

Thin shell analysis from scattered points with *maximum-entropy* approximants

D. Millán, A. Rosolen and M. Arroyo^{*, †}

Laboratori de Càlcul Numèric (LaCàN), Departament de Matemàtica Aplicada III (MA3), Universitat Politècnica de Catalunya (UPC), Campus Nord UPC-C2, E-08034 Barcelona, Spain

SUMMARY

We present a method to process embedded smooth manifolds using sets of points alone. This method avoids any global parameterization and hence is applicable to surfaces of any genus. It combines three ingredients: (1) the automatic detection of the local geometric structure of the manifold by statistical learning methods; (2) the local parameterization of the surface using smooth meshfree (here *maximum-entropy*) approximants; and (3) patching together the local representations by means of a partition of unity. Mesh-based methods can deal with surfaces of complex topology, since they rely on the element-level parameterizations, but cannot handle high-dimensional manifolds, whereas previous meshfree methods for thin shells consider a global parametric domain, which seriously limits the kinds of surfaces that can be treated. We present the implementation of the method in the context of Kirchhoff–Love shells, but it is applicable to other calculations on manifolds in any dimension. With the smooth approximants, this fourth-order partial differential equation is treated directly. We show the good performance of the method on the basis of the classical obstacle course. Additional calculations exemplify the flexibility of the proposed approach in treating surfaces of complex topology and geometry.

KEY WORDS: point-set surfaces; meshfree methods; maximum-entropy approximants; thin shells

1. INTRODUCTION

Over the last years, there has been a growing interest in the computer graphics community on point-based surface processing, which presents attractive features as compared with the conventional mesh-based processing [1, 2]. In mesh-based methods, the mesh serves two useful purposes: it describes the geometry of the surface, and the elements provide local parametric spaces where the shape functions and the local parameterizations of the surface can be defined, and where the required calculations on the surface can be performed, e.g. for thin-shell analysis. In these methods, the mesh generation can be difficult, they are not natural for point-based data, and they seem unpractical for embedded manifolds in high-dimensional spaces. On the other hand, in the absence of a mesh, the notion of a surface defined from a set of scattered nodes becomes difficult to grasp. In particular, as noted in [3], a fundamental difficulty in defining basis functions and performing calculations on an embedded surface, as compared with open sub-sets in the Euclidean space, is the absence in general of a single parametric domain. A simple example is the sphere, which does not admit a single singularity-free parameterization. Mesh-based method, consisting of a collection of local parameterizations from the parent element to the physical elements, does not have

* Correspondence to: M. Arroyo, Laboratori de Càlcul Numèric (LaCàN), Departament de Matemàtica Aplicada III (MA3), Universitat Politècnica de Catalunya (UPC), Campus Nord UPC-C2, E-08034 Barcelona, Spain.

†E-mail: marino.arroyo@upc.edu

any difficulty in this respect at the expense of reduced smoothness across the element boundaries. In meshfree methods, such a natural parametric domain is not available, and the description of surfaces with a topology different to that of an open set on \mathbb{R}^2 , such as a sphere or a torus, becomes challenging. Even for surfaces homeomorphic to open two-dimensional sets, such as that depicted in Figure 18, the geometric complexity can make it very difficult to produce well-behaved global parameterizations.

In the computer graphics literature, Levoy and Whitted [4] pioneered using points as primitives for geometric modeling and rendering of surfaces. Existing methods for describing a surface from a set of scattered points are generally based on implicit representations of the surface [5, 6]. Moving least-squares (MLS) surfaces are a noteworthy example of point-based implicit surface representation, where the surface is defined as a set of fixed points of suitable projections [3, 7]. This idea has been very successful in the computer graphics community for rendering, either up or down-sampling, and manipulating point-set models, see e.g. [1, 2, 8–10]. Despite the common themes and challenges, these developments have remained largely unconnected to the computational mechanics community. In this field, meshfree methods have been applied to thin-shell analysis, and the difficulty of defining an appropriate parametric space has been overcome by considering either a support mesh or very simple surfaces admitting a single parametric space [11–14].

Thin-shell theory requires the approximation of the deformation field to have second-order square integrable derivatives. For general unstructured meshes, it is very difficult to construct C^1 finite element approximations, which has prompted many techniques that avoid this requirement. Most alternative approaches requiring only C^0 elements are based on either Reissner–Mindlin theories for thick plates and shells or on hybrid and mixed methods [15–17]. Excellent reviews and insightful remarks can be found in these references or in [18, 19]. For an extended review, the reader is referred to [20].

Following ideas from computer graphics, Cirak *et al.* [18, 21] presented a direct numerical treatment of thin-shell analysis based on the smooth approximants provided by subdivision surfaces technology. Subdivision finite elements represent unified framework for geometric modeling and thin-shell analysis. On the other hand, discontinuous Galerkin (DG) formulations have been proposed recently for plates, beams and Kirchhoff–Love thin-shells [22–24]. These methods avoid the C^1 continuity requirement by designing suitable numerical fluxes conjugate to the deformation jumps. An advantage of this method is the ease in the imposition of the rotation essential boundary conditions. As disadvantages, the formulation and implementation of these methods is cumbersome, and they typically exhibit a poorer accuracy for a given number of degrees of freedom as compared with methods based on smooth approximants.

Here, we present a method to perform numerical calculations on smooth manifolds described in terms of a set of scattered points. The proposed method avoids any global parametric domain, required in previous meshfree methods, which greatly expands its range of applicability. Although we exercise here the method for surfaces in \mathbb{R}^3 , it is applicable to perform calculations on embedded manifolds in any space dimension, unlike mesh-based methods. The implications of this attribute of the method are hinted in the closing remarks. The method results from combining three ingredients. First, the local geometric structure of the manifold is detected from the node set using statistical learning methods. Here a weighted Principal Component Analysis (wPCA) identifies the (hyper-) plane closest to the points in a given neighborhood that we call patch. This plane is then used as the local parametric space to construct the meshfree maximum-entropy (*max-ent*) basis functions [25, 26] and the local smooth parameterization of the manifold, and it can be seen as the analogous of the parent element in finite element methods. This smooth parameterization in each patch can be realized with a variety of methods, from other mesh-free methods such as MLS approximants to mesh-based methods such as subdivision finite elements. In the latter case, no global mesh is required. Here the local *max-ent* approximants [25] are chosen, due to their smoothness, robustness, and the relative ease of quadrature relative to other meshfree approximants. The different local parameterizations are then glued together with a partition of unity (PU) defined in the ambient space, which consequently is also a PU in the embedded manifold. Specifically, functionals defined on the surface are readily split into local contributions, each involving a single local parameterization.

The outline of the paper is as follows. Section 2 describes the proposed methodology for point-set manifold processing. This methodology is illustrated with the simple yet illustrative calculation of the Willmore energy of a sphere. Next, Section 3, provides a short account of the Kirchhoff–Love shell theory. Throughout the present work, we confine our attention to the linear theory of shells under static loading. Numerical experiments to evaluate the performance of the method are presented in Section 4, with particular attention to the obstacle course of benchmark tests proposed by Belytschko *et al.* [27]. Other numerical examples illustrate the ability of the method to treat shells with complex topology or geometry. Some remarks and conclusions are collected in Section 5.

2. MANIFOLD DESCRIPTION FROM SCATTERED POINTS

We consider a smooth d -manifold \mathcal{M} embedded in \mathbb{R}^D , $d \leq D$. Our aim is to obtain a numerical representation of \mathcal{M} , and make computations on it. Manifolds are important objects in mathematics and physics because they allow us to understand complicated structures in each neighborhood in terms of Euclidean spaces. The first of the three steps in the methodology presented here precisely captures numerically the local Euclidean structure of the manifold in a set of neighborhoods that we call patches.

Let $P = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N\} \subset \mathbb{R}^D$ be a set of points representing \mathcal{M} . We detect the local Euclidean structure and build local parameterizations around the patch points, $Q = \{\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_M\}$, typically a subset of P but not necessarily. For simplicity, we will denote the points in P and its associated objects with a lower case subindex, e.g. \mathbf{P}_a , for $a = 1, 2, \dots, N$, and the patch points in Q and its associated objects with an upper case subindex, e.g. \mathbf{Q}_A , for $A = 1, 2, \dots, M$.

We recall the construction of a Shepard PU, for later reference. Let $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L\} \subset \mathbb{R}^D$ be a set of points and consider a set of non-negative reals $\{\beta_a\}_{a=1,2,\dots,L}$ associated with each point in Y . We define the Shepard PU with the Gaussian weight related to the set Y as the functions $w_a^Y: \mathbb{R}^D \rightarrow \mathbb{R}$ for $a = 1, 2, \dots, L$ given by

$$w_a^Y(\mathbf{x}) = \frac{\exp(-\beta_a |\mathbf{x} - \mathbf{y}_a|^2)}{\sum_{b=1}^L \exp(-\beta_b |\mathbf{x} - \mathbf{y}_b|^2)}. \quad (1)$$

For efficiency, and given the fast decay of the Gaussian functions, it is convenient to consider numerically these functions to be compactly supported. We define for each point \mathbf{x} the neighborhood index set from Y as

$$\mathcal{N}_x^Y = \{a \in \{1, 2, \dots, L\} \mid \exp(-\beta_a |\mathbf{x} - \mathbf{y}_a|^2) > TOL\}, \quad (2)$$

where TOL is a numerical tolerance. Then, the evaluation of the Shepard functions becomes

$$w_a^Y(\mathbf{x}) = \begin{cases} \frac{\exp(-\beta_a |\mathbf{x} - \mathbf{y}_a|^2)}{\sum_{b \in \mathcal{N}_x^Y} \exp(-\beta_b |\mathbf{x} - \mathbf{y}_b|^2)} & \text{for } a \in \mathcal{N}_x^Y, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

A dual notion of neighborhood index list is given by the points in a given set $Z = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_O\}$ that lie within the support of a given Shepard function w_a^Y , i.e.

$$\hat{\mathcal{N}}_{y_a}^Z = \{b \in \{1, 2, \dots, O\} \mid w_a^Y(\mathbf{z}_b) > TOL\}. \quad (4)$$

Following [25], we define a dimensionless parameter $\gamma_a = \beta_a h_a^2$, the *aspect ratio parameter* of the weighting or shape functions, where h_a is a measure of the point spacing around \mathbf{y}_a . As the value of γ increases, the Shepard functions become narrower. For a systematic method for defining the typical point spacing, see [10].

In the following, we will consider Shepard functions associated with the point-set P , which will be denoted as $w_a^P(\mathbf{x})$, $a = 1, 2, \dots, N$, and with the set of patch points Q , denoted by $w_A^Q(\mathbf{x})$, $A = 1, 2, \dots, M$. Note that each of these two PUs are defined not only from different node sets, but also from different sets of aspect ratio parameters. We also note that each of these sets of functions sums to one in every point in \mathbb{R}^D .

2.1. Local Euclidean structure

We turn now to a fundamental element in the proposed method, that of computing the local geometric structure of the manifold around Q_A . For this purpose, we use a fundamental idea in statistical learning: principal component analysis (PCA). PCA is a standard tool in computer graphics [2], data analysis [28], manifold learning [29], or model reduction techniques in computational mechanics based on proper orthogonal decomposition (POD) methods [30, 31]. PCA identifies the low-dimensional subspace that best explains the variance of a higher-dimensional data set. The original data are transformed into a new orthogonal coordinate system such that the projection of the data on the subspace defined by the first m coordinate directions maximizes the variance as compared with any other projection onto an m -dimensional subspace.

To describe the manifold properties around a patch point Q_A , a weighted local version, which we call wPCA (also kernel PCA), is more suitable. The main idea is to define a small neighborhood around Q_A , and compute the covariance matrix of the points of P in this neighborhood. If the local structure of the point-set around Q_A is that of a d -manifold, then the first d eigenvalues of the covariance matrix will strongly dominate the spectrum. If this is not the case, it is possible that the noise in the data is too large, and a larger domain of influence of the weighting function may be adequate. If this domain of influence is too large, the drawback is a loss of the local features of the manifold.

The average of the points in the neighborhood of Q_A is then

$$\bar{Q}_A = \sum_{a \in \mathcal{N}_{Q_A}^P} w_a^P(Q_A) P_a,$$

where $\mathcal{N}_{Q_A}^P$ denotes the neighbor index set of Q_A relative to the Shepard functions of P . Arranging the points as column vectors, we define the matrix $X_A \in \mathbb{R}^{D \times \text{card}(\mathcal{N}_{Q_A}^P)}$ whose columns are $P_a - \bar{Q}_A$, for $a \in \mathcal{N}_{Q_A}^P$. The covariance matrix is then

$$C_A = X_A \text{diag}\{w_a^P(\bar{Q}_A) | a \in \mathcal{N}_{Q_A}^P\} (X_A)^T \in \mathbb{R}^{D \times D}.$$

This positive- (semi-) definite symmetric matrix has real eigenvalues and diagonalizes in an orthonormal basis of eigenvectors. We define $V_A \in \mathbb{R}^{D \times d}$ as the eigenvector matrix formed by the d eigenvectors corresponding to the largest d eigenvalues. These eigenvectors form an orthogonal basis of the local Euclidean structure of the manifold, which can be informally thought of as a numerical tangent space at Q_A . This plane, T_A (see Figure 1), passes through Q_A and is parallel to these first d eigenvectors. The matrix V_A defines an orthogonal projection relative to Q_A onto the reduced space of dimension d , i.e.

$$\begin{aligned} \Pi_A : \mathbb{R}^D &\longrightarrow \mathbb{R}^d \\ \mathbf{z} &\longmapsto (V_A)^T (\mathbf{z} - \bar{Q}_A). \end{aligned}$$

Note that in general $\bar{Q}_A \neq Q_A$; hence $\Pi_A(Q_A) \neq \mathbf{0}$.

2.2. Local parameterization

The local Euclidean structure around Q_A provides a convenient parametric space for the embedded manifold, as is shown in Figure 2. Indeed, consider the reduced dimensionality point-set

$$\Xi_A = \{\xi_a = \Pi_A(P_a) | a \in \hat{\mathcal{N}}_{Q_A}^P\} \subset \mathbb{R}^d. \quad (5)$$

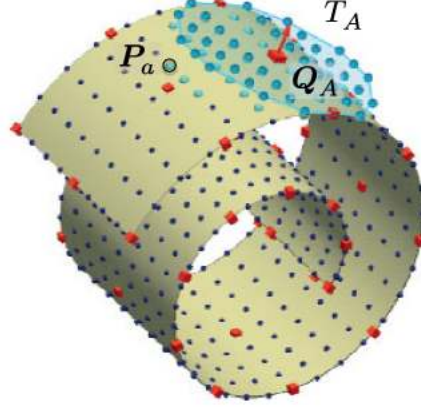


Figure 1. Local Euclidean structure (numerical tangent plane) T_A at the patch point Q_A (box) obtained by applying wPCA. The small spheres represent the N points in P , whereas the big spheres are the points in the neighborhood $\mathcal{N}_{Q_A}^P$ and their projection onto T_A . For this example, the eigenvalues of the covariance matrix are $\lambda = \{0.514, 0.485, 0.001\}$ (arbitrary units). The last eigenvalue is associated with the eigenvector normal to T_A .

