Subdivision schemes for smooth contact surfaces of arbitrary mesh topology in 3D

M. Stadler and G. A. Holzapfel*†

Institute for Structural Analysis – Computational Biomechanics, Graz University of Technology, Schiesstattgasse 14B, A-8010 Graz, Austria

SUMMARY

This paper presents a strategy to parameterize contact surfaces of arbitrary mesh topology in 3D with at least $C^1$-continuity for both quadrilateral and triangular meshes. In the regular mesh domain, four quadrilaterals or six triangles meet in one node, even $C^2$-continuity is attained. Therefore, we use subdivision surfaces, for which non-physical pressure jumps are avoided for contact interactions. They are usually present when the contact kinematics is based on facet elements discretizing the interacting bodies. The properties of subdivision surfaces give rise to basically four different implementation strategies. Each strategy has specific features and requires more or less efforts for an implementation in a finite element program. One strategy is superior with respect to the others in the sense that it does not use nodal degrees of freedom of the finite element mesh at the contact surface. Instead, it directly uses the degrees of freedom of the smooth surface. Thereby, remarkably, it does not require an interpolation. We show how the proposed method can be used to parameterize adaptively refined meshes with hanging nodes. This is essential when dealing with finite element models whose geometry is generated by means of subdivision techniques. Three numerical 3D problems demonstrate the improved accuracy, robustness and performance of the proposed method over facet-based contact surfaces. In particular, the third problem, adopted from biomechanics, shows the advantages when designing complex contact surfaces by means of subdivision techniques. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: contact; smooth surface; subdivision schemes

1. INTRODUCTION

For the finite element simulation of contact problems various methods are used to incorporate contact constraints in the variational formulation (see the pioneering work [1], and [2–4] among others). However, the application of these methods may cause sliding of nodes over edges of the
master surface and may lead, consequently, to discontinuities in the distance function when the contact surface is discretized with facet elements. This is particularly the case when the surface to be discretized retains a high curvature throughout the simulation. Then, the assumptions for proofing uniqueness and convexity of the contact problem do not hold anymore [5]. The consequences may be oscillations of the contact forces, non-realistic pressure jumps, contact cycling (i.e. consecutive changes of contact partners during a Newton iteration) and loss of quadratic convergence of the non-linear solution scheme.

The goal of the present work is to avoid these types of problems by specifying a smooth interpolating surface on one of the contacting bodies. Several authors have demonstrated that for rigid bodies the use of $C^2$-continuity of the contact surfaces is advantageous (see, for example, References [6–8]). Recent papers [9, 10] claim that it would also be a desirable property for deformable bodies, partly due to the fact that $C^2$-continuity is a necessary condition for quadratic rate of convergence of non-linear solvers. This is especially the case if a contact problem involves dynamics—and thus acceleration is present, which may jump at the element interfaces—and if the interpolation is not $C^2$-continuous [11].

Hence, also for deformable bodies a number of smoothing techniques were developed: they include cubic Hermite interpolation [12, 13], cubic Bézier spline [13, 14], Overhauser (or Catmull-Rom) spline [9, 10] and NURBS [15, 16]. These techniques were originally developed for 2D problems. Their bivariate extension to describe contact surfaces in 3D has the inherent disadvantage that it cannot represent surfaces with arbitrary mesh topology. One approach to circumvent this problem is the use of NURBS. Thereby, the surfaces are decomposed into a set of trimmed NURBS patches, i.e. the use of portions of the originally rectangular patches, which are finally assembled. However, this results in computationally expensive algorithms which are prone to numerical errors and complex continuity constraints. Hence, smoothing techniques for contact surfaces in the 3D domain represent a more challenging task.

To the authors’ knowledge two different approaches address this issue. One is by Puso and Laursen [17] who presented a technique using Gregory patches, which are applicable to both structured and unstructured meshes of quadrilaterals. The applicability to unstructured meshes was shown for arbitrary valences. The valence of a vertex is the number of edges coming out of it. However, it only achieves $G^1$-continuity resulting in a discontinuous surface metric at element boundaries. Geometrically this means that the direction of the tangent vector is continuous (not its length). Puso and Laursen reconcile this by introducing a modified frictional formulation. The other approach is documented by Krstulović-Opara et al. [18, 19]. They employ quartic Bézier surfaces for the interpolation through the nodes and the centroid of triangular elements. This approach leads to quasi-$C^1$-continuous surfaces, which are everywhere continuous except at the element nodes, where no derivatives are continuous ($C^0$-continuous).

In the present work we use so-called subdivision surfaces for the interpolation. They originate from the computer-aided-geometric-design community, introduced in 1978 by Catmull and Clark [20], and Doo and Sabin [21]. Both works generalize tensor product B-splines of bi-degree two and three to arbitrary topologies by extending the refinement rules to irregular parts (valence $N \neq 4$ for quadrilateral-based meshes or $N \neq 6$ for triangular meshes) of the control polyhedron. Later, in 1987, Loop [22] has generalized triangular Box splines [23] of total degree four to arbitrary triangular meshes. These schemes have the desirable property that they admit a polynomial representation of the regular part of the mesh. Further developments produced schemes, which possess properties such as interpolations without matrix inversion, true $C^1$ or $C^2$-continuity, and the ability to interpolate meshes of arbitrary topology as well as
arbitrary genus (quadrilateral/triangle) with $C^1$-continuity at the quadrilateral/triangle interface. These features make this approach very attractive for smoothing 3D contact surfaces. For a general introduction to subdivision surfaces see Reference [24].

An additional advantage of the proposed approach is that it allows to base the geometrical modelling and the finite element analysis on an identical representation paradigm. Hence, both the deformed and the undeformed geometries of the contact surfaces are described by the subdivision surface. The importance of such a unified framework for geometrical design and mechanical analysis is discussed in References [25, 26] and is consequently exploited for shells in References [27, 28]. It yields a robust interface between geometrical design and mechanical analysis, whereby a number of difficulties are avoided, inherent in the heterogeneous treatment of smooth surfaces of current software tools.

Although subdivision surfaces were developed more than 20 years ago they were not applied until recently. One reason for that might be that there did not exist a solid mathematical foundation of their convergence and smoothness properties. This gap was closed most notably by the works [29–31]. Due to the same reason, subdivision surfaces have only recently been applied in the FE community. The beginning is marked by the seminal paper [27], which uses subdivision surfaces for thin-shell structures within the geometrically linear regime. Successively, subdivision surfaces were also used for thin shells in the non-linear regime (see References [28, 32]), hierarchical mesh refinement [33] and mesh generation (see References [26, 34, 35]). To the authors’ knowledge, the approach documented in this paper is the first attempt to employ this class of surface description for contact problems.

The organization of the paper is as follows: Section 2 briefly reviews the governing equations and the variational framework for frictional contact problems. In Section 3, we provide a detailed description of four different subdivision schemes, which are appropriate for the design of contact surfaces. In addition, we also discuss some aspects of smoothness and surface evaluation. Section 3 also serves to provide an overview upon the existing literature regarding certain aspects of subdivision surfaces. With this in mind we have all necessary prerequisites to assemble the contact algorithm. Hence, Section 4 discusses four different strategies for the implementation of the subdivision-based contact algorithm into a finite element program. In Section 5, we discuss how subdivision surfaces can be used to operate on two cases of $h$-adaptivity. In Section 6, we show three applications of the proposed smoothing technique, where the particular advantages will become clear. Finally, in Section 7 we provide concluding remarks. The appendix gives some details of the necessary data structures for subdivision surfaces.

2. CONTACT FORMULATION

We consider a contact problem of two bodies $B^x$, $x = 1, 2$. Each of them occupies the open set $\Omega^x_0 \subset \mathbb{R}^3$ with smooth boundaries $\partial \Omega^x_0$, where $\Omega^x_0$ denotes the reference configuration of body $x$ at (fixed) reference time $t = 0$. During some closed time interval $t \in [0, T]$ of interest the two bodies undergo the motions expressed by the maps $\chi^x: \hat{\Omega}^x_0 \times [0, T] \rightarrow \mathbb{R}^3$, where $\hat{\Omega}^x_0 = \Omega^x_0 \cup \partial \Omega^x_0$ denotes the closure of the open sets $\Omega^x_0$. Note that the motions $\chi^x$ transform referential positions $X^x \in \hat{\Omega}^x_0$ into points $x^x \in \hat{\Omega}^x$ located at the current configuration $\Omega^x$ of body $x$ at time $t$, where $x^x = \chi^x(X^x, t)$. Subsequently, we divide the current (smooth)
boundaries $\partial \Omega^2$ of the bodies into the three regions $\gamma_u$, $\gamma_\sigma$ and $\gamma_c$ so that $\partial \Omega^2 = \gamma_u \cup \gamma_\sigma \cup \gamma_c$, at which Dirichlet, von Neumann and contact boundary conditions are prescribed. In addition, we consider a parametrization of the current boundary surface $\gamma_c^2$ of body two. The parameter plane with region $\mathcal{A} \subset \mathbb{R}^2$ is characterized by the convective co-ordinates $\xi$ and $\eta$. Hence, a map $\Psi_t : \mathcal{A} \to \mathbb{R}^3$ gives the relation $x^2 = \Psi_t(\xi, \eta)$. For the geometrical situation see Figure 1.

2.1. Governing equations and normal contact of two bodies

Here we briefly review the field equations and the boundary conditions of body $z$. The displacement solution $u^z$ must satisfy the momentum balance principles and the boundary conditions. Thus, following, for example, References [36, 37], strong forms read

$$ \begin{align*}
\text{div } \sigma^z + b^z &= \rho^z \ddot{u}^z \quad \text{in } \Omega^z \\
\dot{t}^z &= \sigma^z n^z = \ddot{t}^z \quad \text{on } \gamma_\sigma^z \\
u^z &= \bar{u}^z \quad \text{on } \gamma_u^z
\end{align*} $$

(1)

where $\ddot{t}^z$ and $\ddot{u}^z$ denote the prescribed Cauchy traction and displacement vectors, respectively. The symmetric Cauchy stress tensor is characterized by $\sigma^z$, the body force defined per unit current volume as $b^z$, the current mass density is $\rho^z$, the Cauchy traction vector (force measured per unit surface area defined in the current configuration) is denoted by $t^z$, while $n^z$ is the outward unit vector field perpendicular to the boundary $\partial \Omega^z$, and $\text{div}(\bullet)$ denotes the divergence operator of ($\bullet$) with respect to the spatial co-ordinates. The initial conditions are

given by
\[ \chi^2|_{t=0} = \mathbf{1} \quad \text{in } \tilde{\Omega}_0^2 \]
\[ \frac{\partial \chi^2}{\partial t} \bigg|_{t=0} = \mathbf{v}_0^2 \quad \text{in } \tilde{\Omega}_0^2 \]
(2)
where \( \mathbf{1} \) is the unit vector and \( \mathbf{v}_0^2 \) is the prescribed initial velocity field.

