Nonlinear manifold learning for meshfree finite deformation thin-shell analysis

Daniel Millán, Adrian Rosolen and Marino 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

Calculations on general point-set surfaces are attractive because of their flexibility and simplicity in the preprocessing but present important challenges. The absence of a mesh makes it nontrivial to decide if two neighboring points in the three-dimensional embedding are nearby or rather far apart on the manifold. Furthermore, the topology of surfaces is generally not that of an open two-dimensional set, ruling out global parametrizations. We propose a general and simple numerical method analogous to the mathematical theory of manifolds, in which the point-set surface is described by a set of overlapping charts forming a complete atlas. We proceed in four steps: (1) partitioning of the node set into subregions of trivial topology; (2) automatic detection of the geometric structure of the surface patches by nonlinear dimensionality reduction methods; (3) parametrization of the surface using smooth meshfree (here *maximum-entropy*) approximants; and (4) gluing together the patch representations by means of a partition of unity. Each patch may be viewed as a meshfree macro-element. We exemplify the generality, flexibility, and accuracy of the proposed approach by numerically approximating the geometrically nonlinear Kirchhoff–Love theory of thin-shells. We analyze standard benchmark tests as well as point-set surfaces of complex geometry and topology.

1. INTRODUCTION

The Kirchhoff–Love theory of thin-shells is very attractive as compared with shear deformable shells because only the middle surface (three degrees of freedom per node) needs to be described. It has been shown to be a very good model for slender surface-like bodies. However, this theory involves both the first and second fundamental forms of the surface, and consequently, the approximation of the deformation needs to have second-order square integrable derivatives. For general unstructured meshes, it is difficult to define C^1 finite element approximations, which has prompted a myriad of techniques that avoid this requirement ([1–3] and references there in). Yet, C^0 approaches do not perform well in the thin-shell limit.

Early meshfree approaches are among the first numerical methods with smooth approximants for Kirchhoff–Love shells beyond Hermite approximations [4]. Following ideas from computer graphics, smooth subdivision surfaces finite elements have been proposed for thin-shells [5, 6]. Subdivision finite elements provide a unified framework for geometric modeling and thin-shell analysis. Along this line of work, isogeometric analysis [7, 8] is a new technology building on computer-aided design smooth approximation methods, such as B-splines and non-uniform rational

---

B-splines (NURBS). They have been successfully applied to beams, plates, and thin-shells [8–10]. These methods exhibit a high fidelity of the geometry representation, and their difficulties are derived from the rigidity of the NURBS framework with regards to the structure of the grid, which are the topic of current research [11]. On the other hand, discontinuous Galerkin formulations have been proposed recently for plates, beams, and thin-shells [12–16]. 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.

Despite the advances made in the area of computer graphics to process point-set surfaces, that is, surfaces discretized merely as a set of points in space [17–24], meshfree methods for thin-shell analysis are still restricted to simple surfaces admitting a single parametrization [4, 25–27]. In a recent work, we presented a methodology based on linear statistical learning techniques to process a general smooth surface defined by a set of points alone [28]. The performance of the method was assessed by confronting a classical obstacle course of linear benchmarks proposed in [29]. The method is in principle applicable to embedded manifolds in any dimension. In our previous work, the method results from combining three ingredients:

1. The local geometric structure of the manifold is detected from the node set using weighted PCA (wPCA), which identifies the hyperplane closest to the points in a given neighborhood that we call patch. The number of patches is in general comparable with the number of points.
2. A smooth local parametrization is defined in the two-dimensional wPCA projection of the points of each patch. This can be realized with a variety of methods, from other meshfree methods such as moving least squares (MLS) approximants to mesh-based methods such as subdivision finite elements. In the latter case, no global mesh is required. Here, the local *maximum-entropy* (*max-ent*) approximants [30] are chosen because of their smoothness, robustness, and relative ease of quadrature compared with other meshfree approximants.
3. The local parametrizations are then glued together with a partition of unity (PU) defined in the ambient space, which consequently is also a PU on the manifold.

In spite of being able to deal with shells of complex topology and geometry, this method presents a serious practical drawback: wPCA produces a massive overlapping between the patches. The patches cannot be too large; otherwise, the local hyperplane approximates poorly the manifold, and the local parametrization may become very distorted or even not injective. The massive overlapping, needed to glue the local patches with the PU, results in great redundancy of quadrature points and a very large computational cost.

In the present work, our goal is to greatly expand the range of applicability of the ideas presented in [28], building more efficient and robust local parametrizations of the point-set surface. We exploit nonlinear dimensionality reduction (NLDR) techniques, instead of the linear PCA. By doing so, the point-set surface can be partitioned into a small number of regions, which exhibit a small overlap. The calculations are significantly faster and more accurate. We illustrate with a numerical example the amount of overlapping with both methods. We consider a scattered set of points distributed on a sphere and compute the ratio between the number of quadrature point needed with a method based on wPCA or NLDR and the number of quadrature points if no overlap at all was needed (Figure 1). It can be observed that the redundancy factor for the method based on wPCA can be as large as 15, whereas with the proposed method, this factor approaches the optimal value of 1 as the discretization is refined.

In Section 2, a concise introduction to dimensionality reduction in the context of data-driven methods is presented. Additionally, two additional steps are proposed to obtain high-quality low-dimensional embeddings, which are motivated with a simple example. Section 3 describes the proposed methodology for point-set manifold processing. Section 4 provides a short account of the geometrically exact Kirchhoff–Love shell theory. Numerical experiments to evaluate the performance of the method are presented in Section 5. Some remarks and conclusions are collected in Section 6.
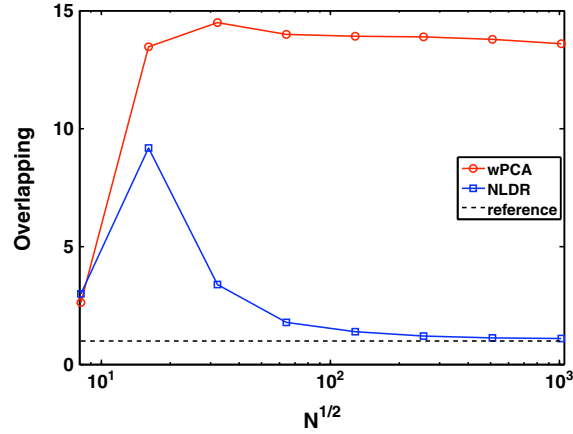
Figure 1. Overlapping ratio comparison between a linear (wPCA) and a nonlinear dimensionality reduction (NLDR) method for a spherical point-set surface. The overlapping measures the amount of redundancy of quadrature points caused by the overlapping of the local parametrizations.

2. DIMENSIONALITY REDUCTION

The goal of statistical learning is extracting meaningful information from empirical data, a general endeavor with many applications in science and engineering. In particular, dimensionality reduction tries to address the curse that dimensionality researchers confront when dealing with high D-dimensional data. It proceeds by finding a lower d-dimensional representation, $d \ll D$, which captures the most relevant features of the data, that is, these methods identify the hidden variables that best explain the behavior of a given system. Generically, given an input data set $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^D$, the problem is finding a lower dimensional representation $\Xi = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_N\} \subset \mathbb{R}^d$ with $d < D$, which in some sense retains the essential information in the original data.

2.1. Linear dimensionality reduction

The most widespread technique of dimensionality reduction was introduced in 1901 by Pearson [31]. In this seminal work, Pearson developed a method to obtain the best affine fit that minimizes the distance from the input data (Figure 2). Throughout the years, this method has been rediscovered and extended in many areas, where it goes by different names: principal components or Hotelling transform [32], Karhunen–Loève transform [33, 34], empirical orthogonal functions [35], and proper orthogonal decomposition [36]. Here, we will refer to it as PCA. See [37] for a modern reference to this subject.

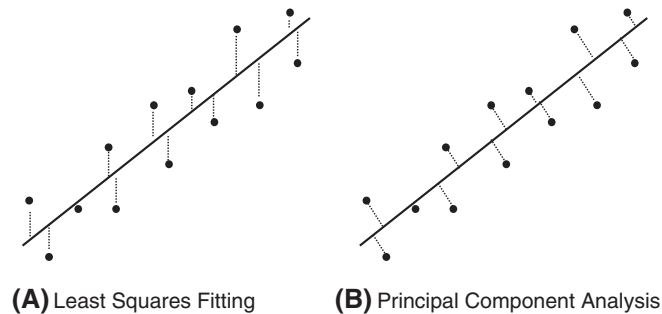**(A)** Least Squares Fitting   **(B)** Principal Component Analysis

Figure 2. (A) Least squares and (B) PCA fits to a set of scattered points in two dimensions. The least-squares fit depends on the choice of axis and requires solving a linear system of equations, whereas the PCA fit is geometrically objective and requires finding eigenvalues and eigenvectors.

Principal component analysis is a standard tool in computer graphics [19], data analysis [38], manifold learning [39], or model reduction techniques in computational mechanics [40, 41]. PCA identifies the low d-dimensional subspace that best explains the variance of a higher D-dimensional data set. The original data are transformed to a new orthogonal coordinate system such that the projection of the data on the subspace defined by the first d coordinate directions, $d \leqslant D$, maximizes the variance as compared with any other projection onto a d-dimensional subspace. Given a data set in high dimension $X = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^D$, we define the matrix $X \in \mathbb{R}^{D \times N}$, where the centered coordinates of the points, $x_a - \bar{x}$ for $a = 1, \ldots, N$, are placed as column vectors. The average position of all the points $\bar{x}$ is

$$\bar{x} = \frac{1}{N} \sum_{a=1}^{N} x_a.$$

The covariance matrix is then

$$C = \frac{1}{N} X X^{\mathrm{T}} \quad \in \mathbb{R}^{D \times D}.$$

This positive (semi-)definite symmetric matrix has real eigenvalues and diagonalizes in an orthonormal basis of eigenvectors. We define $V \in \mathbb{R}^{D \times d}$ as the eigenvector matrix formed by the d eigenvectors corresponding to the largest d eigenvalues. These vectors span the affine space of dimension d passing through $\bar{x}$, which best describes the data. The matrix V defines an orthogonal projection Π_{PCA} relative to $\bar{x}$ onto the reduced space of dimension d, that is,

$$\Pi_{\mathrm{PCA}} : \mathbb{R}^D \longrightarrow \mathbb{R}^d$$
$$x \longmapsto V^{\mathrm{T}}(x - \bar{x}).$$

The low-dimensional representation of the data is then given by $\xi_a = \Pi_{\mathrm{PCA}}(x_a)$ for $a = 1, \ldots, N$.