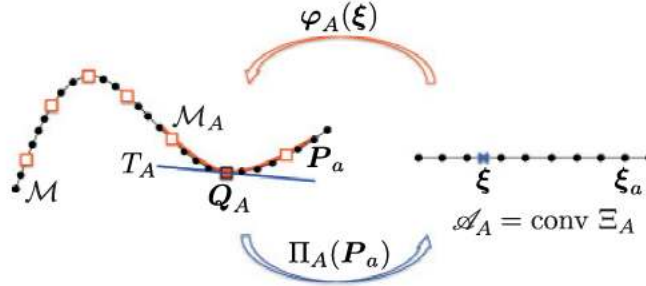


Figure 2. Illustration of the local parameterization φ_A of \mathcal{M} around Q_A . \mathcal{M}_A is a local approximation to the surface in the vicinity of Q_A , T_A is the numerical tangent, and $\mathcal{A}_A = \text{conv} \Xi_A$ is the local parametric space. Small solid points represent the N points in P , whereas the open square symbols represent the M patches in Q associated with the local parameterizations. Each local parameterization has an associated function in the PU w_A^Q .

The convenience of choosing this neighborhood list (the points in P lying within the support of w_A^Q) will become clear below. Suppose we have smooth approximants $p_a(\xi)$ associated with this point-set on a subset \mathcal{A}_A of \mathbb{R}^d , here the convex hull of the projected node set $\text{conv} \Xi_A$, and we locally parameterize the manifold around Q_A as

$$\begin{aligned} \varphi_A: \mathbb{R}^d \supset \text{conv} \Xi_A &\longrightarrow \mathbb{R}^D \\ \xi &\longmapsto \sum_{a \in \mathcal{N}_{Q_A}^P} p_a(\xi) P_a. \end{aligned}$$

The image of $\mathcal{A}_A = \text{conv} \Xi_A$ through the parameterization φ_A , \mathcal{M}_A , is a local approximation of the manifold around Q_A , see Figure 2.

2.3. PU to evaluate integrals on \mathcal{M}

A PU is a classical technique to patch together local constructions on a manifold [32]. We consider now a Shepard PU referred to a set of patch points, i.e. the functions $w_A^Q(x)$, $A = 1, 2, \dots, M$. These functions form a PU in \mathbb{R}^D , and consequently also in \mathcal{M} . Other partitions of unity are perfectly valid, including the local *max-ent* approximants described below used for the local parameterizations.

Our choice, the Shepard basis functions, is straightforward to evaluate and allow us to easily control the extent of the support.

Consider now the integral of a scalar function f over a manifold \mathcal{M} , $f : \mathcal{M} \rightarrow \mathbb{R}$. This function can also depend on other fields, as in a functional over the manifold or its associated weak form. Then we have the following identity:

$$\int_{\mathcal{M}} f(x) d\mathcal{M} = \sum_{A=1}^M \int_{\mathcal{M}} w_A^Q(x) f(x) d\mathcal{M}. \quad (6)$$

Combining the PU with the local parameterization around Q_A , we can approximate numerically integrals over the manifold \mathcal{M} described by a set of scattered points as

$$\int_{\mathcal{M}} f(x) d\mathcal{M} \simeq \sum_{A=1}^M \int_{\mathcal{A}_A} w_A^Q(\varphi_A(\xi)) f(\varphi_A(\xi)) J_A(\xi) d\xi,$$

where \mathcal{A}_A is the parametric space associated with the A th patch, and $J_A = \sqrt{\det[(D\varphi_A)^T D\varphi_A]}$ is the Jacobian determinant of the parameterization. In this way, similar to finite element methods, we have split the integral into local contributions that can be evaluated using local parameterizations.

The last integral can be subsequently approximated by numerical quadrature on the local parametric space. Here, we use the Gauss quadrature on a support triangulation defined over Ξ_A . The integrand often does not depend explicitly on x , and we express f in terms of ξ directly. This is not the case for the PU functions, which are defined on the ambient space \mathbb{R}^D . Defining a PU intrinsically on \mathcal{M} would be useful but seems difficult to implement. Also, from the above expression it is clear that the PU functions need to be compactly supported in \mathcal{M}_A , which motivates the choice of neighborhood index set in Equation (5). As a matter of fact, in computations we ensure this by choosing a slightly larger index set in Equation (5), see Figure 3.

We also note that in the present work, we establish a one-to-one correspondence between local parameterizations or patches and PU Shepard functions. This is not a strict requirement, and other possibilities may prove convenient.

2.4. Maximum-entropy approximants

The smooth approximants $p_a(\xi)$ used here are local *max-ent* (LME) meshfree approximants [25]. We briefly outline their construction and basic properties here. These approximation schemes fall into the class of convex approximants, such as natural neighbor approximants [33], subdivision approximants [18], or NURBS and B-splines basis functions [34]. Convex basis functions, which we

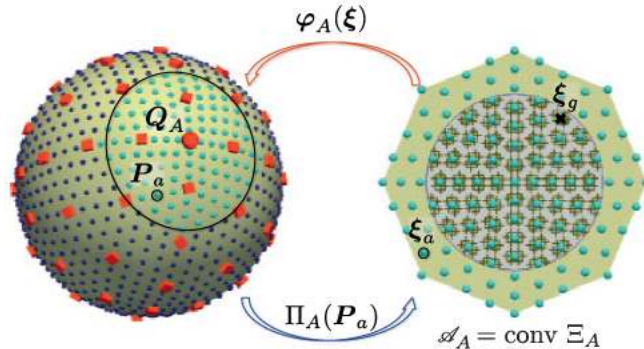


Figure 3. Illustration of the local parameterization φ_A of a sphere around a patch point Q_A . Small solid spheres represent the N points in P , whereas the box symbols represent the M patches where the local parameterizations are constructed (left). The points $\xi_a = \Pi_A(P_a)$ (the projections of the points $P_a, a \in \hat{\mathcal{N}}_{Q_A}^P$ onto the local parametric space \mathcal{A}_A) are represented by spheres, whereas brown cross symbols stand for the Gauss points ξ_g within in the support of $w_A^Q \circ \varphi_A$, displayed in gray (right).

will denote by $p_a(\xi), a = 1, \dots, n$ with $\xi \in \mathbb{R}^d$, are non-negative linearly reproducing approximants intimately related to convex geometry. To simplify the notation in this section, we renumber the point indices for a given patch $\hat{\mathcal{N}}_{\mathcal{Q}_A}^P$, from 1 to $n = \text{card}(\hat{\mathcal{N}}_{\mathcal{Q}_A}^P)$.

The local *max-ent* approximants used here are first-order consistent. Extensions to second-order approximation schemes based on adding extra constraints to the convex program below can be found in [26] Rosolen *et al.* (in preparation), whereas González *et al.* [35] propose a method based on the de Boor's algorithm. For a set of nodal values $\{u_a\}_{a=1, \dots, n}$ associated with a nodal set $\{\xi_a\}_{a=1, \dots, n}$, the approximation of a function $u(\xi)$ is given by

$$u(\xi) \approx \sum_{a=1}^n p_a(\xi) u_a,$$

where the approximants p_a are non-negative and fulfill the zeroth and first-order consistency conditions:

$$p_a(\xi) \geq 0, \quad \sum_{a=1}^n p_a(\xi) = 1, \quad \sum_{a=1}^n p_a(\xi) \xi_a = \xi.$$

The definition of the *max-ent* approximants is not explicit, but rather follows from an optimization problem set up at each evaluation point ξ , where the unknowns are the values of the basis functions at this point $p_a(\xi), a = 1, \dots, n$, and where the above conditions are seen as constraints on these unknowns. As shown in [25], the constraints are only feasible within the convex hull of the node set.

As the approximants are non-negative and form a PU (zeroth-order consistency condition) at each point ξ , they can be interpreted as a probability distribution. This information-theoretical viewpoint allows us to pose a statistical inference problem where the approximants p_a are the unknowns. A canonical measure of the uncertainty associated with a discrete probability distribution is the entropy, and the principle of maximum entropy provides the least biased approximation schemes consistent with the constraints. As proposed in [25], it proves convenient to select a compromise between maximizing the entropy and minimizing the width or support-size of the basis functions, which leads to the following convex program [36]:

$$\begin{aligned} \text{For fixed } \xi, \text{ minimize} \quad & \sum_{a=1}^n \beta_a p_a |\xi - \xi_a|^2 + \sum_{a=1}^n p_a \ln p_a \\ \text{subject to} \quad & p_a \geq 0, \quad a = 1, \dots, n \\ & \sum_{a=1}^n p_a = 1, \quad \sum_{a=1}^n p_a \xi_a = \xi. \end{aligned}$$

This convex optimization problem is solved efficiently and robustly with duality methods [25], obtaining

$$p_a(\xi) = \frac{1}{Z(\xi, \lambda^*(\xi))} \exp[-\beta_a |\xi - \xi_a|^2 + \lambda^*(\xi) \cdot (\xi - \xi_a)],$$

where

$$Z(\xi, \lambda) = \sum_{b=1}^n \exp[-\beta_b |\xi - \xi_b|^2 + \lambda \cdot (\xi - \xi_b)]$$

is the partition function and λ^* is the solution at each point ξ of the convex and unconstrained optimization problem

$$\lambda^*(\xi) = \arg \min_{\lambda \in \mathbb{R}^d} \ln Z(\xi, \lambda). \quad (7)$$

These approximants are meshfree since they are entirely defined by the node set $\{\xi_a\}_{a=1, \dots, n}$ and the parameters $\{\beta_a\}_{a=1, \dots, n}$. The parameter $\beta_a > 0$ is a measure of the locality of the shape

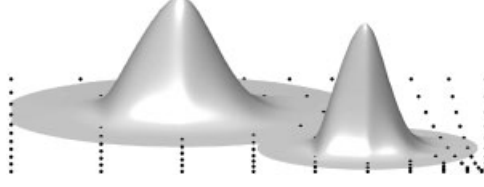


Figure 4. Local *max-ent* approximants for a non-uniform nodal distribution. The locality parameter $\beta_a = \gamma/h_a^2$ at each node is selected to achieve a uniform aspect parameter $\gamma = 1.6$.

functions. As mentioned in Section 2.1, a dimensionless aspect parameter $\gamma_a = \beta_a h_a^2$ characterizes the aspect ratio of the shape functions relative to the typical nodal spacing h_a in the vicinity of ξ_a . As the value of γ_a increases, the corresponding shape function is sharper and more local, and in the limit $\gamma_a \rightarrow \infty$ coincides with the Delaunay hat shape function. For highly unstructured point-sets, it is easy to produce local *max-ent* approximants of uniform aspect ratio by appropriately selecting a non-uniform value of the parameters β_a , resulting in a uniform value for the parameters γ_a (see Figure 4).

It has been shown that these approximants are C^∞ . Their gradients can be computed analytically. The expressions for the first and second spatial derivatives of local *max-ent* approximants are given in Appendix A, see also [36]. These approximants are non-interpolating, except at the boundary of the convex hull of the node set, where a weak Kronecker-delta property holds. This property makes it straightforward to impose essential boundary conditions unlike other meshfree methods such as those based on the MLS approximations [37]. Finally, we note that local *max-ent* approximants deal robustly with training data infected with zero-mean additive noise, as proven for non-parametric supervised classification [38].

2.5. Post-processing of the surface geometry and fields on the surface

The proposed method does not provide a way to patch together the local overlapping parameterizations of the surface or of fields on the surface. Indeed, the patching is performed at the level of the integrals. We describe next the procedure to post-process shapes and displacement fields used here. We assume that a number of points in space are sampling the true surface, that in general does not coincide with the sets P and Q . Given such a point $\mathbf{y} \in \mathcal{M}$, the numerical approximation of the surface takes the form

$$\boldsymbol{\varphi}^h(\mathbf{y}) = \sum_{A \in \mathcal{N}_y^Q} w_A^Q(\mathbf{y}) \sum_{a \in \mathcal{N}_A^P} p_a(\Pi_A(\mathbf{y})) \mathbf{P}_a, \quad (8)$$

where the patch projection Π_A has been defined in Section 2.1. Similarly, other fields on the surface described in terms of nodal values can be evaluated, for instance the displacement field

$$\mathbf{u}^h(\mathbf{y}) = \sum_{A \in \mathcal{N}_y^Q} w_A^Q(\mathbf{y}) \sum_{a \in \mathcal{N}_A^P} p_a(\Pi_A(\mathbf{y})) \mathbf{u}_a.$$

We note that in Equation (8), the points \mathbf{P}_a can be viewed as control points, following the B-Spline terminology. Indeed, as in B-Splines or NURBS, a point of the numerical surface is obtained as a convex combination of the control points, whose coefficients are the basis functions evaluated at a particular point in parameter space. Also, in the present context the numerical approximation of the surface does not pass through the control points either, and a control point has a local effect on the surface in its vicinity. Thus, in general, the control points should be chosen such that the actual surface, given analytically or through a set of sample points, is accurately reproduced, for instance with a least-square fit. In the examples below, we do not perform this pre-processing step, and directly place the points in P on the surface. This results in a small error in the description of the geometry that disappears as the discretization is refined.

2.6. Example: the Willmore energy of a sphere

The computation of the Willmore energy (related to the curvature strain energy of thin elastic sheets) of a spherical surface is presented next to illustrate and assess the methodology. The expression of the energy is

$$E = \int_{\Omega} H^2 d\Omega, \quad (9)$$

where H is the mean curvature and $d\Omega$ is the area element of the surface Ω . This functional involves the computation and integration of second-order spatial derivatives. For a sphere, we have $E = 16\pi$.

