, 17]. Most of these extensions were motivated by computer graphics or animation applications, which usually have a more lenient shape control and accuracy requirements than geometric design. Based on the initial ad-hoc extensions, Biermann et al. [3] introduced a comprehensive set of extended subdivision schemes with the corresponding convergence and smoothness analysis. The Biermann et al. extensions enable, amongst others, interpolation of points and/or normals along the domain boundaries and the interpolation of selected points and/or normals within the domain. The refined control over domain boundaries facilitates the representation of non-manifold surfaces, such as T-joints, which require that several surface patches are attached along their common boundary [53].

In addition to being a versatile geometry representation technique, smooth subdivision surfaces are also ideal for discretizing fourth order boundary value problems. In [8, 11] shape functions derived from subdivision surfaces have been used for discretizing the Kirchhoff-Love shell energy functional, which is a function of the first and second order derivatives of the deformed configuration. In the resulting finite elements, referred to as subdivision shells, the reference and deformed geometries are represented with subdivision surfaces in line with the isogeometric approach. During the finite element solution procedure the only degrees of freedom are the nodal positions of the control mesh in the deformed configuration and no rotation degrees-of-freedom are used. Such, so-called, rotation-free shell elements are very appealing for problems with large rigid body rotations and have also been developed using alternative approaches, see e.g. [2, 29, 35, 39]. In order to analyse structures with non-manifold joints and creases using rotation-free shell elements Lagrange parameters and bending strips have been used [21, 28]. Subdivision shells have

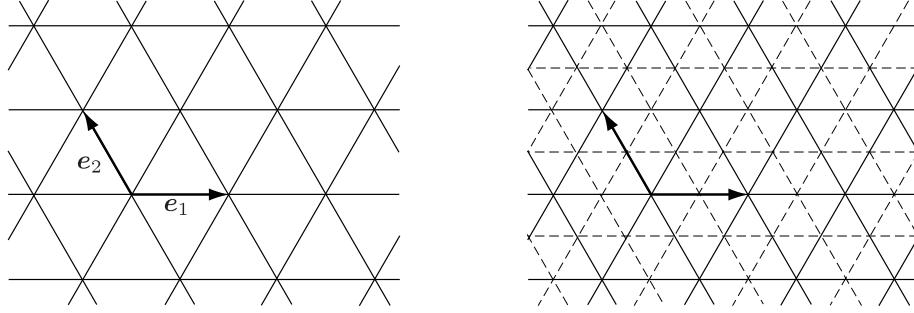


Figure 1. The left figure shows the tiling of the plane with equilateral triangles and an edge-aligned non-orthogonal coordinate system for parameterising the plane. The shown mesh is also known as a three-direction mesh. Note that all the vertices have integer values as coordinates. The right figure shows the refinement of the coarse tiling by bisection of edges and quadrisecion of triangles. The triangles on the coarse and fine tiling are congruent to each other.

so far been applied to a range of shell and membrane applications, including large deformation analysis of shells and membranes [8, 1, 20], fluid-structure interaction [10, 27], integrated design and analysis [11] and contact analysis [12]. Finally, it is worth mentioning that subdivision shells can be readily incorporated into a conventional finite element software by providing the subdivision shape functions as part of the input as proposed by Benson et al. [2].

The outline of the paper is as follows. In Section 2 we begin by reviewing box-splines on triangular three-direction meshes and clarify their relationship to subdivision surfaces. Subsequently, smooth subdivision schemes and their extensions for representing non-smooth and non-manifold surfaces and surfaces with conic sections as boundaries are discussed. Moreover, a computationally efficient algorithm is proposed for parameterising the introduced extended subdivision surfaces. In Section 3, first the relevant aspects of the Kirchhoff-Love thin-shell theory for finite displacements are reviewed and subsequently the enforcement of normal constraints at rigid joints using Lagrange multipliers are described. Finally, in Section 4 a number of test examples are represented to demonstrate the proposed theoretical and algorithmic approach.

2. Subdivision surfaces

2.1. Box-splines and their refinement

In the present work only surfaces based on triangle meshes are used. In contrast to quadrilateral meshes, splines on triangular meshes cannot be generated as the tensor product of one-dimensional b-splines. There are however box-splines for rectangular and non-rectangular meshes which are defined in a recursive manner similar to the one-dimensional b-splines. It can be shown that the box-splines, see e.g. de Boor et al. [15], have the same approximation properties as the one-dimensional and the tensor-product b-splines. For deriving the particular box-splines used in this paper, a tiling of the plane with equilateral triangles as shown in Figure 1 is considered, which is usually referred to as a three-direction mesh due to the presence of three independent directions. Note that using a NURBS terminology this three-direction mesh represents the knot space of the considered box-splines. As will be important later, each vertex in such a mesh has six incident edges or, in other terms, has the *valence* six. Furthermore, the non-orthogonal base vectors e_1 and e_2 shown in Figure 1 are used for parameterising the plane and the associated coordinates are denoted with $\theta^T = (\theta^1, \theta^2)$.

In the following the recursive definition of box-splines using repeated integration is briefly reviewed. Further details like generalisation to arbitrary dimensions and mathematical proofs can be found, for example, in [15] and [41].

The lowest order box spline is a constant with

$$N^0(\theta) = \begin{cases} 1 & \text{if } 0 \leq \theta^1 < 1 \text{ and } 0 \leq \theta^2 < 1 \\ 0 & \text{else} \end{cases} \quad (1)$$

Higher order box-splines are generated by repeated convolutions along direction vectors v_1, v_2, \dots, v_b . For example,

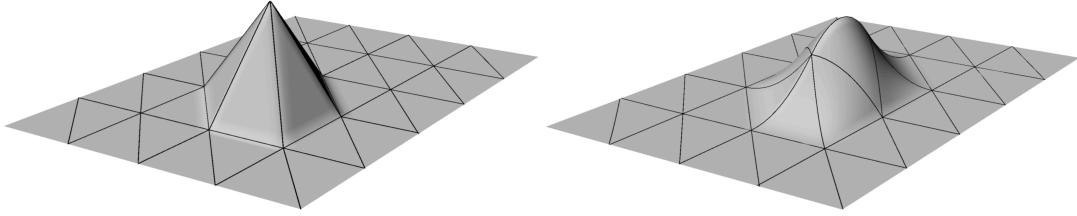


Figure 2. Box spline shape functions created with the direction vector $\mathbf{v}_1^T = (1, 1)$ (left) and with the direction vectors $\mathbf{v}_1^T = (1, 1)$ and $\mathbf{v}_2^T = (1, 0)$ (right).

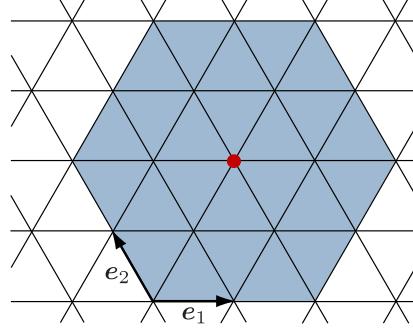


Figure 3. Support of a quartic box spline for the vertex marked with the dot.

the convolution along the direction vector $\mathbf{v}_1^T = (1, 1)$

$$N^1(\boldsymbol{\theta}) = \int_0^1 N^0(\boldsymbol{\theta} + t\mathbf{v}_1) dt \quad (2)$$

yields the hat function (see Fig. 2). Or, a subsequent convolution along the direction vector $\mathbf{v}_2^T = (1, 0)$

$$N^2(\boldsymbol{\theta}) = \int_0^1 N^1(\boldsymbol{\theta} + t\mathbf{v}_2) dt \quad (3)$$

gives a box spline which is quadratic in the e_1 mesh direction and linear in the other two mesh directions (Fig. 2). It can be verified that the generated surface is C^1 continuous in the e_1 direction and C^0 continuous along the other two directions.

In the remainder of the paper only (quartic) box spline basis functions $N(\boldsymbol{\theta})$ generated by successive convolutions along the following set of direction vectors

$$(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_5) = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix} \quad (4)$$

are used. The upper index indicating the polynomial degree of the basis function has been dropped for notational simplicity. This particular choice of direction vectors yields a basis function which is C^2 -continuous across the element boundaries. Since the convolution operation is commutative the order of the vectors is not relevant. Furthermore, there are recursive closed form expressions available so that the convolution integrals have never to be evaluated in practice [15]. The support of a quartic box spline basis function consists of a two ring of elements around a vertex as shown in Figure 3. During convolution the smoothness and the polynomial order of the box spline increases along the direction vectors. Each box-spline is comprised of piecewise polynomial surface patches which are smoothly connected across the triangle boundaries.

Due to the special (shift-invariant) structure of the used three-direction mesh, the translation of the above specified shape function yields the box spline shape functions for the other vertices. Owing to the partition of unity property of box-splines the combination of all shape functions can be used, e.g., for interpolating a scalar field

$$p = \sum_{\mathbf{J}} N(\boldsymbol{\theta} - \mathbf{J}) p_{\mathbf{J}}^0 \quad (5)$$

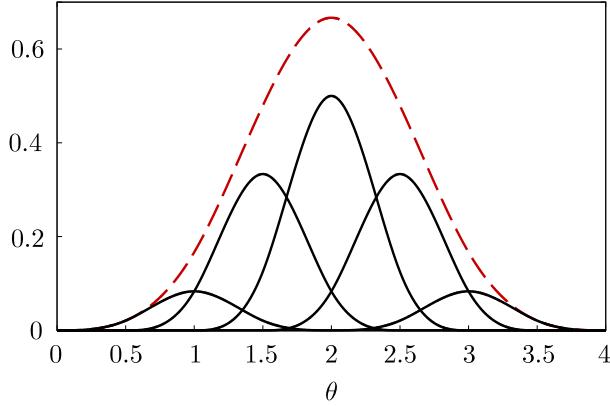


Figure 4. Illustrative example: A one-dimensional cubic b-spline on a coarse mesh (dashed line) can be composed out of five cubic b-spline shape functions on a finer mesh (solid lines). The coarse mesh has an uniform element length of 1.0 and the fine mesh has an uniform element length of 0.5. The relationship between the coarse and fine b-spline shape functions is

$$N(\theta) = \frac{1}{8}N(2\theta) + \frac{1}{2}N(2\theta - 1) + \frac{3}{4}N(2\theta - 2) + \frac{1}{2}N(2\theta - 3) + \frac{1}{8}N(2\theta - 4).$$

where $\mathbf{J} \in \mathbb{Z}^2$ is a multi-index so that the summation is over all the vertices in the $e_1 e_2$ -plane. The corresponding vertex values to be interpolated are denoted with $p_{\mathbf{J}}^0$. The superscript 0 can be ignored at the moment, but is going to be used soon to indicate the refinement level.

In the context of subdivision surfaces, an important property of box spline shape functions is their refinability, which enables to represent a box-spline on a coarse mesh as a linear combination of box-splines on a finer mesh. The fine mesh is obtained from the coarse mesh by subdividing its edges at their midpoint and subsequently introducing new triangles by connecting the newly created vertices (see Fig. 1). The relationship between the coarse and fine mesh shape functions $N(\theta - \mathbf{I})$ and $N(2\theta - \mathbf{J})$, respectively, can be expressed as

$$N(\theta - \mathbf{I}) = \sum_{\mathbf{J}} S_{\mathbf{IJ}} N(2\theta - \mathbf{J}) \quad (6)$$

The scalars $S_{\mathbf{IJ}} \in \mathbb{R}$ depend only on the set of used direction vectors or in other terms on the polynomial order of the shape functions. There are closed form expressions for computing the components of $S_{\mathbf{IJ}} \in \mathbb{R}$, which can be found, e.g., in [15]. As an illustrative example, in Figure 4 the refinability of cubic one-dimensional b-splines is shown. Although not shown here, similar plots can also be produced for box-splines.

Next, the interpolation of the scalar field p of equation (5) on the fine mesh is considered

$$p = \sum_{\mathbf{J}} N(2\theta - \mathbf{J}) p_{\mathbf{J}}^1 \quad (7)$$

where $p_{\mathbf{J}}^1$ are the vertex values on the finer mesh. The refinement relationship (6) can be used for establishing a relationship between the coarse and fine mesh vertex values $p_{\mathbf{J}}^0$ and $p_{\mathbf{J}}^1$, respectively. First, the refinement relationship is introduced into the interpolation equation (5)

$$p = \sum_{\mathbf{I}} N(\theta - \mathbf{I}) p_{\mathbf{I}}^0 = \sum_{\mathbf{I}} \left(\sum_{\mathbf{J}} S_{\mathbf{IJ}} N(2\theta - \mathbf{J}) \right) p_{\mathbf{I}}^0 \quad (8)$$

Comparing this with equation (7) yields an equation for computing the fine mesh vertex values from the coarse mesh vertex values

$$p_{\mathbf{I}}^1 = \sum_{\mathbf{J}} S_{\mathbf{IJ}} p_{\mathbf{J}}^0 \quad (9)$$

This relationship can be recursively used in order to generate vertex values for successively finer meshes. It can be shown that the vertex values converge under repeated refinement towards the scalar field p . Or in other terms, the discrete vertex values on the successively refined meshes provide a denser and denser discrete representation of the continuous field p . Note that the spline shape functions, with the exception of the constant and the linear ones, are due to their non-local support non-interpolating, i.e. $p(\mathbf{I}) \neq p_{\mathbf{I}}^i$ with the subdivision level $i \in \mathbb{N}$ in general.