Another classical method of linear dimensionality reduction is multidimensional scaling (MDS). MDS is more suitable when, instead of data coordinates, we have similarity scores between each pair of *stimuli*. MDS is a standard tool in psychophysics and sensory analysis. Here, we describe the classical metric MDS [42]. Extensions and details about non-metric MDS can be found for instance in [43] and references therein. The data are given in terms of a full $N \times N$ matrix D containing the squares of the similarities between each pair of points. MDS projects the data set into the affine space that best preserves the similarity scores between the data points. If the similarity score is simply the distance derived from the Euclidean metric, that is, $D_{ab} = |x_a - x_b|^2$, then the method tries to find a nearly isometric low-dimensional embedding of the data, and it can be shown to coincide with PCA. In this case, the Gram matrix formed by the scalar products between pairs of centered data is simply

$$S = X^{\mathrm{T}} X \quad \in \mathbb{R}^{N \times N}.$$

In general, the Gram matrix is obtained from the matrix of squared distances (similarities) D through a double-centering step, $S = -1/2 H D H$, where $H_{ij} = \delta_{ij} - 1/N$. The most expensive step in the method is the singular value decomposition of the Gram matrix $S = U \Lambda U^{\mathrm{T}}$, where the eigenvalues are sorted in decreasing order. The data points expressed in the latent variables $\Xi = \{\xi_1, \xi_2, \ldots, \xi_N\} \subset \mathbb{R}^d$ are computed in matrix form as

$$\Xi = I d_{dN} \Lambda^{1/2} U^{\mathrm{T}} \quad \in \mathbb{R}^{d \times N},$$

where $I d_{dN}$ is a $d \times N$ matrix formed by the first d rows of the $N \times N$ identity matrix. The low-dimensional embedding of out-of-sample points $x \in \mathbb{R}^D$ can be obtained from the MDS projection Π_{MDS} (see [44] for details).

The PCA and MDS share important features: both define a linear implicit mapping from the high-dimensional space into the low-dimensional embedding, and the core operation for both methods is the computation of the eigenvalues and eigenvectors of a full matrix. If the data are not too high dimensional but the number of points N is very large, PCA is preferable in terms of

memory and computational cost because the covariance matrix is $D \times D$. By contrast, MDS is better suited when dimensionality is very high but the number of points is moderate, as the Gram matrix is $N \times N$.

2.2. Nonlinear dimensionality reduction

In many cases, the high-dimensional data do not conform to an affine subspace but rather to a nonlinear manifold embedded in $\mathbb{R}^D$. Linear methods are not able to capture this manifold and instead detect its affine hull, which can be of significantly higher dimension. This results in a less compact representation of the data, and what is more, the linear representation glosses over the intrinsic and meaningful structure of the data. NLDR methods have emerged over the last decades for the statistical learning of complex data sets with hidden nonlinear structures in areas such as multivariate data analysis [45], pattern recognition [38], and image processing [46, 47] to mention a few. For an extended NLDR literature and insightful remarks, the reader is referred to [44, 48]. Some years ago, two landmark methods have invigorated this field: Isomap [49] and locally linear embedding (LLE) [50, 51]. Each of these papers has collected some 4000 citations over the last decade. They have expanded the range of the applicability of NLDR and have motivated new developments in the field [39, 52–57]. Successful applications of these techniques include climate data analysis [58], the study of the conformation dynamics of molecules [59, 60], and galaxy spectra classification [61].

2.2.1. Isomap.
Isomap tries to embed isometrically the high-dimensional data in $\mathbb{R}^d$, where the distance between data points is an approximate geodesic distance on the manifold. It is a mere application of MDS, and the core of the method is in the calculation of the matrix $\boldsymbol{D}$. This method builds a graph whose vertices are the data points and whose links are the union of the k-nearest-neighbor connections for each vertex. The graph is weighted by the Euclidean distance in $\mathbb{R}^D$ between nearby neighbors. The underlying assumption is that, if the manifold is sufficiently well sampled, the k-neighborhood of a point will look Euclidean. Then, the geodesic distance on the manifold is approximated as the shortest path distance on this graph and is used to fill in the matrix of squared distances. Isomap has been shown to be robust (more than LLE) for data polluted with noise or for non-uniformly distributed data points. However, as a corollary of Gauss's *Theorema Egregium* [62], we know that it is not possible to isometrically embed in two dimensions a surface with non-zero Gaussian curvature. This fact leads to a frustration in the algorithm, which can become unstable. As discussed previously for MDS, Isomap has a very high computational cost and memory requirements when the number of points is high but can deal easily with very high-dimensional data.

2.2.2. LLE.
Locally linear embedding assumes that the manifold can be locally approximated by a linear small patch, and hence, each point $\boldsymbol{x}_a$, $a = 1, \ldots, N$ can be reconstructed from its nearest neighbors as

$$\boldsymbol{x}_a = \sum_{b \in \mathcal{N}_{\boldsymbol{x}_a}} W_{ab}\, \boldsymbol{x}_b,$$

where $\mathcal{N}_{\boldsymbol{x}_a}$ is the list of indices of the closest k-neighbors to $\boldsymbol{x}_a$ and W_{ab} are the elements of an $N \times N$ sparse matrix containing unknown weights. The sparsity comes from the fact that each point is reconstructed in terms of its nearest neighbors alone. LLE computes these weights by minimizing the cost function

$$\mathcal{E}(\boldsymbol{W}) = \sum_{a=1}^{N} \left| \boldsymbol{x}_a - \sum_{b \in \mathcal{N}_{\boldsymbol{x}_a}} W_{ab}\, \boldsymbol{x}_b \right|^2.$$

In this optimization program, each row of the weight matrix $\boldsymbol{W}$ is constrained to sum up to 1 to enforce the invariance with respect to translations. These linear weights contain the intrinsic information about the local geometrical structure of the manifold. They are invariant with

respect to translations, scaling, and rotations. Then, with the weights fixed to the solution of the aforementioned program, the low-dimensional embedding is obtained from minimizing

$$\mathcal{F}(\boldsymbol{\Xi}) = \sum_{a=1}^{N} \left| \boldsymbol{\xi}_a - \sum_{b \in \mathcal{N}_{\boldsymbol{x}_a}} W_{ab}\, \boldsymbol{\xi}_b \right|^2 ,$$

subject to $(1/N)\boldsymbol{\Xi}\,\boldsymbol{\Xi}^{\mathrm{T}} = \boldsymbol{I}$. This constraint removes the affine invariance of the solution, by requiring the low-dimensional points to have unit covariance. Thus, this method computes a low-dimensional embedding that respects the local geometric structure contained in the sparse matrix $\boldsymbol{W}$. This minimization problem has a global minimum, which can be solved by finding the smallest eigenvalues and eigenvectors of the $N \times N$ sparse matrix $\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{W})^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{W})$. The reader is referred to [50, 51] for full details and to [63] for a concise mathematical description.

Locally linear embedding does not try to be isometric and, in fact, ignores metric information altogether by producing a low-dimensional embedding of unit covariance. It is more sensitive to the number of neighbors than Isomap. From an efficiency viewpoint, in contrast with Isomap, the $N \times N$ matrix whose eigenvalues and eigenvectors need to be computed is sparse for the LLE method. Therefore, this method is applicable to problems with large numbers of sampling points. LLE presents instabilities due to the ill conditioning of the optimization problem that selects the weights, leading often to spurious distortions. The stability of the LLE method has been enhanced in subsequent modifications, such as Hessian LLE [54] or the modified LLE (MLLE) method [57]. These methods produce better-quality embeddings of low dimension in general, and the MLLE method has a small computational overhead as compared with LLE.

2.2.3. Intrinsic dimensionality. In the algorithms outlined earlier, the manifold dimensionality is assumed to be given, and for thin-shells, $d = 2$. Yet, in other applications, the manifold dimensionality may not be known, and we cannot resort to visual inspection in high dimensions. The estimation of the intrinsic dimensionality underlying a high-dimensional data set is an important question in data exploration. There are many methods to estimate d; here, we give a brief account of the correlation dimension method, local PCA, and tracking the reconstruction error [44, Chapter 3].

The correlation dimension method, originally proposed in [64], considers a closed ball of radius ϵ at the center of each data point and counts the number of points inside this ball. The dimension is estimated by noting that the average number of counts should grow linearly with ϵ for 1D objects, quadratically for two-dimensional entities, and so on. Details about practical implementations and insightful remarks can be found in [44]. Local PCA analyzes small local subsets of the data. The intrinsic dimension d for each of these patches is selected such that it preserves a given fraction of the variance of the original data, for example,

$$0.95 \leqslant \frac{\sum\limits_{i=1}^{d} \lambda_i}{\sum\limits_{i=1}^{D} \lambda_i},$$

where λ_i are the eigenvalues of the covariance matrix arranged in decreasing order. In contrast with the global correlation dimension, this method provides a local estimation of the intrinsic dimension. The reconstruction error is a measure of the dissimilarity between the original high-dimensional data points $\boldsymbol{x}_a \in \mathbb{R}^D$ and their reconstruction from the low-dimensional points, generally computed as a weighted average of the neighbors of $\boldsymbol{\xi}_a \in \mathbb{R}^d$. Whereas for PCA, the notion of reconstruction is straightforward, this is not the case for NLDR methods.