Following the method described above and for a surface described by a set of points $P = \{P_1, P_2, \dots, P_N\} \subset \mathbb{R}^3$, the elastic energy E is approximated by

$$\begin{aligned} E &= \int_{\Omega} H^2 d\Omega \\ &\simeq \sum_{A=1}^M \int_{\mathcal{A}_A} w_A^Q(\boldsymbol{\varphi}_A(\boldsymbol{\xi})) H^2(\boldsymbol{\xi}) J_A(\boldsymbol{\xi}) d\boldsymbol{\xi}, \end{aligned} \quad (10)$$

where for a surface we can compute the Jacobian determinant as $J_A = \|\boldsymbol{\varphi}_{A,1} \times \boldsymbol{\varphi}_{A,2}\|$. In the equation above, H is not an explicit function of $\boldsymbol{\xi}$, but rather of the first and second derivatives of the local parameterization $\boldsymbol{\varphi}_A$. The energy in Equation (10) is thus split into patch contributions $E \simeq E^h = \sum_{A=1}^M E_A^h$, which are evaluated using numerical quadrature

$$E_A^h = \sum_{g=1}^{n_g} w_A^Q(\boldsymbol{\varphi}_A(\boldsymbol{\xi}_g)) H^2(\boldsymbol{\xi}_g) J_A(\boldsymbol{\xi}_g) w_g,$$

where $\{\boldsymbol{\xi}_g\}_{g=1, \dots, n_g}$ are the quadrature point coordinates in the parametric space \mathcal{A}_A , with the corresponding weights w_g . We define the relative error for the elastic energy of the sphere as

$$e_{\text{rel}} = \frac{|16\pi - E^h|}{16\pi}.$$

We investigate the influence in the computation of E^h of (i) the accuracy of the integration rule and (ii) some key numerical parameters, see Table I. We consider the Gauss–Legendre cubature rules [39] of order 1, 2, 3, 4, 5, 6, 8, 10, and 15 for the Delaunay triangulation supported on the projected node sets Ξ_A ; these schemes have 1, 3, 4, 6, 7, 12, 16, 25, and 54 quadrature points per triangle, respectively. The numerical parameters entering the calculation, collected in the table, are a set of tolerances and aspect ratio parameters. More specifically, the tolerance TOL that sets the cutoff of the neighbor search, see Equations (2,4), can be chosen independently for the Shepard functions involved in the wPCA step, w_a^P , for the Shepard functions defining the PU that splits the integrals, w_A^Q , and for the local *max-ent* (LME) approximants p_a . Similarly, the aspect ratio parameter can be selected independently for each of these three approximants, i.e. γ_{wPCA} , γ_{PU} and γ_{LME} . Additionally, a tolerance must be set for the Newton–Raphson solution of the dual optimization problem in Equation (7) required to evaluate the local *max-ent* shape functions.

In setting the tolerances, it is important to note that those involved in the neighbor search, see Equations (2,4), have a noticeable effect on the computational cost of the method. The values

Table I. Numerical parameters used to perform the computations.

Parameter	wPCA	PU	LME
γ	1.8	3.0, 6.0	0.6, 1.0, 1.4
TOL	10^{-8}	10^{-6}	10^{-10}
TOL_{NR}	—	—	10^{-12}

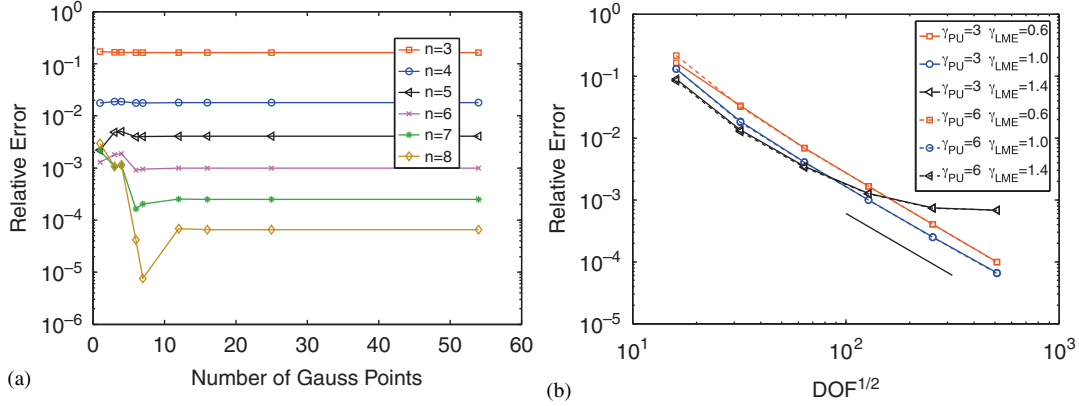


Figure 5. Relative error for the Willmore energy of a sphere. Error sensitivity with respect to (a) the number of quadrature points and (b) to the aspect-ratio parameter of the LME and PU shape functions. Also, a line of slope two for the visual inspection of (b) is shown.

chosen here provide good accuracy at a reasonable cost. The tolerance for the Newton–Raphson iterations influences very slightly the computational cost due to the fast quadratic convergence. As for the aspect ratio parameters, the calculations are robust for a wide range of values of γ_{wPCA} . The value selected in the Willmore energy calculations is shown in Table I, although a smaller value might be needed for data points affected by noise, or a higher value and more patch points for surfaces with sharp features. We analyze in more detail the sensitivity of the results with respect to the two remaining parameters, γ_{PU} and γ_{LME} .

We consider different levels of refinement of the point-set describing the sphere and of the set of patches. Specifically, we consider subdivisions of an octahedron following Loop’s scheme, and relocate the resulting points on the unit sphere. We consider the levels of refinement $n=3, 4, 5, 6, 7, 8$ with $N=2^{2(n+1)}+2$ points in P , and one level of refinement less for the patch points in Q , with $M=2^{2n}+2$ patches (see Figure 3). For this example, the refinement of the patches relative to the point-set P can be lowered by two or more levels without noticeable changes in the results.

Figure 5(a) shows the relative error e_{rel} for different levels of refinement and different number of cubature points. It can be observed that for coarse point-sets, the interpolation error dominates the quadrature error and the results are insensitive to the number of integration points. For finer point-sets, it can be observed that the accuracy is affected by the numerical quadrature, and up to 12 points per triangle are needed for the finest point-set, containing 262 146 points, in order to keep the quadrature error negligible relative to other errors.

Convergence plots in a log–log scale are shown in Figure 5(b) for different choices of the aspect ratio parameters γ_{LME} and γ_{PU} . The relative error in the Willmore energy converges with a slope of 2, i.e. $e_{\text{rel}} \propto h^2$. The insensitivity of the results with respect to γ_{PU} is apparent. Nevertheless, this parameter has a significant impact on the computational cost. Indeed, low values cause much overlap and redundancy in the numerical quadrature, whereas higher values greatly reduce the number of quadrature points in each patch (see also Figure 3 for an illustration). However, excessively large values may result in poor overlapping and degradation of the accuracy. From our experience, the interval between 4 and 6 is a suitable range for γ_{PU} . As for γ_{LME} , we know that for very large values, the surface \mathcal{M}_A will become polyhedral since the approximants will converge to the Delaunay shape functions. Therefore, for functionals depending on the curvature, lower values can be expected to produce accurate results. Indeed, we find numerically that the convergence is degraded for $\gamma_{\text{LME}}=1.4$. In thin-shell calculations, lower values of γ_{LME} result in a denser structure of the stiffness matrix, hence in a higher computational cost. To conclude this section, we highlight the fact that accurate results and numerical evidence of convergence is provided throughout the paper for functionals depending on second-order spacial derivatives and their corresponding fourth-order partial differential equations, while using linearly consistent approximation schemes.

3. THIN-SHELL MODEL

In this section, we review the mechanics of thin shells [11, 18], based on the geometrically exact formulation in [17, 19]. Here we restrict our attention to the linearized Kirchhoff–Love theory of shells, i.e. we neglect the shearing and stretching deformation normal to the shell mid-surface. In this theory, the shell director remains normal to the mid-surface during the deformation.

We follow the usual convention for Latin and Greek indices (i.e. $i = 1, 2, 3$; $\alpha = 1, 2$) a comma denotes partial differentiation, subscripts refer to covariant components, and superscripts denote contravariant components.

3.1. Kinematics of the shell

We next describe the kinematics of a thin-shell body $\mathcal{S} \subset \mathbb{R}^3$ in three-space. We assume that this body can be described by the pair (φ, t) , where the mapping φ defines the shell middle surface, Ω , and t is a field of unit vectors (a field of directors). We assume the thickness h of the shell to be uniform for simplicity. The thin shell body \mathcal{S} is given by

$$\mathcal{S} = \left\{ x \in \mathbb{R}^3 \mid x = \varphi(\xi^\alpha) + \xi t(\xi^\alpha), -\frac{h}{2} \leq \xi \leq \frac{h}{2}, (\xi^1, \xi^2) \in \mathcal{A} \right\}, \quad (11)$$

where $\mathcal{A} \subset \mathbb{R}^2$ is the parametric space for the middle surface, see Figure 6. Hence, we view a configuration x as a mapping from a parametric domain $\mathcal{A} \times [-h/2, h/2]$ into \mathbb{R}^3 . The parametric domain is described by the coordinates $\{\xi^1, \xi^2, \xi^3\}$ (where we identify $\xi = \xi^3$), whose corresponding dual basis is $\{E^i\}$. The area element of the middle surface can be computed as $d\Omega = \bar{j} d\xi^1 d\xi^2$, where $\bar{j} = \|\varphi_{,1} \times \varphi_{,2}\|$. The tangent map of a given configuration Tx can be computed from the convective basis vectors g_i as

$$Tx = \frac{\partial x}{\partial \xi^i} \otimes E^i = g_i \otimes E^i,$$

with $g_\alpha = \partial x / \partial \xi^\alpha = \varphi_{,\alpha} + \xi t_{,\alpha}$ and $g_3 = \partial x / \partial \xi = t$. The covariant components of the metric tensor in convected coordinates are given by $g_{ij} = g_i \cdot g_j$.

Hereinafter, subscript 0 denotes quantities in the reference configuration, for instance φ_0 is a point on the reference middle surface. A deformation mapping is a mapping from a reference body into \mathbb{R}^3 , $x \circ x_0^{-1}$. Consequently, the deformation gradient is $F = Tx(Tx_0)^{-1}$, and the Jacobian is $J = \det(F) = j/j_0$, where $j = \det(Tx) = g_3 \cdot (g_1 \times g_2)$.

The shell director in the reference configuration t_0 coincides with the normal to the undeformed middle surface of the shell and hence has the properties

$$t_0 = \frac{\varphi_{0,1} \times \varphi_{0,2}}{\bar{j}_0}, \quad \varphi_{0,\alpha} \cdot t_0 = 0, \quad |t_0| = 1, \quad t_0 \cdot t_{0,\alpha} = 0.$$

In general, the director in the deformed configuration of the shell, t , is allowed to be an arbitrary vector field over $\Omega = x(\mathcal{A} \times \{0\})$.

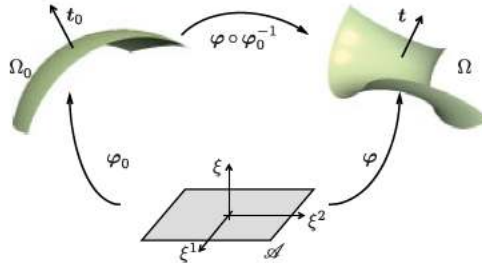


Figure 6. Schematic description of the reference, deformed and parametric configurations of the middle shell surface.

The local shell deformations can be characterized by the Green–Lagrange strain tensor. Since the convected components of the metric tensor coincide with the components of $(T\mathbf{x})^T T\mathbf{x}$ in the basis associated with $\{\zeta^i\}$, the Green–Lagrange strain tensor can be expressed as the difference between the metric tensors on the deformed and undeformed configurations of the shell, i.e.

$$E_{ij} = \frac{1}{2}(g_{ij} - g_{0ij}) = \frac{1}{2}(\mathbf{x}_{,i} \cdot \mathbf{x}_{,j} - \mathbf{x}_{0,i} \cdot \mathbf{x}_{0,j}).$$

Plugging the basic kinematic ansatz $\mathbf{x} = \boldsymbol{\varphi}(\zeta^\alpha) + \zeta \mathbf{t}(\zeta^\alpha)$ into the above expression and grouping terms, we obtain

$$E_{ij} = \varepsilon_{ij} + \zeta \rho_{ij} + (\zeta)^2 \vartheta_{ij}, \quad (12)$$

which admits the following interpretation in terms of the symmetric tensors ε_{ij} , ρ_{ij} , and ϑ_{ij} :

- The membrane strain tensor $\varepsilon_{\alpha\beta} = \frac{1}{2}(\boldsymbol{\varphi}_{,\alpha} \cdot \boldsymbol{\varphi}_{,\beta} - \boldsymbol{\varphi}_{0,\alpha} \cdot \boldsymbol{\varphi}_{0,\beta})$, which lives on the middle surface, measures the in-plane deformation of the surface; the components $\varepsilon_{\alpha 3} = \frac{1}{2} \boldsymbol{\varphi}_{,\alpha} \cdot \mathbf{t}$ measure the shearing of the director \mathbf{t}_0 ; and the component $\varepsilon_{33} = \frac{1}{2}(\mathbf{t} \cdot \mathbf{t} - 1)$ measures the stretching of the director \mathbf{t}_0 .
- The bending or change in curvature of the shell is measured by the tensor $\rho_{\alpha\beta} = \boldsymbol{\varphi}_{,\alpha} \cdot \mathbf{t}_{,\beta} - \boldsymbol{\varphi}_{0,\alpha} \cdot \mathbf{t}_{0,\beta}$, and $\rho_{\alpha 3} = \frac{1}{2} \mathbf{t}_{,\alpha} \cdot \mathbf{t}$ measures the shearing originated from the director elongation; the in-plane tensor $\vartheta_{\alpha\beta} = \frac{1}{2}(\mathbf{t}_{,\alpha} \cdot \mathbf{t}_{,\beta} - \mathbf{t}_{0,\alpha} \cdot \mathbf{t}_{0,\beta})$ is exclusively related to changes of the middle surface directors. The rest of the components vanish, $\rho_{33} = \vartheta_{3i} = \vartheta_{i3} = 0$.

3.2. Kirchhoff–Love hypothesis

In the remainder of this section, we restrict our attention to the Kirchhoff–Love theory of thin shells, i.e. we constrain the deformed director \mathbf{t} to coincide with the unit normal of the deformed middle surface of the shell, i.e.

$$\mathbf{t} = \frac{\boldsymbol{\varphi}_{,1} \times \boldsymbol{\varphi}_{,2}}{j}, \quad \boldsymbol{\varphi}_{,\alpha} \cdot \mathbf{t} = 0, \quad |\mathbf{t}| = 1, \quad \mathbf{t} \cdot \mathbf{t}_{,\alpha} = 0.$$

Consequently, the theory can be formulated exclusively in terms of the shell middle surface. We introduce its first and second fundamental forms expressed in convected components

$$a_{\alpha\beta} = \boldsymbol{\varphi}_{,\alpha} \cdot \boldsymbol{\varphi}_{,\beta},$$

$$\kappa_{\alpha\beta} = \boldsymbol{\varphi}_{,\alpha} \cdot \mathbf{t}_{,\beta} = -\boldsymbol{\varphi}_{,\alpha\beta} \cdot \mathbf{t}.$$

Here we have identified the director with the normal. With the Kirchhoff–Love hypothesis, the only remaining non-zero components of the Green–Lagrange strain tensor are

$$E_{\alpha\beta} = \frac{1}{2}(a_{\alpha\beta} - a_{0\alpha\beta}) + \zeta(\kappa_{\alpha\beta} - \kappa_{0\alpha\beta}) + \frac{(\zeta)^2}{2}(\mathbf{t}_{,\alpha} \cdot \mathbf{t}_{,\beta} - \mathbf{t}_{0,\alpha} \cdot \mathbf{t}_{0,\beta})$$

$$= \varepsilon_{\alpha\beta} + \zeta \rho_{\alpha\beta} + (\zeta)^2 \vartheta_{\alpha\beta}. \quad (13)$$

3.3. Small displacements

As we say at the beginning of this section, we assume that the deformation field for the shell is restricted to account only for small displacement. Then, if the displacement vector of the middle surface of the shell is defined as $\mathbf{u} = \boldsymbol{\varphi} - \boldsymbol{\varphi}_0$, the linear membrane and bending strain measures can be derived from the kinematic variables up to first order in \mathbf{u} in the form

$$\varepsilon_{\alpha\beta} = \frac{1}{2}(\boldsymbol{\varphi}_{0,\alpha} \cdot \mathbf{u}_{,\beta} + \boldsymbol{\varphi}_{0,\beta} \cdot \mathbf{u}_{,\alpha}),$$

$$\rho_{\alpha\beta} = -\mathbf{t}_0 \cdot \mathbf{u}_{,\alpha\beta} - \boldsymbol{\varphi}_{0,\alpha\beta} \cdot \Delta \mathbf{t},$$

$$\vartheta_{\alpha\beta} = \frac{1}{2}(\mathbf{t}_{0,\alpha} \cdot \Delta \mathbf{t}_{,\beta} + \Delta \mathbf{t}_{,\alpha} \cdot \mathbf{t}_{0,\beta}),$$

with

$$\begin{aligned} \mathbf{t}_{0,\alpha} &= -\bar{j}_0^{-1} \{ \mathbf{t}_0 \times [\mathbf{t}_0 \times (\boldsymbol{\varphi}_{0,1\alpha} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,2\alpha})] \}, \\ \Delta \mathbf{t} &= -\bar{j}_0^{-1} \{ \mathbf{t}_0 \times [\mathbf{t}_0 \times (\mathbf{u}_{,1} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \mathbf{u}_{,2})] \}, \end{aligned}$$

where $\Delta \mathbf{t}$ is the approximation to $\mathbf{t} - \mathbf{t}_0$ up to first order in \mathbf{u} . We refer to the Appendix B for a detailed derivation of the linearized strains.