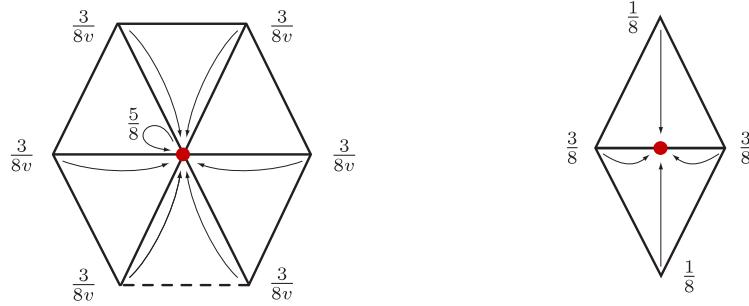


Figure 5. Vertex and edge masks (left and right, respectively) for standard vertices and edges (i.e., non-crease and non-corner). The valence v of a vertex is the number of its incident edges. The dashed edge emphasises that the vertex mask applies to vertices with arbitrary valence.

2.2. Review of subdivision surfaces

As discussed in the previous section, it is feasible to create spline surfaces through repeated refinement of a given coarse mesh representation. This inherently discrete approach of creating a surface through successive refinement is the key to subdivision surfaces. The well-known spline refinement relations are only applicable to shift-invariant meshes, such as triangular three-direction meshes or quadrilateral tensor product meshes. In 1978, Catmull and Clark [6] and Doo and Sabin [18] independently realised that the spline refinement relations can be generalised to unstructured meshes. The Catmull and Clark approach generalises the cubic tensor-product b-splines and the Doo and Sabin approach generalises the quadratic tensor-product b-splines to arbitrary connectivity meshes. Later, in 1987 Loop [33] introduced a subdivision approach which generalises quartic box-splines defined on triangular three-direction meshes to arbitrary triangular meshes. In the following the Loop's scheme is described which is the only scheme used in the present work.

In Loop subdivision, the box-spline refinement relations of the last section are written in a form more suitable for software implementation. To this end, one subdivision step is decomposed into a refinement and a smoothing step. In the refinement step the mesh is refined by quadrisecting each triangle. Subsequently, the coordinates of all the vertices are recomputed as a weighted average of the vertex coordinates on the coarse mesh. Due to the restricted support size of the underlying box-spline basis functions it is evident that the fine mesh vertices depend only on coarse mesh vertices on their immediate topological neighbourhood. In the subdivision literature, the weights are usually given in form of *subdivision masks* as shown in Figure 5. In case of a three-direction mesh, where each vertex has six incident edges, the weights given in Figure 5 represent the scalars S_{JI} in equation (9) [33]. The vertex mask is used for recomputing the coordinates of vertices which have already existed on the coarse mesh. The edge mask is used for computing the coordinates of vertices introduced on the edges during the quadrisection.

The general idea behind subdivision surfaces is to devise subdivision masks, such as the ones in Figure 5, which converge to smooth surfaces even if applied to arbitrary connectivity meshes. Although in practice most subdivision schemes reduce to splines on shift-invariant meshes, it is theoretically possible to devise subdivision surfaces which converge in the limit to a surface different from a spline surface. In fact there is considerable scope in designing specific subdivision masks for modelling various surface characteristics. As will be discussed in the following sections, it is possible to include non-smooth features, such as creases, or exact boundaries by just locally modifying the subdivision masks. It is even possible to combine one refinement step with several smoothing steps whereby the polynomial order of the spline surface is increased during each smoothing step [54, 51].

2.3. Extended subdivision surfaces

The input of extended subdivision surfaces introduced by Biermann et al. [3] is a so-called *tagged* control mesh on which some edges and vertices have certain tags. More precisely, triangle edges along creases and domain boundaries are tagged as *crease*. In addition, vertices at domain corners are tagged as *corner*. The tags determine the subdivision masks to be applied during the smoothing step of subdivision. The refinement step for extended subdivision surfaces is identical to the refinement step for smooth subdivision surfaces. The tags of the coarse mesh entities are during the

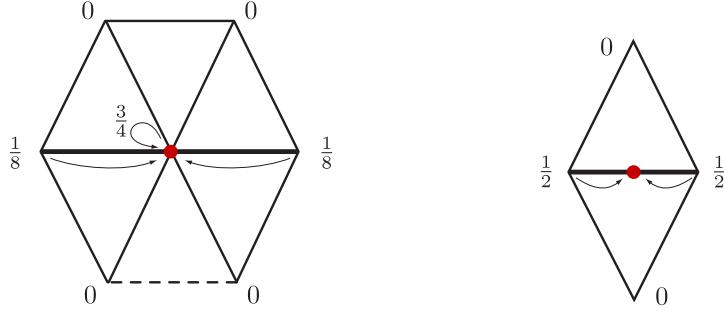


Figure 6. Vertex and edge masks (left and right, respectively) for crease vertices and edges. The bold lines represent crease edges. Note that the vertex mask is independent of the valence of the vertex and only vertices on the crease have a weight different from zero.



Figure 7. Vertex mask for corner vertices (left) and edge mask for inserting a vertex on an edge next to a corner (right). The solid square denotes the corner node and the bold edges the domain boundary. For convex corners of the domain the factor γ is chosen as $\frac{3}{8}$ and for non-convex corners it depends on the corner-angle. See Biermann et al. [3] for details.

refinement inherited to the corresponding fine mesh entities. For example, the two edges resulting from the refinement of a crease edge are again tagged as a crease.

During the smoothing step, if none of the edges attached to a vertex has a tag, the smooth subdivision masks given in Figure 5 are applied. If two of the attached edges to a vertex have tags, the edges represent either a physical crease or boundary and the masks shown in Figure 6 are applied. Although in the present implementation the case of vertices with one attached tagged edge has been excluded, their treatment is discussed by Biermann et al. [3]. Note that in Figure 6 the new vertex coordinates along a crease are only the weighted average of the vertex coordinates of the coarse mesh crease vertices. As a result, along a crease two smooth surface patches join only in a C^0 -continuous sense. Therefore, the crease masks can also be applied to vertices on the domain boundaries and no special tags and masks are necessary for domain boundaries. It is noteworthy that the crease weights given in Fig. 6 are identical to the refinement weights of one-dimensional cubic b-splines. Furthermore, there are parallels between the introduced crease masks with zero weights and the knot repetition in case of NURBS.

In addition to crease masks, there are special masks for vertices located at the domain corners and newly created vertices next to the corners. To this end the vertices at the domain corners are tagged as corners. As visible from the corner mask in Figure 7 the position of the corner vertices are fixed during the entire subdivision refinement. As discovered by Biermann et al. [3] the standard edge refinement masks applied around a non-convex corner lead to surfaces which fold over on themselves. Therefore, Biermann et al. proposed the modified edge masks given in Figure 7 which include the parameter γ which depends on the corner angle. In case of convex corner corners γ is $\frac{3}{8}$ and the edge mask identical to the smooth subdivision edge mask in Figure 5.

2.4. Non-manifold subdivision surfaces

In engineering many industrially relevant surface geometries have non-manifold topology in the sense that the surface cannot be mapped onto a disc at some surface points. The most common non-manifold feature is the connection of several surface pieces along their edges, such as T-joints (see Fig. 8). Non-manifold surfaces can be thought as the union of intersecting manifold surfaces [53]. In particular in subdivision surfaces, this view is facilitated by extended subdivision surfaces as they allow to exactly control the position and tangent of the subdivision surface. More specifically, non-manifold geometries can be modelled by tagging all element edges attached to a joint as

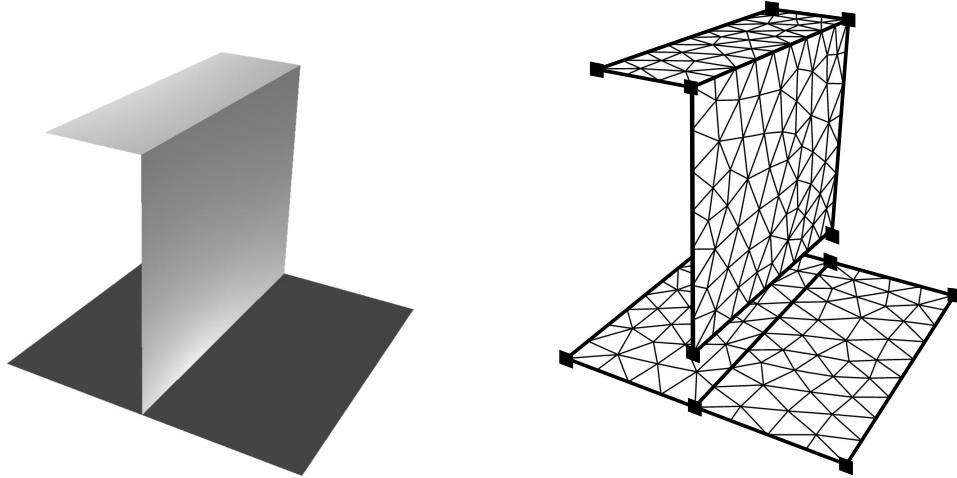


Figure 8. A non-manifold, non-smooth surface (left) and its control mesh composed of two tagged control meshes (right). The bold lines on the control mesh denote the crease edges along which the crease subdivision masks of Figure 6 are used.

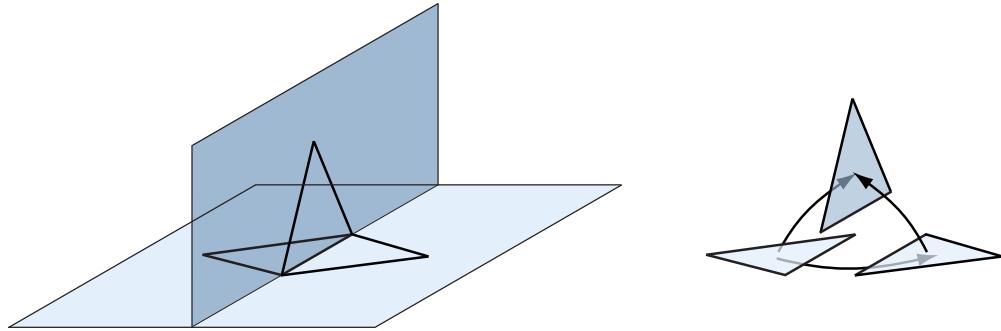


Figure 9. A T-joint with three representative elements shown (left) and the corresponding adjacency relationship as stored in the data structure (right).

crease, as discussed in Section 2.3.

The major difference between the smooth and non-manifold subdivision concerns the implementation of the mesh handling data structures. The most basic data structure for subdivision stores for each triangle its three vertices and three neighbour triangles [7]. This basic data structure can be modified to incorporate non-manifold edges, if the following rule is taken into account (see also [53]). According to this rule the connectivity across the triangles is stored in a ring of pointers (see Figure 9). Considering a joint with $r > 1$ attached triangles T_0, T_1, \dots, T_{r-1} , the triangle T_i contains a pointer to T_{i+1} , the next triangle on the non-manifold, where $i < (r-1)$. The last triangle T_{r-1} contains a pointer to the first triangle T_0 . This way of referencing the connectivity of the joint permits to reuse the original basic triangle data structure. The joint-free manifold case is recovered with $r = 2$.