The preceding techniques are easy to implement and robust for data not polluted with noise. For noisy measurements, the notion of intrinsic dimension becomes strongly scale dependent, and these methods cannot be used as a black box. In such situations, physical understanding of the system should be combined with dimension estimation methods.
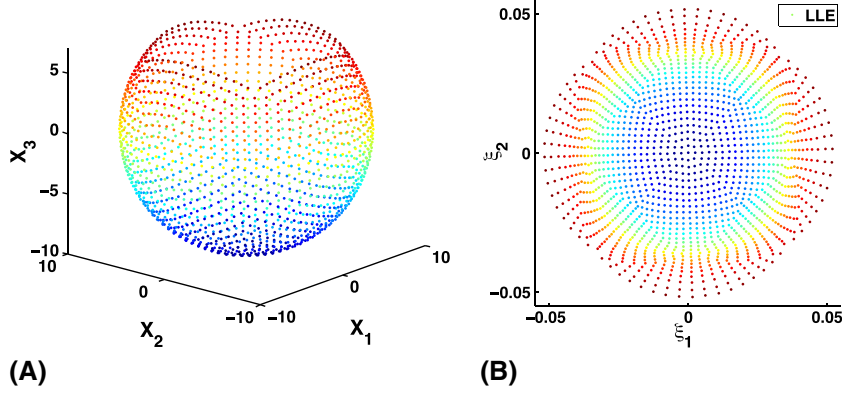
Figure 3. (A) Set of scattered points quasi-uniformly distributed on a truncated sphere. (B) Two-dimensional embedding obtained by locally linear embedding (LLE), exhibiting unavoidable high distortions. The color coding is a visual guide.

2.3. Illustrative examples, point-set partitioning, and affine isometric correction

We show here a few toy examples illustrating the challenges that need to be addressed to use the low-dimensional embeddings given by NLDR techniques as local parametric patches in processing point-set manifolds. We first consider uniformly sampled points lying on a sphere, describing a large portion of the full sphere (Figure 3A). Isomap is not able to embed this cloud of points in two dimensions, despite them clearly representing a surface with the topology of a two-dimensional open set. The frustration arising from the impossibility of embedding isometrically such a large portion of a sphere is too large, and the method fails. LLE provides satisfactory results, at the expense of generating very large local distortions (Figure 3B), which are unavoidable and a corollary of Gauss's *Theorema Egregium*. Another notable feature of the results of LLE in this example is the lack of a clear metric relationship between the low-dimensional embedding and the original data, as noted earlier. Both the distortions and the lack of metric correspondence are problematic for our purposes. Indeed, a good-quality sampling of a surface may become strongly distorted or scaled in one direction if the surface is elongated. As a result, the H^1 norm of the parametrization from the low-dimensional embedding into three dimensions may become very large and non-uniform, leading to inaccurate numerical calculations.

We address the unavoidable distortions by partitioning the original data set into a small number of patches (Figure 4A). We then embed each partition in two dimensions with NLDR techniques. We use the METIS library to partition the data [65]. Not only is partitioning advisable to obtain good-quality embeddings of low dimension, it also becomes unavoidable for surfaces of general topology, as NLDR methods are applicable only to surfaces of trivial topology. It is always possible to recursively partition a data set until all the partitions have trivial topology. Figure 4B–D shows the low-dimensional embedding generated by Isomap, LLE, and MLLE, respectively. Isomap provides a good-quality embedding of low dimension, whereas LLE produces spurious distortions. MLLE corrects these distortions yet introduces a uniform stretching of the domain in one direction as a result of its affine invariance and the unit covariance constraint.

To address the uniform distortions of LLE-based methods, that is, the lack of metric correspondence between the low-dimensional embedding and the original data, we propose a simple iterative method that we call affine isometric correction (AIC). This method finds the affine transformation T that minimizes a measure of isometry [66], given by the discrete stress function

$$E_{\mathrm{S}}(\boldsymbol{T}) = \frac{1}{2} \sum_{a=1}^{N} \sum_{b \in \mathcal{N}_{\boldsymbol{x}_a}} \left(\frac{|\boldsymbol{T}(\boldsymbol{\xi}_a - \boldsymbol{\xi}_b)| - |\boldsymbol{x}_a - \boldsymbol{x}_b|}{|\boldsymbol{x}_a - \boldsymbol{x}_b|} \right)^2,$$

where $\boldsymbol{\xi}_a$ are the low-dimensional points given by a direct application of a NLDR method and $\boldsymbol{T}$ is a $d \times d$ matrix with a positive determinant. By the polar decomposition theorem, it is sufficient to
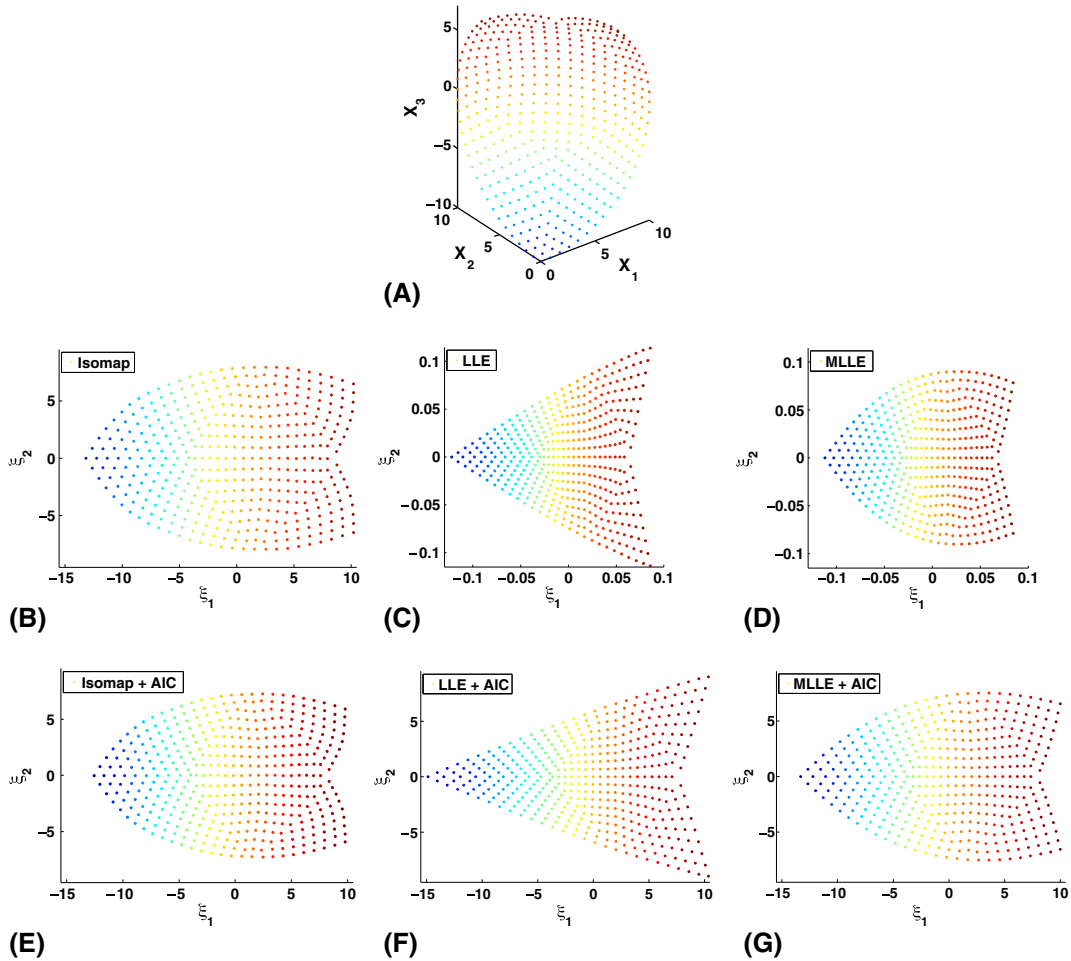
Figure 4. (A) Partition of the data set in Figure 3. Two-dimensional embeddings given by a direct application of (B) Isomap, (C) locally linear embedding (LLE), and (D) modified LLE (MLLE). Two-dimensional embeddings given by (E) Isomap, (F) LLE, and (G) MLLE after the affine isometric correction step.

Table I. Stress function E_S for the quarter of a truncated sphere (Figure 4).

T	Isomap	LLE	MLLE
I	25.9	30.6×10^4	26.8×10^4
arg min E_S	11.7	96.5	6.22

Here, we consider a k-rule search for neighbors [67], with $k = 12$.
LLE, locally linear embedding; MLLE, modified locally linear embedding.

minimize the function over symmetric positive-definite matrices. Computationally, we implement Newton's method with line search that converges very rapidly. The low-dimensional embedding resulting from post-processing the raw NLDR results with AIC are shown in Figure 4E–G, and the values of the stress function are given in Table I. It is clear that this simple procedure significantly improves the isometric quality of LLE-based methods *a posteriori*, while it changes only marginally the results of metric methods such as Isomap. In subsequent calculations, we choose the MLLE method combined with the AIC, as it provides good-quality embedding of low dimension metrically related to the original data, and it involves calculations on sparse matrices.