3.4. Equilibrium configuration of thin shells

The potential energy of an elastic shell body with internal energy density W can be expressed by the functional

$$\Pi[\mathbf{u}] = \int_{\mathcal{S}_0} W(\mathbf{u}) dV_0 + \Pi_{ext}[\mathbf{u}],$$

where Π_{ext} is the potential energy of the external loads. For concreteness, we consider an isotropic Kirchhoff–St. Venant elastic material, with an internal energy density expressed as [40]

$$W = \frac{1}{2} C^{ijkl} E_{ij} E_{kl},$$

where C^{ijkl} are the contravariant components of the elasticity tensor.

For thin-shell bodies, the Green–Lagrange tensor components are commonly retained up to first order in h , see Equation (13), and the effect of curvature on the configuration Jacobian away from the middle surface is neglected, that is $j_0/\bar{j}_0 = 1$ [17, 19]. Assuming that the elasticity tensor does not vary through-the-thickness, the internal energy density can be integrated through-the-thickness, resulting in an internal energy density per unit area

$$\mathcal{W} = \frac{1}{2} \int_{-h/2}^{h/2} C^{\alpha\beta\gamma\delta} E_{\alpha\beta} E_{\gamma\delta} \frac{j_0}{\bar{j}_0} d\xi \simeq \frac{1}{2} C^{\alpha\beta\gamma\delta} \left(h \varepsilon_{\alpha\beta} \varepsilon_{\gamma\delta} + \frac{h^3}{12} \rho_{\alpha\beta} \rho_{\gamma\delta} \right),$$

with

$$C^{\alpha\beta\gamma\delta} = \frac{E}{(1-\nu^2)} \left[\nu a_0^{\alpha\beta} a_0^{\gamma\delta} + \frac{1}{2} (1-\nu) \left(a_0^{\alpha\gamma} a_0^{\beta\delta} + a_0^{\alpha\delta} a_0^{\beta\gamma} \right) \right],$$

where $a_0^{\alpha\gamma} (a_0)_{\gamma\beta} = \delta_{\beta}^{\alpha}$, E is Young's modulus, and ν the Poisson ratio. Thus, the internal potential energy can be written as an integral over the reference middle surface

$$\Pi_{int}[\mathbf{u}] = \int_{\Omega_0} \mathcal{W}(\mathbf{u}) d\Omega_0,$$

and the external potential becomes

$$\Pi_{ext}[\mathbf{u}] = - \int_{\Omega_0} \mathbf{q} \cdot \mathbf{u} d\Omega_0 - \int_{\partial\Omega_0} \mathbf{h} \cdot \mathbf{u} d\ell_0,$$

where \mathbf{q} is the external body load per unit area, \mathbf{h} the forces per unit length applied on the boundary of the middle surface, and $d\ell_0$ is the line element of the boundary of the middle surface.

The equilibrium displacement field is obtained by requiring stationarity of the total potential energy with respect to all admissible variations $\boldsymbol{\eta}$

$$\delta \Pi[\mathbf{u}, \boldsymbol{\eta}] = \frac{d}{d\varepsilon} \Pi(\mathbf{u} + \varepsilon \boldsymbol{\eta}) \Big|_{\varepsilon=0} = 0.$$

Here, the admissible displacement fields and variations must be consistent with the essential boundary conditions. For thin shells, these can be in terms of the displacements, in $\partial\Omega_0^u$, or the rotations of the director about the tangent to the boundary, in $\partial\Omega_0^\theta$. Note that $\partial\Omega_0^u$ and $\partial\Omega_0^\theta$ can

overlap. The above is a statement of the principle of virtual work, which can be expressed in terms of integrals over the parametric space \mathcal{A} as follows:

$$\begin{aligned}\delta\Pi_{int}[\mathbf{u}, \boldsymbol{\eta}] &= \int_{\mathcal{A}} C^{\alpha\beta\gamma\delta} \left(h\varepsilon_{\gamma\delta}(\mathbf{u})\varepsilon_{\alpha\beta}(\boldsymbol{\eta}) + \frac{h^3}{12}\rho_{\gamma\delta}(\mathbf{u})\rho_{\alpha\beta}(\boldsymbol{\eta}) \right) \bar{j}_0 d\xi^1 d\xi^2, \\ \delta\Pi_{ext}[\boldsymbol{\eta}] &= - \int_{\mathcal{A}} \mathbf{q} \cdot \boldsymbol{\eta} \bar{j}_0 d\xi^1 d\xi^2 - \int_{\partial\mathcal{A}} \mathbf{h} \cdot \boldsymbol{\eta} \|\boldsymbol{\varphi}_{0,t}\| d\ell_{\xi},\end{aligned}$$

where $\boldsymbol{\varphi}_{0,t}$ is the partial derivative of the middle surface mapping in the tangential direction to the boundary curve.

Following [17], we introduce the elastic constitutive relations between the shell stresses and the strains as

$$\begin{aligned}n^{\alpha\beta} &= \frac{\partial \mathcal{W}}{\partial \varepsilon_{\alpha\beta}} = hC^{\alpha\beta\gamma\delta} \varepsilon_{\gamma\delta}, \\ m^{\alpha\beta} &= \frac{\partial \mathcal{W}}{\partial \rho_{\alpha\beta}} = \frac{h^3}{12} C^{\alpha\beta\gamma\delta} \rho_{\gamma\delta},\end{aligned}$$

where $n^{\alpha\beta}$ is the effective membrane stress and $m^{\alpha\beta}$ is the effective bending stress, which can be interpreted as force and moment resultants. Further, by recourse to Voigt's notation, we obtain the following convenient expressions:

$$\mathbf{n} = \begin{pmatrix} n^{11} \\ n^{22} \\ n^{12} \end{pmatrix} = h\mathbf{C}\boldsymbol{\varepsilon}, \quad \mathbf{m} = \begin{pmatrix} m^{11} \\ m^{22} \\ m^{12} \end{pmatrix} = \frac{h^3}{12}\mathbf{C}\boldsymbol{\rho}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{pmatrix}, \quad \boldsymbol{\rho} = \begin{pmatrix} \rho_{11} \\ \rho_{22} \\ 2\rho_{12} \end{pmatrix},$$

where the matrix \mathbf{C} is given by the expression

$$\mathbf{C} = \frac{E}{1-\nu^2} \begin{pmatrix} (a_0^{11})^2 & \nu a_0^{11} a_0^{22} + (1-\nu)(a_0^{12})^2 & a_0^{11} a_0^{12} \\ & (a_0^{22})^2 & a_0^{22} a_0^{12} \\ \text{symm} & & \frac{1}{2}[(1-\nu)a_0^{11} a_0^{22} - (1+\nu)(a_0^{12})^2] \end{pmatrix}.$$

Finally, with the above definitions, we can write the principle of virtual work as

$$0 = \delta\Pi[\mathbf{u}, \delta\mathbf{u}] = \int_{\mathcal{A}} (\delta\boldsymbol{\varepsilon} \cdot \mathbf{n} + \delta\boldsymbol{\rho} \cdot \mathbf{m}) \bar{j}_0 d\xi^1 d\xi^2 + \delta\Pi_{ext}[\delta\mathbf{u}]. \quad (14)$$

3.5. Galerkin discretization

We consider now the discrete equilibrium equations $\mathbf{K}\mathbf{U}=\mathbf{f}$ for surfaces numerically represented with the procedure described before, in terms of a set of nodes $P=\{\mathbf{P}_a\}$, $a=1, \dots, N$, and a set of patches identified with the patch points $Q=\{\mathbf{Q}_A\}$, $A=1, \dots, M$. We consider a specific patch A . Let $\boldsymbol{\varphi}_{0A}$ be a configuration mapping for the middle surface for this patch, defined over the parametric space \mathcal{A}_A

$$\boldsymbol{\varphi}_{0A}(\boldsymbol{\xi}) = \sum_{a \in \hat{\mathcal{N}}_{\mathbf{Q}_A}^P} p_a(\boldsymbol{\xi}) \mathbf{P}_a,$$

as described in Section 2.2. We represent the displacement field in a given patch A as

$$\mathbf{u}_A(\boldsymbol{\xi}) = \sum_{a \in \hat{\mathcal{N}}_{\mathbf{Q}_A}^P} p_a(\boldsymbol{\xi}) \mathbf{u}_a.$$

Virtual displacements are represented likewise. With the strategy presented in Section 2.3, we can split the integrals in the principle of virtual work into patch contributions, e.g.

$$\delta\Pi_{int}[\mathbf{u}, \delta\mathbf{u}] = \sum_{A=1}^M \int_{\mathcal{A}_A} [(\delta\boldsymbol{\varepsilon} \cdot \mathbf{n} + \delta\boldsymbol{\rho} \cdot \mathbf{m}) \bar{j}_0]_A (w_A^Q \circ \boldsymbol{\varphi}_0) d\xi^1 d\xi^2. \quad (15)$$

Here, $[\cdot]_A$ means that the expression within the brackets is evaluated with the A th patch approximation of the undeformed middle surface, the displacement field, and the virtual displacement field. A simple calculation yields the Galerkin stiffness matrix. The interaction between nodes a and b is given by

$$\mathbf{K}^{ab} = \sum_{A=1}^M \int_{\mathcal{A}_A} \left[\left(h\mathbf{M}^{aT} \mathbf{C}\mathbf{M}^b + \frac{h^3}{12} \mathbf{B}^{aT} \mathbf{C}\mathbf{B}^b \right) \bar{j}_0 \right]_A (w_A^Q \circ \boldsymbol{\varphi}_0) d\xi^1 d\xi^2,$$

where \mathbf{M}^a and \mathbf{B}^a are the membrane and bending strain–displacement matrices for the a th node. Note that $\mathbf{M}^a, \mathbf{B}^a \in \mathbb{R}^{3 \times 3}$. See Appendix C for a detailed description. The force contribution of the a th node is

$$\mathbf{f}^a = \sum_{A=1}^M \int_{\mathcal{A}_A} [\mathbf{q} \ p_a \ \bar{j}_0]_A (w_A^Q \circ \boldsymbol{\varphi}_0) d\xi^1 d\xi^2 + \sum_{A=1}^M \int_{\partial\mathcal{A}_A} [\mathbf{h} \ p_a \|\boldsymbol{\varphi}_{0,t}\|]_A (w_A^Q \circ \boldsymbol{\varphi}_0) d\ell_\xi.$$

The Dirichlet displacement and rotation boundary conditions are imposed with the Lagrange multipliers, as in [11], see Appendix D.

4. NUMERICAL EXAMPLES

We confront next the proposed method with a set of standard numerical benchmark tests. We consider a square plate loaded with a point force, and the shell obstacle course of Belytschko *et al.* [27] for the linear analysis of thin shells. The test problems in the shell obstacle course are Scordelis–Lo’s roof loaded by self-weight, a pinched short cylinder, and a hemisphere loaded with two pairs of facing concentrated forces. At the end of this section, the flexibility of the proposed methodology to deal with the shells of complex topology and geometry is illustrated by two additional examples. Before presenting the numerical examples, a few remarks concerning the numerical implementation of the method are collected.

4.1. Numerical aspects

In all the calculations of this section, we fix $\gamma_{wPCA} = 1.8$ and $TOL_{wPCA} = 10^{-8}$ for the tangent plane weight functions, $\gamma_{PU} = 4.0$ and $TOL_{PU} = 10^{-6}$ for the PU shape functions, and $TOL_{LME} = 10^{-10}$ and $TOL_{NR} = 10^{-12}$ for the LME approximants. As for γ_{LME} , we discuss its influence with the first numerical example, and then fix it to $\gamma_{LME} = 0.8$, which provides accurate solutions at a moderate computational cost. We note that the present approach neither requires an external specification of the normals to the surface nor of the tangent vectors to the shell boundary [11].

4.1.1. Numerical quadrature. Numerical quadrature is one of the main outstanding issues in mesh-free methods. Here, we follow the straightforward approach of building a Delaunay triangulation in the parametric domain of each patch \mathcal{A}_A , and placing quadrature points supported on this local planar triangulation. More advanced technologies exist that probably result in more efficient simulations [41]. In all the examples, we used a standard Gauss–Legendre cubature rule of 12 points (order 6) per triangle. In the boundary curves, a quadrature scheme of four Gauss–Legendre points per integration cell was used.

4.1.2. Boundary conditions. In principle, with the weak Kronecker-delta property of the local *max-ent* approximants [25] at the boundary, the displacement Dirichlet boundary conditions can be treated directly. However, we have noticed through numerical experiments that this results in

locking, as the local *max-ent* approximants become flat (with zero curvature) near the boundary of the domain. This could be avoided using second-order *max-ent* approximants, a topic of current investigation.

We bypass this difficulty introducing ghost nodes, like in subdivision finite elements [18, 42]. This destroys the weak Kronecker-delta property on the boundary, but produces convergent approximations to shapes with non-vanishing curvature near the boundary. Consequently, we use the Lagrange multipliers both for the slope and for the displacement boundary conditions, see [11] and Appendix D.