By assuming that the two bodies \( B^2 \) come into contact we ask for the gap function \( g_N \) as indicated in Figure 1. It is the normal projection of \( x^1 \) onto \( \gamma_c^2 \) according to
\[ g_N = \|x^1 - x_2^*\| = \min_{x_2 \subseteq \gamma_c^2} \|x^1 - x_2(\zeta, \eta)\| \]
(3)
where \( x^1 \in \gamma_c^2 \) is considered to be a (slave) point, with \( x^1 = \chi^1(X^1, t) \), while \( x_2^* = x_2(\zeta^*, \eta^*) \) on \( \gamma_c^2 \) is the associated (master) point. The minimum distance problem is performed, in particular, by a local Newton iteration of the form
\[ \xi_{i+1} = \xi_i - \left. \frac{\partial g_N}{\partial \xi} \right|_{\xi=\xi_i} \]
(4)
where the solution vector \( \xi^* \) stands for the set \((\zeta^*, \eta^*)\). For subsequent use in Section 2.2 we note that
\[ \xi^* = \xi^*[g_N(u)] \]
(5)
and denote all geometric quantities \((\bullet)\), which are evaluated at \((\zeta^*, \eta^*)\), by \((\bullet)^*\).

2.1.1. Path integration for frictional contact. Frictional contact is based on the knowledge of the relative tangential slip. If the frictional behaviour is characterized by the local parametrization induced by the FE triangulation, the frictional time integration becomes meaningless when the incremental slip path involves more than one surface element. An efficient method to avoid this problem is documented in Reference [38]. Therein, the slip path length is computed in terms of two positions and the local normal vectors associated with a slave point at the beginning and the end of a time increment. Based on this, an assumed slip path is generated. Geometrically, the assumed slip path can be viewed as a second-order approximation of the geodesic (i.e. a locally length-minimizing curve) defined by these two points.

Subsequently, we briefly review the basic ideas of this concept and adapt the algorithm, as proposed in Reference [38]. Within the iterative solution scheme, we require to refer to quantities from the previous step. Therefore, subsequently, we denote quantities of the previous and current time steps as \( -{\bullet} \) and \( +{\bullet} \), respectively. If all quantities within an equation refer to a single time step and if no ambiguity can arise, then the superscripts \( +{\bullet} \) and \( -{\bullet} \) are omitted for convenience. Time steps are used synonymous for load steps within the iterative solution scheme.

The assumed slip path is based on two positions which belong to different time steps. Therefore, the distance between these steps cannot be evaluated by standard methods. Hence, in order to compute the length of the assumed slip path within a single time step, we require a mapping, \((\bullet)\) say, of a geometric quantity \((\bullet)\) in the current time step, back to the previous time step. Usually, a quantity is evaluated by means of global variables (nodal displacement
Figure 2. Mapping of the point of contact from the current time frame back to the previous time frame for computing the assumed slip path length $\Gamma$ within a single time frame.

$\mathbf{u}$, and local variables $(\xi^*, \eta^*)$ from the same time frame. However, the symbol (**) means that it is evaluated with the global variable from the previous time step and local variables from the current time step. Hence, for example, the map of the current projection point $+\mathbf{x}^2 = \mathbf{x}^2(+/\xi^*, +\eta^*)$ back to the previous time step has the form

$$
+\mathbf{x}^2(+/\xi^*, +\eta^*) = \mathbf{X}^2(+\xi^*, +\eta^*) + -\mathbf{u}^2(+\xi^*, +\eta^*)
$$

where $-\mathbf{u}^2(+\xi^*, +\eta^*)$ means that we compute a displacement vector by projecting the local co-ordinates $(+/\xi^*, +\eta^*)$ of the current time step onto the nodal configuration (i.e. the finite element mesh) of the previous time step. The geometrical situation is illustrated in Figure 2. Subsequently, the arguments of the functions are omitted for simplicity.

Before proceeding it is necessary to define the assumed slip path, for which the orthonormal frames attached to $+\mathbf{x}^2$ and $-\mathbf{x}^2$ are required. Therefore, we define the quantities (see Figure 2)

$$
\mathbf{d} = +\mathbf{x}^2 - -\mathbf{x}^2
$$

$$
d = \|\mathbf{d}\|, \quad \mathbf{m} = \frac{\mathbf{d}}{d}
$$

$$
-\tau_2 = -\mathbf{n} \times \mathbf{m}
$$

$$
-\tau_1 = -\tau_2 \times -\mathbf{n}
$$

$$
+\tau_2 = +\mathbf{n} \times \mathbf{m}
$$

$$
+\tau_1 = +\tau_2 \times +\mathbf{n}
$$

Copyright © 2004 John Wiley & Sons, Ltd.

Then, for the path parametrization we define the approximation to the geodesic as [38]

$$+\Xi(\zeta) = B_1(\zeta) x^2 + \frac{d}{2 - \tau_1 \cdot m} B_1(\zeta) - \tau_1 + B_2(\zeta) + \bar{x}^2 + \frac{d}{2 + \bar{\tau}_1 \cdot m} B_2(\zeta) + \bar{\tau}_1$$  \hspace{1cm} (8)

where $B_i, \bar{B}_i, \ i = 1, 2$ are Hermite shape functions, defined in the domain $\zeta \in [-1, 1]$ as

$$B_1(\zeta) = \frac{1}{4} (2 + \zeta)(1 - \zeta)^2, \quad \bar{B}_1(\zeta) = \frac{1}{4} (1 + \zeta)(1 - \zeta)^2$$

$$B_2(\zeta) = \frac{1}{4} (2 - \zeta)(1 - \zeta)^2, \quad \bar{B}_2(\zeta) = -\frac{1}{4} (1 - \zeta)(1 + \zeta)^2$$  \hspace{1cm} (9)

The length of the assumed slip path, $\Gamma$ say, can then be computed as the arc length of the geodesic

$$+\Gamma = \int_{-1}^{1} \left| \frac{\partial^+ \Xi(\zeta)}{\partial \zeta} \right| \ d\zeta$$  \hspace{1cm} (10)

which is evaluated numerically by, for example, Gauss quadrature. For later use, we define the direction $+\tau_1$ of the increment of the total tangential slip, i.e.

$$+\tau_1 = \pm \tau_2 \times +n^* \quad \text{with} \quad +\tau_2 = +n^* \times (+x^2 - x^2)$$  \hspace{1cm} (11)

Therein, we have employed the operator $\cdot$ ($= (\cdot)/\| (\cdot) \|$). The approximation to the geodesic and the computation of the length $\Gamma$ of the assumed slip path is illustrated in Figure 2.

2.2. Variational formulation

The goal of this section is to develop the finite element framework for frictional contact so that it can take advantage of the introduced surface parametrization. We start with the virtual work contribution due to contact. The contact constraint is enforced by the Karush–Kuhn–Tucker conditions, which transform the virtual work principle into an inequality variational principle. After regularization we arrive at equations, which describe the contact-related quantities necessary for the implementation into a finite element program.

2.2.1. External virtual work—contact contribution. The state of tangential contact is characterized either by stick (or adhesion) or slip, in the following indicated by $\bullet^{st}$ or $\bullet^{sl}$, respectively. Stick means that a node, which is in contact, does not move in the tangential direction, while for slip the node moves. Note that this distinction is not required for normal contact. Consequently, it turns out that it is convenient to decompose the contact-related variables into normal and tangential components according to, for example, Reference [4]. For the contact contribution $\delta W_{ext,c}$ to the external virtual work we may write

$$\delta W_{ext,c} = \int_{\gamma_c^l} (t_N \delta g_N + t_T \cdot \delta g_T) \ d\gamma_c^l$$  \hspace{1cm} (12)

whereby the Cauchy traction vector $t$ is decomposed into a normal component $t_N n$, with $t_N = t \cdot n \leq 0$ (no adhesion), and a tangential component $t_T = t - (t \cdot n)n$, while $g_T$ denotes the tangential gap vector to be defined below, and $\delta(\bullet)$ is the variation of $(\bullet)$.

The conditions upon admissibility of $\chi^l$ regarding normal contact are summarized in the Karush–Kuhn–Tucker conditions, i.e.

$$g_N \leq 0, \quad t_N \geq 0, \quad t_N g_N = 0 \quad \text{on } \gamma_c$$

(13)

where $g_N = 0$, $t_N > 0$ means contact (no contact means $g_N < 0$, $t_N = 0$). The three relations (13) represent conditions for impenetrability, compressive normal interaction and complementarity of gap and contact pressure, respectively. Since they yield an inequality variational principle, a regularization method is required. Here we employ the penalty method. For normal contact, it replaces the conditions $(13)_1$–$(13)_3$ by the expression $t_N = \varepsilon_N(g)$, where $(\cdot)$ denotes the Macauley bracket, representing the positive part of its argument and $\varepsilon_N$ is the non-negative normal penalty parameter.