3. MANIFOLD DESCRIPTION FROM SCATTERED POINTS

3.1. General setting and partition of unity

We consider a smooth d-manifold $\mathcal{M}$ embedded in $\mathbb{R}^D$, $d < D$. Our goal is to numerically represent $\mathcal{M}$ from a set of points and make computations on it. Let $P = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_N\} \subset \mathbb{R}^D$ be a set of control points representing $\mathcal{M}$. We consider another set of geometric markers, $Q = \{\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_M\} \subset \mathbb{R}^D$, 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, for example, $\boldsymbol{P}_a$, for $a = 1, 2, \ldots, N$, and the geometric markers in Q and its associated objects with an upper case subindex, for example, $\boldsymbol{Q}_A$, for $A = 1, 2, \ldots, M$ $(M \leqslant N)$.

We partition these geometric markers into L groups on the basis of proximity (METIS domain decomposition with a k-nearest neighbor graph). We represent these groups of geometric markers with index sets $\mathcal{I}_\kappa, \kappa = 1, \ldots, L$ with $\cup_{\kappa=1}^L \mathcal{I}_\kappa = \{1, 2, \ldots, M\}$ and $\mathcal{I}_\kappa \cap \mathcal{I}_\iota = \emptyset$ when $\kappa \neq \iota$, and use Greek subindices to refer to entities associated with these groups of markers. As it will become clear below, there is a one-to-one correspondence between these groups of geometric markers and the local parametrizations of the surface, which we call patches.

We consider a Shepard PU associated with the geometric markers. Consider a set of non-negative reals $\{\beta_A\}_{A=1,2,\ldots,M}$ associated with each point in Q. We define the Shepard PU with Gaussian weight associated to the set Q as the functions $w_A : \mathbb{R}^D \to \mathbb{R}$ given by

$$w_A(\boldsymbol{x}) = \frac{\exp\left(-\beta_A |\boldsymbol{x} - \boldsymbol{Q}_A|^2\right)}{\sum_{B=1}^M \exp\left(-\beta_B |\boldsymbol{x} - \boldsymbol{Q}_B|^2\right)}.$$

For efficiency and given the fast decay of the Gaussian functions, these functions are numerically treated as compactly supported.

We aggregate these PU functions by patches, as it is depicted in Figure 5 for $d = 1$ and $D = 2$, yielding a coarser PU given by the functions

$$\psi_\kappa(\boldsymbol{x}) = \sum_{A \in \mathcal{I}_\kappa} w_A(\boldsymbol{x}).$$

These functions form a PU in $\mathbb{R}^D$ and consequently also in $\mathcal{M}$. We consider the index sets of all control points influencing each patch, $\mathcal{J}_\kappa$, with $\cup_{\kappa=1}^L \mathcal{J}_\kappa = \{1, 2, \ldots, N\}$, but now $\mathcal{J}_\kappa \cap \mathcal{J}_\iota \neq \emptyset$ because of the overlap between patch PU functions. Roughly speaking, these sets are $\{a \mid \boldsymbol{P}_a \in \mathrm{sup}\,\psi_\kappa\}$, slightly enlarged so that the patch parametrization is smooth on the boundary of the support of ψ_κ. Note that, as can be seen in Figure 5, the partition unity functions ψ_κ restricted to the manifold are quite insensitive to the position of the geometrical markers $\boldsymbol{Q}_A$ in the direction normal to the curve. Figure 6 illustrates the overlap regions for a partition of the sphere.
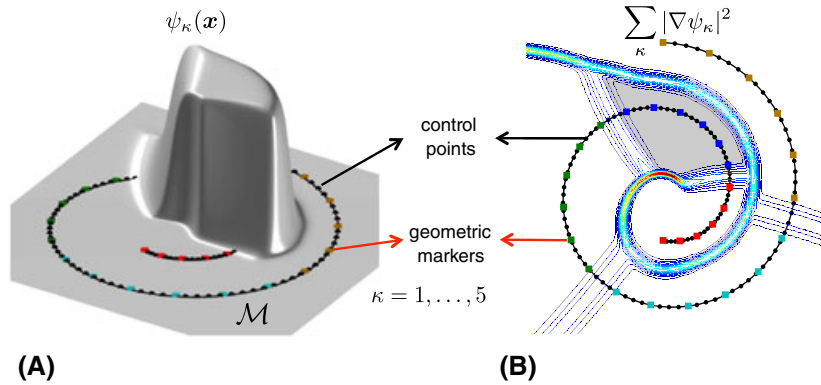

(A)   **(B)**

Figure 5. Illustration of the proposed method for a curve ($d = 1$) in the plane $D = 2$. (A) Illustration of a function $\psi_\kappa(\boldsymbol{x})$ of the coarse partition of unity tied to the patches. (B) Visualization of the coarse partition-of-unity overlap regions. The partition of the geometric markers is color coded.
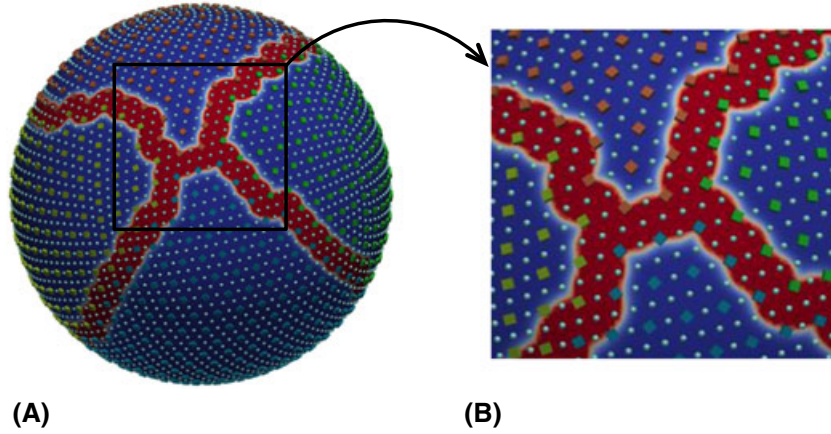
Figure 6. (A) Visualization of a coarse partition of unity overlap regions on a sphere (contour map of $\sum_\kappa |\nabla \psi_\kappa|^2$). (B) Zoom from a selected region on the sphere; the partitioning of the geometric markers ($\diamond$, color coded by the partitions), the width of the overlapping, and the control points $\circ$ can be appreciated.

3.2. Patch manifold learning and surface parametrization

For each patch, we obtain a low-dimensional embedding R_k of the points $P_\kappa = \{\boldsymbol{P}_a\}_{a \in \mathcal{J}_\kappa} \subset \mathbb{R}^D$ with an NLDR technique, resulting in the representation $\Xi_\kappa = \{\boldsymbol{\xi}_a\}_{a \in \mathcal{J}_\kappa} \subset \mathbb{R}^d$, this is

$$R_\kappa : \mathbb{R}^D \longrightarrow \mathbb{R}^d$$
$$P_\kappa \longmapsto \Xi_\kappa.$$

For instance, in meshfree thin-shell analysis, MLLE with AIC produces in general a low-dimensional embedding of good geometric quality and can be computed efficiently. See Figure 7 for an illustration in a complex example. If the automatic partitioning of the data creates patches of complex topology, for example, a tubular partition in an ear of the bunny in the figure or patches of high geometric complexity leading to excessive geometric distortions, such as the blue partition at the tip of one ear, we proceed by recursively re-partitioning such patches.
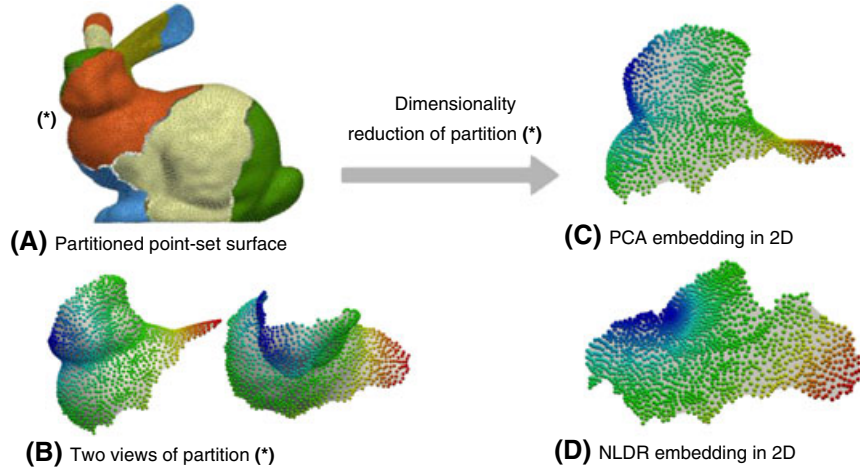

Figure 7. (A) The automatic partitioning of a point-set surface representing the Stanford bunny, performed with METIS, can create patches of complex geometry and topology, for example, a tubular partition in an ear. We recursively partition such patches. (B) Two views of the patch (*). Low-dimensional embedding of the patch (*) by PCA (C) and by a nonlinear method (D). The colors of the points are provided to guide the visual inspection of the embeddings of low dimension. PCA collapses large regions of the patch, whereas the NLDR method successfully 'irons' the curved patch into a moderately distorted low-dimensional embedding.

(A) Basis functions

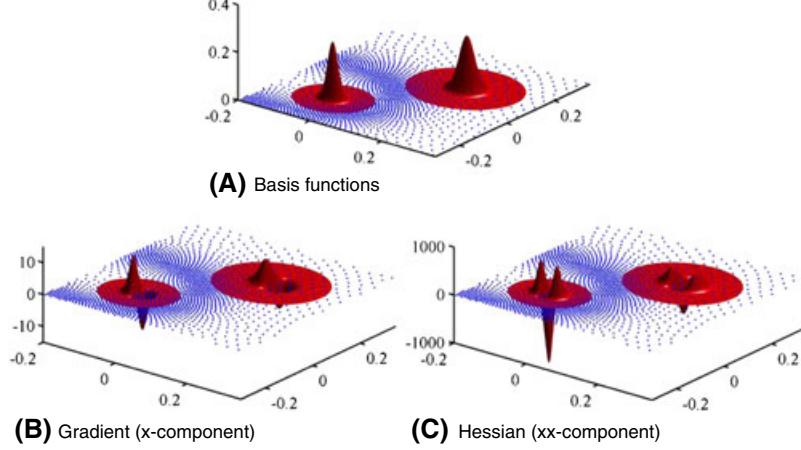**(B)** Gradient (x-component)   **(C)** Hessian (xx-component)

Figure 8. (A) Local *maximum-entropy* basis functions computed with an aspect ratio parameter $\gamma = 0.8$ in an unstructured and non-uniform two-dimensional distribution of points. (B, C) The x-components of the gradient and the Hessian.