2.5. Limit analysis and surface properties

The convergence and smoothness properties of subdivision schemes can be analysed without referring to their spline origins. This is particularly useful for studying the properties of subdivision schemes around irregular vertices and close to tagged vertices and edges. Note that around such vertices and edges the mesh does not have the three-direction structure crucial for box-splines, as discussed in Section 2.1. For example, at irregular vertices, sometimes also referred to as extraordinary or star vertices, the number of incident edges is different from six. To study the convergence properties of subdivision schemes at a vertex, its one-ring vertices are considered. The one-ring of a vertex is the set of vertices which lie one edge away from it. For ease of notation, the coordinates of the considered

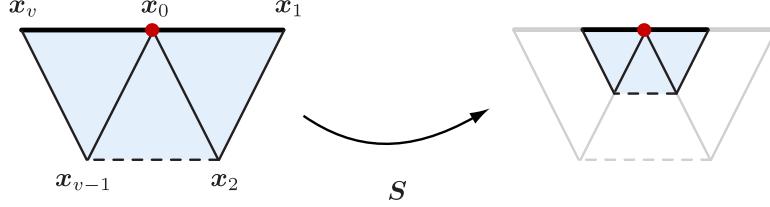


Figure 10. A one-ring of vertices at the boundary of the control mesh (left) and the refined one-ring at subdivision level 1 (right). The refinement is the linear mapping of the coordinates of the one-ring on the left to the one-ring on the right.

vertex x_0 and the coordinates of its one-ring vertices are assembled into a matrix

$$\mathbf{X}^0 = (x_0, \ x_1, \ x_2, \ \dots, \ x_v)^T \quad (10)$$

where \mathbf{X}^0 is a $(v + 1) \times 3$ matrix in which v is the valence of the vertex. The superscript 0 indicates again the refinement level. The valence v of the vertex is the number of its incident edges. As shown in Figure 10, this one-ring of vertices can be subdivided with the subdivision masks given in Sections 2.2 and 2.3 yielding

$$\mathbf{X}^1 = \mathbf{S}\mathbf{X}^0 \quad (11)$$

The vector \mathbf{X}^1 represents the coordinates of the considered neighbourhood on the subdivision level 1 and the subdivision matrix \mathbf{S} contains the subdivision weights previously given in form of masks. The matrix \mathbf{X}^1 is of size $(v + 1) \times 3$ and \mathbf{S} is of size $(v + 1) \times (v + 1)$. For computing the vector of coordinates \mathbf{X}^k on subdivision level k , the original vector \mathbf{X}^0 is multiplied k times with the subdivision matrix

$$\mathbf{X}^k = \mathbf{S}^k \mathbf{X}^0 \quad (12)$$

It is advantageous to perform this multiplication in the eigenspace of the subdivision matrix in which the following standard relationship holds

$$\mathbf{S} = \sum_{I=0}^v \mathbf{L}_I \lambda_I \mathbf{R}_I^T \quad (13)$$

where \mathbf{L}_I and \mathbf{R}_I are the I 'th left and right eigenvectors, respectively, and λ_I is the I 'th eigenvalue. It is now possible to decompose the vectors \mathbf{X}^0 with respect to the eigenvectors \mathbf{R}_I

$$\mathbf{X}^0 = \sum_{I=0}^v \mathbf{R}_I \mathbf{a}_I^T \quad (14)$$

with the 3×1 components (in the three-dimensional case)

$$\mathbf{a}_I = \mathbf{X}^{0T} \mathbf{L}_I \quad (15)$$

With these definitions, the subdivision equation (12) can be rewritten as

$$\mathbf{X}^k = \mathbf{S}^k \mathbf{X}^0 = \sum_{I=0}^v (\lambda_I)^k \mathbf{R}_I \mathbf{a}_I^T \quad (16)$$

For a convergent subdivision scheme, it can be expected, \mathbf{X}^k converges towards a limit surface as $k \rightarrow \infty$. By inspecting equations (16) and (15), it can be concluded that the properties of the subdivision surface must be intrinsically linked to the eigenvalues and eigenvectors of subdivision matrix \mathbf{S} . A more detailed analysis, as, e.g., presented in [42], shows that the following relationships must hold for the largest eigenvalues of the subdivision matrix \mathbf{S} of a convergent smooth subdivision scheme: $\lambda_0 = 1$, $\lambda_1 = \lambda_2 < 1$. Furthermore, for λ_0 the corresponding right eigenvector is $\mathbf{R}_0 = (1, 1, \dots, 1)^T$ so that \mathbf{a}_0 gives the limit position

$$\mathbf{a}_0 = \mathbf{X}^{0T} \mathbf{L}_0 \quad (17)$$

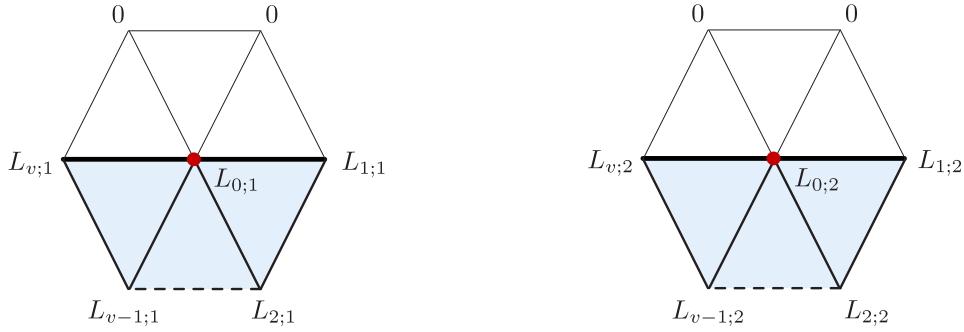


Figure 11. Masks for computing the two tangent vectors to the surface at crease or boundary vertices. The indicated weights give the tangents to the lower shaded surface. The weights for the vertices across the crease are zero so that the tangents on both sides of the crease can be determined independently from each other.

Similarly, it can be shown that \mathbf{a}_1 and \mathbf{a}_2 give two tangent vectors to the surface

$$\mathbf{a}_1 = \mathbf{X}^{0T} \mathbf{L}_1 \quad \mathbf{a}_2 = \mathbf{X}^{0T} \mathbf{L}_2 \quad (18)$$

It should be noted that the subdivision matrix \mathbf{S} and its eigenstructure depend only on the valence of the vertex and are independent of the vertex coordinates. Therefore, it is straightforward to compute the eigenvectors, \mathbf{L}_0 , \mathbf{L}_1 and \mathbf{L}_2 , of the subdivision and to store them in form of masks.

For the purpose of this paper, in particular the tangents of the surface at the domain boundaries and at creases are relevant. In Section 3 the tangents will be used for determining the shell normals and enforcing Dirichlet type boundary conditions on them. The tangent masks for Loop subdivision scheme at boundaries and creases are shown in Figure 11. The components of the eigenvector \mathbf{L}_1 are given by Biermann et al. [3] with

$$\mathbf{L}_1 = (0, \frac{1}{2}, 0, \dots, 0, -\frac{1}{2})^T \quad (19)$$

and the components of the eigenvector $\mathbf{L}_2 = (L_{0;2}, L_{1;2}, \dots, L_{v;2})^T$ with

$$L_{0;2} = -\frac{2}{v-1} \left(\left(\frac{2}{3} - c_1 \right) c_3 - c_2 c_4 \right) \quad (20)$$

$$L_{1;2} = L_{v;2} = -\frac{2}{v-1} \left(\left(\frac{c_1}{2} + \frac{1}{6} \right) c_3 + \frac{1}{2} c_2 c_4 \right) \quad (21)$$

$$L_{i;2} = \frac{2}{v-1} \sin(\xi(i-1)) \quad \text{for } i = 2, \dots, v-1 \quad (22)$$

in which ξ , c_1 , c_2 , c_3 and c_4 are constants which depend on the valence v of the considered vertex

$$\xi = \frac{\pi}{v-1} \quad \beta = \arccos(\cos(\xi) - 1) \quad (23)$$

$$c_1 = \frac{\frac{1}{4}(1 + \cos \xi)}{3(\frac{1}{2} - \frac{1}{4}\xi)} \quad c_2 = \frac{\frac{2}{3} - c_1}{\cos \frac{(v-1)\beta}{2}} \quad c_3 = \frac{\sin \xi}{1 - \xi} \quad c_4 = \frac{\cos \frac{(v-1)\beta}{2} \sin \xi}{\cos \beta - \cos \xi} \quad (24)$$

Furthermore, in Section 3.3 the tangent masks for corner vertices are needed which are given in Figure 12.

2.6. Rational subdivision surfaces

Rational subdivision surfaces are necessary for representing exact conic sections, such as circles, which are abundant in industrially relevant geometries. It is possible to derive rational subdivision surfaces along the lines of the derivation of rational b-splines. To this end it is instructive to first briefly review the related derivations. Rational b-splines R_I are defined by assigning a scalar weight $w_I \in \mathbb{R}$ to each b-spline shape function and at the same time enforcing partition of unity

$$\sum_I R_I = 1 \quad (25)$$



Figure 12. Masks for computing the two tangent vectors to the surface at corner vertices.

which leads to the following rational shape functions

$$R_I = \frac{w_I N_I}{\sum_J w_J N_J} \quad (26)$$

A rational b-spline is the interpolation of a given set of control mesh vertex coordinates \mathbf{x}_I with the shape functions R_I

$$\mathbf{x} = \sum_I R_I \mathbf{x}_I = \sum_I \left(\frac{w_I N_I}{\sum_J w_J N_J} \right) \mathbf{x}_I \quad (27)$$

The scalar weights w_I provide additional control over the shape of the b-spline and are extensively used in geometric b-spline applications.

Alternatively, in CAD literature rational b-splines are introduced as the perspective projection of a b-spline from a space \mathbb{R}^{n+1} which has one more dimension than the embedding space \mathbb{R}^n [40]. The b-spline in \mathbb{R}^{n+1} is defined with

$$\tilde{\mathbf{x}} = \begin{pmatrix} w\mathbf{x} \\ w \end{pmatrix} = \sum_I N_I \begin{pmatrix} w_I \mathbf{x}_I \\ w_I \end{pmatrix} = \sum_I N_I \tilde{\mathbf{x}}_I \quad (28)$$

In this equation only the nodal vector $\tilde{\mathbf{x}}_I$ is different from the ordinary b-spline interpolation. In $\tilde{\mathbf{x}}$ the control vertex coordinates \mathbf{x}_I have been multiplied with the corresponding weights w_I and an additional row with w_I has been introduced. The perspective projection of $\tilde{\mathbf{x}}$ onto the plane with $w = 1$ gives the physical b-spline surface

$$\mathbf{x} = \frac{w\mathbf{x}}{w} = \sum_I \left(\frac{w_I N_I}{\sum_J w_J N_J} \right) \mathbf{x}_I \quad (29)$$

which is identical to equation (27).

As discussed in Section 2.2, all b-splines have a corresponding subdivision scheme. Although the opposite is not true, the most widely used subdivision schemes reduce to splines in the regular setting. This means according to equation (28), rational subdivision schemes can be defined by simply applying the subdivision in the higher dimensional space \mathbb{R}^{n+1} and computing its perspective projection into the embedding space \mathbb{R}^n .

In the following the proposed approach is illustrated by means of a surface with a circular hole (Fig. 13). In the context of rational b-splines, circles are composed of several circular arcs and each arc is represented, for example, as a quadratic non-uniform rational b-spline curve, which is, in fact, a quadratic rational Bézier curve (see, e.g., [43]). In the same way, in rational subdivision schemes the circle can also be composed out of several quadratic Bézier segments. For example, in Figure 13 the circular boundary is composed out of four quadratic Bézier curves with the control polygons $8 - 1 - 2$, $2 - 3 - 4$, $4 - 5 - 6$ and $6 - 7 - 8$. In order to obtain the corresponding four Bézier curves by subdivision, first the begin or end vertices 8 , 2 , 4 and 6 need to be tagged as corner. Recall that assigning the corner tag enforces that the vertex coordinates remain the same throughout the subdivision. Subsequently, subdividing the control polygons with the masks given in Figure 14 leads in the limit to four quadratic Bézier segments. In CAD, it is well known that rational Bézier curves with the weights $w_1 = w_3 = w_5 = w_7 = \frac{\sqrt{2}}{2}$ and $w_8 = w_2 = w_4 = w_6 = 1$ reproduce an exact circle. By assigning the same weights and applying the subdivision on the augmented vertex vector $\tilde{\mathbf{x}}$ and projecting the result back to \mathbb{R}^n (eq. 29) leads to a circle which is created by subdivision.

In Figure 15 the internal circular boundary has been created with the described rational subdivision scheme. Inside the domain the standard smooth subdivision masks of Figure 5 and at the external boundary the crease masks of Figure 6 have been used. In order to obtain the mesh given in Figure 15 the initial control mesh has been subdivided three

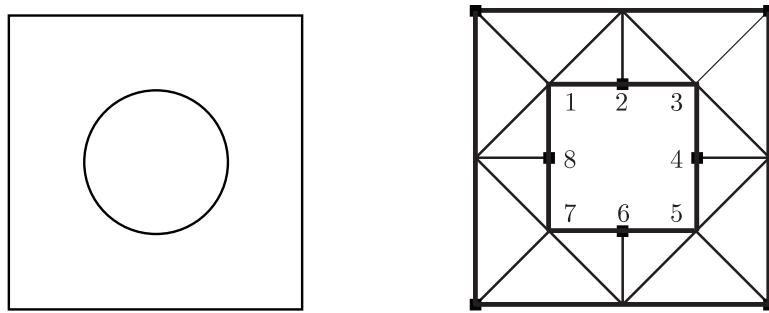


Figure 13. Geometry of a plate with a circular hole and its discretization with a coarse triangular mesh. Note that the bold edges indicate boundaries and squares corner vertices. On the external boundary the mask given in Figure 6, within the domain the mask given in Figure 5 and on internal circular boundary the mask given in Figure 14 is used. It is noteworthy that all the (non corner) internal boundary vertices are irregular as their valence is different from four.



Figure 14. Vertex and edge masks for creating a quadratic Bézier curve by subdividing its control polygon (see, e.g., [43] for details).