The frictional state, characterized either by stick (or adhesion) or slip, can be formally modelled by using the concept of elasto-plasticity theory (see Reference [3] or [4] among others). Therein, plastic flow is governed by an evolution equation and a flow criterion. Since it is not known in advance if plastic flow occurs, a trial state is computed. This state is then checked if it satisfies the flow criterion or not. If yes, the trial state is accepted as an elastic state. If not, plastic flow occurs and the flow criterion is satisfied by means of a return-mapping algorithm [39]. For the related algorithm it is convenient to split the deformation into its elastic and plastic components. Hence, in an analogous manner, we additively decompose the tangential slip $g_T$ into an (elastic) part $g_{T}^{st}$, responsible for stick or adhesion, and a (plastic) part $g_{T}^{sl}$, responsible for slip. Thus,

$$g_T = g_T^{st} + g_T^{sl}$$

(14)

The ideal stick constraint $g_{T}^{st} = 0$ can then be introduced into the variational framework by a penalty regularization according to

$$t_T = \varepsilon_T g_{T}^{st}$$

(15)

where $\varepsilon_T$ denotes the non-negative tangential penalty parameter associated with tangential sliding. The sliding process is modelled by an evolution law which is introduced through a slip potential $\Phi(t_T)$ (the analogue of the yield function in plasticity theories). From the maximum dissipation principle one obtains the constitutive equation for the actual slip path

$$\dot{g}_{T}^{sl} = \dot{\gamma} \frac{\partial \Phi(t_T)}{\partial t_T} = \dot{\gamma} n_T \quad \text{with } n_T = |t_T|$$

(16)

where we have introduced the unknown consistency parameter $\dot{\gamma}$ (slip rate). Here, we particularize the slip potential with Coulomb’s law (equivalent to the plastic slip criterion) as

$$\Phi(t_T) = \|t_T\| - \mu g_N^N$$

(17)

where $\mu$ denotes the coefficient of friction. For tangential contact, the Karush–Kuhn–Tucker conditions can be specified as

$$\dot{\gamma} \geq 0, \quad \Phi \leq 0, \quad \dot{\gamma} \Phi = 0$$

(18)

Next we have to perform the algorithmic update of the tangential gap vector $+t_T$, which is performed by the unconditionally stable backward-Euler integration of the evolution equation.
and the elastic predictor/plastic corrector concept [40]. In a first step, a trial traction solution, $\tilde{t}_T$ say, is then computed, which does, in general, not fulfill the slip criterion (18)$_2$. The total slip $\tilde{g}_T = \tilde{g}_T + \Delta g_T$ is computed in terms of stick and slip parts according to (14). Then from (15) we may derive the trial traction vector at the current time step according to

$$\tilde{t}_T = \tilde{t}_T + \varepsilon_T \Delta g_T$$  \hspace{1cm} (19)$$

where we have used the relation

$$-\tilde{t}_T = \varepsilon_T (-\tilde{g}_T - \tilde{g}_{sl, T})$$  \hspace{1cm} (20)$$

of the tangent traction vector $-\tilde{t}_T$ of the previous time step. Note that the increment $\Delta g_T = +\Gamma + \tau_1$ of the total tangential slip vector is given by its direction $+\tau_1$, defined in (11)$_1$, and its length $+\Gamma$ provided in (10). The slip potential (17) takes on the form $+\Phi^{fr} = || +\tilde{t}_T || - \mu \varepsilon_N^+ g_N$.

In a second step, the criterion (18)$_2$, i.e. stick/slip, can be checked. For the case $+\Phi^{fr} \leq 0$ no friction takes place (pure stick), while for $+\Phi^{fr} > 0$ sliding occurs in the tangential direction.

Hence, we may distinguish between stick and slip as follows:

- For the case of stick (i.e. ‘elastic’), the tangential traction vector $+t_T$ is given by $+t_T^r$, as provided in (19)$_2$. Consequently, by using (19)$_2$ we find from (15) that

$$g_T^{sl} = -\frac{t_T}{\varepsilon_T} + \frac{\varepsilon_T}{\varepsilon_T} \Delta g_T$$  \hspace{1cm} (21)$$

which defines the tangential stick vector.

- For the case of sliding (i.e. ‘plastic’), a return mapping to Coulomb’s friction cone has to be performed. By using backward-Euler integration in order to approximate (16), we obtain

$$g_T^{sl} = -g_T^{sl} + \lambda + n_T \quad \text{with} \quad +n_T = +t_T$$  \hspace{1cm} (22)$$

where $\lambda = \dot{\lambda} \Delta t$ denotes the unknown incremental slip (plastic) parameter. Hence, with the use of (18) the incremental Karush–Kuhn–Tucker conditions read $\dot{\lambda} > 0$, $\Phi < 0$, $\dot{\lambda} \Phi = 0$, which determine $\lambda$. The traction vector is then

$$t_T = t_T^r - \lambda \varepsilon_T + n_T$$  \hspace{1cm} (23)$$

where we have introduced the definition $+n_T = +t_T = +t_T^r$. For a non-linear slip potential, the return-mapping algorithm, required to determine the unknown incremental slip parameter $\lambda$, leads to an iterative solution procedure. However, for the case of Coulomb’s law, $\lambda$ can be determined explicitly. By taking the norm of (23) we deduce from (17), by setting $\Phi = 0$, that

$$\lambda = \frac{1}{\varepsilon_T} (|| t_T^r || - \mu \varepsilon_N^+ g_N)$$  \hspace{1cm} (24)$$

Finally, with the known $\lambda$, the current traction vector $t_T$ and the tangential slip vector $g_T^{sl}$ can be obtained by substituting (24) into (23) and (22). Thus,

$$t_T = \mu \varepsilon_N^+ g_N + n_T$$  \hspace{1cm} (25)$$

$$g_T^{sl} = g_T^{sl} - \frac{1}{\varepsilon_T} (|| t_T^r || - \mu \varepsilon_N^+ g_N) + n_T$$  \hspace{1cm} (26)$$
The components of the history vector, which are needed as input for the next load step, are the tangential traction vector \( +t_T \), i.e. \( +t_T^f \) for stick and (25) for slip, and the current co-ordinates of the solution point, i.e. \( +x^{2*}(+\xi^*, +\eta^*) \).

2.2.2. Aspects of the finite element implementation. In the finite element context, we need to solve a non-linear problem for which we employ Newton’s method. The MATHEMATICA package ACEGen [14] is used for the automatic derivation of the matrix formulae needed to describe the finite element discretization of the smooth contact element.

We consider a smooth master surface, defined by a control polyhedron, which consists of \( r \times r \) master nodes and an associated slave node. Hence, the total number of degrees of freedom of these contact partners is in three dimensions \( w = (r^2 + 1)3 \). The displacements of all involved nodes are given by the vector \( [u]_i, i = 1, \ldots, w \). According to (12), the residual vector \( f_c \) is then given as

\[
[f_c]_i = t_N \frac{\partial g_N}{\partial [u]_i} + [t_T] \frac{\partial [g_T]}{\partial [u]_i}, \quad i = 1, \ldots, w
\]

where we have omitted the superscripts \((\bullet)^{\text{sl}}\) and \((\bullet)^{\text{st}}\), because the equation is valid for both cases when the appropriate quantities are taken into account. Since \( g_N = g_N([u], \xi^*, \eta^*) \) and \( g_T = g_T([u], \xi^*, \eta^*) \), the implicit derivatives \( \partial \xi^*/\partial [u]_i \) and \( \partial \eta^*/\partial [u]_i \) are required. By application of the chain rule, they can be obtained from (5) as

\[
\frac{\partial \xi^*}{\partial [u]_i} = \frac{\partial g_N}{\partial [u]_i} \frac{\partial \xi^*}{\partial g_N}, \quad \frac{\partial \eta^*}{\partial [u]_i} = \frac{\partial g_N}{\partial [u]_i} \frac{\partial \eta^*}{\partial g_N}
\]

This allows to rewrite (27) as

\[
[f_c]_i = t_N \frac{\partial g_N}{\partial [u]_i} + [t_T] \left( \frac{\partial [g_T]}{\partial [u]_i} - \frac{\partial [g_T]}{\partial g_N/\partial [u]_i} \frac{\partial g_N}{\partial \xi^*} - \frac{\partial [g_T]}{\partial g_N/\partial [u]_i} \frac{\partial g_N}{\partial \eta^*} \right), \quad i = 1, \ldots, w
\]

where, for the frictionless case, we just have the first term \( t_N \partial g_N/\partial [u]_i \). The corresponding tangent matrix is obtained as

\[
[K_c]_{ij} = \frac{\partial [f_c]_i}{\partial [u]_j}, \quad i = 1, \ldots, w, \quad j = 1, \ldots, w
\]

which, for frictional slip, is a non-symmetric matrix, since Coulomb’s laws is non-associative. These expressions are still independent of the smooth surface description, which we are going to particularize in the following section.

3. SUBDIVISION SCHEMES FOR CONTACT PROBLEMS

Until recently, NURBS were the defacto standard for computational geometry. They represent a versatile tool to model free-form surfaces. In addition, they possess the property of local support which means that modification of a single node of the control polyhedron changes the surface only within a bounded region. In regard to contact mechanics, this means that the element arrays (i.e. tangent matrix and the residual vector), which are associated with a...
particular finite element, are of limited size [16]. However, NURBS are bivariate functions, and, therefore, they are not designed to directly represent surfaces of arbitrary topology. A possible work-around is to decompose the surface into (trimmed) NURBS patches. Unfortunately, this suffers from two drawbacks:

- Trimming two NURBS patches to match along their common boundary involves the complex computation of surface-to-surface intersection. These algorithms are, in general, computationally expensive and prone to numerical errors due to ill-conditioning.
- Complex and less intuitive continuity constraints across adjacent (trimmed) patches must be enforced throughout the deformation process.

These drawbacks have led to the development of subdivision surfaces (see References [20, 21]). Historically, they were derived from B-spline knot insertion rules to address the challenge of designing smooth surfaces of arbitrary topology. We start with an arbitrary connectivity control polyhedron which is described by control vertices and connecting edges in between. It forms a topological 2-manifold—possibly with boundary. The eventual surface (called the limit surface) is constructed through a limiting process of repeated refinement. The subdivision process consists of two parts, (i) a topological split rule describing how the existing control polyhedron is subdivided into additional polyhedrons, and (ii) some geometric rule with which the new control vertex positions are determined from the old positions. Most subdivision schemes use smoothing filters of finite support and local definition with constant coefficients depending only on the valence (i.e. the number of elements surrounding a vertex) in the support of the filter. This leads to very efficient implementations. The valence of the vertices of the control polyhedron allows to divide the mesh into regular and irregular regions. For quadrilateral-based schemes, discussed here, the valence required for regularity is $N = 4$, for triangle-based schemes it is $N = 6$. Other valences indicate irregular regions. Some schemes produce limit surfaces with different properties in these two regions, as noted below.