The low-dimensional space where the data points of a partition are embedded is a convenient parametric space for the corresponding patch. It is important to note that the embedded points are in general unstructured, and that, although here $d = 2$, the methodology is applicable to higher-dimensional embedded manifolds. This is the topic of current research. The patch parametrizations often need to be smooth, here because of the requirements of the Kirchhoff–Love theory. Thus, a general method to process embedded smooth manifolds demands a smooth approximation scheme for general unstructured nodes in multiple dimensions. There are not many available methods fulfilling these requirements. In two dimensions, subdivision approximations may be used. Here, we consider a general meshfree method to produce such approximants in any spatial dimension d, the local *max-ent* approximants. See [28, 30] for the formulation, properties, and evaluation of the basis functions and their derivatives. The local *max-ent* basis functions in two dimensions are illustrated in Figure 8.

Let $p_a(\boldsymbol{\xi})$ denote the local *max-ent* approximants associated to the point set Ξ_κ on a domain $\mathscr{A}_\kappa \subset \mathbb{R}^d$, a subset of the convex hull of the reduced node set conv Ξ_κ. We locally parametrize the manifold in this patch as

$$\boldsymbol{\varphi}_\kappa : \mathscr{A}_\kappa \longrightarrow \mathbb{R}^D$$
$$\boldsymbol{\xi} \longmapsto \sum_{a \in \mathcal{J}_\kappa} p_a(\boldsymbol{\xi}) \, \boldsymbol{P}_a.$$

Note that the images of the local patches, $\boldsymbol{\varphi}_\kappa(\mathscr{A}_\kappa)$, overlap in the vicinity of the partition boundaries and need not exactly coincide in these regions.

3.3. Partition of unity to evaluate integrals on $\mathcal{M}$

A PU is a classical technique to patch together local objects on a manifold [62]. Consider for instance the integral of a scalar function f over a manifold $\mathcal{M}$, $f : \mathcal{M} \to \mathbb{R}$. Then, we have the following identity

$$\int_{\mathcal{M}} f(\boldsymbol{x}) \, \mathrm{d}\mathcal{M} = \sum_{\kappa=1}^{L} \int_{\mathcal{M}} \psi_\kappa(\boldsymbol{x}) \, f(\boldsymbol{x}) \, \mathrm{d}\mathcal{M}.$$

Combining the PU with the local parametrization of the κth patch, we can approximate numerically integrals over the manifold $\mathcal{M}$ described by a set of scattered points as

$$\int_{\mathcal{M}} f(\boldsymbol{x})\,\mathrm{d}\mathcal{M} \simeq \sum_{\kappa=1}^{L} \int_{\mathscr{A}_{\kappa}} \psi_{\kappa}(\boldsymbol{\varphi}_{\kappa}(\boldsymbol{\xi}))\, f(\boldsymbol{\varphi}_{\kappa}(\boldsymbol{\xi}))\, J_{\kappa}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi},$$

where $J_{\kappa} = \sqrt{\det\left[(D\boldsymbol{\varphi}_{\kappa})^{\mathrm{T}} D\boldsymbol{\varphi}_{\kappa}\right]}$ is the Jacobian determinant of the parametrization. In this way, similar to finite element methods, we have split the integral into local contributions, which can be evaluated using local parametrizations. The last integral can be subsequently approximated by a numerical quadrature on the local parametric space. Here, we resort to Gauss quadrature on a support triangulation defined over Ξ_{κ}.

All variational models describing the mechanics of thin-shells are formulated in terms of integrals over the mid-surface, for example, the elastic energy functional, or the weak form. Consequently, all these theories can be approximated numerically following the preceding ideas. In such cases, the function does not depend explicitly on $\boldsymbol{x}$, but rather on other fields on the manifold, yet the method is still applicable.

3.4. Single representation of a manifold described by multiple patches

We have avoided so far a precise definition of a numerical surface in the overlapping regions. Although a single representation of the manifold given by multiple overlapping patches, which do not coincide exactly at the overlap regions, is not needed to compute integrals and functionals on the manifold, such a representation is useful in a number of situations such as visualization, contact detection, or imposition of displacement at interior points. We describe now how we proceed. As a starting point, we have an out-of-sample point $\boldsymbol{x} \in \mathbb{R}^{D}$ in the vicinity of the embedded manifold. This point could be a sampling of the actual surface or the image of a point in parametric space by a patch parametrization. Our goal is to define an operator mapping $\boldsymbol{x}$ onto the manifold. We first identify the patches that have an influence on $\boldsymbol{x}$ by building the index set

$$\mathcal{N}_{\boldsymbol{x}} = \{\kappa \in \{1, 2, \ldots, L\} \mid \psi_{\kappa}(\boldsymbol{x}) > \mathrm{TOL}\}.$$

We can then find the preimage of the closest point projection of $\boldsymbol{x}$ onto the multiple patch representations, which we denote by $\boldsymbol{\xi}_{\kappa} = \boldsymbol{\varphi}_{\kappa}^{-1}(\boldsymbol{\pi}_{\kappa}(\boldsymbol{x}))$ for $\kappa \in \mathcal{N}_{\boldsymbol{x}}$, where $\boldsymbol{\pi}_{\kappa}(\boldsymbol{x})$ is the closest point projection of $\boldsymbol{x}$ at each patch (Appendix A). We can then define the operator (almost a projection)

$$\boldsymbol{\pi}(\boldsymbol{x}) = \sum_{\kappa \in \mathcal{N}_{\boldsymbol{x}}} \psi_{\kappa}(\boldsymbol{x})\, \boldsymbol{\varphi}_{\kappa}(\boldsymbol{\xi}_{\kappa}(\boldsymbol{x})),$$

which averages the position of the point as represented by the different overlapping parametrizations. Similar to the definition of MLS point-set surfaces [18, 20], we can formally define the numerical surface at overlapping regions as the fixed points of this operator. For MLS point-set surfaces, the properties of the fixed points of a related operator have been mathematically analyzed, and the properties of the resulting manifold established. Our numerical experiments indicate that successive iterations of $\boldsymbol{\pi}(\boldsymbol{x})$ converge extremely fast. In practice, we do not iterate the operator.

4. THIN-SHELL MODEL

In this section, we review the mechanics of thin-shells [5, 6], on the basis of a geometrically exact formulation [3, 68]. We restrict our attention to the Kirchhoff–Love theory of shells, that is, 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 (that is, $i = 1, 2, 3$; $\alpha = 1, 2$), a comma denotes partial differentiation, subscripts refer to covariant components, and superscripts denote contravariant components.

4.1. Kinematics of the shell

We next describe the kinematics of a thin-shell body $\mathscr{S}$ in three-space (Figure 9). We assume that this body can be described by the pair $(\boldsymbol{\varphi}, \boldsymbol{t})$, where the mapping $\boldsymbol{\varphi}$ defines the shell middle surface, Ω, and $\boldsymbol{t}$ is a field of unit vectors (a field of directors). We assume the thickness h of the shell to be uniform for simplicity, and also we assume that the change in shell thickness after deformation is negligible. Then, the thin-shell body $\mathscr{S}$ is given by

$$\mathscr{S} = \left\{ \boldsymbol{\Phi} \in \mathbb{R}^3 \mid \boldsymbol{\Phi} = \boldsymbol{\varphi}(\xi^\alpha) + \xi\, \boldsymbol{t}(\xi^\alpha), \; -\frac{h}{2} \leqslant \xi \leqslant \frac{h}{2}, \; (\xi^1, \xi^2) \in \mathscr{A} \right\},$$

where $\mathscr{A} \subset \mathbb{R}^2$ is the parametric space for the middle surface. Hence, we view a configuration $\boldsymbol{\Phi}$ as a mapping from a parametric domain $\mathscr{A} \times [-h/2, h/2]$ 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 $\{\boldsymbol{E}^i\}$. The area element of the middle surface can be computed as $d\Omega = \bar{j}\, d\xi^1 d\xi^2$, where $\bar{j} = |\boldsymbol{\varphi}_{,1} \times \boldsymbol{\varphi}_{,2}|$. The tangent map of a given configuration $T\boldsymbol{\Phi}$ can be computed from the convective basis vectors $\boldsymbol{g}_i$ as

$$T\boldsymbol{x} = \frac{\partial \boldsymbol{\Phi}}{\partial \xi^i} \otimes \boldsymbol{E}^i = \boldsymbol{g}_i \otimes \boldsymbol{E}^i,$$

with $\boldsymbol{g}_\alpha = \partial \boldsymbol{\Phi}/\partial \xi^\alpha = \boldsymbol{\varphi}_{,\alpha} + \xi\, \boldsymbol{t}_{,\alpha}$ and $\boldsymbol{g}_3 = \partial \boldsymbol{\Phi}/\partial \xi = \boldsymbol{t}$. The covariant components of the metric tensor in convected coordinates are given by $g_{ij} = \boldsymbol{g}_i \cdot \boldsymbol{g}_j$.