4.1.3. Patch vs node points. A systematic study of the optimal relation between the number and location of the patch points relative to the node points has not been performed here. Related issues have been addressed in the area of computer graphics for surface reconstruction and rendering from unorganized set of points [6]. In the later work [43], a methodology to obtain the patch points as a down-sampling of the node points was presented in an MLS context. Through numerical tests, we have observed that the number of patch points needs to be large enough to capture the geometry of the surface. In our examples, we define refinement rules for the node sets, and choose a lower order of refinement for the patches than for the nodes. To give an idea of the extent of the patches and their overlapping, for $\gamma_{\text{PU}} \in [4, 6]$ and $TOL_{\text{PU}} = 10^{-6}$, the support of the patch PU functions w_A^Q is of the order of $[1.52, 1.86]h_A$, where h_A is the characteristic spacing of the patch points.

4.2. Pinched plate

We consider a clamped square plate with a concentrated central force, see Figure 7(a). In this test, the bending behavior completely dominates the solution. The solution of the plate center deflection is $\delta_z = 0.00560FL^2/D$, where $D = Eh^3/(12(1-\nu^2))$ is the flexural rigidity of the plate [44]. The side length is $L=10$, the thickness $h=0.1$, the concentrated load applied at its center $F=200$, Young's modulus $E=10^6$, and the Poisson ratio $\nu=0.3$. Material and geometrical parameters coincide with those used by Noels and Radovitzky [24], where a Discontinuous Galerkin method for thin shells is developed. Since the plate center deflection does not contain enough significant digits, we use a converged displacement obtained with a very fine discretization, $\delta_z = 0.0056122FL^2/D$, which is similar to the value reported in [24].

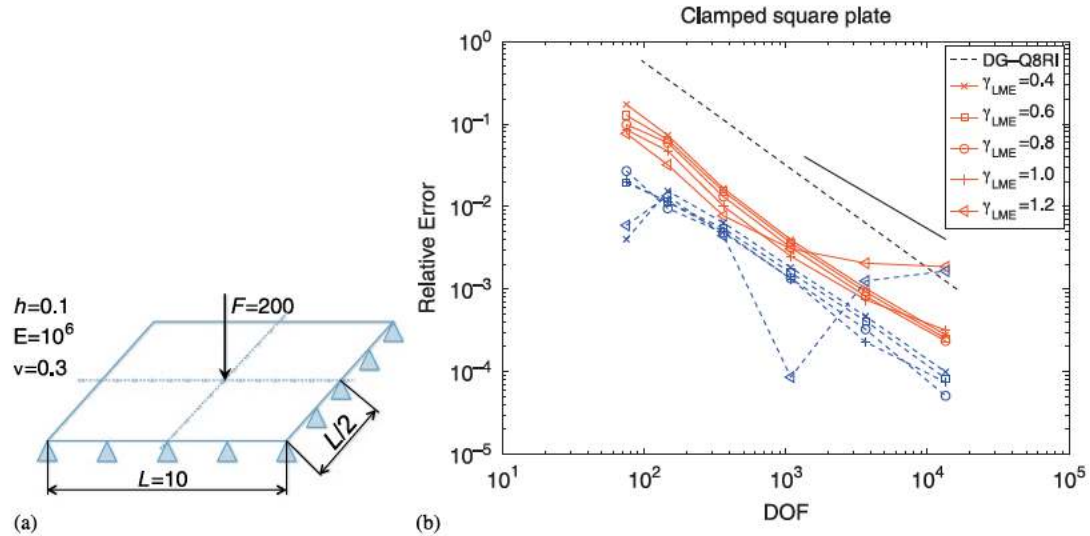


Figure 7. (a) Sketch for the pinched plate problem and (b) relative error for the center deflection. Results for the full model (—) and quarter model cases (---) are depicted. In (b) a line segment of slope two is shown to help in the visual inspection of the curves.

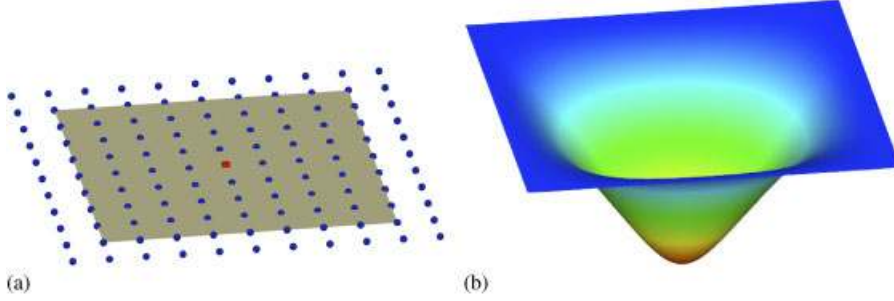


Figure 8. Full model of the clamped plate: (a) discretization and reference configuration and (b) computed deformed surface, $\gamma_{\text{LME}}=0.8$. In the picture, the high smoothness of the deflected plate is noteworthy.

Owing to the symmetry of the problem, a quarter of the plate with appropriate symmetry boundary conditions can be considered. The convergence plots for the relative error of the center deflection for both full and quarter clamped plate models are illustrated in Figure 7(b). As expected, the models that take symmetry into account and consider one quarter of the plate have a denser node distribution for a given number of degrees of freedom, hence produce more accurate solutions. The results are compared with those obtained by using a DG method (values from [24]). The figure illustrates the high accuracy of the local *max-ent* approximations for $\gamma_{\text{LME}}=0.4, 0.6, 0.8$. We observe a degradation of the convergence for $\gamma_{\text{LME}} \geq 1.0$, similar to the behavior reported in Section 2.6. The superior accuracy of the proposed approach as compared with the DG method in this example is clear from the figure, with more than one order of magnitude of more accurate results. Note that these authors report the results for the quarter model; hence, their results need to be compared with the dashed lines in the figure. As the rate of convergence of the DG method is higher, this is not expectable for extremely fine discretizations. We note again that our method is based on smooth approximants that are only first-order consistent. Structured and unstructured sets of nodes have been used, with no significant difference in the result. Figure 8 illustrates the node set, with the ghost nodes, and the smooth deformation. In the remaining examples, by simplicity, a constant value of $\gamma_{\text{LME}}=0.8$ has been selected.

4.3. Scordelis–Lo roof

In this example, a cylindrical roof is loaded by uniform gravity $g=90$, and supported by rigid diaphragms in the arched sides, $u_x=u_y=0$. Material and geometrical parameters are detailed in Figure 9(a). In the Scordelis–Lo’s roof test, both the membrane and the flexural strain energies dominate the overall behavior of the model; hence, representation of inextensional modes is not crucial in this problem. Therefore, this test evaluates the inadequacies of the numerical method to compute membrane stresses, which severely could inhibit convergence.

The reference value for the maximum vertical displacement at the mid-side if the free edge reported in the literature presents a wide scatter, e.g. with a 3% difference between the converged value found here and that reported in [42]. For an overkill discretization we find $\delta_z=0.300575$, which is 0.6% lower than the value given in [27] of 0.3024. Figure 9(b) shows the convergence results obtained in both the full and quarter case, which exhibit a power-law asymptotic convergence behavior. Figure 10 shows typical node and patch point-sets, as well as the deformation for this example.

4.4. Pinched hemisphere

In this example, a hemispherical shell of radius $R=10$ and thickness $h=0.04$ is subjected to radial loads $F=2$ applied on two diametral directions, see Figure 11. This is a challenging test for assessing the method’s ability to represent inextensional deformations under complex shell bending conditions involving curvature in two directions. The ability to bend without developing parasitic membrane strains is essential for good performance in this problem. Furthermore, the

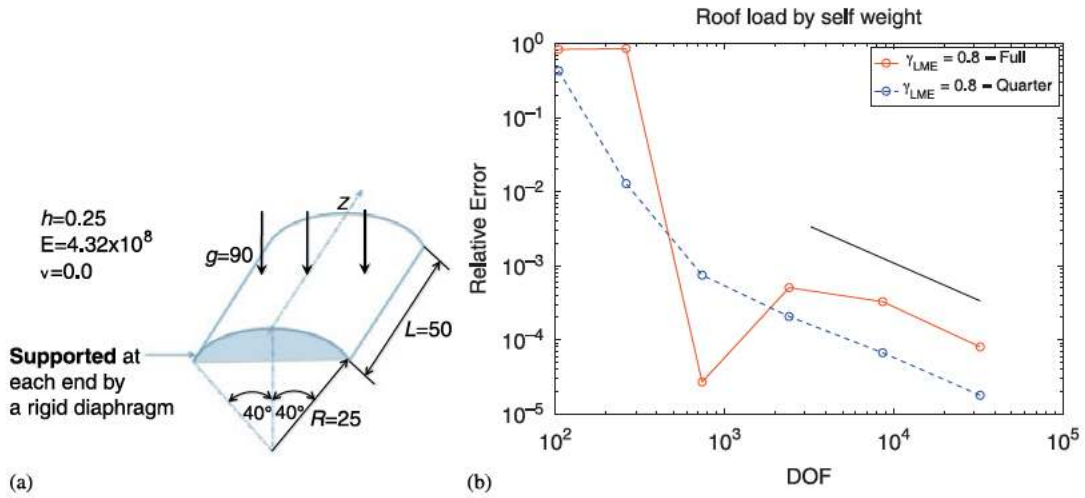


Figure 9. (a) Sketch for the Scordelis-Lo roof problem and (b) normalized deflection of the vertical displacement at the mid-side of the free edge, and line segment of slope two. Results are given for the full model (top —) and for the quarter model (bottom ---).

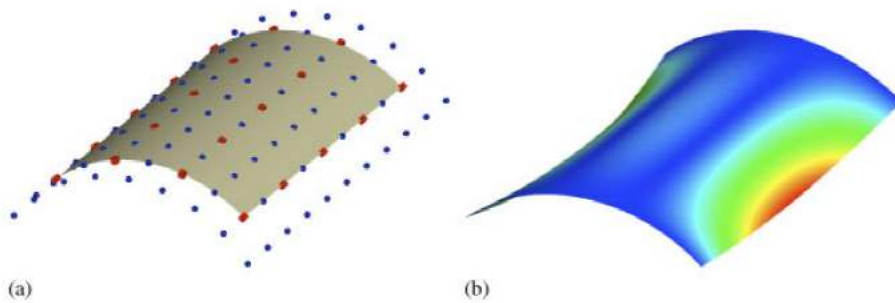


Figure 10. Full model of the Scordelis-Lo's roof: (a) discretization of both patches \diamond and nodes \circ , and the reference configuration and (b) computed deformed surface, $\gamma_{LME} = 0.8$.

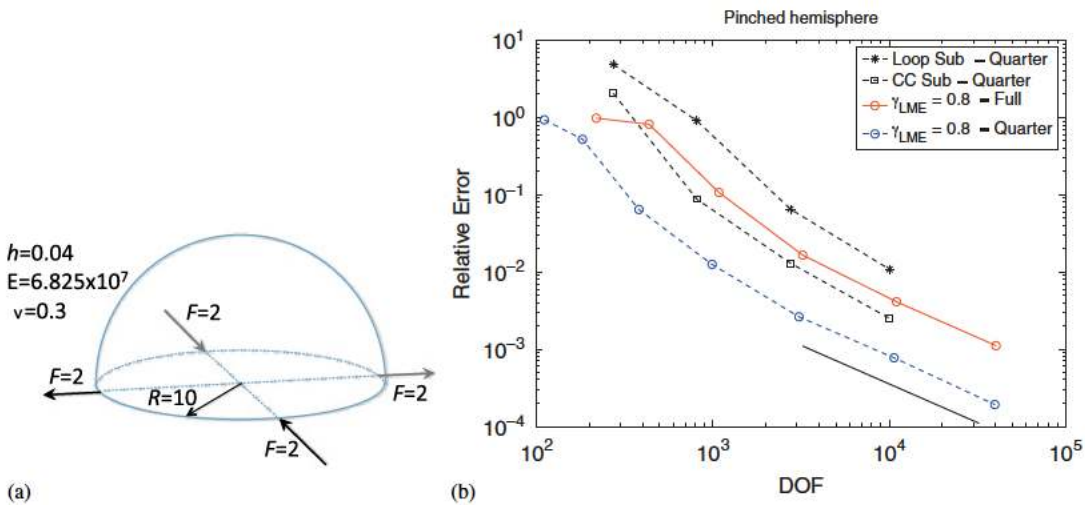


Figure 11. (a) Sketch for the pinched hemisphere problem and (b) pinched hemispherical shell problem. Results for the convergence of the normalized radial displacement are given the full model (top —) and for the quarter model (bottom ---).

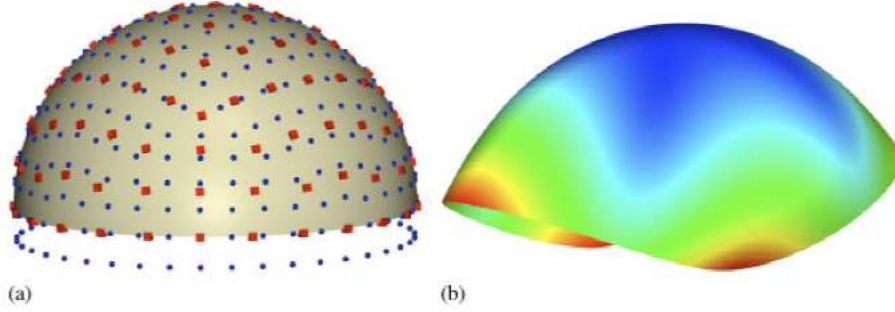


Figure 12. Full model of the pinched hemisphere: (a) discretization of both patches \diamond - and nodes \circ -, and reference configuration and (b) computed deformed surface, $\gamma_{LME}=0.8$.

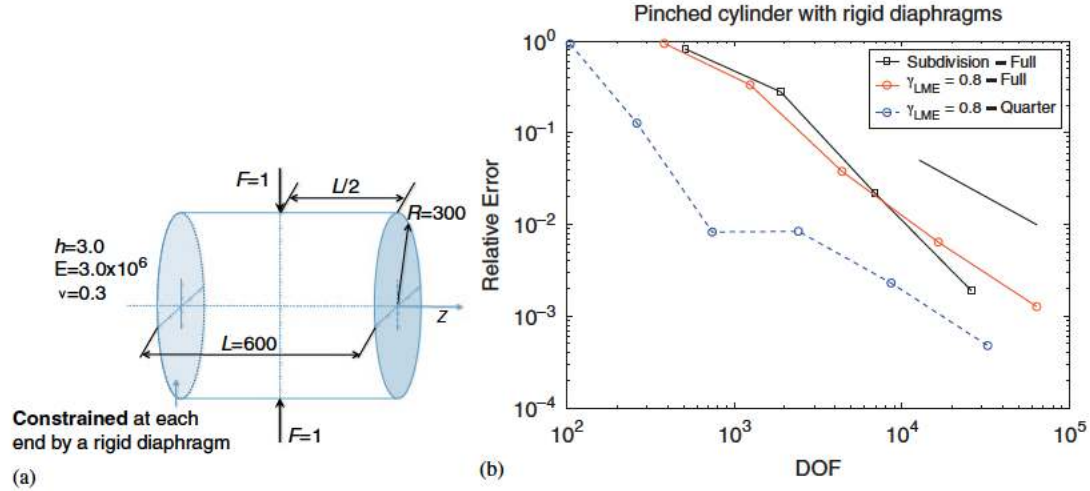


Figure 13. (a) Sketch for the pinched cylinder with rigid diaphragms and (b) pinched cylinder with rigid diaphragms problem. Convergence rates for the relative error of the deflection under the loads are given for the full model (top —) and for an eighth of the model (bottom ---).

case of a hemispherical shell provides an example of a surface that presents necessarily irregular nodes, i.e. which cannot be meshed in a structured way without singularities. For instance, in the presence of such nodes, simple box-spline methods are not applicable, and a special treatment in subdivision finite element methods is required [18].