To illustrate the process of applying a subdivision scheme, we consider a simple one-dimensional example and use the Catmull–Clark subdivision scheme [20]. We denote the vertices of a polygon, which is the result of $n$ subdivision steps, by $x_n^i$. Thereby, $i$ is the index (starting from zero) of the vertex within one level (an overview of all indices used in this text is given in Appendix A.1). For 1D, for example, the topological split rule is simply described as an insertion of a new vertex between two existing vertices. The geometric rule for newly introduced vertices is

$$x_{2i+1}^{n+1} = \frac{1}{2}(x_i^n + x_{i+1}^n)$$ (31)

while the new co-ordinates of existing vertices are computed as

$$x_{2i}^{n+1} = \frac{1}{8}(x_{i-1}^n + 6x_i^n + x_{i+1}^n)$$ (32)

For existing terminal vertices, we use the rule

$$x_0^{n+1} = \frac{1}{4}(3x_0^n + x_1^n) \quad \text{or} \quad x_2^{n+1} = \frac{1}{4}(x_{i-1}^n + 3x_i^n)$$ (33)

for the left end or for the right end, respectively. Hence, the vertex positions of the refined polygon are computed as weighted averages of the unrefined polygon of the previous subdivision step. Figures 3(a)–(d) shows the repeated application of these rules in two dimensions. Within each step, the unrefined polygon of the previous step is shown by dashed lines. In the limiting
process for $n \to \infty$, we obtain smooth curves, which are actually $C^2$-continuous cubic splines. It can be observed that the resulting curve in Figure 3(d) is approximating the polygon in Figure 3(a). For one-dimensional problems, subdivision schemes do not offer an advantage over traditional approaches such as the methods discussed in the introduction. Their strengths become apparent in the two-dimensional case and for problems with arbitrary-topology.

To illustrate this, we consider a coarse control polyhedron, as seen in Figure 4(a). Its irregular vertices (of valence $N = 3$) are shown as black dots. By repeated application of the subdivision scheme, the surface converges to the smooth continuous limit surface, as shown in Figure 4(c). For this example, the two-dimensional version of the Catmull–Clark [20] scheme was used, whose refinement rules will be described later in Section 3.1. The reader who is interested in the topic of subdivision surfaces with a focus on geometrical modelling is referred to the standard introductory literature as mentioned in Section 1, and to Reference [42].

There exist a number of subdivision schemes, which can be basically classified into two groups:

1. **Interpolating schemes**: Within each refinement step, the nodal positions of the coarser mesh are not moved, while only the positions of the new vertices are computed.
Therefore, vertex positions of the initial mesh as well as all vertices generated during refinement lie on the limit surface. Such schemes have been introduced for quadrilaterals by Dyn et al. [43], Kobbelt et al. [44] and for triangles by Zorin et al. [45].

(2) **Approximating schemes**: These schemes compute new nodal positions for both the newly generated vertices and the vertices inherited from the coarser mesh. Therefore, the vertices of the control polyhedron do not lie on the limit surface. So, at each of subdivision, the existing vertices in the control polyhedron are moved closer to the limit surface. Approximating schemes are generally the favored schemes for use in high-end animation. Candidates which fall into this category were developed by Catmull and Clark [20] and Loop [22].

Further criteria for classifications are (i) the type of the refinement rule (interpolating versus approximating), (ii) the type of the generated mesh (triangular or quadrilateral), and (iii) the level of smoothness of the limit surface for regular and irregular regions of the mesh ($C^1$, $C^2$, ...). We use these criteria to characterize four schemes presented in the following section (see Figure 5).

### 3.1. Refinement rules

In this text we discuss the Catmull–Clark (see Reference [20]), Loop (see Reference [22]), Kobbelt (see Reference [44]), and the modified Butterfly schemes (see Reference [45], which goes back to the works by Dyn et al. [43,46]). Their properties as well as their so-called subdivision masks are summarized in Figure 5. Among the many other schemes available in the literature we have chosen these, because they are well-established and simple to implement. Furthermore, their intrinsic properties, when used to describe smooth contact surfaces, become more obvious. This might be not so clear with the more elaborate schemes. Figure 5 shows the subdivision masks for each scheme and presents the geometric (subdivision) rules for regular and irregular vertices in the first and second rows. Thereby, the numbers as well as the symbols $a, \ldots, d, q, \beta, \gamma$ and $s_j, j = 0, \ldots, N - 1$ represent the variables associated with the weights, which are provided below. The third row in Figure 5 provides the rules for subdivision of a vertex on the boundary or at a crease (i.e. a part of the surface with $C^0$-continuity by design). The remaining rows summarize the specific properties of the subdivision schemes regarding the continuity at regular and irregular vertices, the behaviour of interpolation or approximation of the control polygon and the mesh genus (i.e. quadrilateral or triangular).

For all of the four schemes except the Kobbelt scheme, the weights for the averaging process depend on the valence $N$ of a particular vertex (see Figure 5). These weights were determined such that $C^1$-continuity is achieved at irregular vertices. In summary, they are:

- **Catmull–Clark scheme**: In Reference [20] it is suggested to use $\beta = 3/(2N)$ and $\gamma = 1/(4N)$.
- **Loop scheme**: Loop [22] proposed the relation
  \[
  \beta = \frac{1}{N} \left[ \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{N} \right)^2 \right]
  \]  \hspace{1cm} (34)

A simpler choice is reported in [23]: $\beta = 3/(8N)$ for $N > 3$ and $\beta = 3/16$ for $N = 3$.
- **Modified Butterfly scheme**: If the edge, where a new node is created, connects two nodes of valence $N = 6$, then it is suggested to use $a = 1/2 - w$, $b = 1/8 + 2w$, $c = -1/16 - w$, $d = 1/16 - w$, $q = 1/(4N)$.
Figure 5. Subdivision schemes with associated subdivision masks and characteristic properties.

\[ d = w, \text{ see Reference [46].} \]

Therein, \( w \) can be chosen ‘suitably small’ or zero, resulting in the same rule set as the eight-point butterfly stencil introduced in Reference [43]. If one node has a valence \( N \neq 6 \), then the values for \( s_j, j = 0, \ldots, N-1 \), are to be determined according to Table I, see Reference [45]. The weight of the center node \( q = 3/4 \) remains constant for any value of \( N \).

3.2. Subdivision surface evaluation

Surface evaluation is the process of taking the control polyhedron, adding vertices, and braking faces into more, smaller faces to find a better polygonal approximation of the limit surface.
Table I. Values for the weights $s_j$, $j = 1, \ldots, N$, of the modified Butterfly scheme when using valences $N \neq 6$, [45].

<table>
<thead>
<tr>
<th>Valence</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 3$</td>
<td>$s_0 = 5/12$, $s_1 = s_2 = -1/12$</td>
</tr>
<tr>
<td>$N = 4$</td>
<td>$s_0 = 3/8$, $s_1 = s_3 = 0$, $s_2 = -1/8$</td>
</tr>
<tr>
<td>$N \geq 5$</td>
<td>$s_j = [1/4 + \cos(2\pi j/N) + 1/2 \cos(4\pi j/N)]/N$, $j = 0, \ldots, N - 1$</td>
</tr>
</tbody>
</table>

Basically there exist three different approaches to evaluate subdivision surfaces: (i) recursive evaluation [20, 22], (ii) exact evaluation [47, 48] and (iii) pre-tabulated basis function composition [49]. In the present work we consider the first two approaches. The latter one is not suitable for the use within a contact algorithm.

3.2.1. Recursive evaluation of subdivision surfaces. Recursive evaluation, which is based on repeated application of subdivision masks, as provided in Figure 5, is the most direct implementation of the standard definition of subdivision. For a curve this approach was briefly described in Section 3. However, within the non-linear solution scheme of the contact problem we require the linearization of geometric quantities based on the surface representation. Since recursive evaluation does not parametrize the surface with local co-ordinates $\xi$ and $\eta$, the resulting smooth surface, which covers a finite element, must be treated by the contact algorithm as if standard facet elements were used. Hence, the normal vector, distance function and shape function derivatives are evaluated from the facets, that were generated by the subdivision method. Although this approach is in principle possible, a recursively evaluated quantity leads to poor performance when an algebraic linearization is required. Therefore, an explicit form of the limit surface is of big advantage. This is only possible for the spline-based subdivision schemes such as Catmull–Clark and Loop, as shown next.

3.2.2. Exact evaluation of subdivision surfaces. Within the regular domain of the mesh, these schemes admit a simple evaluation of surface points since the control vertices of the Catmull–Clark scheme coincide with those of rectangular B-splines and the control vertices of the Loop scheme coincide with those of triangular Box splines. However, at irregular nodes, these schemes cannot be applied. Until recently, the proper parametrization of subdivision surfaces in the vicinity of irregular vertices has been an unsolved problem. A solution, which is based on the eigendecomposition of the subdivision matrix, is presented for (quadrilateral) Catmull–Clark surfaces in Reference [47], and for (triangular) Loop surfaces in Reference [48]. A modified version of this technique, which involves only one subdivision step, is documented in the recent work [27]. This is possible, because after one subdivision step the integration point of interest lies within a regular mesh region. Since for contact problems the contact point may lie anywhere on the surface, this technique is not applicable for our purposes. Here we briefly review this technique and apply it to the Catmull–Clark surfaces. It differs only slightly from the Loop scheme, which is described in detail in Reference [48].

We consider a quadrilateral patch (shaded region in Figure 6(a)) with one irregular vertex (see node 1), without loss of generality. This is due to the fact that after a single subdivision, even for a quadrilateral patch with four irregular vertices, the geometric situation, as shown in Figure 6(a), can always be recovered. The geometry is then defined by control vertices,
Figure 6. Parametrization of a Catmull–Clark subdivision surface near an irregular vertex of valence $N$: (a) numbering scheme for the set of control vertices $C_0$, whereby $[C_0]^T = [c_{0,1}, \ldots, c_{0,K}]$ (black dots), which defines a surface region (grey shaded) near an irregular vertex; (b) numbering scheme for the control vertices $\bar{C}_1$, which define the B-spline surface regions $\Omega_i^1$, $i = 1, 2, 3$; and (c) partition of the unit square into an infinite family of tiles (c). Therein, we used the shift-operator: $(\bullet)$, defined as: $(\bullet) = 2N + \bullet$.