The subscript 0 denotes quantities in the reference configuration, for instance, $\boldsymbol{\varphi}_0$ parametrizes the reference middle surface. A deformation mapping is a mapping from a reference body into $\mathbb{R}^3$, $\boldsymbol{\Phi} \circ \boldsymbol{\Phi}_0^{-1}$. Consequently, the deformation gradient is $\boldsymbol{F} = T\boldsymbol{\Phi}\,(T\boldsymbol{\Phi}_0)^{-1}$, and the Jacobian is $J = \det(\boldsymbol{F}) = j/j_0$, where $j = \det(T\boldsymbol{\Phi}) = \boldsymbol{g}_3 \cdot (\boldsymbol{g}_1 \times \boldsymbol{g}_2)$.

The shell director in the reference configuration $\boldsymbol{t}_0$ coincides with the normal to the undeformed middle surface of the shell and hence

$$\boldsymbol{t}_0 = \frac{\boldsymbol{\varphi}_{0,1} \times \boldsymbol{\varphi}_{0,2}}{\bar{j}_0}, \quad \boldsymbol{\varphi}_{0,\alpha} \cdot \boldsymbol{t}_0 = 0, \quad |\boldsymbol{t}_0| = 1, \quad \boldsymbol{t}_0 \cdot \boldsymbol{t}_{0,\alpha} = 0.$$

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

The local shell deformations can be characterized by the Green–Lagrange strain tensor. Because the convected components of the metric tensor coincide with the components of $(T\boldsymbol{\Phi})^{\mathrm{T}} T\boldsymbol{\Phi}$ in the basis associated with $\{\xi^i\}$, the Green–Lagrange strain tensor can be expressed as the difference between the metric tensors on the deformed and undeformed configurations of the shell, that is,

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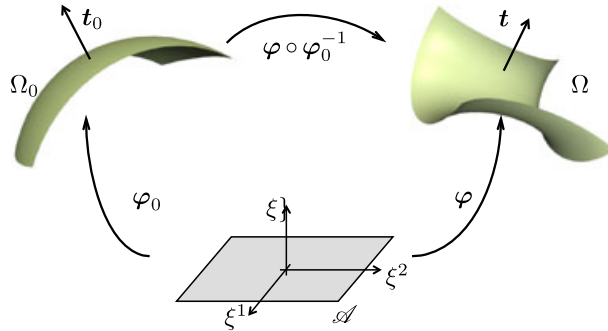

Figure 9. Reference, deformed and parametric configurations of the middle shell surface.

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

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

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

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

4.2. Kirchhoff–Love hypothesis

In the remainder of this section, we restrict our attention to the Kirchhoff–Love theory of thin-shells, that is, we constrain the deformed director $\boldsymbol{t}$ to coincide with the unit normal of the deformed middle surface of the shell, that is,

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

This assumption is well suited when the ratio between the shell thickness and its characteristic size is $\ll 1$. With the Kirchhoff–Love hypothesis, the only remaining non-zero components of the Green–Lagrange strain tensor are

$$
\begin{aligned}
E_{\alpha\beta} &= \frac{1}{2}(a_{\alpha\beta} - a_{0\alpha\beta}) + \xi(\kappa_{\alpha\beta} - \kappa_{0\alpha\beta}) + \frac{(\xi)^2}{2}(\boldsymbol{t}_{,\alpha} \cdot \boldsymbol{t}_{,\beta} - \boldsymbol{t}_{0,\alpha} \cdot \boldsymbol{t}_{0,\beta}) \\
&= \varepsilon_{\alpha\beta} + \xi\, \rho_{\alpha\beta} + (\xi)^2\, \vartheta_{\alpha\beta},
\end{aligned}
\tag{1}
$$

where we have introduced the first and second fundamental forms expressed in convected components

$$
\begin{aligned}
a_{\alpha\beta} &= \boldsymbol{\varphi}_{,\alpha} \cdot \boldsymbol{\varphi}_{,\beta}, \\
\kappa_{\alpha\beta} &= \boldsymbol{\varphi}_{,\alpha} \cdot \boldsymbol{t}_{,\beta} = -\boldsymbol{\varphi}_{,\alpha\beta} \cdot \boldsymbol{t}.
\end{aligned}
$$

Thus, the Kirchhoff–Love kinematic assumption leads to a formulation of the shell exclusively in terms of the middle surface.

4.3. Equilibrium configurations

The potential energy of an elastic shell body with bulk internal energy density W can be expressed as

$$\Pi[\mathbf{\Phi}] = \int_{\mathscr{S}_0} W(E_{ij})\, \mathrm{d}V_0 + \Pi_{\text{ext}}[\mathbf{\Phi}],$$

where Π_{ext} is the potential energy of the external loads. For concreteness, we consider an isotropic Kirchhoff–St. Venant elastic material [69]

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

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 (Equation (1)), and the effect of curvature on the Jacobian away from the middle surface is neglected, that is, $j_0/\bar{j}_0 = 1$ [3, 68]. 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

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

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 is Poisson's ratio. Thus, the internal potential energy is a functional of the middle-surface configuration, which can be written as an integral over the reference middle surface

$$\Pi_{\text{int}}[\boldsymbol{\varphi}] = \int_{\Omega_0} \mathcal{W}(E_{\alpha\beta}) \, \mathrm{d}\Omega_0,$$

and the external potential becomes

$$\Pi_{\text{ext}}[\boldsymbol{\varphi}] = -\int_{\Omega_0} \boldsymbol{q} \cdot \boldsymbol{\varphi} \, \mathrm{d}\Omega_0 - \int_{\partial\Omega_0} \boldsymbol{h} \cdot \boldsymbol{\varphi} \, \mathrm{d}\ell_0,$$

where $\boldsymbol{q}$ is the external body load per unit area, $\boldsymbol{h}$ are 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. Distributed torques can be also applied at the boundary of the thin-shell.

Following [3], we introduce the elastic constitutive relations between the shell stresses and the strains as

$$n^{\alpha\beta} = \frac{\partial \mathcal{W}}{\partial \varepsilon_{\alpha\beta}} = h \, C^{\alpha\beta\gamma\delta} \varepsilon_{\gamma\delta},$$

$$m^{\alpha\beta} = \frac{\partial \mathcal{W}}{\partial \rho_{\alpha\beta}} = \frac{h^3}{12} C^{\alpha\beta\gamma\delta} \rho_{\gamma\delta},$$

where $n^{\alpha\beta}$ is the membrane stress resultant and $m^{\alpha\beta}$ is the bending stress resultant.

The stable equilibrium configurations of the shell minimize the total potential energy, subject to the boundary conditions, and, consequently, satisfy the principle of virtual work, expressed here in terms of integrals over the parametric space $\mathscr{A}$:

$$0 = \delta\Pi[\boldsymbol{\varphi}, \delta\boldsymbol{\varphi}] = \int_{\mathscr{A}} (\delta\boldsymbol{\varepsilon} \cdot \boldsymbol{n} + \delta\boldsymbol{\rho} \cdot \boldsymbol{m}) \, \bar{j}_0 \, \mathrm{d}\xi^1 \mathrm{d}\xi^2 + \delta\Pi_{\text{ext}}[\delta\boldsymbol{\varphi}].$$

4.4. Ritz–Galerkin discretization

We consider now the discrete equilibrium equations for a shell whose middle surface in the reference configuration is numerically represented with the procedure described before, in terms of a set of nodes $P_0 = \{\boldsymbol{P}_{01}, \dots, \boldsymbol{P}_{0N}\}$ and a set of L patches. We follow a total Lagrangian approach, with the same parametric space and basis functions for the reference and deformed configurations. Let $\boldsymbol{\varphi}_{0\kappa}$ be the reference configuration mapping for the middle surface of a specific patch κ, defined over the parametric space $\mathscr{A}_\kappa$

$$\boldsymbol{\varphi}_{0\kappa}(\boldsymbol{\xi}) = \sum_{a \in \mathcal{J}_\kappa} p_a(\boldsymbol{\xi}) \, \boldsymbol{P}_{0a},$$

as described in Section 3.2. We represent the deformed configuration in a given patch κ as

$$\boldsymbol{\varphi}_\kappa(\boldsymbol{\xi}) = \sum_{a \in \mathcal{J}_\kappa} p_a(\boldsymbol{\xi}) \, \boldsymbol{P}_a.$$

With the strategy presented in Section 3.3 and the preceding definitions, the internal elastic energy of the discretized shell can be split into patch contributions

$$\Pi_{\text{int}}^h(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_N) = \sum_{\kappa=1}^L \int_{\mathscr{A}_\kappa} (\psi_\kappa \circ \boldsymbol{\varphi}_{0\kappa}) \, \mathcal{W}(E_{\alpha\beta}) \, \bar{j}_0 \, \mathrm{d}\xi^1 \mathrm{d}\xi^2,$$

where $E_{\alpha\beta}$ and $\bar{j}_0$ are evaluated with the κth patch approximation of the undeformed and deformed configurations. Note that only $E_{\alpha\beta}$ depends on the unknown control points defining the deformed configuration. The external potential is numerically computed likewise. Equilibrium configurations satisfy that the out-of-balance forces vanish

$$0 = \frac{\partial \Pi^h}{\partial \boldsymbol{P}_a}(P) = \boldsymbol{f}_{\text{int}}^a(P) - \boldsymbol{f}_{\text{ext}}^a(P).$$

Stable equilibrium configurations are obtained by numerically minimizing $\Pi^h(P)$, where the essential displacement and rotation boundary conditions are imposed with Lagrange multipliers in an augmented Lagrangian framework. Within the augmented Lagrangian loop, we first obtain a coarse and robust approximation of the equilibrium point with a limited-memory Broyden–Fletcher–Goldfarb–Shanno method and then switch to Newton's method combined with line search to refine the minimization. Details about the calculation of the out-of-balance forces, the tangent stiffness matrix, the boundary constraints, and the solution method are given in Appendices C, D, and E, respectively.