The convergence of the relative error for the radial displacement is shown in Figure 11, for both the full and quarter models. The displacements are normalized by a deflection value of $\delta_r=0.092414$, which is in agreement with the lower bound value given by MacNeal and Harder [15, 27] and Belytschko *et al.* [27], that is $\delta_r=0.0924$. In this figure, we plot the convergence results reported in [42] for subdivision finite elements based on Loop's scheme (triangular elements) and on Catmull-Clark's scheme (quadrilateral elements). The superior accuracy of the proposed meshfree approach is clear from the figure, with over one order of magnitude better accuracy than the flexible approach based on Loop's scheme. The control and patch points, as well as the deformation, are shown in Figure 12.

4.5. Pinched cylinder

In the last example of the obstacle course, a short cylinder with rigid end diaphragms, $u_x=u_y=\theta_z=0$, is subjected to compressive normal loads $F=1$ applied along one diameter at the center of the cylinder, see Figure 13(a). The pinched-cylinder problem is a severe test for both inextensional bending modes and complex membrane states. The solution involves highly localized deformations

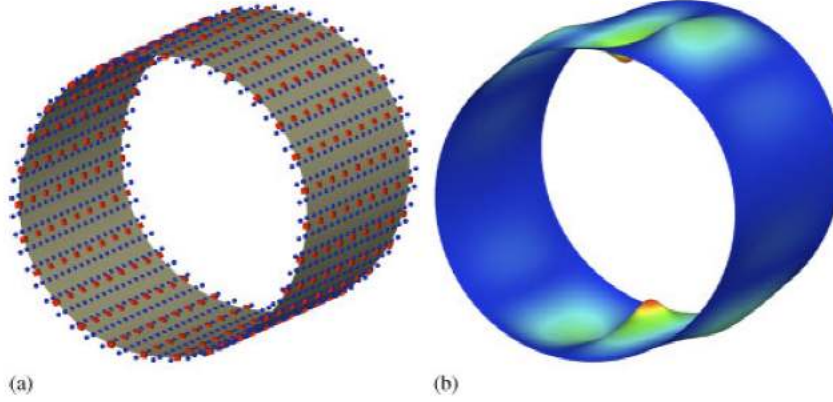


Figure 14. Full model of the pinched cylinder with rigid diaphragms: (a) discretization of both patches \diamond - and nodes \circ -, and reference configuration and (b) computed deformed surface, $\gamma_{LME}=0.8$.

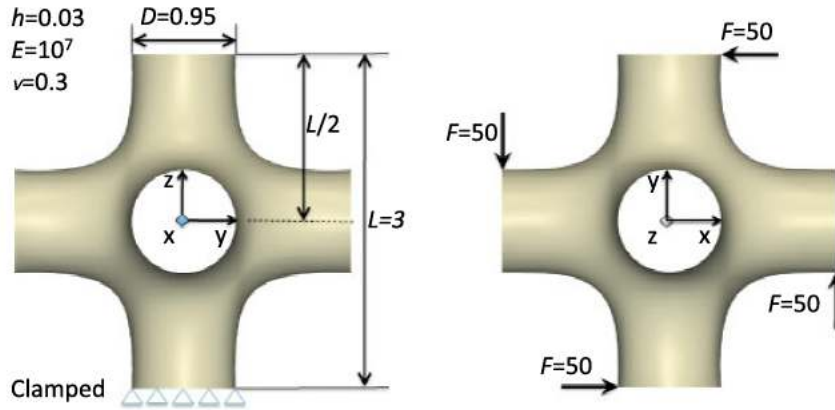


Figure 15. Sketch for the pinched pipes problem. Material and geometrical parameters are detailed. Also, the prescribed boundary conditions and the applied punctual forces are shown.

around the load, which preclude fast convergence. An exact solution for the displacement under the loads follows from a double Fourier series solution of Flugge's equations by Lindberg *et al.* [45], $\delta_r = 1.8248 \times 10^{-5} F$. Figure 13(b) presents the results for the full model case and for an eighth of the cylinder with the appropriate symmetry boundary conditions. In this case, the Catmull-Clark subdivision solutions reported in [42] is particularly accurate, slightly more than our approach for the finest meshes. The computed deformed limit surface is shown in Figure 14(b). The converged solution used as reference is $\delta_r = 1.7973 \times 10^{-5}$.

4.6. Pinched pipes

Here, we study the flexibility of the proposed approach in a shell with complex topology. Figure 15 shows the surface, consisting of a set of six connected open pipes subjected to four concentrated loads. In the figure both the location and the value of the loads are given, as well as the material and geometrical properties. The boundary curve of the bottom pipe is clamped. Figure 16 illustrates the discretization of the patch and the node points, which consist of two arrangements of 1628 and 6716 unstructured set of points, respectively. The original coarse point-set has been obtained from the MATLAB Central File exchange. Figure 17 shows the physical deformation obtained, which has been amplified by a factor of 2 to give a better idea of the resulting smooth field of displacements. We insist on the fact that this thin shell cannot be studied with the previous meshfree approaches because this surface does not admit a single parametric space.

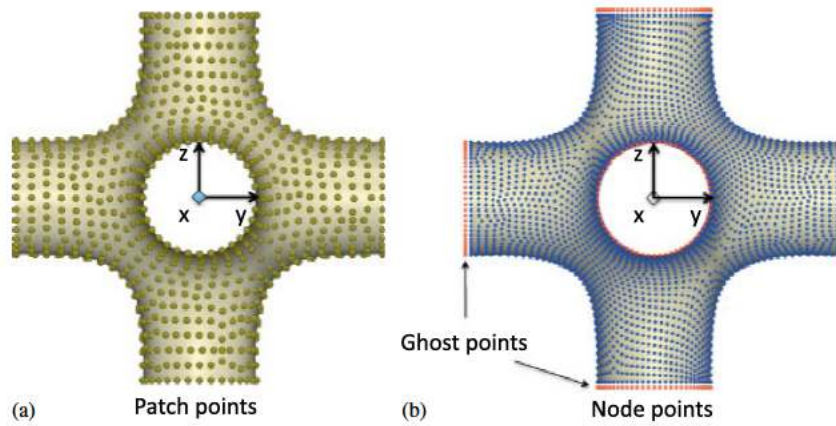


Figure 16. (a) Discretization used by computing the local parametric spaces, 1628 patch points and (b) node set distribution consisting of 6716 points, also the ghost nodes are detailed.

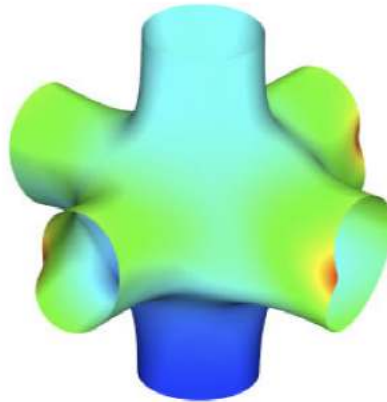


Figure 17. Physical deformation for the pinched pipes problem, $\gamma_{LME}=0.8$.

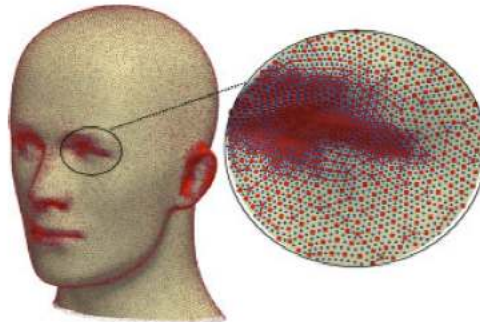


Figure 18. Discretization of the mannequin thin-shell problem, with 24 380 patch points and 97 675 node points.

4.7. Mannequin

This last example illustrates the ability of the method to deal with shells of complex geometry, defined by a set of points alone, without the need for a surface mesh. Of course, the quality of the node and patch sets needs to be sufficiently good, and adapted to complex geometric features, here the ears and the eyes. Figure 18 shows a mannequin elastic thin-shell under (a) a uniform gravity load of magnitude $g=9.0$ and (b) two concentrated forces of value $F=100$ at the crown of the head and at the nose. Material parameters have been selected as $E=10^6$ and $\nu=0.3$, whereas the

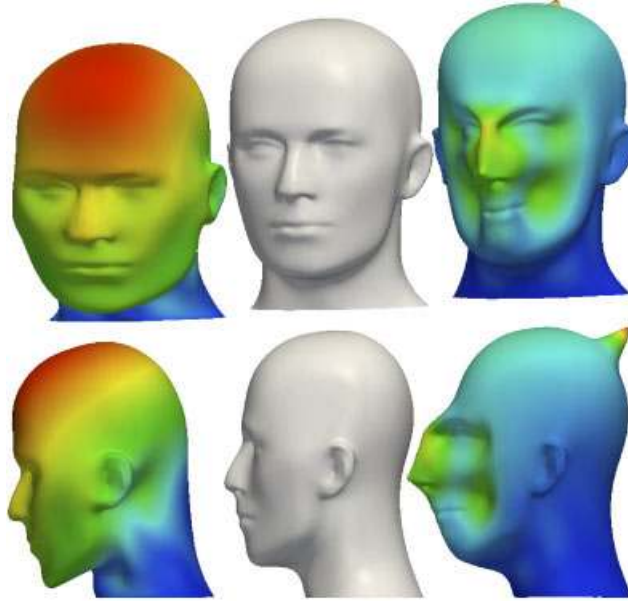


Figure 19. Mannequin thin-shell deformations under the action of a uniform vertical load (left), and by applying two point forces (right).

thickness is $h=0.08$. The boundary curve at the bottom is clamped. Figure 19 shows the smooth deformations obtained, showing strong effect of geometry on the deformation morphology.

5. CONCLUSIONS

We have presented a new methodology for processing d -dimensional point-set manifolds embedded in \mathbb{R}^D , which avoids a global parameterization or mesh. This approach relies on three ingredients: (1) the automatic detection of the local Euclidean structure of the manifold around a set of predefined patch points, i.e. the numerical tangent space to the manifold, (2) the local, smooth parameterization of the manifold around these patches using local *max-ent* approximants, and (3) a PU to split integrals into patch contributions. Each of these steps is general in dimension. We have applied the method to the Kirchhoff–Love thin-shell analysis. The performance of the method, assessed by the classical obstacle course, is excellent. It exhibits better accuracy for a given number of degrees of freedom than DG approaches to thin shells, and better or comparable results than subdivision finite elements. The proposed method significantly extends the applicability of meshfree methods to thin-shell analysis; in that it liberates such methods from the burden of requiring a single parametric space or imposing cumbersome patching conditions between meshfree macro-elements. This feature is illustrated by an example of a shell of complex topology. The method’s applicability depends crucially on the quality of the sampling of the surface, specifically on the density of the sampling relative to the feature size. Such concepts have been formalized in the computer graphics literature [46].

Current research includes combining the proposed method with second-order *max-ent* approximants [26] to increase the order of convergence, improving the accuracy and efficiency of the numerical quadrature with stabilized nodal integration techniques [41]. More importantly, we are developing methods that dramatically reduce the number of patches (i.e. the number of local parameterizations and PU functions) relative to the number of nodes. With the current method, the density of patches is limited by the geometric features of the surface; in that the projections Π_A should not distort too much the node geometry. Finally, once the performance of the method has been assessed numerically, a mathematical analysis of the method shedding light on the rational choice of the numerical parameters would be highly desirable.

APPENDIX A: DERIVATIVES OF THE SHAPE FUNCTIONS

We detail here the calculation of the derivatives of local *max-ent* approximants. We denote spatial gradients of scalar functions by ∇ , whereas for vector-valued functions we denote by $D\mathbf{y}(\boldsymbol{\xi})$ the matrix of partial derivatives. The symbol ∂ denotes partial differentiation. The subindexes a , b , and c refer to nodes. Within the scope of the appendix, we define the following functions

$$f_a(\boldsymbol{\xi}, \boldsymbol{\lambda}) = -\beta_a |\boldsymbol{\xi} - \boldsymbol{\xi}_a|^2 + \boldsymbol{\lambda} \cdot (\boldsymbol{\xi} - \boldsymbol{\xi}_a), \quad (\text{A1})$$

$$p_a(\boldsymbol{\xi}, \boldsymbol{\lambda}) = \frac{\exp[f_a(\boldsymbol{\xi}, \boldsymbol{\lambda})]}{\sum_b \exp[f_b(\boldsymbol{\xi}, \boldsymbol{\lambda})]} = \frac{\exp[f_a(\boldsymbol{\xi}, \boldsymbol{\lambda})]}{Z(\boldsymbol{\xi}, \boldsymbol{\lambda})}, \quad (\text{A2})$$

$$\mathbf{r}(\boldsymbol{\xi}, \boldsymbol{\lambda}) = \sum_a p_a(\boldsymbol{\xi}, \boldsymbol{\lambda}) (\boldsymbol{\xi} - \boldsymbol{\xi}_a), \quad (\text{A3})$$

$$\mathbf{J}(\boldsymbol{\xi}, \boldsymbol{\lambda}) = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\lambda}} = \sum_a p_a(\boldsymbol{\xi}, \boldsymbol{\lambda}) (\boldsymbol{\xi} - \boldsymbol{\xi}_a) \otimes (\boldsymbol{\xi} - \boldsymbol{\xi}_a) - \mathbf{r}(\boldsymbol{\xi}, \boldsymbol{\lambda}) \otimes \mathbf{r}(\boldsymbol{\xi}, \boldsymbol{\lambda}). \quad (\text{A4})$$

The dependence on the evaluation point $\boldsymbol{\xi}$ and on the Lagrange multiplier $\boldsymbol{\lambda}$ is dropped for notational simplicity. The symbol $*$ is used to denote that a function is evaluated in $\boldsymbol{\lambda}^*(\boldsymbol{\xi}) = \arg \min_{\boldsymbol{\lambda} \in \mathbb{R}^d} \ln Z(\boldsymbol{\xi}, \boldsymbol{\lambda})$. This introduces explicit and implicit dependences on $\boldsymbol{\xi}$ in all functions with $*$. Note that what has been denoted by p_a in the remainder of the paper is denoted by p_a^* in the appendix. No implied sum is assumed for repeated node indices.