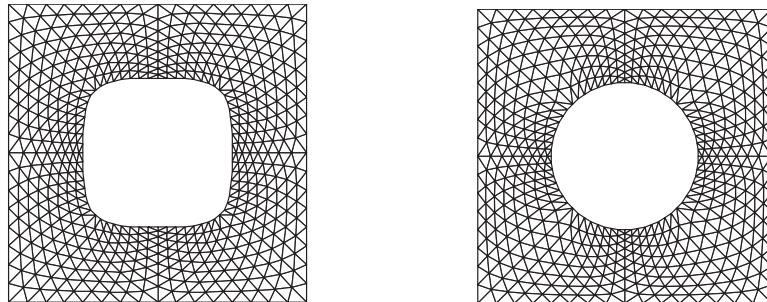


Figure 15. Three times subdivided meshes. In the left mesh standard subdivision masks were used. In the right mesh rational subdivision masks have been used along the internal boundary in order to create the exact circle.

times. It is evident that, in particular, for very coarse initial control meshes the rational subdivision schemes are able to reproduce the desired geometry much more faithfully. For comparison purposes Figure 15 shows the mesh resulting from standard subdivision during which the internal boundary has been subdivided with the crease masks given in Figure 6, which lead in the limit to a cubic b-spline.

2.7. Parameterisation of subdivision surfaces

The parameterisation of subdivision surfaces enables to compute quantities, such as limit positions, tangents and curvatures, at arbitrary locations on the surface. In the finite element context, this is necessary for evaluating surface properties at quadrature points and for computing the element integrals. The limit masks introduced in Section 2.5 provide the surface position and tangent plane only at the vertices of the mesh and, hence, cannot be used when arbitrary integration rules are used.

2.7.1. Basic approach The Loop subdivision coincides with box spline surfaces away from crease edges and irregular vertices as discussed in Section 2.1. Hence, the box spline shape functions can be used for evaluating regular element patches. It is also possible to use instead of the vertex centric box-spline shape functions the element centric Bézier shape functions [50]. Note that it is in computer-aided geometric design well known that spline shape functions can be expressed with the corresponding Bézier shape functions. As shown in Figure 16, the interpolation within one

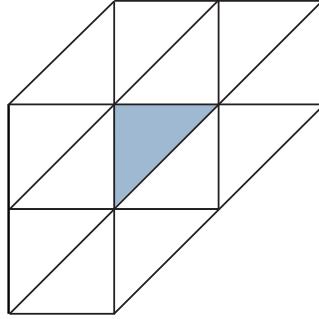


Figure 16. Control patch of a triangular element. In the shaded element only the shape functions associated to the shown nodes are non-zero.

triangular element ultimately depends on a patch of elements containing twelve vertices

$$\mathbf{x} = \sum_{I=0}^{11} N_I \mathbf{x}_I = (\mathbf{N} \mathbf{X}_P^0)^T \quad (30)$$

where N_I are the twelve Bézier shape functions[†] and \mathbf{x}_I are the corresponding vertex coordinates [50, 9]. Equation (30) also shows an equivalent concise matrix form in which $\mathbf{N} = (N_I)$ is a (1×12) row vector and the matrix $\mathbf{X}_P^0 = (\mathbf{x}_I)$ contains the co-ordinates of the twelve vertices in the patch. As an aside, recall from Section 2.1 that the support of one quartic-box spline shape function consists of a two ring of elements around a vertex. Also from this, it is evident that the interpolation within one element is influenced by twelve vertices which have a shape function overlapping the element.

As proposed by Stam [49, 50], patches containing irregular vertices with valence different from $v \neq 6$ can also be parameterised with box spline shape functions. To this end, it is crucial to realise that during the refinement of the original coarse triangle mesh by quadrissection only regular vertices are created (i.e., all new vertices have the valence $v = 6$). In fact, the total number of irregular vertices in the mesh remains constant and their area of influence converges in the limit to zero. Hence, during repeated subdivision refinement more and more regular patches are created. Based on this observation Stam proposed an algorithm for parameterising subdivision surfaces which coincide with spline surfaces on regular patches. Although the original Stam algorithm was restricted to smooth irregular patches far away from boundaries, it can be generalised to extended subdivision surfaces. This modification is particularly straightforward in the case of finite elements where the limit surface needs only to be evaluated at few distinct quadrature points within each element.

As can be deduced from the introduced extended subdivision masks (Figs. 5, 6 and 7), the interpolation within one element depends on the set of vertices in the one-neighbourhood of the element, see also the example in Figure 17. In order to parameterise the surface over an element, the subdivision of the corresponding patch is considered, which can be expressed in matrix notation as

$$\mathbf{X}^1 = \mathbf{A}^0 \mathbf{X}^0 \quad (31)$$

Again, the upper indices indicate the subdivision level so that \mathbf{X}^0 are the vertex coordinates of the coarse patch and \mathbf{X}^1 are the vertex coordinates of the once subdivided patch. Each row of \mathbf{X}^0 and \mathbf{X}^1 represent one vertex with its three coordinates. The matrix \mathbf{A}^0 represents the subdivision operation and contains the subdivision weights previously shown as masks. For example, for the subdivision shown in Figure 18, \mathbf{X}^0 has the dimension (9×3) , \mathbf{X}^1 has the dimension (20×3) and \mathbf{A}^0 has the dimension (20×9) . The subdivision is applied until the point of interest (e.g. quadrature point) lies within a quartic box spline patch. As will be discussed in Section 3, in subdivision shells element integrals are evaluated with a single quadrature point located at the barycentre of the element. In that case the barycentre of the coarse element always lies after two subdivision steps within a regular patch.

$$\mathbf{X}^2 = \mathbf{A}^1 \mathbf{X}^1 = \mathbf{A}^1 \mathbf{A}^0 \mathbf{X}^0 \quad (32)$$

[†]With a slight abuse of notation box-splines as well as Bézier shape functions are denoted with N .

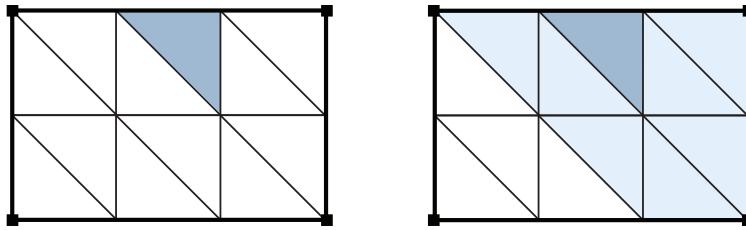


Figure 17. A coarse control mesh for a flat quadrilateral domain consisting out of twelve triangular elements (left). The element edges on the domain boundary are tagged as *crease* and the four corners of the domain are tagged as *corner*. As an example, the interpolation within the dark shaded element is considered. The interpolation depends on the one-neighbourhood of the element shown on the right and contains nine vertices.

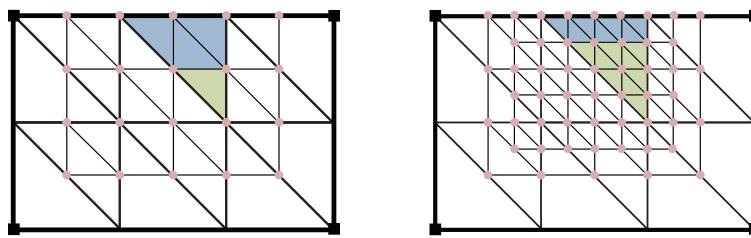


Figure 18. Subdivision refinement of the domain shown in Figure 17 for the purpose of parameterising the limit surface within the dark shaded parent element. As can be seen from the left figure, after one subdivision step one of the four newly created elements within the parent element belong to a regular patch. After two subdivision steps (right figure) nine of the newly created elements within the parent element belong to a regular patch.

Continuing with the example shown in Figure 18, \mathbf{X}^2 has the dimension (53×3) and \mathbf{A}^1 has the dimension (53×20) . For extracting the twelve vertices of a patch over the barycentre of the element, a (12×53) picking matrix \mathbf{P} is defined.

$$\mathbf{X}_P^{k=2} = \mathbf{P} \mathbf{X}^2 = \mathbf{P} \mathbf{A}^1 \mathbf{X}^1 = \underbrace{\mathbf{P} \mathbf{A}^1 \mathbf{A}^0}_{\tilde{\mathbf{S}}} \mathbf{X}^0 \quad (33)$$

where $\mathbf{X}_P^{k=2}$ is the (12×3) matrix with the coordinates of the regular patch, cf. Equation (30). Note that each row of the picking matrix is filled with zeros and a single one. Finally introducing the vertex coordinates of the extracted patch into (30) yields the result

$$\mathbf{x} = (\mathbf{N} \mathbf{X}_P^{k=2})^T = (\mathbf{N} \tilde{\mathbf{S}} \mathbf{X}^0)^T \quad (34)$$

where \mathbf{N} is a row vector of the twelve shape functions.

2.7.2. Implementation Although the matrix $\tilde{\mathbf{S}}$ was explicitly constructed in Stam's original method, see [49], the same is practically impossible for the extended subdivision schemes due to the multitude of masks and tags. This leads to too many different subdivision matrices \mathbf{A} and picking matrices \mathbf{P} depending on the tags present in the patch. However, a computationally oriented and easy-to-implement approach can be devised in which the general subdivision refinement algorithm is used directly to determine the components of the matrix $\tilde{\mathbf{S}}$. Thus, an explicit construction of various \mathbf{A} -matrices can be avoided completely.

To this end, it is important to realise that each row of the matrix $\tilde{\mathbf{S}}$ corresponds to a vertex in the original coarse control patch. Hence, each row of $\tilde{\mathbf{S}}$ can be determined by subdividing a scalar field which is non-zero only at the corresponding coarse patch control vertex. The values of the scalar field on the refined mesh constitute the components of the matrix. This is easiest understood by thinking of the subdivision refinement as a matrix vector multiplication. For example, assigning the value 1 to the vertex J and 0 to the other vertices on the control patch yields the J 'th column of $\tilde{\mathbf{S}}$. Using Kronecker delta δ_{JM} , this can be denoted concisely as

$$\tilde{S}_{KJ} = \sum_M \tilde{S}_{KM} \delta_{JM} \quad (35)$$

with twelve row-indices K forming the regular patch (30). The number of column indices M depends on the connectivity of the coarse patch and is, for example, nine for the example in Figure 17. In terms of implementation,

the subdivision refinement algorithm is considered as a "black-box" for determining the components of \tilde{S} so that the described approach works independently of the particular masks and tags used.

During the finite element solution procedure the limit surface has to be evaluated at pre-tabulated quadrature points θ_{GP} . The parametric coordinates θ_{GP} refer to the parameter space of the triangle on the control patch. If the limit surface is evaluated using equation (34) it is necessary to establish a mapping between the parameter space of the triangle on the control patch and the triangle on which the actual evaluation is performed. For the details of this mapping and computing the derivatives of the shape functions it is referred to [50, 9].

3. Review of thin-shell equations

In this section a brief summary of the thin shell equations is given as necessary for the subsequent derivations. Since the focus of the present paper is on thin shells, the Kirchhoff-Love assumption is used. The presented equations are restricted to large displacements and small strains. Further details can be found in [9, 8, 13, 4, 47, 48].

3.1. Kinematics

In the following, the mid-surface of the shell in the reference configuration is denoted with $\bar{\Omega} \subset \mathbb{R}^3$ and in deformed configuration with Ω . The respective boundaries are $\partial\bar{\Omega} = \bar{\Gamma}$ and $\partial\Omega = \Gamma$. Any configuration of the shell is assumed to be defined as

$$\varphi(\theta^1, \theta^2, \theta^3) = \mathbf{x}(\theta^1, \theta^2) + \theta^3 \mathbf{n}(\theta^1, \theta^2) \quad (36)$$

where $\varphi(\theta^1, \theta^2, \theta^3)$ are the position vector of a material point with the convective coordinates $(\theta^1, \theta^2, \theta^3)$. Similarly, the vector $\mathbf{x}(\theta^1, \theta^2)$ is the position vector of a material point with the coordinates (θ^1, θ^2) on the shell mid-surface with $\theta^3 = 0$. Furthermore, \mathbf{n} is the unit normal to the mid-surface. The corresponding vectors in the reference configuration are denoted with $\bar{\varphi}(\theta^1, \theta^2, \theta^3)$, $\bar{\mathbf{x}}(\theta^1, \theta^2)$ and $\bar{\mathbf{n}}(\theta^1, \theta^2)$. In the assumed kinematics the normals $\bar{\mathbf{n}}$ and \mathbf{n} are functions of the mid-surface and are not independent vectors.

The covariant base vectors on the mid-surface are defined as[‡]

$$\mathbf{a}_\alpha = \frac{\partial \mathbf{x}}{\partial \theta^\alpha} \quad (37)$$

and the corresponding contravariant vectors \mathbf{a}^α are defined as

$$\mathbf{a}_\alpha \cdot \mathbf{a}^\beta = \delta_\alpha^\beta \quad (38)$$

where the operation \cdot denotes the scalar product and δ_α^β is the Kronecker delta. Again, the corresponding vectors in the reference configuration are denoted with $\bar{\mathbf{a}}_\alpha$ and $\bar{\mathbf{a}}^\beta$.