5. NUMERICAL EXAMPLES

We exercise the proposed method with some standard numerical linear and nonlinear benchmark tests. For the linear analysis of thin-shells, we consider the classical problem of a hemisphere loaded with two pairs of facing concentrated forces [29]. Then, we analyze two popular nonlinear problems [70], an open hemispherical shell subjected to alternating radial forces and the pullout of a cylindrical shell with open ends. Finally, the flexibility of the proposed methodology to deal with shells of complex topology and geometry is illustrated by two additional examples at the end of this section. A collection of videos highlighting the nonlinear mechanics of these geometrically exact shells can be found at [71].

5.1. Numerical aspects

We refer to [28] for a detailed account on the max-ent basis functions and the numerical parameters involved. We only note here, from this reference, that linearly reproducing local *max-ent* approximants with relatively wide support can very accurately approximate thin-shell problems with functionals involving second-order derivatives. The smoothness or aspect ratio of the basis functions is controlled by a nondimensional parameter, γ_{LME}. We choose $\gamma_{\text{LME}} = 0.8$, which provides accurate solutions at a moderate computational cost. Similarly, for the PU Shepard functions, we select $\gamma_{\text{PU}} = 4.0$, which results in moderately narrow overlap regions. In all the examples, we build a Delaunay triangulation in the low-dimensional embedding of each meshfree macro-element and generate a standard Gauss–Legendre cubature rule of 12 points (order 6) per triangle, an overkill integration rule. At the boundary curves, to integrate the boundary constraints, we choose a quadrature scheme of four Gauss–Legendre points per integration cell. As in [28, 72], we resort to ghost nodes at the boundaries of the middle surface to avoid the loss of accuracy caused by the excessive flattening of the *max-ent* approximants at the boundary of the convex hull of the nodes. The number of marker points is one order of refinement lower than the number of control points.

Smooth convex approximants, such as local *max-ent* methods, B-splines, and NURBS basis functions, are in general not interpolating. Therefore, if the set of control points lies on the manifold, a systematic error is introduced, which for shells generally results in a stiffer behavior. Here, we fit the control points so that the reconstruction error of the original surface is minimized in a least-squares sense (Appendix B). This procedure improves the accuracy of the method, although it is not required for optimal convergence rate, as shown in the next example.

5.2. Pinched closed hemisphere

In this example, a hemispherical shell of radius $R = 10$ and thickness $h = 0.04$ is subjected to two pairs of radial loads $F = 2$ acting along diametral directions (Figure 10A). This is a challenging test, which assesses the method's ability to represent inextensional deformations under complex shell bending conditions with curvature in two directions. The convergence of the relative error for the radial displacement is shown in Figure 10B. The displacements are normalized by a deflection of $\delta_r = 0.09241$ obtained by an overkill calculation, which agrees with the lower bound given in [1, 29], that is, $\delta_r = 0.0924$. In this figure, we plot the convergence results reported in [72] for subdivision finite elements based on Loop's scheme (triangular elements) and on Catmull–Clark's scheme (quadrilateral elements), as well as results with the previous version of our method reported in [28]. The excellent convergence properties of the proposed method is clear from the figure. We obtain more accurate results for a given number of degrees of freedom than arguably the most competitive method for thin-shells. Our method is more expensive than subdivision finite elements because of the quadrature and the larger sparsity pattern of the stiffness matrix, which makes a full comparison difficult. We can also see that the results of the present method are very similar to

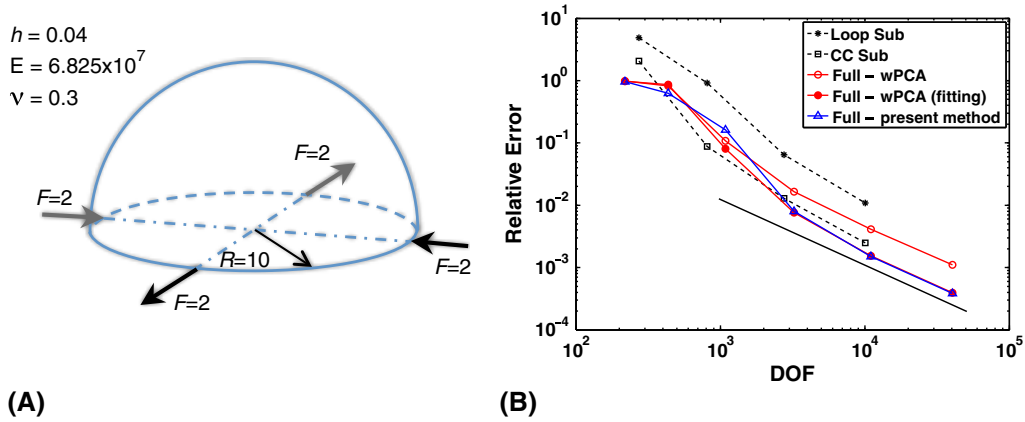
(A)   **(B)**

Figure 10. (A) Sketch of the pinched closed hemisphere shell test. (B) Convergence of the normalized radial displacement for two subdivision schemes [72], for the weighted PCA (wPCA) method proposed in [28] and for the present method based on modified locally linear embedding.
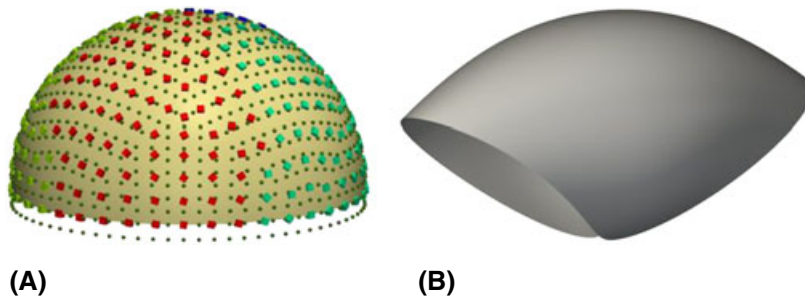

(A)   **(B)**

Figure 11. Pinched hemisphere: geometric markers ◇, control points ○, and reference configuration (A) and deformed surface (B). In this linear problem, the deformation has been amplified by a factor of 100.

those obtained with our previous local linear manifold learning method, which nevertheless is much more expensive, as discussed earlier (Figure 1). Finally, we note the effect of fitting the control points to better represent the geometry. The control and patch points and the deformation are shown in Figure 11.

5.3. Pinched open hemisphere

This is a classical nonlinear benchmark analyzing a hemispherical shell with a hole of 18° in its pole. The shell is pinched by two opposite pairs of forces, as shown in Figure 12A. Figure 12B plots the radial displacements under the loads. We compare our results against the results given by ABAQUS's S4R four-node shell element [70]. The agreement is remarkable.

5.4. Pullout of an open-ended cylindrical shell.

It is a challenging nonlinear benchmark, which shows the capabilities of the method to deal with problems with comparable membrane and bending energies. The material and geometrical properties for this benchmark are indicated in Figure 13A, whereas the radial displacements of the test points A, B, and C are plotted in Figure 13B. Again, we compare our results against those in [70], and the agreement is excellent. The biggest difference is at the indentation points.
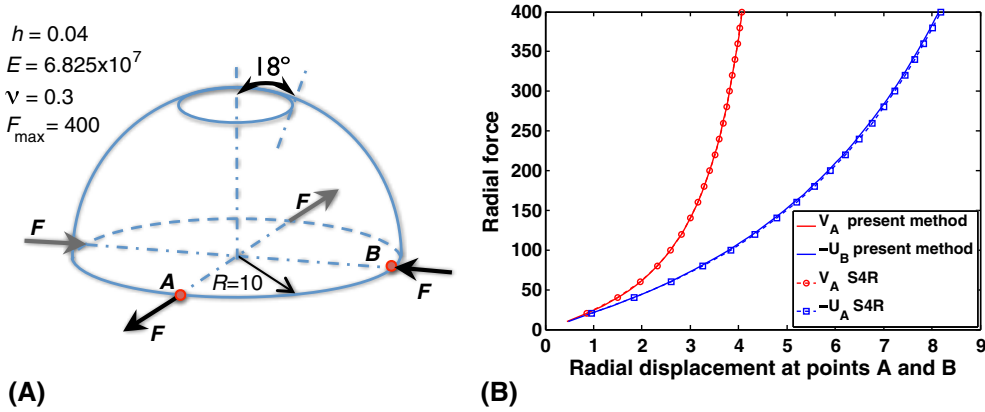

Figure 12. (A) Sketch for the pinched open hemisphere problem. (B) Load–deflection curves for the open hemispherical shell subjected to two pairs of facing concentrated forces.
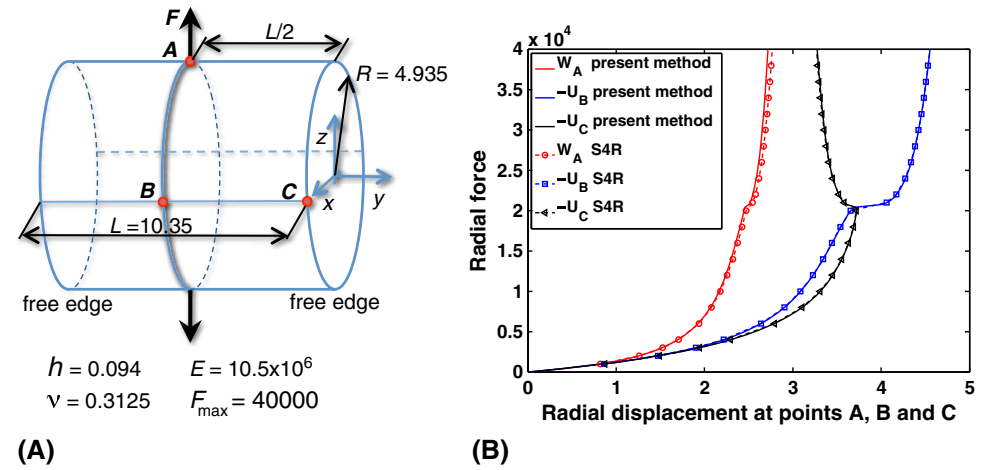

Figure 13. (A) Sketch for the pulled cylinder with free edges. (B) Load–deflection curves of an open-ended cylindrical shell under radial forces.