$[C_0]^T = [c_{0,1}, \ldots, c_{0,K}]$ say, which are shown in Figure 6(a) as black dots. Therein, $K = 2N + 8$ denotes the number of control vertices that influence the shape of the grey shaded region in Figure 6(a). Through the topological splitting process, new control vertices (small circles in Figures 6(a) and (b)) can be generated, which allow to parameterize three quarters $\Omega_p^1$, $p = 1, 2, 3$ (see Figure 6(b)) of the region, and to evaluate them as simple B-splines. These new control vertices are defined as

$$[\bar{C}_1]^T = [[C_1]^T, c_{1,K+1}, \ldots, c_{1,M}]$$  \hspace{1cm} (35)$$

with

$$[C_1]^T = [c_{1,1}, \ldots, c_{1,K}]$$  \hspace{1cm} (36)$$
where \( M = K + 9 \). These control vertices for subdivision level \( n = 1 \) can be computed via an \( M \times K \) extended subdivision matrix \([\tilde{A}]\) as \([\tilde{C}_n] = [\tilde{A}][C_0]\), where

\[
[\tilde{A}] = \begin{bmatrix} [S] & [0] \\
[S_{11}] & [S_{12}] \\
[S_{21}] & [S_{22}] \end{bmatrix}
\]  

(37)

Therein, \([S]\) is the common \((2N + 1) \times (2N + 1)\) subdivision matrix for an irregular vertex of valence \( N \). All submatrices are provided in Appendix A2. To advance to any further level of subdivision \( n > 1 \), a smaller \( K \times K \) subdivision matrix

\[
[A] = \begin{bmatrix} [S] & [0] \\
[S_{11}] & [S_{12}] \end{bmatrix}
\]  

(38)

can be used. Hence, for \( n \geq 1 \), the control vertices for the B-splines can be obtained as

\[
[\tilde{C}_n] = [\tilde{A}][C_{n-1}] = [\tilde{A}][A]^{n-1}[C_0]
\]  

(39)

where we used \([C_i] = [A]^i[C_0]\). For example, if the origin in Figure 6(c), denoted by a double circle, represents an irregular vertex, the evaluation of the surface at node \( A \) requires three subdivisions. Hence, the prerequisites are obtained for defining the three surface regions, denoted as \( s(\xi, \eta)|\Omega_p^n \), \( p = 1, 2, 3 \), for any level of subdivision \( n \), as

\[
s(\xi, \eta)|\Omega_p^n = [\tilde{C}_n]^T [Q_p]^T [b(t_{p,n}(\xi, \eta))], \quad p = 1, 2, 3
\]  

(40)

where \((\xi, \eta) \in \Omega\) (see Figure 6(c)) and \( n \geq 1 \). Therein, we have used the \( 16 \times M \) ‘picking matrix’ \([Q_p]\), \( p = 1, 2, 3 \), which selects the control vertices from \([\tilde{C}_n]\), associated with the \( p \)th region, and \([b(\xi, \eta)]\) is the vector containing the 16 cubic B-spline basis functions (see Appendix A3). In addition we have used the transformations

\[
t_{1,n}(\xi, \eta) = (2^n \xi - 1, 2^n \eta)
\]  

(41)

\[
t_{2,n}(\xi, \eta) = (2^n \xi - 1, 2^n \eta - 1)
\]  

(42)

\[
t_{3,n}(\xi, \eta) = (2^n \xi, 2^n \eta - 1)
\]  

(43)

which map the tile \( \Omega_p^n \), \( p = 1, 2, 3 \), onto the unit square \( \Omega \) (see Figure 6(c)). However, the evaluation, as described in Equation (40), is not very efficient, since it involves \( n - 1 \) multiplications of the matrix \([A]\), as seen in Equation (39). As a possible remedy the subdivision is transformed into its eigenspace. Then, the subdivision is equivalent to a simple scaling of its eigenvectors by their associated eigenvalues. This technique allows to compute limit points and limit normals.

Subsequently, we aim to rewrite Equation (40) in a way that utilizes the transformation of the subdivision matrix into its eigenspace in order to make the computation of this equation for large values of \( n \) more efficient. Therefore, we consider the \( K \times K \) subdivision matrix \([A]\). For the Catmull–Clark and the Loop schemes it is non-defective for any valence, which means that the eigenvectors are complete. This is not the case for several other schemes. The property
that the eigenvectors are complete is important for the diagonalization of $[A]$. Consequently, there exist always $K$ linearly independent eigenvectors. The corresponding eigensystem of $[A]$ can be written as $([V], [A])$, where $[A]$ is the diagonal matrix containing the eigenvalues of $[A]$, and $[V]$ is an invertible matrix whose columns are the corresponding right eigenvectors. The computation of the eigensystem is equivalent to the solution of Equation (47)

$$[A][V] = [V][A]$$  \hspace{1cm} (44)

Since $[V]$ is invertible, we can rewrite $[A]$ as

$$[A] = [V][A][V]^{-1}$$  \hspace{1cm} (45)

which enables to rewrite Equation (39) as

$$[\tilde{C}_n] = [\tilde{A}][V][A]^{n-1}[V]^{-1}[C_0]$$  \hspace{1cm} (46)

The last two matrices in Equation (46) may be collected as $[\tilde{C}_0] = [V]^{-1}[C_0]$, representing the projection of the $K$ control vertices into the eigenspace of the subdivision matrix. Hence, with (46), Equation (40) can now be rewritten as

$$s(\xi, \eta)|_{\Omega_p} = [\tilde{C}_0]^T[A]^{n-1}([Q_p][\tilde{A}][V])^T[b(t_{p,n}(\xi, \eta))], \quad p = 1, 2, 3$$  \hspace{1cm} (47)

For simplicity, by collecting the terms in Equation (47), which are independent of the control vertices and the power $n$, we obtain

$$s(\xi, \eta)|_{\Omega_p} = [\tilde{C}_0]^T[A]^{n-1}[x(\xi, \eta, p)], \quad p = 1, 2, 3$$  \hspace{1cm} (48)

where the definition

$$[x(\xi, \eta, p)] = ([Q_p][\tilde{A}][V])^T[b(t_{p,n}(\xi, \eta))]$$  \hspace{1cm} (49)

has been introduced, and $(\xi, \eta) \in \Omega$.

Finally, we have an efficient technique for the evaluation of contact points on the interior of the master surface. However, at the boundary of the subdivision surface, several control vertices are missing. They are computed as a reflection of the existing nodes. For the reflecting plane, we use a plane that contains (i) the boundary edge and (ii) the surface normal vector at the boundary. For example, we consider the grey shaded quadrilateral, as shown in Figure 6(a), with valence $N = 4$, and assume that its right edge is part of the boundary. Hence, the vertices 5–7 and 14 describe the boundary. To provide an analytical description of the subdivision surface, we require the missing vertices with the indices 10–13. Then, for example, vertex $c_{0,11}$ is computed as $c_{0,11} = 2c_{0,5} - c_{0,4}$. A similar treatment applies to corners of the subdivision surface, where two reflecting planes are used. Note that this approach is different from the one we have used in the one-dimensional setting in Equation (33). This completes our discussion of the exact evaluation of subdivision surfaces.

3.3. Surface continuity

Continuity is a major concern in contact mechanics. In a recent paper [16] we have shown that it may influence the rate of convergence significantly. The analysis of the level of continuity of subdivision schemes may cause problems, partly due to the fact that the limit surface cannot
be evaluated explicitly for all schemes. Although, Fourier methods were developed to analyse smoothness properties for regular mesh regions, these are not applicable at irregular mesh vertices. The breakthrough was recently published in References [31, 50] to develop methods for proving the level of continuity. There exist schemes, which are $C^2$-continuous for a regular mesh and $C^1$ at irregular vertices (Catmull–Clark and Loop schemes), schemes which are everywhere $C^1$-continuous (Kobbelt and modified Butterfly schemes), and schemes which are $C^1$-continuous for a regular mesh and $C^0$-continuous at irregular vertices (Butterfly scheme). These properties are summarized in Figure 5.

4. IMPLEMENTATION STRATEGIES

The properties of the various subdivision schemes suggest four reasonable implementations strategies regarding the contact surface. In particular, the different approaches use (i) the interpolating schemes directly, (ii) the approximating schemes without interpolation, (iii) the approximating schemes with interpolation, (iv) the approximating schemes by defining an interface finite continuum element, where the vertices of the control polyhedron form a part of its degrees of freedom. The specific merits of each implementation strategy is discussed in the following:

(i) **Interpolating schemes**: The Kobbelt and the modified Butterfly schemes can be used directly for interpolating the finite element nodes of quadrilateral or triangular genus, respectively. They have $C^1$-continuity everywhere. A disadvantage is that the only way to evaluate the surfaces is by application of the subdivision mask. Even by the use of automatic differentiation and code optimization techniques, such as proposed in Reference [41], this may lead to long computation times for the element arrays. However, a reasonable limit of accuracy may help to avoid this.

(ii) **Approximating schemes without interpolation**: Without any interpolation technique, the Loop and Catmull–Clark schemes can be used to approximate contact surfaces of quadrilateral or triangular genus, respectively. Since they do not pass through the finite element nodes, mesh refinement changes the contact surface.

(iii) **Approximating schemes with interpolation**: As an extension to the method (ii), an interpolation technique as described in Reference [51] can be used. If the method is local, as it is the case for the quasi-interpolation technique, then the element arrays are of limited size. Another method to avoid non-local element arrays, when interpolating, is to include the vertices of the control polyhedron in the global system of equations and carry out the fitting procedure simultaneously to the solution of the non-linear system of equilibrium equations, as described in Reference [16].

(iv) **Define an interface finite continuum element with the control vertices as part of its degrees of freedom (composite contact element)**: Special finite continuum elements can be defined in order to describe the outermost layer of elements forming the contact surface. Since they represent a kind of interface between the continuum body and the contact surface, some of their degrees of freedom are the element nodes and some are the vertices of the control polyhedron. This leads to an identical representation paradigm (see References [25, 26]), whereby both the deformed and the undeformed geometries of the contact surfaces are described by the subdivision surface. It yields a robust interface between geometrical
design and mechanical analysis, helping to avoid a number of difficulties inherent in the heterogeneous treatment of smooth surfaces of current software tools. Due to the property of local support of subdivision surfaces, only a few nodes of the control polyhedron need to be considered for the element arrays. This is illustrated in Figure 7 for a simple 8-noded hexahedral element, described by nodes $P_i$, which are associated with current co-ordinates $x_i$, referential co-ordinates $X_i$ and nodal displacements $u_i, i = 1, \ldots, 8$.