The kinematic assumption (36) leads to the following Green-Lagrange strain tensor for the shell

$$\mathbf{E} = \boldsymbol{\alpha} + \theta^3 \boldsymbol{\beta} \quad (39)$$

where the higher-order terms in θ^3 have been neglected. The first tensor $\boldsymbol{\alpha}$ is related to the in-plane deformation of the mid-surface

$$\boldsymbol{\alpha} = \frac{1}{2} (\mathbf{a}_\alpha \cdot \mathbf{a}_\beta - \bar{\mathbf{a}}_\alpha \cdot \bar{\mathbf{a}}_\beta) \bar{\mathbf{a}}^\alpha \otimes \bar{\mathbf{a}}^\beta \quad (40)$$

and the second tensor $\boldsymbol{\beta}$ is related to the curvature change of the mid-surface

$$\boldsymbol{\beta} = (\bar{\mathbf{a}}_{\alpha,\beta} \cdot \bar{\mathbf{n}} - \mathbf{a}_{\alpha,\beta} \cdot \mathbf{n}) \bar{\mathbf{a}}^\alpha \otimes \bar{\mathbf{a}}^\beta \quad (41)$$

[‡]In this paper the Greek indices take only the values 1 or 2 and the summation convention over repeated indices applies.

3.2. Thin-shell energy functional and its discretization

The potential energy of a smooth hyper-elastic shell is given by

$$\Pi(\boldsymbol{x}) = \int_{\bar{\Omega}} (W^m(\boldsymbol{\alpha}) + W^b(\boldsymbol{\beta})) \mu d\bar{\Omega} + \Pi^{ext}(\boldsymbol{x}) \quad (42)$$

where W^m is the membrane energy density, W^b is the bending energy density, μ is the jacobian associated to the integration over the thickness and $\Pi^{ext}(\boldsymbol{x})$ is the potential of the external forces. An example of a hyper-elastic material is the St. Venant–Kirchhoff material

$$W^m(\boldsymbol{\alpha}) = \frac{1}{2} \frac{Et}{1-\nu^2} \boldsymbol{\alpha} : \boldsymbol{H} : \boldsymbol{\alpha} \quad W^b(\boldsymbol{\beta}) = \frac{1}{2} \frac{Et^3}{12(1-\nu^2)} \boldsymbol{\beta} : \boldsymbol{H} : \boldsymbol{\beta} \quad (43)$$

where E is Young's modulus, ν is the Poisson's ratio, t is the shell thickness and \boldsymbol{H} is the (constant) fourth order tensor with components

$$H^{\alpha\beta\gamma\delta} = \nu \bar{a}^{\alpha\beta} \bar{a}^{\gamma\delta} + \frac{1}{2}(1-\nu)(\bar{a}^{\alpha\gamma} \bar{a}^{\beta\delta} + \bar{a}^{\alpha\delta} \bar{a}^{\beta\gamma}) \quad (44)$$

with the contravariant metric in the reference configuration $\bar{a}^{\alpha\beta} = \bar{\boldsymbol{a}}^\alpha \cdot \bar{\boldsymbol{a}}^\beta$. The corresponding (second Piola–Kirchhoff) stress resultants of the shell are

$$\boldsymbol{p} = \frac{\partial W^m}{\partial \boldsymbol{\alpha}} = \frac{Et}{1-\nu^2} \boldsymbol{H} : \boldsymbol{\alpha} \quad (45)$$

$$\boldsymbol{q} = \frac{\partial W^b}{\partial \boldsymbol{\beta}} = \frac{Et^3}{12(1-\nu^2)} \boldsymbol{H} : \boldsymbol{\beta} \quad (46)$$

The energy functional (42) is discretized by introducing an interpolation of the form

$$\bar{\boldsymbol{x}}_h = \sum N_I(\theta^1, \theta^2) \bar{\boldsymbol{x}}_I \quad \boldsymbol{x}_h = \sum N_I(\theta^1, \theta^2) \boldsymbol{x}_I \quad (47)$$

where N_I are the subdivision shape functions and $\bar{\boldsymbol{x}}_I$ and \boldsymbol{x}_I are nodal position vectors of the shell mid-surface in reference and deformed configurations, respectively. Note that there is no closed form expression for N_I close to irregular vertices. However, the subdivision formalism guarantees that there is a smooth limit surface and that it can be evaluated as described in Section 2.7.

Introducing the interpolation (47) into the energy functional (42) and computing its stationary points with respect to the nodal coordinates in the deformed configuration \boldsymbol{x}_I leads to the discrete equilibrium equations

$$\frac{\partial \Pi(\boldsymbol{x}_I)}{\partial \boldsymbol{x}_I} = \int_{\bar{\Omega}} \left(\frac{\partial W^m(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{x}_I} + \frac{\partial W^b(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{x}_I} \right) \mu d\bar{\Omega} + \frac{\partial \Pi^{ext}(\boldsymbol{x})}{\partial \boldsymbol{x}_I} \quad (48)$$

$$= \int_{\bar{\Omega}} \left(\boldsymbol{p} : \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{x}_I} + \boldsymbol{q} : \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{x}_I} \right) \mu d\bar{\Omega} + \frac{\partial \Pi^{ext}(\boldsymbol{x})}{\partial \boldsymbol{x}_I} = \mathbf{0} \quad (49)$$

where the integral on the right hand side yields the nodal internal force vector and the second term is the external force vector at node I . The derivatives of the strain tensors with respect to the nodal coordinates follow from equations (40), (41) and (47)

$$\frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{x}_I} = \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{a}_\gamma} \frac{\partial \boldsymbol{a}_\gamma}{\partial \boldsymbol{x}_I} = \boldsymbol{a}_\alpha N_{I,\beta} \bar{\boldsymbol{a}}^\alpha \otimes \bar{\boldsymbol{a}}^\beta \quad (50)$$

$$\frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{x}_I} = \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{a}_{\alpha,\beta}} \frac{\partial \boldsymbol{a}_{\alpha,\beta}}{\partial \boldsymbol{x}_I} + \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{n}} \frac{\partial \boldsymbol{n}}{\partial \boldsymbol{x}_I} = - \left(\boldsymbol{n} N_{I,\alpha\gamma} + \boldsymbol{a}_{\alpha,\beta} \frac{\partial \boldsymbol{n}}{\partial \boldsymbol{x}_I} \right) \bar{\boldsymbol{a}}^\alpha \otimes \bar{\boldsymbol{a}}^\beta \quad (51)$$

The derivative of the normal vector \boldsymbol{n} with respect to the nodal positions can be found in the appendix.

3.3. Rigid non-manifold joints and rotational Dirichlet boundary conditions

Non-manifold shells, as shown in Fig. 8, are modelled as the composition of surface patches, which are forced to coherent displacements and rotations along their joint boundary. Displacement continuity is automatically achieved

by subdividing the boundary of each surface patch with the crease subdivision stencils introduced in Section 2.3. In case of rigid-joints or clamped boundaries it is necessary to also constrain the deformation of surface normals at patch boundaries. To this end, a constrained minimisation problem is considered for deriving the discrete equilibrium equations

$$\Pi^C(\boldsymbol{x}_I, \boldsymbol{\lambda}) = \Pi(\boldsymbol{x}_I) + \boldsymbol{\lambda}^T \mathbf{C}(\boldsymbol{x}_I) \quad (52)$$

where $\boldsymbol{\lambda}$ is a Lagrange multiplier vector and \mathbf{C} is the constraint matrix which depends on the normals of the joined surface patches. In turn, the normals of the limit surface depend on the vertex coordinates \boldsymbol{x}_I of the control mesh as discussed in Section 2.3. Note that instead of the Lagrange multiplier method any other method for solving constrained minimisation problems, such as the penalty or Nitsche methods, can be used in general.

As an example, we consider the case of a rigid joint between two smooth surface patches with a common boundary. To enforce the joint rigidity, the surface normals along the common boundary have to be constrained during the finite element solution procedure so that

$$\mathbf{C}^{(1,2)}(\boldsymbol{x}_I) = \mathbf{n}^{(1)} \cdot \mathbf{n}^{(2)} - \bar{\mathbf{n}}^{(1)} \cdot \bar{\mathbf{n}}^{(2)} = 0 \quad (53)$$

with the indices (1) and (2) denoting surface patch one and two and the bar denoting normals in the reference configuration. The normals of both patches follow from the respective tangents $\mathbf{t}^{(1)}$ and $\mathbf{t}^{(2)}$ of the limit surface

$$\mathbf{n}^{(1)} = \frac{\mathbf{t}_1^{(1)} \times \mathbf{t}_2^{(1)}}{|\mathbf{t}_1^{(1)} \times \mathbf{t}_2^{(1)}|} \quad \mathbf{n}^{(2)} = \frac{\mathbf{t}_1^{(2)} \times \mathbf{t}_2^{(2)}}{|\mathbf{t}_1^{(2)} \times \mathbf{t}_2^{(2)}|} \quad (54)$$

where $\mathbf{t}_1^{(1)}$ and $\mathbf{t}_2^{(1)}$ are the two tangents of limit surface (1) and $\mathbf{t}_1^{(2)}$ and $\mathbf{t}_2^{(2)}$ are the two tangents of the limit surface (2). The masks for computing the tangents are given in Figure 11 and 12. Finally, the constraint $\mathbf{C}^{(1,2)}(\boldsymbol{x}_I)$ can be reformulated as

$$\mathbf{C}^{(1,2)}(\boldsymbol{x}_I) = (\mathbf{t}_1^{(1)} \times \mathbf{t}_2^{(1)}) \cdot (\mathbf{t}_1^{(2)} \times \mathbf{t}_2^{(2)}) - \bar{\mathbf{n}}^{(1)} \cdot \bar{\mathbf{n}}^{(2)} |\mathbf{t}_1^{(1)} \times \mathbf{t}_2^{(1)}| |\mathbf{t}_1^{(2)} \times \mathbf{t}_2^{(2)}| = 0 \quad (55)$$

Introducing this equation into the constrained energy functional (52) and computing its stationary points give the equilibrium configurations of the constrained shell

$$\begin{pmatrix} \frac{\partial \Pi^C(\boldsymbol{x}_I, \boldsymbol{\lambda})}{\partial \boldsymbol{x}_I} \\ \frac{\partial \Pi^C(\boldsymbol{x}_I, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} \end{pmatrix} = \begin{pmatrix} \frac{\partial \Pi(\boldsymbol{x}_I)}{\partial \boldsymbol{x}_I} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{C}^{(1,2)}(\boldsymbol{x}_I)}{\partial \boldsymbol{x}_I} \\ \mathbf{C}^{(1,2)}(\boldsymbol{x}_I) \end{pmatrix} = \mathbf{0} \quad (56)$$

The resulting discrete non-linear system of equations can be, e.g., solved with the Newton-Raphson method.

The discussed approach can be generalised to joints with arbitrary number of attached surface patches, in which the case of one surface patch represents a clamped boundary. In each case, a suitable constraint matrix can be assembled using the tangent masks for the limit surface. For the case of a clamped boundary the constraint matrix is

$$\mathbf{C}^{(1,1)} = \mathbf{n}^{(1)} \cdot \bar{\mathbf{n}}^{(1)} = 1 \quad (57)$$

and for the case of a rigid joint with n attached patches the constraint matrix is composed from

$$\mathbf{C}^{(1,2,3,\dots,n)} = \{\mathbf{C}^{(1,2)}, \mathbf{C}^{(2,3)}, \dots, \mathbf{C}^{(n-1,n)}\} \quad (58)$$

4. Examples

In this section we introduce a number of representative examples to investigate the performance of the non-manifold subdivision shell finite elements with exact boundary control. In all examples the reference shell surface consists either of a single or a collection of planar, cylindrical and spherical shapes which have an analytical description. If, however, the shell surface is not given as a subdivision or analytic surface, a subdivision surface can be obtained by using one of the surface fittings algorithms, as e.g. introduced in [31, 52].

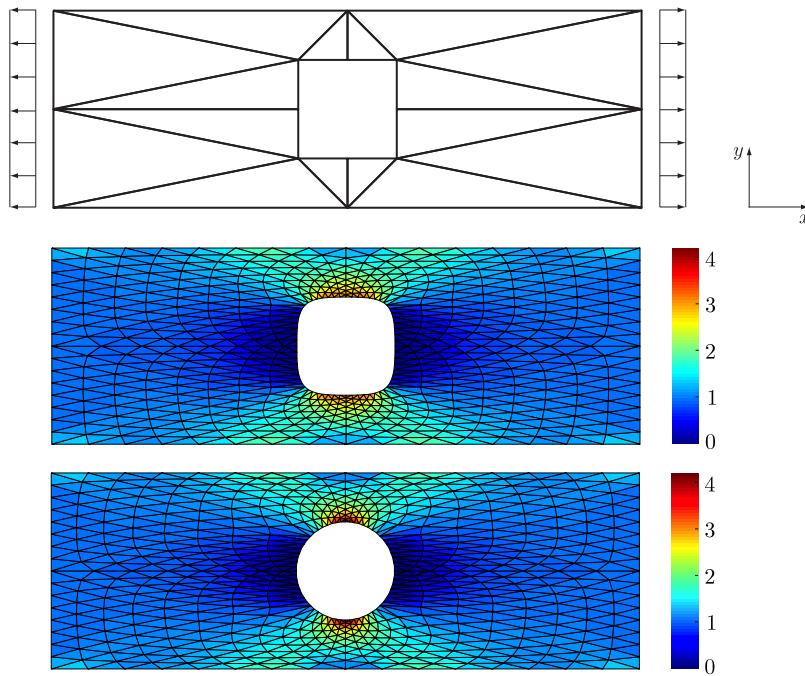


Figure 19. Control mesh and loading of the plate with circular hole (top). σ_{xx} iso-contours for the plate parameterised with ordinary Loop subdivision scheme (middle) and with the rational Loop subdivision scheme (bottom).