The first spatial derivative of the shape functions will be referred as ∇p_a^* . It is readily verified [25] that

$$\nabla p_a^* = p_a^* \left(\nabla f_a^* - \sum_c p_c^* \nabla f_c^* \right). \quad (\text{A5})$$

Applying the chain rule, we have

$$\nabla f_a^* = \left(\frac{\partial f_a}{\partial \boldsymbol{\xi}} \right)^* + D\boldsymbol{\lambda}^* \left(\frac{\partial f_a}{\partial \boldsymbol{\lambda}} \right)^*, \quad (\text{A6})$$

where

$$\left(\frac{\partial f_a}{\partial \boldsymbol{\xi}} \right)^* = -2\beta_a (\boldsymbol{\xi} - \boldsymbol{\xi}_a) + \boldsymbol{\lambda}^*, \quad \left(\frac{\partial f_a}{\partial \boldsymbol{\lambda}} \right)^* = (\boldsymbol{\xi} - \boldsymbol{\xi}_a).$$

The only term that is not available explicitly in Equation (A6) is $D\boldsymbol{\lambda}^*$. In order to compute it we note that, since \mathbf{r}^* is identically zero,

$$0 = D\mathbf{r}^* = \left(\frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}} \right)^* + D\boldsymbol{\lambda}^* \left(\frac{\partial \mathbf{r}}{\partial \boldsymbol{\lambda}} \right)^*,$$

where

$$\left(\frac{\partial \mathbf{r}}{\partial \boldsymbol{\lambda}} \right)^* = \mathbf{J}^*, \quad \left(\frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}} \right)^* = -\mathbf{J}_\beta + \mathbf{I}, \quad \mathbf{J}_\beta = 2 \sum_a \beta_a p_a^* (\boldsymbol{\xi} - \boldsymbol{\xi}_a) \otimes (\boldsymbol{\xi} - \boldsymbol{\xi}_a).$$

It follows that

$$D\boldsymbol{\lambda}^* = (\mathbf{J}_\beta - \mathbf{I})(\mathbf{J}^*)^{-1}.$$

Rearranging terms, we finally obtain the spatial gradients of the shape functions as

$$\nabla p_a^* = p_a^* (\mathbf{r}_\beta - \mathbf{M}_a (\boldsymbol{\xi} - \boldsymbol{\xi}_a)),$$

where

$$\mathbf{r}_\beta = 2 \sum_a \beta_a p_a^* (\boldsymbol{\xi} - \boldsymbol{\xi}_a), \quad \mathbf{M}_a = 2\beta_a \mathbf{I} - D\boldsymbol{\lambda}^*.$$

The second spatial derivative of the shape functions, i.e. their Hessian, will be referred as $(Hp_a)^*$. We calculate the derivative of ∇p_a as

$$(Hp_a)^* = \underbrace{\nabla p_a^* \otimes \left(\nabla f_a^* - \sum_b p_b^* \nabla f_b^* \right)}_A + \underbrace{p_a^* \left((D\nabla f_a)^* - \sum_b p_b^* (D\nabla f_b)^* \right)}_B - \underbrace{p_a^* \sum_b \nabla p_b^* \otimes \nabla f_b^*}_C,$$

where

$$\begin{aligned} A &= p_a^* [\mathbf{r}_\beta - \mathbf{M}_a(\boldsymbol{\xi} - \boldsymbol{\xi}_a)] \otimes [\mathbf{r}_\beta - \mathbf{M}_a(\boldsymbol{\xi} - \boldsymbol{\xi}_a)], \\ B &= 2p_a^* \left(\sum_b \beta_b p_b^* - \beta_a \right) \mathbf{I} + p_a^* D^2 \boldsymbol{\lambda}^*(\boldsymbol{\xi} - \boldsymbol{\xi}_a), \\ C &= p_a^* \mathbf{r}_\beta \otimes \mathbf{r}_\beta - p_a^* \sum_b p_b^* \mathbf{M}_b(\boldsymbol{\xi} - \boldsymbol{\xi}_b) \otimes \mathbf{M}_b(\boldsymbol{\xi} - \boldsymbol{\xi}_b). \end{aligned}$$

The term $D^2 \boldsymbol{\lambda}^*$ is computed by using again the fact that \mathbf{r}^* is identically zero, which also implies that $D^2 \mathbf{r}^* = 0$. Lengthy but simple calculations lead to

$$D^2 \boldsymbol{\lambda}^*(\boldsymbol{\xi} - \boldsymbol{\xi}_a) = \mathbf{r}_\beta \otimes \mathbf{j}_a + \mathbf{j}_a \otimes \mathbf{r}_\beta + (\mathbf{r}_\beta \cdot \mathbf{j}_a) \mathbf{I} - \sum_b p_b^* \Delta_{ab} \mathbf{M}_b(\boldsymbol{\xi} - \boldsymbol{\xi}_b) \otimes \mathbf{M}_b(\boldsymbol{\xi} - \boldsymbol{\xi}_b),$$

where

$$\Delta_{ab} = (\boldsymbol{\xi} - \boldsymbol{\xi}_b) \cdot (\mathbf{J}^*)^{-1} (\boldsymbol{\xi} - \boldsymbol{\xi}_a), \quad \mathbf{j}_a = (\mathbf{J}^*)^{-1} (\boldsymbol{\xi} - \boldsymbol{\xi}_a).$$

Finally, the second spatial derivative of the shape functions can be written as

$$\begin{aligned} (Hp_a)^* &= p_a^* [\mathbf{r}_\beta - \mathbf{M}_a(\boldsymbol{\xi} - \boldsymbol{\xi}_a)] \otimes [\mathbf{r}_\beta - \mathbf{M}_a(\boldsymbol{\xi} - \boldsymbol{\xi}_a)] + 2p_a^* \left(\sum_b \beta_b p_b^* - \beta_a \right) \mathbf{I} \\ &\quad + p_a^* [\mathbf{r}_\beta \otimes \mathbf{r}_\beta + \mathbf{r}_\beta \otimes \mathbf{j}_a + \mathbf{j}_a \otimes \mathbf{r}_\beta + (\mathbf{r}_\beta \cdot \mathbf{j}_a) \mathbf{I}] \\ &\quad - p_a^* \sum_b p_b^* (1 + \Delta_{ab}) \mathbf{M}_b(\boldsymbol{\xi} - \boldsymbol{\xi}_b) \otimes \mathbf{M}_b(\boldsymbol{\xi} - \boldsymbol{\xi}_b). \end{aligned} \tag{A7}$$

APPENDIX B: STRAIN TENSORS FOR LINEARIZED KINEMATICS

In this appendix, we derive the bending strain tensor components for linearized kinematics. From Section 3, with the Kirchhoff–Love hypothesis and defining the displacement vector $\mathbf{u} = \boldsymbol{\varphi} - \boldsymbol{\varphi}_0$, the bending strains in Equation (12) become

$$\rho_{\alpha\beta} = \boldsymbol{\varphi}_{0,\alpha\beta} \cdot \mathbf{t}_0 - (\boldsymbol{\varphi}_{0,\alpha\beta} + \mathbf{u}_{,\alpha\beta}) \cdot \mathbf{t}, \tag{B1}$$

where

$$\mathbf{t} = \bar{j}^{-1} (\boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,2} + \mathbf{u}_{,1} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \mathbf{u}_{,2} + \mathbf{u}_{,1} \times \mathbf{u}_{,2}).$$

Neglecting higher-order terms in \mathbf{u} , \bar{j}^{-1} expressed in the reference configuration takes the form

$$\bar{j}^{-1} \approx \bar{j}_0^{-1} - \bar{j}_0^{-2} \mathbf{t}_0 \cdot (\mathbf{u}_{,1} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \mathbf{u}_{,2}),$$

which allows us to calculate the normal director increment $\Delta \mathbf{t} = \mathbf{t} - \mathbf{t}_0$ as

$$\Delta \mathbf{t} \approx \bar{j}_0^{-1} (\mathbf{u}_{,1} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \mathbf{u}_{,2}) - \bar{j}_0^{-1} [\mathbf{t}_0 \cdot (\mathbf{u}_{,1} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \mathbf{u}_{,2})] \mathbf{t}_0.$$

This expression can be written more compactly after introducing $\mathbf{v} = \mathbf{u}_{,1} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \mathbf{u}_{,2}$ and the identity $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$:

$$\Delta \mathbf{t} = \bar{j}_0^{-1} [\mathbf{v}(\mathbf{t}_0 \cdot \mathbf{t}_0) - (\mathbf{t}_0 \cdot \mathbf{v}) \mathbf{t}_0] = \bar{j}_0^{-1} [\mathbf{t}_0 \times (\mathbf{v} \times \mathbf{t}_0)].$$

Replacing \mathbf{t} by $\mathbf{t}_0 + \Delta\mathbf{t}$ in Equation (B1), rearranging terms, applying the identities $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a})$ and $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$, and neglecting higher-order terms in \mathbf{u} , the bending strains can be expressed as

$$\begin{aligned} \rho_{\alpha\beta} = & -\mathbf{t}_0 \cdot \mathbf{u}_{,\alpha\beta} + \bar{j}_0^{-1} [(\boldsymbol{\varphi}_{0,\alpha\beta} \times \boldsymbol{\varphi}_{0,2}) \cdot \mathbf{u}_{,1} + (\boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,\alpha\beta}) \cdot \mathbf{u}_{,2}] \\ & + \bar{j}_0^{-1} (\mathbf{t}_0 \cdot \boldsymbol{\varphi}_{0,\alpha\beta}) [(\boldsymbol{\varphi}_{0,2} \times \mathbf{t}_0) \cdot \mathbf{u}_{,1} + (\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,1}) \cdot \mathbf{u}_{,2}]. \end{aligned} \quad (\text{B2})$$

The derivatives of the normal vector \mathbf{t}_0 are computed as

$$\mathbf{t}_{0,\alpha} = \bar{j}_0^{-1} (\boldsymbol{\varphi}_{,1\alpha} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{,2\alpha}) - \bar{j}_0^{-1} [\mathbf{t}_0 \cdot (\boldsymbol{\varphi}_{,1\alpha} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{,2\alpha})] \mathbf{t}_0,$$

which can be compactly re-written by applying the previous procedure for $\Delta\mathbf{t}$:

$$\mathbf{t}_{0,\alpha} = -\bar{j}_0^{-1} \{ \mathbf{t}_0 \times [\mathbf{t}_0 \times (\boldsymbol{\varphi}_{0,1\alpha} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,2\alpha})] \}.$$

APPENDIX C: MEMBRANE AND BENDING STRAIN-DISPLACEMENT MATRICES

In this appendix we describe the membrane and bending strain-displacement matrices, which are needed for the computation of the stiffness matrix \mathbf{K} (Section 3). We consider a specific patch A . Let $\boldsymbol{\varphi}_{0A}$, $\boldsymbol{\varphi}_0$ to keep the notation light, be a configuration mapping for the middle surface for this patch, defined over the parametric space \mathcal{A}_A

$$\boldsymbol{\varphi}_0(\boldsymbol{\xi}) = \sum_{a \in \hat{\mathcal{N}}_{\mathcal{Q}_A}^P} p_a(\boldsymbol{\xi}) \mathbf{P}_a,$$

with derivatives

$$\boldsymbol{\varphi}_{0,\alpha}(\boldsymbol{\xi}) = \sum_{a \in \hat{\mathcal{N}}_{\mathcal{Q}_A}^P} p_{a,\alpha}(\boldsymbol{\xi}) \mathbf{P}_a, \quad \boldsymbol{\varphi}_{0,\alpha\beta}(\boldsymbol{\xi}) = \sum_{a \in \hat{\mathcal{N}}_{\mathcal{Q}_A}^P} p_{a,\alpha\beta}(\boldsymbol{\xi}) \mathbf{P}_a.$$

The membrane and bending strain-displacement matrices for the a th nodal point, \mathbf{M}^a and \mathbf{B}^a respectively, take the form

$$\mathbf{M}_{ij}^a = \mathbf{M}_i^a \cdot \mathbf{e}_j \quad \text{and} \quad \mathbf{B}_{ij}^a = \mathbf{B}_i^a \cdot \mathbf{e}_j,$$

where

$$\begin{aligned} \mathbf{M}_\alpha^a &= p_{a,\alpha} \boldsymbol{\varphi}_{0,\alpha}, \\ \mathbf{M}_3^a &= p_{a,2} \boldsymbol{\varphi}_{0,1} + p_{a,1} \boldsymbol{\varphi}_{0,2}, \\ \mathbf{B}_\alpha^a &= -p_{a,\alpha\alpha} \mathbf{t}_0 + \bar{j}_0^{-1} [(\boldsymbol{\varphi}_{0,\alpha\alpha} \times \boldsymbol{\varphi}_{0,2}) p_{a,1} + (\boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,\alpha\alpha}) p_{a,2}] \\ &\quad + \bar{j}_0^{-1} (\mathbf{t}_0 \cdot \boldsymbol{\varphi}_{0,\alpha\alpha}) [(\boldsymbol{\varphi}_{0,2} \times \mathbf{t}_0) p_{a,1} + (\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,1}) p_{a,2}], \\ \mathbf{B}_3^a &= -2p_{a,12} \mathbf{t}_0 + 2\bar{j}_0^{-1} [(\boldsymbol{\varphi}_{0,12} \times \boldsymbol{\varphi}_{0,2}) p_{a,1} + (\boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,12})] p_{a,2} \\ &\quad + 2\bar{j}_0^{-1} (\mathbf{t}_0 \cdot \boldsymbol{\varphi}_{0,12}) [(\boldsymbol{\varphi}_{0,2} \times \mathbf{t}_0) p_{a,1} + (\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,1}) p_{a,2}], \end{aligned}$$

and \mathbf{e}_j represent the canonical basis vectors of \mathbb{R}^3 . Note that repeated indices in the expressions for \mathbf{M}_α^a and \mathbf{B}_α^a do not imply summation.

APPENDIX D: ESSENTIAL BOUNDARY CONDITIONS

We describe here the imposition of the essential boundary conditions, both for the displacements and the rotations. We describe the variational formulation with the Lagrange multipliers (a slight variation of [11]), as well as the discretization.

Let us consider the integral of a function f over the boundary surface $\partial\mathcal{S}_0$ of a thin shell object \mathcal{S}_0 . In fact, $\partial\mathcal{S}_0$ is the only thin part of the boundary of \mathcal{S}_0 , i.e. it excludes the surfaces parallel to the middle surface. Assuming that the function does not change through-the-thickness, we have

$$\int_{\partial\mathcal{S}_0} f \, dS_0 = \int_{\partial\Omega_0} f \left(\int_{-h/2}^{h/2} \frac{\left\| \frac{\partial \mathbf{x}_0}{\partial \xi} \times \frac{\partial \mathbf{x}_0}{\partial t} \right\|}{\|\boldsymbol{\varphi}_{0,t}\|} d\xi \right) d\ell_0,$$

where t is a tangent coordinate along the boundary curve $\partial\mathcal{A}$. By introducing $\partial \mathbf{x}_0 / \partial t = \boldsymbol{\varphi}_{0,t} + \xi \mathbf{t}_{0,t}$, we obtain

$$\int_{-h/2}^{h/2} \frac{\left\| \mathbf{t}_0 \times \frac{\partial \mathbf{x}_0}{\partial t} \right\|}{\|\boldsymbol{\varphi}_{0,t}\|} d\xi = h \frac{\|\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,t}\|}{\|\boldsymbol{\varphi}_{0,t}\|}.$$

With the previous expressions and the PU, the integral of a function f on the boundary surface $\partial\mathcal{S}_0$ becomes

$$\begin{aligned} \int_{\partial\mathcal{S}_0} f \, dS_0 &= \int_{\partial\Omega_0} h f \frac{\|\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,t}\|}{\|\boldsymbol{\varphi}_{0,t}\|} d\ell_0 \\ &= \sum_{A=1}^M \int_{\partial\mathcal{A}_A} [h(f \circ \boldsymbol{\varphi}_0) \|\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,t}\|]_A (w_A^Q \circ \boldsymbol{\varphi}_0) d\ell_\xi. \end{aligned}$$

Here subindex A means that the expression between the brackets is computed with the local parameterization of the A th patch.