The nodes $P_1, \ldots, P_4$ are part of the finite element mesh, while the nodes $P_5, \ldots, P_8$ depend on the vertices of the control polyhedron denoted by $c_{n,i}, i = 1, \ldots, 16$ (with the associated displacements defined by $u_{c_{n,i}}, i = 1, \ldots, 16$, where the superscript $c$ indicates a displacement field of a node located at the control polyhedron). Hence, the geometry-related quantities $x_5, \ldots, x_8, X_5, \ldots, X_8, u_5, \ldots, u_8$ are evaluated via (40) (or (48)) as $s([\xi])|_{\Omega_n}, i = 1, \ldots, 4, p = 1, 2, 3$, where $[\xi]_i = [(0,0), (1,0), (1,1), (0,1)]_i$, and $n$ is the level of subdivision. This method bypasses the need to interpolate the subdivision surface through the finite element nodes, because they do not exist at the global element level. Hence the displacement-related variables of an 8-noded continuum element are described by the vector

$$[a] = [u_1, \ldots, u_4, u_{c_{n,1}}, \ldots, u_{c_{n,16}}]$$ (50)

Suppose that the material model is defined by some strain-energy function $\psi$, then the material contribution (indicated by the subscript $m$) to the element arrays are given by

$$[f_{m}]_i = \frac{\partial \psi}{\partial [a]_i}, \quad i = 1, \ldots, 16$$ (51)

$$[K_{m}]_{ij} = \frac{\partial}{\partial [a]_i} \left( \frac{\partial \psi}{\partial [a]_j} \right), \quad i = 1, \ldots, 16, \quad j = 1, \ldots, 16$$ (52)

The contact related parts are provided through Equations (29) and (30). We call the element a composite contact element, because, for a continuum element, this formulation is only meaningful when it also acts as a contact element.
In summary, the advantage of the interpolating scheme, i.e. (i), is its simplicity. It can easily be implemented into an existing framework of a finite element system. The only ingredients required by the element is the mesh topology (i.e. element neighbourhood information). However, the highest achievable level of continuity is $C^1$ for the regular and irregular domains of the mesh. Approaches (ii)–(iv) have in common the salient feature of approximating subdivision schemes, i.e. they achieve $C^2$-continuity everywhere except at the irregular vertices, where the continuity is $C^1$. In addition, they allow an explicit evaluation, as described in Section 3.2, which accounts for a short computation time of the element arrays. Approach (ii) shares the simplicity with approach (i), however, it does not interpolate the finite element nodes. This difficulty is overcome in approach (iii) by means of an interpolation technique. The first three approaches have in common that the contact constraint is applied by an overlay element (see References [3, 4]). This is different for the composite contact element, i.e. approach (iv), where the nodes $P_5, \ldots, P_8$, of the continuum element, which form a part of the contact surface, do not appear as unknowns in the global element matrix, instead they depend on the nodes $c_{n,i}, \ i = 1, \ldots, 16,$ of the control polyhedron. The composite contact element demands the largest implementation effort when used within a standard finite element environment. However, it provides the most attractive features regarding the contact algorithm.

Note that since Equation (40) is linear in $[C_n]$, the nodal displacements $u_{i+4}$ can efficiently be computed as

$$
\begin{align*}
    u_{i+4} &= x_{i+4} - X_{i+4} \\
    &= [C_n^x]^T[Q_p]^T[b(t_{p,n}(\xi_i))] - [C_n^x]^T[Q_p]^T[b(t_{p,n}(\xi_i))] \\
    &= ([C_n^x]^T - [C_n^x]^T)[Q_p]^T[b(t_{p,n}(\xi_i))], \ i = 1, \ldots, 4, \ p = 1, 2, 3
\end{align*}
$$

Therein, we used right superscripts ($\bullet^x$) and ($\bullet^X$) to signify polyhedral co-ordinates associated with the current and the reference configurations, respectively.

As a general remark, it should be noted that the valence $N$ influences the number of entries in the element arrays. However, this does not necessitate to derive the characteristic equations of the element arrays for any value of $N$ that might occur. Instead, it is sufficient to derive these equations for the regular case $N = 4$. Subsequently, the element arrays can be supplemented by additional equations, which include the additional degrees of freedom for irregular cases ($N \neq 4$). This is illustrated as part of Section 5.

Regarding terminology, it should also be noted that the term ‘approximating subdivision scheme’ refers to the fact that the limit surface approximates the vertices of the control polyhedron and does not pass through them. It has nothing to do with the approximation or interpolation of the finite element nodes. In fact, it passes through the finite element nodes with arbitrary precision in method (iii) and exactly in method (iv).

5. ADAPTIVITY

In this paper we only discuss $h$-adaptivity. Other approaches such as $p$ or $r$-adaptivities do not change the mesh topology and are therefore not of interest here. $h$-adaptivity can be classified
Figure 8. Non-uniform (adaptive) refinement of a mesh shown in: (a) with different strategies; (b) hierarchically structured with hanging nodes; and (c) unstructured without hanging nodes.

into two approaches, one which does not generate hanging nodes and one which does. Clearly, the first approach leads to unstructured meshes with arbitrary topology. This case is naturally handled by subdivision surfaces. It remains to discuss the second case, which leads to structured meshes with hanging nodes. Both cases are illustrated in Figure 8.

Adaptively refined meshes with hanging nodes are usually described by a hierarchical data structure. This has the advantage that neighbourhood relations of elements can easily be generated, which are essential for the present contact algorithm. For more details about the associated data structures, the reader is referred to References [52, 53]. Here we will only consider refined meshes with no more than one hanging node per element edge, which is common practice.

The parametrization with non-spline-based (Kobbelt or modified Butterfly) subdivision schemes is trivial, since the additional finite element nodes of a refined mesh represent only information, which does not need to be computed by the subdivision scheme. However, when parameterizing an adaptively refined mesh with a spline-based subdivision scheme (Catmull–Clark or Loop), we encounter the situation that at some mesh locations the set of control vertices $c_{1,m}$, $m = 1, \ldots, 16$, required to describe the B-spline, is not complete. This is discussed next and illustrated for two cases in Figure 9. The element area to be parameterized, shown as the dark region $\Omega$, is adjacent to (a) coarser elements and (b) finer elements. However, these missing support vertices (hollow circles in Figure 9) can easily be evaluated as a function of existing vertices (black dots in Figure 9). It must be noted that the highest level of mesh refinement involved determines how the region $\Omega$ must be partitioned so that it can be parameterized. This is obvious from Figure 9(b), where $\Omega$ has to be partitioned into four regions (the one currently parameterized is denoted by $\tilde{\Omega}$ and indicated by a dashed line) to attain the same level of subdivision as the highest level of its neighbours. If this procedure is omitted, important information about the surface, and, therefore, continuity, is lost.

To introduce the necessary dependencies between the missing nodes and the known $c_{0,j}$, $j = 1, \ldots, K$, we set up the vector $[I]_i$, $i = 1, \ldots, d$, which contains the index $i$ of the unknown $c_{1,[i]}$, $i = 1, \ldots, d$. Therein, $d$ is the number of missing support vertices. To implement these additional equations in the stiffness matrix, it is convenient to define a matrix $[D]$, which describes these dependencies. Thus, we may write

$$[D]^T = [\tilde{A}]^T[G]$$  \hfill (54)
where \([G]\) is an \(M \times d\) matrix defined as

\[
[G]_{mn} = \begin{cases} 
1 & \text{if } [I]_m = n \\
0 & \text{otherwise}
\end{cases}
\] (55)

Hence, similar to hanging node constraints for finite element meshes, \([K_c]\) is supplemented by \([D]\), which allows to eliminate the unknown degrees of freedom.

6. NUMERICAL EXAMPLES

Three representative numerical examples are selected to demonstrate the features of the proposed contact algorithm. The examples aim to show the accuracy, which may be achieved with the presented algorithm. The first example, first published in Reference [17], involves three concentric spherical shells, which are in contact. It uses smooth surfaces with an unstructured mesh. The second example illustrates the use of triangular composite contact elements by considering a cylindrical contactor sliding in a half-tube with friction. The distribution of the tangential traction along the path of the contactor is analysed for different contact surface types. The third example illustrates the proposed algorithm more from the modelling point of view. It aims to show the strengths when designing complex contact surfaces by means of subdivision techniques, and simulates the contact interaction between the facet joints of two lumbar vertebral bodies, which exhibits a research area by its own in the field of biomechanics. In order to
avoid confusion between vertebral facet joints and facet contact elements, we subsequently use the term piecewise planar elements for facet contact elements.

6.1. Interference fit of concentric spheres

According to Reference [17] we consider three concentric spherical shells, where the inner (I) and outer shells (O) are rigid, while the middle shell (M) is deformable (see Figure 10(a)). For the discretization we use smooth surfaces with an unstructured mesh. Shell (M) has an inner radius $R_i = 1.0$, an outer radius $R_o = 1.1$ and consists of a material with the properties $E = 5.0 \times 10^{10}$, $\nu = 0.0$, $\alpha = 1.0 \times 10^{-5}$, which are the Young’s modulus, Poisson’s ratio and the thermal expansion coefficient, respectively. The outer surface of shell (M) is tied to shell (O), while the inner surface of (M) is in contact with shell (I). This contact surface is characterized by a Coulomb frictional parameter $\mu = 0.1$.