For the considered examples the control meshes are created either manually or with the Gmsh software [22] and the control nodes are always placed on the exact surface. This implies an approximation of the shell reference configuration by the subdivision surface, which introduces as discussed in [9] an error of higher order than the finite element error. All computations use the St. Venant-Kirchhoff material model and are performed using one-point Gaussian quadrature. As our current and previous computations indicate, one quadrature point is sufficient for obtaining highly accurate and numerically stable results [9, 8]. However, in order to obtain the optimal convergence rate with mesh refinement it is necessary to use more than one integration point [32]. The degrees of freedom in the presented computations are the displacements of the shell mid-surface and additional Lagrange parameters in case of clamped boundaries and rigid non-manifold joints. The nonlinear discrete equations are solved with the Newton-Raphson method.

4.1. Plate with circular hole

A plate with a circular hole is considered to study the effectiveness of exact geometry description using rational subdivision schemes, Fig. 19. The length of the plate is 24; the width is 8; the circular hole radius is 2; the thickness is $t = 0.1$; the Young's modulus is $E = 200000$ and the Poisson's ratio is $\nu = 0.3$. The coarse control mesh and the loading of the plate are shown in Figure 19. The magnitude of the uniform applied line loading is 1.

Two sets of computations are performed in which the circular hole is represented either with ordinary or rational subdivision schemes. With ordinary Loop subdivision the circular hole is approximated with a cubic spline, which gives a poor approximation if the control mesh is very coarse as in the present example (c.f. middle snapshot in Figure 19). In contrast, with rational Loop subdivision the circular hole is exactly represented independent of the control mesh resolution (c.f. bottom snapshot in Figure 19). In certain instances, such as stress concentration computations, the quality of the geometry approximation can have a significant adverse effect on the solution accuracy. As can be seen in Figure 19, for the three-times subdivided meshes the σ_{xx} stress iso-contours are significantly different. According to Howland [25], the analytic solution of this example yields a maximum σ_{xx} of 4.32 and a stress concentration factor of 2.16. In Figure 20, the error in the computed stress concentration factor for three different mesh sizes for ordinary and rational subdivision are given. As to be expected, in case of rational subdivision the computed stress concentration converges to the analytic value and in case of ordinary subdivision to a value which is off by about 25%.

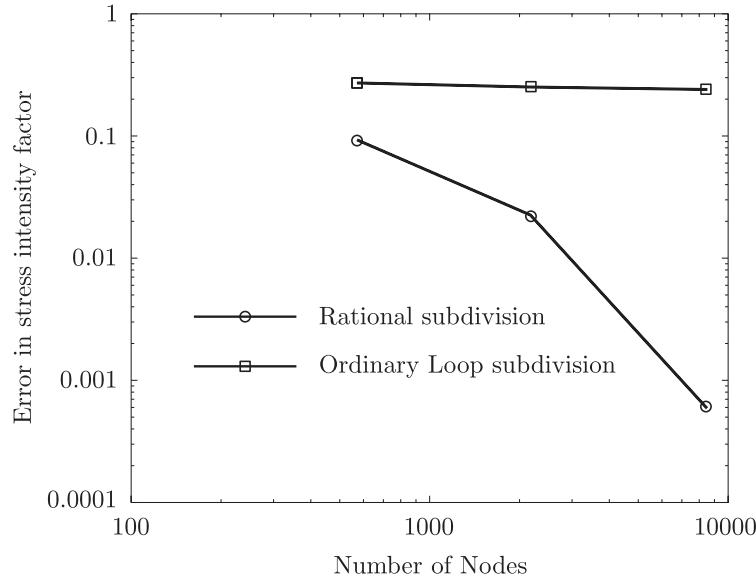


Figure 20. Convergence of the error in the stress concentration factor of the plate with circular hole.



Figure 21. Geometry and loading of the pinched cylinder (left). Deformed limit surface at two different load values (middle and right).

4.2. Pinched cylinder

The pinched-cylinder is a widely used benchmark example which tests a method's ability to deal with large inextensional bending deformations and complex membrane states. The cylinder is loaded by two diametrically opposite concentrated forces located within the middle section of the shell, Fig. 21. The length of the cylinder is 10.35; the radius is 4.953; the thickness is $t = 0.094$; the Young's modulus is $E = 10.5 \times 10^6$; and the Poisson's ratio is $\nu = 0.3125$. The discretization of the cylinder consists of 1600 triangles and 840 vertices all of which are regular ($v = 6$). In the subdivision algorithm the two boundary edges of the cylinder have been tagged as crease. In contrast to our earlier work [9] no ghost elements or vertices have been used for modelling the boundary. In Figure 22, the load-displacement curve computed with the present method is compared with the results of Gruttmann et al. [23] and Peric et al. [37]. Note that in the load-displacement curves, the points A and B refer to the points indicated in Figure 21, left. As evident from Figure 22, the load-displacement curve computed with the subdivision shells is within the range provided by the two reference solutions.

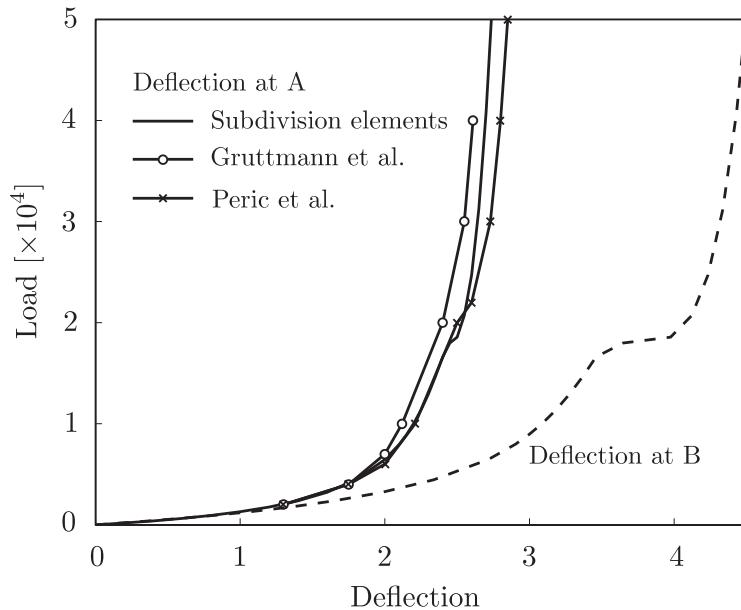


Figure 22. Load-displacement curve of the pinched cylinder for the points A and B on the cylinder.

4.3. Pinched square tube

The pinched square tube demonstrates the treatment of shells with rigid joints. The tube consists of one single mesh with 2438 triangles and 4684 vertices. The length, width and height of the tube are 100. The four sharp tube edges have been realised by tagging selected element edges as *crease*. The shell thickness is $t = 1$; the Young's Modulus is $E = 1 \times 10^7$ and the Poisson's ratio is $\nu = 0$. The tube is loaded by two diametrically opposite concentrated forces located at the top and bottom faces, Fig. 23. During the deformation, the angle between the patch normals at crease vertices is enforced to be constant. To this end, Lagrange parameters are introduced to solve the resulting constrained minimisation problem. As can be seen in Figure 23, the tube exhibits large membrane deformations and localised bending deformations around the rigid creases and two load application points. The high degree of smoothness of the deformed limit surfaces and the faithful representation of the rigid joint is noteworthy. Figure 24 compares the computed displacements with a reference solution generated with the Abaqus finite element package using the element type S3R, which agree very well over the entire deformation range.

4.4. Plate with stiffeners

A plate with stiffeners is considered as an example in which several surface patches are attached along rigid joints, Fig. 25. The square plate has an edge length of 25.4 and the eccentric stiffeners are $h_s = 1.27$ high. The stiffeners are attached to the top of the plate so that their location is eccentric with respect to the plate thickness. The plate and the stiffeners have a uniform thickness of $t = 0.254$. The Young's modulus is $E = 117.25$ and the Poisson's ratio is $\nu = 0.3$. The plate is loaded with uniform pressure and is simply supported along its four edges. A quarter of the plate has been discretized with 3200 elements and the corresponding stiffeners are meshed with 160 elements. Figure 26 compares the centre maximum deflection computed with subdivision shells with the finite element solution of Ojeda et al. [36]. In [36], the stiffeners were modelled with beam elements. If the stiffeners are modelled with beam elements instead of shell elements, the discretized plate structure is inherently more stiff. In order to better compare the results also computations with stocky stiffeners discretized as shell elements were performed. The results of the stocky stiffeners agree very well with the results of [36].

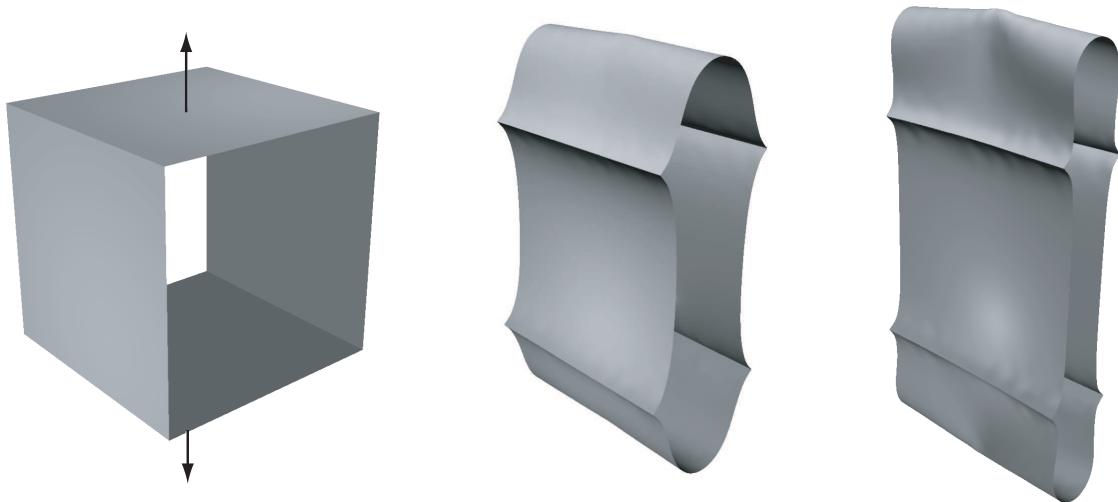


Figure 23. Geometry and loading of the pinched square tube (left). Deformed limit surface at two different load values (middle and right).

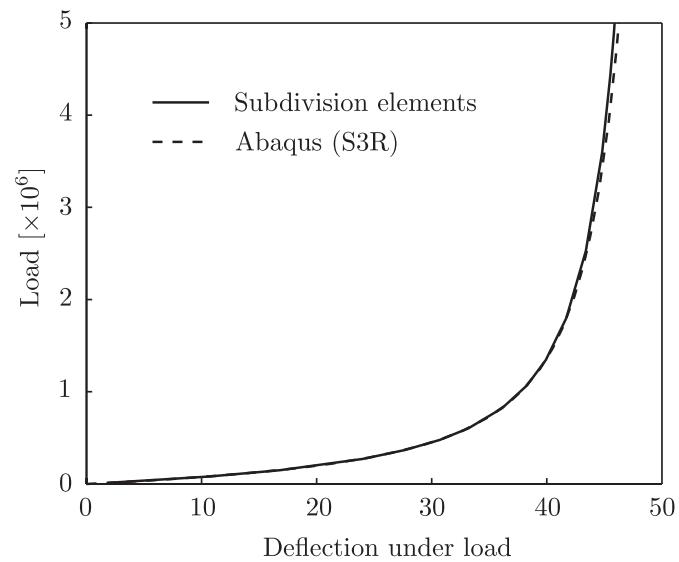


Figure 24. Load-displacement curve of the pinched square tube for the load attachment point.

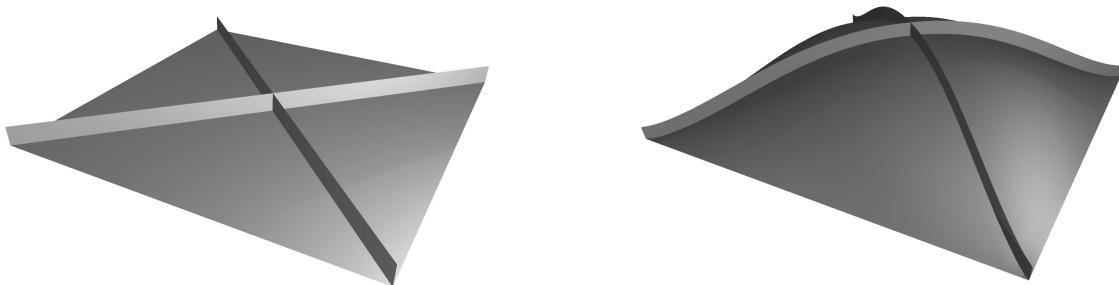


Figure 25. Geometry of the plate with eccentric stiffeners (left). Scaled deflection of the plate under pressure loading (right).