D.1. Displacement constraints

Let λ_u be the Lagrange multipliers associated with the displacement constraints $\mathbf{u} = \bar{\mathbf{u}}$ on $\partial\Omega_0^u$; the additional terms for Equation (14) are

$$- \int_{\partial\Omega_0^u} h [(\mathbf{u} - \bar{\mathbf{u}}) \cdot \delta \lambda_u + \lambda_u \cdot \delta \mathbf{u}] \frac{\|\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,t}\|}{\|\boldsymbol{\varphi}_{0,t}\|} d\ell_0.$$

With the PU, we obtain

$$- \sum_{A=1}^M \int_{\partial\Omega_{0A}^u} h [(\mathbf{u} - \bar{\mathbf{u}}) \cdot \delta \lambda_u + \lambda_u \cdot \delta \mathbf{u}] \frac{\|\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,t}\|}{\|\boldsymbol{\varphi}_{0,t}\|} w_A^Q d\ell_0,$$

where $\partial\Omega_{0A}^u = \partial\Omega_0^u \cap \text{supp}(w_A^Q)$. Then, the final expression in the parametric space for the displacement constraints is

$$- \sum_{A=1}^M \int_{\partial\mathcal{A}_A^u} \{h [(\mathbf{u} - \bar{\mathbf{u}}) \cdot \delta \lambda_u + \lambda_u \cdot \delta \mathbf{u}] \|\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,t}\|\}_A (w_A^Q \circ \boldsymbol{\varphi}_0) d\ell_\xi.$$

D.2. Rotation constraints

We express a unit vector tangent to the boundary curve of the middle surface Ω_0 as $\boldsymbol{\tau}_0 = \boldsymbol{\varphi}_{0,t} / \|\boldsymbol{\varphi}_{0,t}\|$, which satisfies $\mathbf{t}_0 \cdot \boldsymbol{\tau}_0 = 0$. The small rotation vector $\boldsymbol{\theta}$ measures the infinitesimal rotation of the director field, i.e. it is given by the relation [11]

$$\boldsymbol{\theta} \times \mathbf{t}_0 \approx \Delta \mathbf{t} = \mathbf{t} - \mathbf{t}_0,$$

where the approximate sign means here up to first order in the displacements. Recalling the linearized expression for Δt of Appendix B

$$\Delta t \approx -\bar{j}_0^{-1} \{[(\mathbf{u}_{,1} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \mathbf{u}_{,2}) \times \mathbf{t}_0] \times \mathbf{t}_0\},$$

and rearranging the term between brackets as

$$(\mathbf{u}_{,1} \times \boldsymbol{\varphi}_{0,2} + \boldsymbol{\varphi}_{0,1} \times \mathbf{u}_{,2}) \times \mathbf{t}_0 = (\mathbf{u}_{,1} \cdot \mathbf{t}_0) \boldsymbol{\varphi}_{0,2} - (\mathbf{u}_{,2} \cdot \mathbf{t}_0) \boldsymbol{\varphi}_{0,1},$$

we finally obtain

$$\theta = \bar{j}_0^{-1} [(\mathbf{u}_{,2} \cdot \mathbf{t}_0) \boldsymbol{\varphi}_{0,1} - (\mathbf{u}_{,1} \cdot \mathbf{t}_0) \boldsymbol{\varphi}_{0,2}]. \quad (\text{D1})$$

The rotation of the shell boundary about the tangential direction is computed as $\theta_\tau = \theta \cdot \boldsymbol{\tau}_0$. The rotation boundary conditions take the form $\theta_\tau = \bar{\theta}_\tau$ on $\partial\Omega_0^\theta$. Denoting with λ_θ the Lagrange multiplier associated with this constraint, the extra terms for the weak form of Equation (14) are

$$- \int_{\partial\Omega_0^\theta} h [(\theta_\tau - \bar{\theta}_\tau) \delta \lambda_\theta + \lambda_\theta \delta \theta_\tau] d\ell_0.$$

With the PU, these terms become

$$- \sum_{A=1}^M \int_{\partial\mathcal{A}_A^\theta} \{h [(\theta_\tau - \bar{\theta}_\tau) \delta \lambda_\theta + \lambda_\theta \delta \theta_\tau] \|\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,t}\| \}_A (w_A^Q \circ \boldsymbol{\varphi}_0) d\ell_\xi.$$

D.3. Numerical imposition of the constraints

The discretization of the displacement and rotational Lagrange multipliers is performed as follows:

$$\lambda_u = \sum_a N_a \Lambda_{ua}, \quad \lambda_\tau = \sum_a N_a \Lambda_{\theta a},$$

where N_a are the basis functions that define the space of Lagrange multipliers. Here, we use the piecewise linear hat shape functions associated with the boundary nodes. The rotational constraints can be written in terms of the displacement nodal values as

$$\theta_\tau = \bar{j}_0^{-1} \sum_{b=1}^N [p_{b,2}(\boldsymbol{\varphi}_{0,1} \cdot \boldsymbol{\tau}_0) - p_{b,1}(\boldsymbol{\varphi}_{0,2} \cdot \boldsymbol{\tau}_0)] (\mathbf{u}_b \cdot \mathbf{t}_0).$$

The solution is computed by solving the following linear system of equations:

$$\begin{pmatrix} \mathbf{K} & \mathbf{C}_u^\top & \mathbf{C}_\theta^\top \\ \mathbf{C}_u & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_\theta & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \Lambda_u \\ \Lambda_\theta \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g}_u \\ \mathbf{g}_\theta \end{pmatrix},$$

For concreteness, only the components of the matrix of rotational constraints \mathbf{C}_θ and the components of the vector \mathbf{g}_θ are given as

$$\mathbf{C}_\theta^{ab} = - \sum_{A=1}^M \int_{\partial\mathcal{A}_A^\theta} [h \bar{j}_0^{-1} N_a [p_{b,2}(\boldsymbol{\varphi}_{0,1} \cdot \boldsymbol{\tau}_0) - p_{b,1}(\boldsymbol{\varphi}_{0,2} \cdot \boldsymbol{\tau}_0)] \|\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,t}\|]_A (w_A^Q \circ \boldsymbol{\varphi}_A) d\ell_\xi,$$

$$\mathbf{g}_\theta^a = - \sum_{A=1}^M \int_{\partial\mathcal{A}_A^\theta} [h N_a \bar{\theta}_\tau \|\mathbf{t}_0 \times \boldsymbol{\varphi}_{0,t}\|]_A (w_A^Q \circ \boldsymbol{\varphi}_A) d\ell_\xi.$$

Note that $\mathbf{C}_\theta^{ab} \in \mathbb{R}^{1 \times 3}$. The tangent boundary vector, defined as $\boldsymbol{\tau}_0 = \boldsymbol{\varphi}_{0,t} / \|\boldsymbol{\varphi}_{0,t}\|$, is computed from the expression

$$\boldsymbol{\varphi}_{0,t} = \sum_a \nabla p_a \cdot \boldsymbol{\tau}_\xi \mathbf{P}_a,$$

where $\boldsymbol{\tau}_\xi$ is a unit vector tangent to $\partial\mathcal{A}^\rho$.

ACKNOWLEDGEMENTS

We acknowledge the support of the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement nr. 240487 and the Ministerio de Ciencia e Innovación (DPI2007-61054). M. A. acknowledges the support received through the prize 'ICREA Academia' for excellence in research, funded by the Generalitat de Catalunya.

REFERENCES

1. Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva C. Point set surfaces. *VIS '01: Proceedings of the Conference on Visualization '01*. IEEE Computer Society: Washington, DC, U.S.A., 2001; 21–28.
2. Pauly M. Point primitives for interactive modeling and processing of 3d geometry. *Ph.D. Thesis*, Federal Institute of Technology (ETH) of Zurich, 2003.
3. Levin D. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, Brunnett H, Mueller (eds). Springer: Berlin, 2003; 37–49.
4. Levoy M, Whitted T. The use of points as displays primitives. *Technical Report TR-85-022*, Computer Science Department, University of North Carolina at Chappel Hill, 1985.
5. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. *SIGGRAPH '92: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*. ACM: New York, NY, U.S.A., 1992; 71–78.
6. Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H. Multi-level partition of unity implicits. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)* 2003; **22**:463–470.
7. Levin D. The approximation power of moving least-squares. *Mathematics of Computation* 1998; **67**(224): 1517–1531.
8. Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva C. Computing rendering point set surfaces. *Transactions on Visualization and Computer Graphics* 2003; **9**(1):3–15.
9. Amenta N, Kil Y. Defining point-set surfaces. *ACM Transactions on Graphics* 2004; **23**(3):264–270.
10. Alexa M, Gross M, Pauly M, Pfister H, Stamminger M, Zwicker M. Point-based computer graphics. *SIGGRAPH 2004 Course Notes*, 2004.
11. Krysl P, Belytschko T. Analysis of thin shells by the element-free Galerkin method. *International Journal of Solids and Structures* 1996; **33**(20–22):3057–3078.
12. Noguchi H, Kawashima T, Miyamura T. Element free analyses of shell and spatial structures. *International Journal for Numerical Methods in Engineering* 2000; **47**(6):1215–1240.
13. Chen J, Wang D. A constrained reproducing kernel particle formulation for shear deformable shell in Cartesian coordinates. *International Journal for Numerical Methods in Engineering* 2006; **68**:151–172.
14. Rabczuk T, Areias P, Belytschko T. A meshfree thin shell method for non-linear dynamic fracture. *International Journal for Numerical Methods in Engineering* 2007; **72**:525–548.
15. MacNeal R, Harder R. A proposed standard set of problems to test finite element accuracy. *Finite Element in Analysis and Design* 1985; **1**(1):3–20.
16. Bucalem M, Bathe J. Higher-order Mitc general shell elements. *International Journal for Numerical Methods in Engineering* 1993; **36**:3729–3754.
17. Simo J, Fox D. On a stress resultant geometrically exact shell model. Part I: formulation and optimal parametrization. *Computer Methods in Applied Mechanics and Engineering* 1989; **72**:267–304.
18. Cirak F, Ortiz M, Schröder P. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering* 2000; **47**:2039–2072.
19. Simo J, Fox D, Rifai M. On a stress resultant geometrically exact shell model. Part II: the linear theory; computational aspects. *Computer Methods in Applied Mechanics and Engineering* 1989; **73**:53–92.
20. Yang HTY, Saigal S, Masud A, Kapania RK. A survey of recent shell finite elements. *International Journal for Numerical Methods in Engineering* 2000; **47**(1–3):101–127.
21. Cirak F, Ortiz M. Fully C^1 -conforming subdivision elements for finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering* 2001; **51**(7):813–833.
22. Engel G, Garikipati K, Hughes T, Larson M, Mazzei L, Taylor R. Continuous/discontinuous finite element approximations of fourth-order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:3669–3750.
23. Wells G, Dung N. A C^0 discontinuous Galerkin formulation for Kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(35–36):3370–3380.
24. Noels L, Radovitzky R. A new discontinuous Galerkin method for Kirchhoff–Love shells. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**(33–40):2901–2929.
25. Arroyo M, Ortiz M. Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering* 2006; **65**:2167–2202.
26. Cyron C, Arroyo M, Ortiz M. Smooth, second order, non-negative meshfree approximants selected by maximum entropy. *International Journal for Numerical Methods in Engineering* 2009; **79**(13):1605–1632.
27. Belytschko T, Stolarski H, Liu W, Carpenter N, Ong J. Stress projection for membrane and shear locking in shell finite-elements. *Computer Methods in Applied Mechanics and Engineering* 1985; **51**:221–258.

28. Jain AK, Duin R, Mao J. Statistical pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2000; **22**(1):4–37.
29. Zhang Z, Zha H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing* 2005; **26**(1):313–338.
30. Lall S, Krysl P, Marsden J. Structure-preserving model reduction for mechanical systems. *Physica D* 2003; **184**:304–318.
31. Niroomandi S, Alfaro I, Cueto E, Chinesta F. Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering* 2009; DOI: 10.1002/nme.2733.
32. do Carmo MP. *Differential Geometry of Curves and Surfaces*. Prentice-Hall: Englewood Cliffs, NJ, 1976.
33. Sukumar N. Construction of polygonal interpolants: a maximum entropy approach. *International Journal for Numerical Methods in Engineering* 2004; **61**(12):2159–2181.
34. Hughes T, Cottrell J, Bazilevs Y. Isogeometric analysis: cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:4135–4195.
35. González D, Cueto E, Doblaré M. A higher order method based on local maximum entropy approximation. *International Journal for Numerical Methods in Engineering* 2010; DOI: 10.1002/nme.2855.
36. Rosolen A, Millán D, Arroyo M. On the optimum support size in meshfree methods: a variational adaptivity approach with maximum entropy approximants. *International Journal for Numerical Methods in Engineering* 2009; DOI: 10.1002/nme.2793.
37. Fernández-Méndez S, Huerta A. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:1257–1275.
38. Gupta MR. An information theory approach to supervised learning. *Ph.D. Thesis*, Stanford, 2003.
39. Stroud AH. *Approximate Calculation of Multiple Integrals*. Prentice-Hall: Englewood Cliffs, NJ, 1971.
40. Ciarlet PG. *Mathematical Elasticity, Volume III: Theory of Shells*. North-Holland: Amsterdam, 2000.
41. Wang D, Chen J. A hermite reproducing kernel approximation for thin plate analysis with sub-domain stabilized conforming integration. *International Journal for Numerical Methods in Engineering* 2008; **74**:368–390.
42. Green S, Turkiyyah G. Second-order accurate constraint formulation for subdivision finite element simulation of thin shells. *International Journal for Numerical Methods in Engineering* 2004; **61**(3):380–405.
43. Ohtake Y, Belyaev A, Seidel HP. Sparse surface reconstruction with adaptive partition of unity and radial basis functions. *Graphical Models* 2006; **68**(1):15–24.
44. Timoshenko S, Woinowsky-Kreiger S. *Theory of Plates and Shells*. McGraw-Hill: New York, 1959.
45. Lindberg G, Olson M, Cowper G. New developments in the finite element analysis of shells. *Quarterly Bulletin, Division of Mechanical Engineering, National Aeronautical Establishment*, vol. 4, 1969; 1–38.
46. Amenta N, Choi S, Kolluri R. The power crust. *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications* 2001; 249–260.