In a first loading step, the temperature of shell (M) is raised up to 1000.0, in a second step, shell (I) is rotated by 45$^\circ$ (as indicated in Figure 10(b)) within 20 steps, while shell (O) is fixed in space. The moment is applied by a prescribed displacement on one node of shell (I). Two other nodes of shell (I) are fixed to define the axis of rotation. In Reference [17], the analytical solution for the resulting moment is given by $4.93071 \times 10^8$. We carried out simulations with 96, 384 and 864 elements for shell (M) (see Figure 10(b)). In Figure 11 the obtained frictional moments are plotted in logarithmic scale over the rotation angles and compared for different meshes and contact surface types. Since only the sliding behaviour is of interest, the transition from stick to slip of the finite element solution is not shown. Hence, we start directly from the slip behaviour. For the different contact surface types, we used (i) standard piecewise planar contact elements (with $C^0$-continuity), (ii) the Kobelt subdivision scheme with implementation strategy (i) (with $C^1$-continuity), and (iii) the Catmull–Clark subdivision scheme with implementation strategy (iv) (with $C^2$-continuity in the regular mesh domain and $C^1$-continuity in the irregular mesh domain). In this problem eight nodes, identified with symbol ‘A’ in Figure 10(b), define irregular mesh domains with valence $N = 3$. Clearly, the piecewise planar-based algorithm does not produce meaningful results, while the subdivision surface techniques quickly converge to the exact solution.
Figure 11. Frictional moment in logarithmic scale over the relative rotation angle of three concentric spherical shells with different contact surface types.

Figure 12. Cylindrical contactor sliding in a half-tube: (a) finite element mesh of the problem, irregular vertices are shown as black dots; (b) distribution of the normalized tangential traction along the path; i.e. the displacement $u$ of the cylindrical contactor, computed with different contact surface types.

6.2. Cylindrical contactor sliding in a half-tube

This example illustrates the use of triangular composite contact elements. A similar version of the example was published in Reference [18]. We consider a half-tube, chosen as the master surface, and a cylindrical contactor, chosen as the slave surface (see Figure 12(a)). The inner
and outer radii of the tube is 3.0 mm and 5.0 mm, respectively, while its length is 10.0 mm. The dimension of the contactor is 2 mm $\times$ 2 mm $\times$ 2 mm. For the purpose to generate an irregular mesh a hole with radius $r = 1.0$ mm was placed at the lower edge of the half-tube. The contactor slides within the region indicated by the dashed line from one end to the other in the direction indicated by the arrow. Thereby, the contactor passes regular (valence $N = 6$) as well as irregular regions (valence $N = 5, \ldots, 8$). Irregular vertices, which influence the surface within the contacting region, are shown as black dots. We used a course mesh in order to make the differences between the used surface representations more apparent. The half-tube/contactor problem was discretized with 882 tetrahedral elements.

The half-tube is fixed at all (cutting) planes. The contactor has prescribed displacements at the upper surface in order to move it along the path. In addition, a normal force of 500.0 N is applied on the upper surface in order to press the contactor against the half-tube. Due to the (uneven) discretized surface, loading was applied in 89 steps (i.e. 89 prescribed displacements). Both bodies are simulated as compressible hyperelastic neo-Hookean material described by the strain-energy function

$$
\psi = \frac{c}{2}(I_1 - 3) + \frac{\kappa}{2}(J - 1)^2
$$

with the material parameters $c = 350.0$ N/mm$^2$ and $\kappa = 3000.0$ N/mm$^2$. In (56) $I_1 = \text{Tr} \, C$ is the first invariant of the right Cauchy–Green tensor $C = F^T F$, where $F$ is the deformation gradient and $J = \text{det} \, F > 0$ is the local volume ratio. The frictional coefficient was chosen to be $\mu = 0.1$, and hence the exact solution for the tangential traction vector is $t_T = 50$ N. For the normal and the tangential penalty parameters we used the value $\varepsilon_N = \varepsilon_T = 10000$.

The numerical simulations we carried out with three different surface representations, i.e. (i) piecewise planar elements, (ii) modified Butterfly, [45, 46], and (iii) Loop subdivision surfaces [22]. We study the distribution of the tangential traction along the path length. Here we are only interested in the distribution of $t_T$, since, for this problem, $t_T$ is independent of the interpolation technique. We normalize $t_T$ by dividing the numerical solution with the analytical solution. The results, as illustrated in Figure 12(b), reveal that the modified Butterfly method already yields a better approximation and a significantly smoother tangential force distribution than the one obtained with piecewise planar elements, which shows oscillations in the normalized tangential traction. The use of the Loop subdivision surface gives a further improvement.

6.3. Contact interaction between two lumbar vertebral bodies

In this example the advantages of modelling complex surfaces with the subdivision technique should become more apparent. We discretize two lumbar vertebral bodies, in particular, we focus on the facet joints, which form the posterior part of the vertebral body. Although this would not necessitate to model the whole body by subdivision surfaces, it has advantages when performing mesh refinement or when contact develops in regions not known a priori.

Since these joints are known to be susceptible to degenerative processes and injuries, numerous publications are devoted to study the underlying mechanisms. The issue was discussed, in particular, as a contact problem in References [54–56]. In a more general context, the problem is investigated for the whole spine in References [57, 58], among others. All of them use piecewise planar contact surfaces. However, it turns out that the facet articular surfaces have a complex curvature [55], which may contribute essentially to the obtained results. For example,
in Reference [57] it was shown that the relationship between torsion moment and torsion angle is a function of the number of elements on the facet joint surface. In addition, the geometry resulting from graded facetectomy (partial removal of the facet joints) may be too complicated to be modelled reasonably by standard discretization methods. Subdivision surfaces allow to capture the associated geometry up to any desired level of accuracy. The goals of this example are twofold: (i) to illustrate the process of modelling such a complicated surface with the subdivision surface technique, and (ii) to assess how far the simplification of piecewise planar contact surfaces is justified. In particular, we study the contact stress within the facet joint of the lumbar vertebral bodies L2 and L3 during physiologically normal loading.

Usually it may be difficult to represent a body of arbitrary topology with smooth surfaces. However, using subdivision surfaces, a whole vertebral body can be modelled by a single surface. Hence, it was not necessary to join individual surface patches as it would have been necessary with NURBS. In particular, for this example we have chosen the Catmull–Clark subdivision scheme and implementation strategy (iv). Since the generated mesh contains hanging nodes, we had to use the technique described in Section 5. We obtained generally $C^2$-continuous surfaces, and only at irregular nodes the level of continuity decreased to $C^1$. The geometry was obtained from the Visible Human Project data set [59]. The subdivision surface was designed using a proprietary software tool. In order to fill the surface with hexahedral finite elements, it is necessary to take some care when laying out the surface topology. The topology is governed by the so-called ‘polygon proxy’, which represents the coarsest control polyhedron of the model. Hence, the polygon proxy must be built purely from quadrilateral polygons, which is, in general, easy to obtain. The polygon proxy and the resulting subdivision surface for the vertebral body are shown in Figures 13(a) and (b), respectively. For the finite element mesh we performed additional refinements as it was necessary for the numerical simulations.

The two vertebral bodies are connected with the following ligaments: intertransverse ligament (ITL), ligamentum flavum (LF), posterior longitudinal ligament (PLL), anterior longitudinal ligament (ALL), capsular ligament (CL), interspinous ligament (ISL) and supraspinous ligament (SSL). They are modelled as transversely isotropic materials, with an isotropic matrix material, defined by the strain-energy function (i.e. in analogy with References [60, 61], see also Reference [37])

$$\psi(C, A) = \psi_m(\bar{C}) + \psi_t(\bar{C}, A) + L(J) \quad (57)$$

where

$$\psi_m(\bar{I}_1) = \frac{c}{2}(\bar{I}_1 - 3) \quad (58)$$

represents the energy stored in the matrix material,

$$\psi_t(\bar{I}^s) = \frac{k_1}{2k_2}\{\exp[k_2(\bar{I}^s - 1)^2] - 1\} \quad (59)$$

represents the contribution from the collagen fibres, and

$$L(J) = \frac{\kappa}{2}(J - 1)^2 \quad (60)$$

is a scalar-valued function, with the property $L(1) = 0$, which is motivated mathematically. It serves as a penalty function enforcing the incompressibility constraint.
Figure 13. Contact interaction between lumbar vertebral bodies L2 and L3: (a) polygon proxy of a half of L2; (b) representation of both bodies by means of subdivision surfaces; and (c) contact stress distribution along the white dashed path (indicated in (b)) for different mesh densities and contact surface types. The ligaments and the intervertebral disc are not shown.

Therein, $\bar{C} = J^{-2/3}C$ is the modified right Cauchy–Green tensor, $\bar{I}_1 = \text{Tr} \bar{C}$ is the first invariant of $\bar{C}$, and the pseudo invariant $\bar{I}^* = \bar{C} : A$, where $A = a \otimes a$ is a structural tensor with the vector $a$ characterizing the direction of the fibres. The symbols $c > 0$, $k_1 > 0$ and $k_2 > 0$ are material parameters, which were obtained by fitting the strain-energy function to the data from Reference [62] (see the summary provided in Table II). The value for $\kappa > 0$ (considered as a positive penalty parameter) was chosen to be 250.0 MPa for all ligaments. For additional information about the model see Reference [60]. For the intervertebral disc as well as for cortical and cancellous bone, the material parameters were adopted from Reference [63]. The articular cartilage was modelled as a 1 mm thick layer of neo-Hookean material with the parameters also adopted from Reference [62]. To model the contact interaction, body L2 was chosen as the slave surface while L3 was the master surface. For the penalty parameters $\varepsilon_N$ and $\varepsilon_T$ associated with contact, the value 1000.0 was chosen, and the frictional coefficient $\mu$ was taken to be 0.06.
Table II. Material parameters for the ligaments, i.e. intertransverse ligament (ITL), ligamentum flavum (LF), posterior longitudinal ligament (PLL), anterior longitudinal ligament (ALL), capsular ligament (CL), interspinous ligament (ISL) and supraspinous ligament (SSL).

<table>
<thead>
<tr>
<th></th>
<th>ITL</th>
<th>LF</th>
<th>PLL</th>
<th>ALL, CL, ISL, SSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$ (MPa)</td>
<td>13.1</td>
<td>6.0</td>
<td>6.0</td>
<td>1.5</td>
</tr>
<tr>
<td>$k_2$ (−)</td>
<td>21.0</td>
<td>0.11</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>$c$ (MPa)</td>
<td>1.3</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

To apply the load, the lower endplate of L3 was fixed, while the upper endplate of L2 was exposed to an overall force of 294 N. In Figure 13(c) the contact stress distribution along the white dashed path (see Figure 13(b)) is shown—the length measurement starts at position ‘A’.