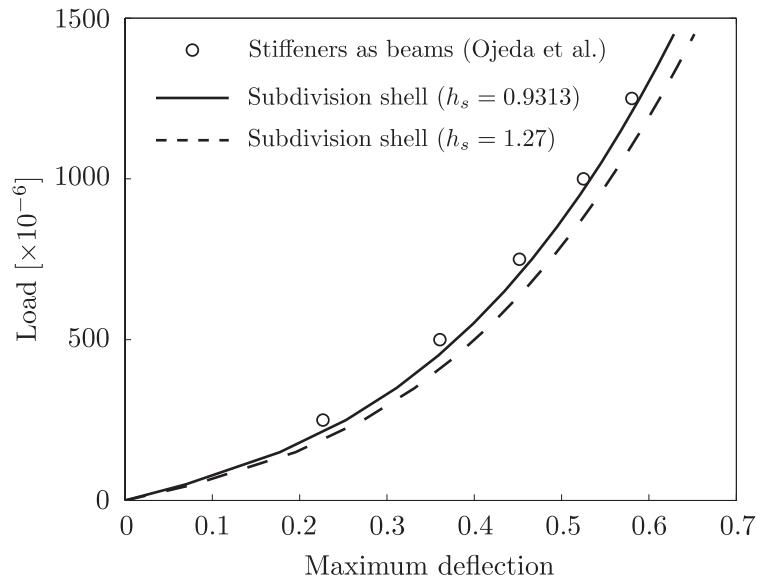


Figure 26. Load-displacement curve of the stiffened plate.

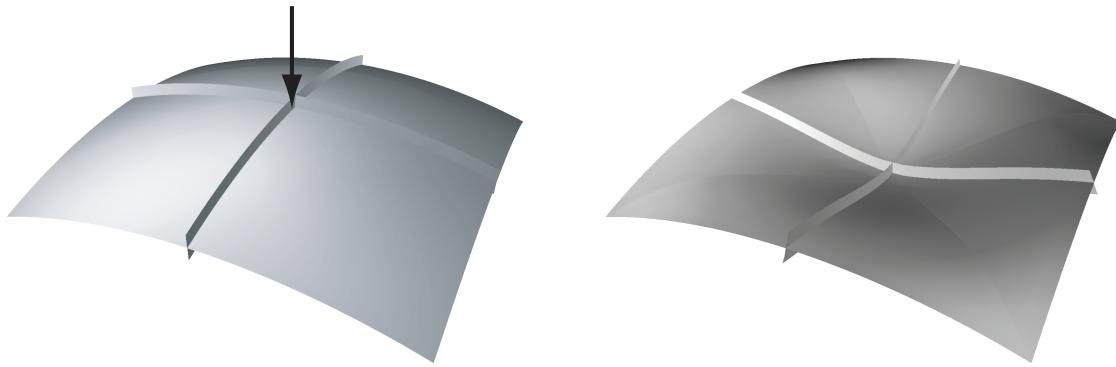


Figure 27. Geometry and loading of the spherical panel with centric stiffeners (left). The deformed panel (right).

4.5. Spherical panel with stiffeners

A spherical panel is considered as an example for a shell with non-zero Gaussian curvature and stiffeners, Fig. 27. The length of the panel is 1.57; the radius of curvature is 2.54; the thickness is $t = 0.09945$; the Young's modulus is $E = 68.97 \times 10^6$ and the Poisson's ratio is $\nu = 0.3$. The stiffeners have a height of 0.1016 and a thickness of $t = 0.0508$ and are symmetrically placed with respect to the thickness of the panel. The panel is simply supported along its four boundary edges and subjected to a radial point load at the centre. The discretization of the panel consists of 2907 vertices and 5600 triangles. Figure 28 compares the centre maximum deflection computed with subdivision shells with the finite element solution of Liao et al. [30]. In [30], the stiffeners were modelled as beams, which is a possible explanation for the slight deviation in results. As is evident from the load displacement curve, the panel exhibits a snap-through instability. The computation is conducted on a quarter of the panel which is shaped as an isosceles right triangle with a stiffener at the centre. Symmetrical boundary conditions are applied on two diagonal boundaries.

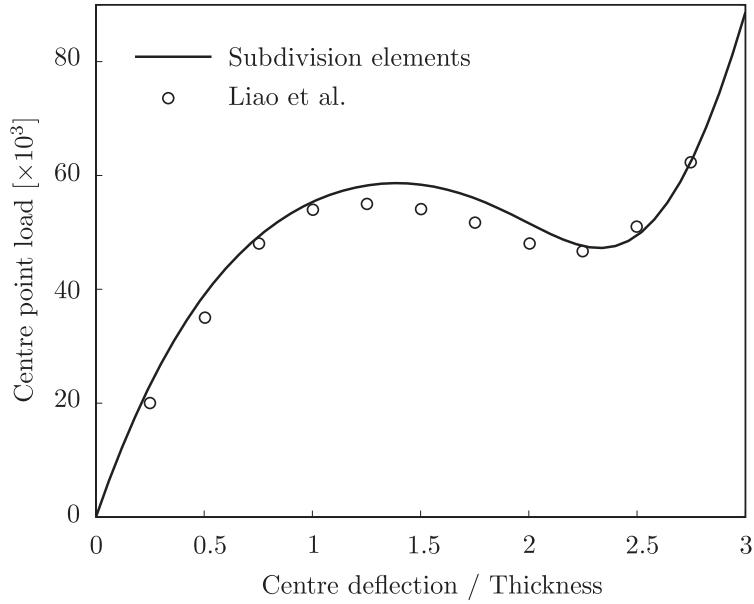


Figure 28. Load-displacement curve of the spherical panel with centric stiffeners for the load attachment point.

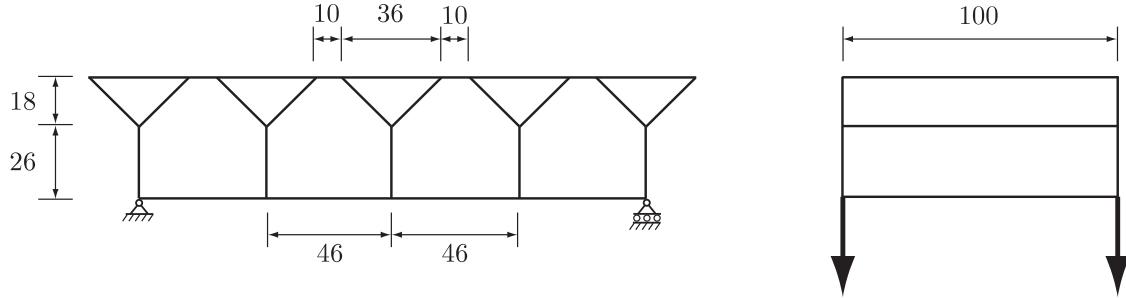


Figure 29. Cross-section, boundary-conditions and dimensions of a Y-core composite plate (left). Side view, dimension, and loading (right).

4.6. Composite plate

The composite plate with a Y-core demonstrates the suitability of the method for computing topologically complex geometries. The cross-section of the plate, its dimensions and boundary conditions are shown in Figure 29. The thickness is uniformly $t = 0.5$; the Young's modulus is $E = 2.1 \times 10^5$ and the Poisson's ratio is $\nu = 0.3$. As loading, controlled displacements are applied at the mid points of the front and back bottom edges of the centre web as sketched in Figure 29, right. An unstructured mesh with 4627 vertices and 9282 triangles is used. The four snapshots in Figure 30 show the reference plate geometry and the plate deformations under successively increasing applied displacements. In Figure 31, the average of the vertical forces at the two displacement application points over the applied displacements is plotted.

5. Conclusions

We presented several extensions to subdivision shells [9, 8, 11] which provide an improved level of control over the boundary geometry in the reference and deformed configurations. To this end, the extended subdivision masks of Biermann et al. [3] and the newly proposed rational subdivision surfaces are used. The input for the introduced

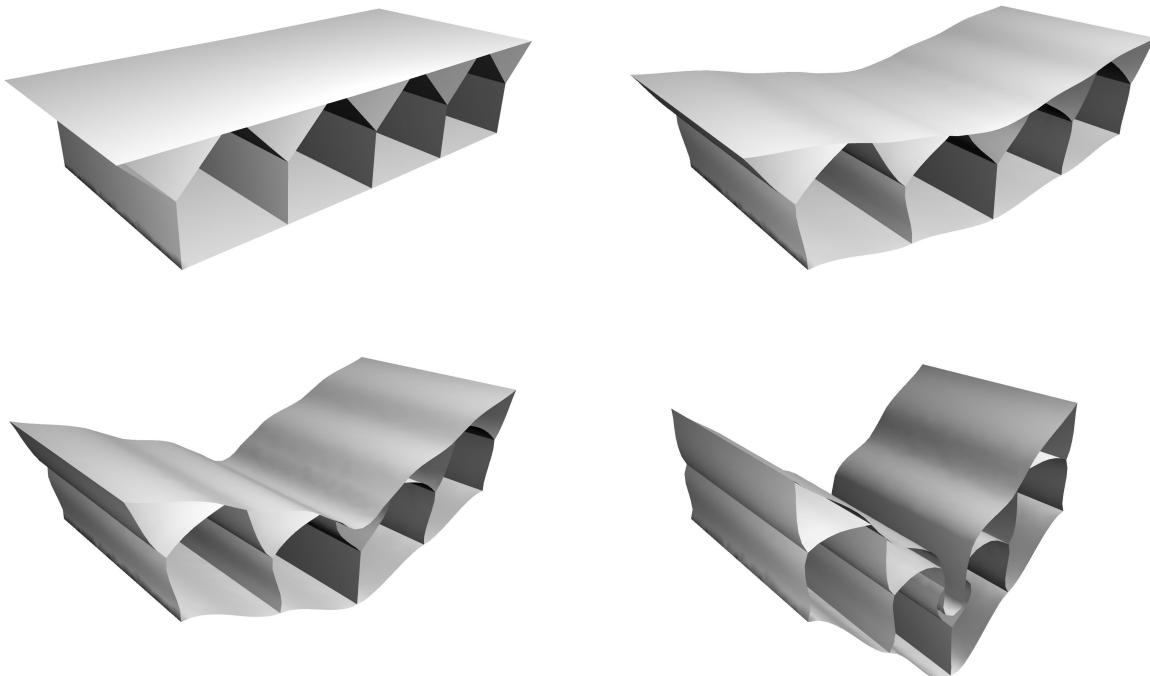


Figure 30. Limit surface of the Y-core composite plate at four different load values.

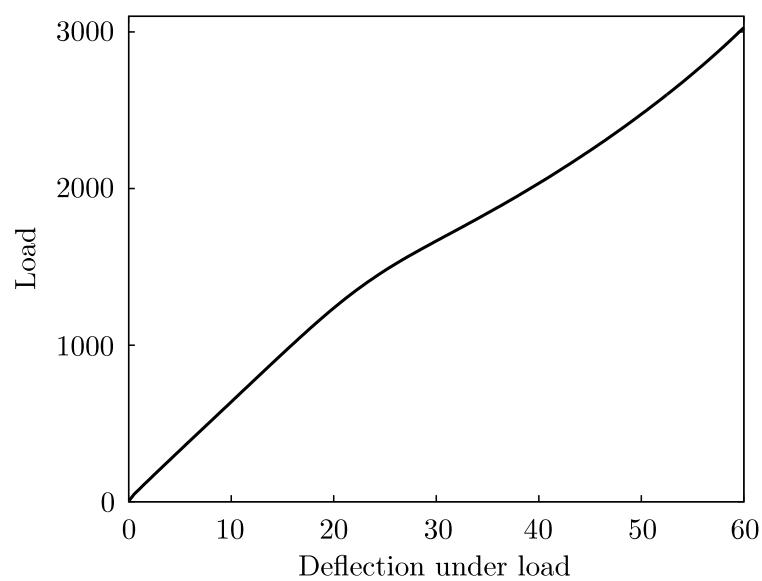


Figure 31. Load-displacement curve of the composite plate for the load attachment point.

subdivision schemes is a tagged control mesh with tagged vertices and edges and additional vertex weights in case of rational subdivision. The application of special subdivision masks close to tagged vertices and edges enables to generate smooth surfaces with non-smooth and non-manifold features without substantially changing the overall refinement algorithm. Similarly, the algorithm for parameterising the shell surface, as needed for the evaluation of the energy integrals, requires only minor modifications in case of general thin shells. Furthermore, the improved boundary control enables to compute shells with non-manifold features consisting of several smooth surface patches.

In closing, it is worthwhile mentioning that subdivision surfaces are still relatively new and, presently predominantly used in computer graphics and animation. In contrast, NURBS are a mature technology and provide the kernel for most CAD packages. However, NURBS are restricted to structured meshes and have limitations in modelling with meshes of general topology. Until recently, subdivision surfaces were only compatible with low order, uniform b-splines and, therefore, could not coexist or compete with NURBS [34]. During the last ten years, there have been significant advances in high degree [54, 51], non-uniform [44] and curvature continuous subdivision schemes, which make subdivision more and more competitive with NURBS. In one particular noteworthy contribution by Cashman et al. [5] it is possible to naturally blend arbitrary degree NURBS with subdivision surfaces. In parallel to these advancements in research, subdivision surfaces have started to make an impact in commercial CAD systems in form of T-splines [46], which use subdivision in order to cope with arbitrary topologies.

Acknowledgements We are grateful to EPSRC (Engineering and Physical Sciences Research Council) for partial support through the Dorothy Hodgkin Postgraduate Awards scheme and to the EU for partial support through the Marie Curie Actions (IAPP) as part of the CASOPT project.

APPENDIX

In a numerical implementation of subdivision shells it is advantageous to use vertex displacements as unknowns instead of vertex coordinates. The first derivative of the Kirchhoff-Love energy functional with respect to nodal displacements yields the internal and external forces and its second derivative yields the stiffness matrix. In computing the derivatives of the Kirchhoff-Love energy functional the derivatives of the shell normal \mathbf{n} with respect to the nodal displacements are required. The first derivative of \mathbf{n} with respect to the i -th component of the displacement of vertex I is

$$\frac{\partial \mathbf{n}}{\partial u_{Ii}} = \frac{1}{j} \mathbf{c} - \frac{\mathbf{n}}{j} (\mathbf{c} \cdot \mathbf{n}) \quad (59)$$

with

$$j = |\mathbf{a}_1 \times \mathbf{a}_2| \quad \text{and} \quad \mathbf{c} = \frac{\partial \mathbf{a}_1}{\partial u_{Ii}} \times \mathbf{a}_2 + \mathbf{a}_1 \times \frac{\partial \mathbf{a}_2}{\partial u_{Ii}} \quad (60)$$

The second derivative of (59) with respect to the j -th component of the displacement of vertex J is

$$\begin{aligned} \frac{\partial^2 \mathbf{n}}{\partial u_{Ii} \partial u_{Jj}} &= -\frac{1}{j^2} \frac{\partial j}{\partial u_{Jj}} (\mathbf{c} - \mathbf{n}(\mathbf{c} \cdot \mathbf{n})) \\ &+ \frac{1}{j} \left(\frac{\partial \mathbf{c}}{\partial u_{Jj}} - \frac{\partial \mathbf{n}}{\partial u_{Jj}} (\mathbf{c} \cdot \mathbf{n}) - \mathbf{n} \left(\frac{\partial \mathbf{c}}{\partial u_{Jj}} \cdot \mathbf{n} \right) - \mathbf{n} \left(\mathbf{c} \cdot \frac{\partial \mathbf{n}}{\partial u_{Jj}} \right) \right) \end{aligned} \quad (61)$$

Furthermore, for shells with rigid non-manifold joints or rotational Dirichlet boundary conditions the constraint matrix \mathbf{C} has to be differentiated. For example, for the discussed rigid joint between two surface patches the derivative of equation (55) is

$$\begin{aligned}
\frac{\partial \mathbf{C}^{(1,2)}}{\partial u_{Ii}} = & \left(\frac{\partial \mathbf{t}_1^{(2)}}{\partial u_{Ii}} \times \mathbf{t}_2^{(1)} + \mathbf{t}_1^{(1)} \times \frac{\partial \mathbf{t}_2^{(1)}}{\partial u_{Ii}} \right) \cdot \left(\mathbf{t}_1^{(2)} \times \mathbf{t}_2^{(2)} \right) \\
& + \left(\mathbf{t}_1^{(1)} \times \mathbf{t}_2^{(1)} \right) \cdot \left(\frac{\partial \mathbf{t}_1^{(2)}}{\partial u_{Ii}} \times \mathbf{t}_2^{(2)} + \mathbf{t}_1^{(2)} \times \frac{\partial \mathbf{t}_2^{(2)}}{\partial u_{Ii}} \right) \\
& - \bar{\mathbf{n}}^{(1)} \cdot \bar{\mathbf{n}}^{(2)} \left[\left(\frac{\partial \mathbf{t}_1^{(1)}}{\partial u_{Ii}} \times \mathbf{t}_2^{(2)} + \mathbf{t}_1^{(1)} \times \frac{\partial \mathbf{t}_2^{(2)}}{\partial u_{Ii}} \right) \cdot \mathbf{n}^{(1)} |\mathbf{t}_1^{(2)} \times \mathbf{t}_2^{(2)}| \right] \\
& - \bar{\mathbf{n}}^{(1)} \cdot \bar{\mathbf{n}}^{(2)} \left[\left(\frac{\partial \mathbf{t}_1^{(2)}}{\partial u_{Ii}} \times \mathbf{t}_2^{(2)} + \mathbf{t}_1^{(2)} \times \frac{\partial \mathbf{t}_2^{(2)}}{\partial u_{Ii}} \right) \cdot \mathbf{n}^{(2)} |\mathbf{t}_1^{(1)} \times \mathbf{t}_2^{(1)}| \right]
\end{aligned} \tag{62}$$

REFERENCES

- M. Arroyo and T. Belytschko. An atomistic-based finite deformation membrane for single layer crystalline films. *Journal of the Mechanics and Physics of Solids*, 50:1941–1977, 2002.
- D.J. Benson, Y. Bazilevs, E. De Luycker, Hsu M.-C., M. Scott, T.J.R. Hughes, and T. Belytschko. A generalized finite element formulation for arbitrary basis functions: From isogeometric analysis to xfem. *International Journal for Numerical Methods in Engineering*, 83:765–785, 2010.
- H. Biermann, A. Levin, and D. Zorin. Piecewise smooth subdivision surfaces with normal control. In *SIGGRAPH 2000 Conference Proceedings*, pages 113–120, July 2000.
- N. Buechter and E. Ramm. Shell theory versus degeneration - A comparison in large rotation finite element analysis. *International Journal for Numerical Methods in Engineering*, 34:39–59, 1992.
- T.J. Cashman, U.H. Augsdörfer, N.A. Dodgson, and M.A. Sabin. Nurbs with extraordinary points: high-degree, non-uniform, rational subdivision schemes. In *SIGGRAPH 2009 Conference Proceedings*, 2009.
- E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10:350–355, 1978.
- F. Cirak and J.C. Cummings. Generic programming techniques for parallelizing and extending procedural finite element programs. *Engineering with Computers*, 24:1–16, 2008.
- F. Cirak and M. Ortiz. Fully C^1 -conforming subdivision elements for finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering*, 51:813–833, 2001.
- F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 47:2039–2072, 2000.
- F. Cirak and R. Radovitzky. A lagrangian-eulerian shell-fluid coupling algorithm based on level sets. *Computers & Structures*, 83:491–498, 2005.
- F. Cirak, M.J. Scott, E.K. Antonsson, M. Ortiz, and P. Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Computer-Aided Design*, 34:137–148, 2002.
- F. Cirak and M. West. Decomposition contact response (dcr) for explicit finite element dynamics. *International Journal for Numerical Methods in Engineering*, 64:1078–1110, 2005.
- P.G. Ciralet. *An Introduction to Differential Geometry with Applications to Elasticity*. Springer, 2005.
- J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric analysis: Toward Integration of CAD and FEA*. John Wiley & Sons Ltd., 2009.
- C. De Boor, K. Höllig, and S. Riemenschneider. *Box splines*. Springer Verlag, 1993.
- G. de Rham. Un peu de mathématiques à propos d'une courbe plane. *Elemente der Mathematik*, 2:73–76, 89–97, 1947.
- T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. In *SIGGRAPH 1998 Conference Proceedings*, 1998.
- D. Doo and M. Sabin. Behavior of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10:356–360, 1978.
- G. Farin. *Curves and Surfaces for CAGD*. Academic Press, 2002.
- F. Feng and W.S. Klug. Finite element modeling of lipid bilayer membranes. *Journal of Computational Physics*, 220:394–408, 2006.
- F.G. Flores and E. Oñate. A rotation-free shell triangle for the analysis of kinked and branching shells. *International Journal for Numerical Methods in Engineering*, 69:1521–1551, 2007.
- C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309–1331, 2009.
- F. Gruttmann, E. Stein, and P. Wriggers. Theory and numerics of thin elastic shells with finite rotations. *Archive of Applied Mechanics*, pages 54–67, 1989.
- H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *SIGGRAPH 1994 Conference Proceedings*, 1994.
- R.C.J. Howland. On the stresses in the neighbourhood of a circular hole in a strip under tension. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229:49–89, 1930.
- T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- K. Karagiozis, R. Kamakoti, F. Cirak, and C. Pantano. A computational study of supersonic disk-gap-band parachutes using large-eddy simulation coupled to a structural membrane. *Journal of Fluids and Structures*, 27:175–192, 2011.

28. J. Kiendl, Y. Bazilevs, Hsu M.-C., Wüchner R., and K.U. Bletzinger. The bending strip method for isogeometric analysis of kirchhoff-love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199:2403–2416, 2010.
29. J. Kiendl, K.U. Bletzinger, J. Linhard, and Wüchner R. Isogeometric shell analysis with kirchhoff-love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.
30. C.-L. Liao and J.N. Reddy. Analysis of anisotropic, stiffened composite laminates using a continuum-based shell element. *Computers & Structures*, pages 805–815, 1990.
31. N. Litke, A. Levin, and P. Schröder. Fitting subdivision surfaces. In *IEEE Visualization 2001*, pages 319–324, 2001.
32. Q. Long, P.B. Bornemann, and F. Cirak. Shear-flexible subdivision shells. *Submitted for publication*, 2010.
33. C.T. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, Department of Mathematics, University of Utah, 1987.
34. W. Ma. Subdivision surfaces for cad – an overview. *Computer-Aided Design*, 37:693–709, 2004.
35. E. Oñate and F.G. Flores. Advances in the formulation of the rotation-free basic shell triangle. *Computer Methods in Applied Mechanics and Engineering*, 194:2406–2443, 2005.
36. R. Ojeda, B.G. Prusty, N. Lawrence, and G. Thomas. A new approach for the large deflection finite element analysis of isotropic and composite plates with arbitrary orientated stiffeners. *Finite Elements in Analysis and Design*, 43:989–1002, 2007.
37. D. Peric and D.R.J. Owen. The Morley thin shell finite element for large deformation problems: simplicity versus sophistication. In *Proceedings of the International Conference on Nonlinear Engineering Computations*, pages 121–142, Swansea, 1991.
38. J. Peters and U. Reif. *Subdivision Surfaces*. Springer Series in Geometry and Computing. Springer, 2008.
39. R. Phaal and C.R. Calladine. A simple class of finite elements for plate and shell problems. II: An element for thin shells, with only translational degrees of freedom. *International Journal for Numerical Methods in Engineering*, 35:979–996, 1992.
40. L. Piegl and W. Tiller. *The NURBS book*. Monographs in visual communication. Springer, 1997.
41. H. Prautzsch, W. Boehm, and M. Paluszny. *Bézier and b-spline techniques*. Springer Verlag, 2002.
42. U. Reif. A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design*, 12:153–174, 1995.
43. D.F. Rogers. *An introduction to NURBS*. Academic Press, 2001.
44. S. Schaefer and R. Goldman. Non-uniform subdivision for b-splines of arbitrary degree. *Computer Aided Geometric Design*, 26:75–81, 2009.
45. J.E. Schweitzer. *Analysis and application of subdivision surfaces*. PhD thesis, University of Washington, Seattle, WA, 1996.
46. T.W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and t-nurccs. In *SIGGRAPH 2003 Conference Proceedings*, 2003.
47. J.C. Simo and D.D. Fox. On a stress resultant geometrically exact shell model. Part I: Formulation and optimal parameterization. *Computer Methods in Applied Mechanics and Engineering*, 72:267–304, 1989.
48. J.C. Simo, D.D. Fox, and M.S. Rifai. On a stress resultant geometrically exact shell model. Part II: The linear theory, computational aspects. *Computer Methods in Applied Mechanics and Engineering*, 73:53–92, 1989.
49. J. Stam. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. In *SIGGRAPH 1998 Conference Proceedings*, pages 395–404, 1998.
50. J. Stam. Evaluation of loop subdivision surfaces. In *SIGGRAPH 1999 Course Notes*, 1999.
51. J. Stam. On subdivision schemes generalizing uniform b-spline surfaces of arbitrary degree. *Computer Aided Geometric Design*, 18:383–396, 2001.
52. H. Suzuki, S. Takeuchi, and T. Kanai. Subdivision surface fitting to a range of points. In *Pacific Graphics 1999*, pages 158–167, 1999.
53. L. Ying and D. Zorin. Nonmanifold subdivision. In *Proceedings of the conference on Visualization '01*, pages 325–332, San Diego, CA, 2001.
54. D. Zorin and P. Schröder. A unified framework for primal/dual quadrilateral subdivision schemes. *Computer Aided Geometric Design*, 18:429–454, 2001.