As can be seen from Figure 13(c), by representing the contact surface with only 7 elements and $C^2$-continuous subdivision surfaces, the distribution of the contact stress is much smoother than for $C^0$-continuous piecewise planar surfaces. To identify the quality of the solutions, a uniform mesh refinement was performed, using twice as many elements (i.e. 14 elements along the path in Figure 13(b)). It shows that the solution with only 7 subdivision surface elements is already a good approximation. The model using piecewise planar elements also seems to converge to the smooth solution, but it shows strong discontinuities even for the refined mesh. Hence, for this type of problem, piecewise planar contact surfaces may only yield reliable results with very high mesh densities, which is inefficient to perform.

7. CONCLUSION

Subdivision surfaces were used to parameterize contact surfaces of arbitrary mesh topology in 3D with at least $C^1$-continuity everywhere, both for quadrilateral and triangular meshes. Therefore, for frictional contact interaction a simple standard formulation, as described in Reference [4], can be used.

We presented four different subdivision schemes, the Catmull–Clark [20], the Loop [22], the Kobbelt [44], and the modified Butterfly schemes [43, 45, 46]. The first two belong to the group of spline-based subdivision schemes, which allow an analytical evaluation of the surface in the regular mesh domain. The other two are not spline-based and can, therefore, only be evaluated by recursive application of the refinement rule, which is less efficient. We also reviewed how such surfaces can be evaluated efficiently near irregular vertices ($N \neq 4$ for quadrilateral-based meshes or $N \neq 6$ for triangular meshes), and on adaptively refined meshes with hanging nodes.

Four different implementation strategies of a contact finite element were presented. These strategies differ in the implementation effort and in the quality of the surface representation (such as level of continuity, interpolation versus approximation). The proposed implementation strategy (iv) is superior with respect to the strategies (i)–(iii) in the sense that it does not use nodal degrees of freedom of the finite element mesh at the contact surface. Instead, it directly uses the degrees of freedom of the smooth surface, which makes an interpolation obsolete when dealing with spline-based subdivision schemes. This leads to an identical representation.
paradigm. We call the resulting contact element a composite contact element (as opposed to an overlay element), because of the fact that this formulation is only meaningful for a continuum element when it also acts as a contact element. The Mathematica package ACEGen was used for the automatic derivation of the matrix formulae required to describe the finite element discretization of the smooth contact element.

When using the Catmull–Clark or Loop schemes, there is no additional numerical cost when compared when other smoothing techniques that consider also all neighbour vertices of the current element in the element arrays. When compared to facet elements, the evaluation of the element arrays is more complex due to the larger number of variables involved. Those schemes, which allow only a recursive evaluation (i.e. the Kobbelt and the modified Butterfly schemes) make the numerical cost of the computation of surface derivatives dependent on the precision to be achieved.

Three numerical examples were used to demonstrate the special merits of the proposed method. They aimed to show the accuracy, which may be achieved with the presented algorithm. The results show a significant improvement over facet-based (i.e. piecewise planar) contact surfaces when compared with analytical solutions. In particular, the non-physical oscillations are avoided and the robustness is significantly increased. Therefore, the additional effort for the evaluation of the element arrays pays off well due to the better convergence behaviour. The third example, adopted from biomechanics, showed, in addition, how easy it is to design complex contact surfaces by means of subdivision techniques and how to use the contact surfaces directly in finite element analyses. To the authors’ knowledge, the detailed curvature of the facet articular joint has not been considered in simulations using such a high level of accuracy before. The obtained high accuracy was clearly pointed out in comparison with piecewise planar contact surfaces. Hence, the presented approach has the potential to open up an area which allows to efficiently simulate three-dimensional contact problems with complex geometry and topology in a numerically accurate manner. This is in virtue of the contact surface (including the finite element mesh), which (i) can be modelled in a highly efficient way, (ii) has at least $C^1$-continuity everywhere, (iii) can deal with surface vertices of any valence, and (iv) does not need any interpolation.

APPENDIX A

A.1 Description of the used indices

An overview of all Indices is given in Table AI.

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>Level of subdivision</td>
</tr>
<tr>
<td>$N$</td>
<td>Valence</td>
</tr>
<tr>
<td>$K = 2N + 8$</td>
<td>Number of control vertices for $n = 0$</td>
</tr>
<tr>
<td>$M = K + 8$</td>
<td>Number of control vertices for $n \geq 1$</td>
</tr>
<tr>
<td>$p = 1, 2, 3$</td>
<td>Regular patch index</td>
</tr>
</tbody>
</table>
A.2 Subdivision matrices

With the given order of the control vertices, the \( (2N+1) \times (2N+1) \) subdivision matrix \([S]\) for the Catmull–Clark scheme, as introduced in Equations (37) and (38), is given by [20]

\[
[S] = \begin{bmatrix}
a_N & b_N & c_N & b_N & c_N & \cdots & b_N & c_N & b_N & c_N \\
d & d & e & e & 0 & 0 & \cdots & 0 & 0 & e & e \\
f & f & f & f & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
d & e & e & d & e & e & \cdots & 0 & 0 & 0 & 0 \\
f & 0 & 0 & f & f & f & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
d & e & 0 & 0 & 0 & 0 & \cdots & e & e & d & e \\
f & f & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & f & f \\
\end{bmatrix}
\]

(A1)

Therein, \( N \) denotes the valence of the irregular vertex and \( a_N = 1 - 7/4N, \ b_N = 3/2N^2, \ c_N = 1/4N^2, \ d = 3/8, \ e = 1/16 \) and \( f = 1/4 \). The submatrices \([S_{11}]\) and \([S_{12}]\) in \([A]\) (see Equation (38)) follow from the B-spline knot insertion rules as [47]

\[
[S_{11}] = \begin{bmatrix}
c & 0 & 0 & b & a & b & 0 & 0 & [0] \\
e & 0 & 0 & e & d & d & 0 & 0 & [0] \\
b & 0 & 0 & c & b & a & b & c & [0] \\
e & 0 & 0 & 0 & d & d & e & [0] \\
b & c & b & a & b & c & 0 & 0 & [0] \\
e & e & d & d & 0 & 0 & 0 & 0 & [0] \\
\end{bmatrix}, \quad \begin{bmatrix}
c & b & c & 0 & b & c & 0 \\
e & 0 & e & 0 & 0 & 0 & 0 \\
0 & c & b & c & 0 & 0 & 0 \\
0 & 0 & e & e & 0 & 0 & 0 \\
0 & 0 & c & b & c & e & e \\
0 & 0 & 0 & 0 & e & e \\
\end{bmatrix}
\]

(A2)

where \( a = 9/16, \ b = 3/32, \ c = 1/64 \) and \([0]_i = 0, \ i = 1, \ldots, 2N - 7 \). For the case \( N = 3 \), there is no control vertex \( c_8 \) (it coincides with \( c_2 \)). Then, the components of the second column of \([S_{11}]\) must be replaced according to \([S_{11}]_{12} = [0, 0, c, e, 0, c, e]^T, \ i = 1, \ldots, 7 \), and the last two columns in \([S_{11}]\) are missing. The submatrices \([S_{21}]\) and \([S_{22}]\) appearing in \([\bar{A}]\)
(see Equation (37)) are \([47]\)

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & f & 0 & 0 & [\mathbf{0}] \\
0 & 0 & 0 & 0 & d & e & 0 & [\mathbf{0}] \\
0 & 0 & 0 & 0 & f & f & 0 & [\mathbf{0}] \\
0 & 0 & 0 & 0 & e & d & e & [\mathbf{0}]
\end{bmatrix}
\quad \begin{bmatrix}
0 & 0 & 0 & 0 & f & f & 0 & 0 \\
e & d & e & 0 & e & 0 & 0 \\
0 & f & f & 0 & 0 & 0 & 0 \\
0 & e & d & e & 0 & 0 & 0
\end{bmatrix}
\]

\[
[S_{21}] = \begin{bmatrix}
0 & 0 & 0 & 0 & f & f & 0 & [\mathbf{0}] \\
0 & 0 & 0 & 0 & e & f & 0 & [\mathbf{0}] \\
0 & 0 & 0 & 0 & f & g & 0 & 0 \\
0 & 0 & e & d & r & 0 & 0 & [\mathbf{0}]
\end{bmatrix}, \quad [S_{22}] = \begin{bmatrix}
0 & 0 & f & f & 0 & 0 & 0 \\
e & e & 0 & 0 & d & e \\
0 & 0 & 0 & 0 & f & f \\
0 & 0 & 0 & 0 & e & d & e
\end{bmatrix}
\]  \hspace{1cm} (A3)

where \([\mathbf{0}]_i = 0, \ i = 1, \ldots, 2N - 6.\)

A.3 B-spline basis functions

The vector \([\mathbf{b}(\xi, \eta)]\) in Equations (40) and (49) contains the 16 cubic B-spline basis functions. The component \(i\) of \([\mathbf{b}(\xi, \eta)]\) is defined as

\[
[\mathbf{b}(\xi, \eta)]_i = N_{(i-1)/4}(\xi)N_{(i-1)/4}(\eta)
\]  \hspace{1cm} (A4)

where ‘\%' denotes the modulo operator returning the remainder of the division between the first and the second argument, and ‘/’ denotes the integer division operator. The functions \(N_i(t)\) are the uniform cubic B-spline basis functions given by

\[
\begin{align*}
N_0(t) & = \frac{1}{6}(1 - 3t + 3t^2 - t^3) \\
N_1(t) & = \frac{1}{6}(4 - 6t^2 + 3t^3) \\
N_2(t) & = \frac{1}{6}(1 + 3t + 3t^2 - 3t^3) \\
N_3(t) & = \frac{1}{6}t^3
\end{align*}
\]  \hspace{1cm} (A5)

where \(t\) stands for either \(\xi\) or \(\eta\).

ACKNOWLEDGEMENTS

Special thanks go to Professor Rudolf Stollberger, Ph D, from the Institute of Magnetic Resonance, Karl-Franzens-University Graz, to Margit Bauer, MD., from the Department of Obstetrics and Gynecology, Karl-Franzens-University Graz, and to Harald Bisail, MSc, from the Institute of Anatomy, Karl-Franzens-University Graz, who assisted us in developing the second example and in generating the necessary data for the vertebral bodies. Financial support for this research was provided by the Austrian Science Foundation under START-Award Y74-TEC. This support is gratefully acknowledged.
REFERENCES