We interpret this small disagreement as an overestimation of the S4R elements caused by their non-smooth interpolating character. Figure 14 shows selected snapshots along the deformation, illustrating the buckling event for a force of around $2 \cdot 10^4$.

Figure 14. Selected snapshots of the deformation process during the pullout of an open-ended cylindrical shell (deformation not magnified), showing the buckling event for a force of about $2 \cdot 10^4$.
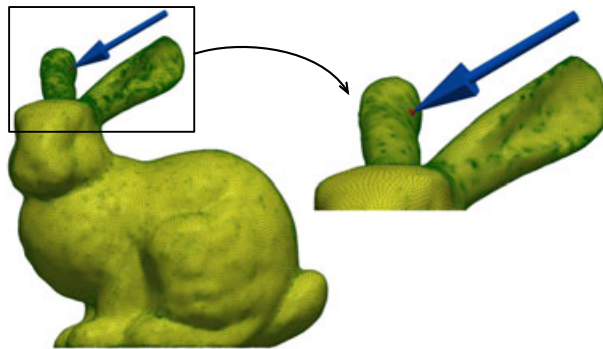

Figure 15. Control points of the bunny and depiction of the imposed displacement, which moves along the blue arrow.
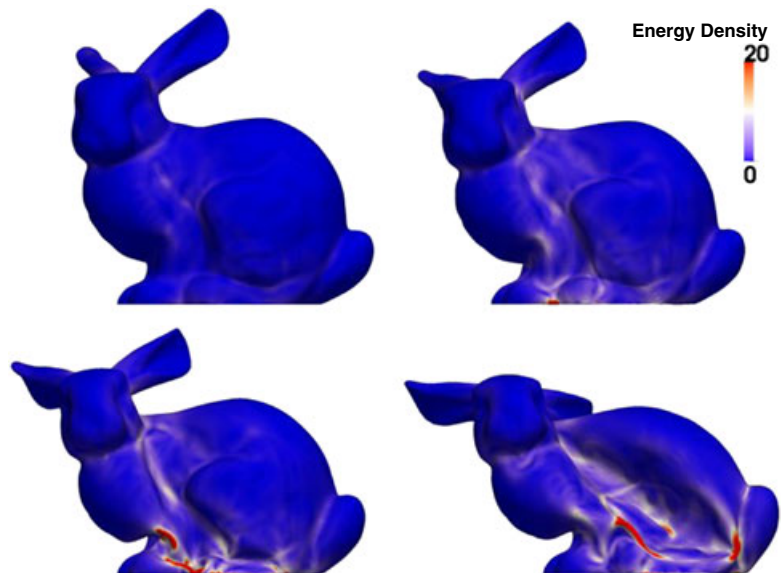

Figure 16. Elastic energy density at selected equilibrium configurations for the bunny thin-shell under a static incremental loading.

5.5. Indentation of a bunny

We illustrate now the ability of the method to deal with shells of complex geometry defined by a set of points alone, without the need for a global surface mesh. The only requirement on the nodes is that they sample sufficiently well the geometric features. This problem is treated with the full geometric nonlinearity. The height of the object is around 1.5, and the thickness of the thin-shell is $h = 0.005$. The material parameters are $E = 10^7$ and $v = 0.3$. Figure 15 shows a sketch of 54,867 control points sampling the Stanford bunny [73], which is deformed by an imposed displacement moving incrementally in the direction of the blue arrow.

Figure 16 shows four snapshots along the deformation process, experiencing a number of buckling events (see [71] for illustrative movies). The deformation is not magnified. The simulation proceeds robustly and exhibits very large deformations and localized creases with strain energy density concentrations, typical of the post-buckling response of thin-shells.

5.6. Connected pipes

We now illustrate the ability of the proposed method to deal with extremely complex topologies (Figure 17). The boundary curves of the bottom pipes are clamped, and the top boundary curves
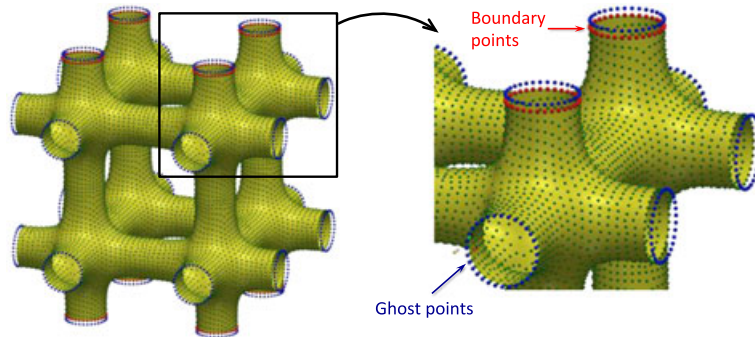

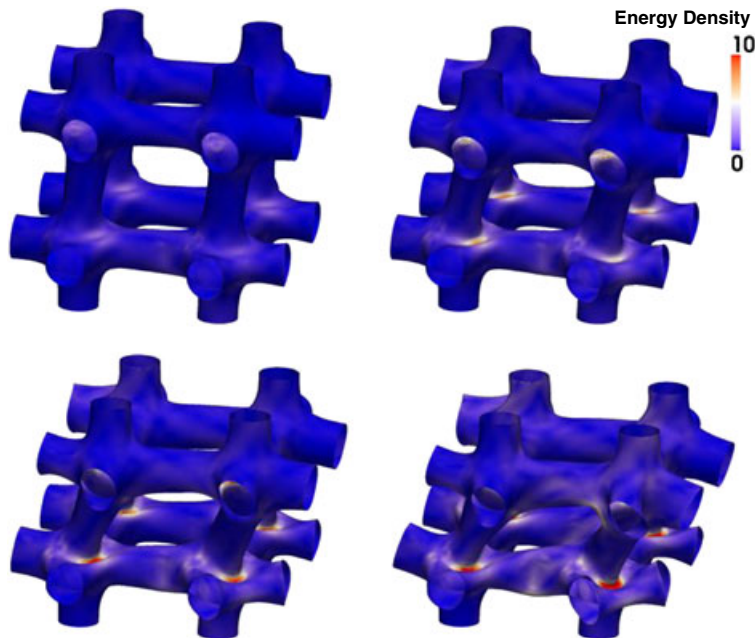Figure 17. Point-set for the example of the connected pipes.

Figure 18. Selected snapshots of the deformation process of the shell with complex topology. Nonmagnified deformation and color map of the strain energy density.

are incrementally displaced in the $(-1, -1, -1)$ direction. The lateral dimension of the system is 6.2, and the shell thickness is $h = 0.03$. Figure 17 illustrates the discretization with 14,176 control points. Figure 18 shows the energy density on the deformed configuration at four snapshots, without magnification of the displacements. Again, the shell undergoes several geometric instabilities and exhibits localized elastic deformations. We insist on the fact that this thin-shell cannot be studied with previous meshfree approaches because it does not admit a single parametric space.

6. CONCLUSIONS

We have extended the methodology proposed in [28] to build smooth numerical representations of d-dimensional point-set manifolds embedded in $\mathbb{R}^D$, which avoids a global parametrization or a mesh. The proposed method exploits modern NLDR techniques, such as Isomap and LLE, to find embeddings of low dimension of good geometric quality for large patches of surface defined by automatic partitioning of the set of points. A meshfree parametrization of the patches is then defined, with local *max-ent* approximants. The different patches can be glued together with a PU associated to the patches, which allows us to split the evaluation of functionals on the manifold in patch-by-patch calculations.

Although the method is applicable in higher dimensions, we have exercised it on the geometrically exact theory of Kirchhoff–Love thin-shells. Our work significantly extends the applicability of meshfree methods to thin-shell analysis. Previous methods were limited to very simple surfaces admitting a single parametric space. The proposed method is very robust and general and can deal very easily with shells of very complex geometry and topology. Furthermore, we have shown that it is very accurate and competitive with state-of-the-art mesh-based methods such as subdivision finite elements. Interestingly, although the local *max-ent* approximants are only linearly reproducing, we obtain excellent results for the fourth-order partial differential equations of thin-shells. We have observed the same behavior in the numerical approximation of a fourth-order phase-field model for biomembranes [74]. This suggests further mathematical analysis of the method [75]. The proposed method can be easily enhanced to account for internal connections or non-manifold shells [72, 76]. We are also working on a boundary representation avoiding ghost nodes, by describing the boundary curves by B-spline or NURBS curves [77].

The general methodology proposed here can be applied to many problems in science and engineering. We are currently exploiting it for the quantitative analysis of swimming strokes in micro-organisms, the compact model reduction of dynamical systems whose near-invariant manifold is nonlinear, or the automatic detection of meaningful collective variables in biomolecular simulations.

APPENDIX A: CLOSEST-POINT PROJECTION ONTO THE MANIFOLD

We describe here how to perform the closest-point projection of a point $x \in \mathbb{R}^D$ close to the manifold onto the manifold. To simplify the notation, we restrict ourselves to a given patch. If $x \in \mathcal{M}^D$, this can be interpreted as 'inverting' the parametrization to find a point $\xi \in \mathbb{R}^d$ such that $\varphi(\xi) = x$. For this purpose, we minimize the cost function $f(\xi) = (1/2)|\varphi(\xi) - x|^2$. We solve this nonlinear optimization problem with Newton's method

$$\xi^{i+1} = \xi^i - J^{-1}(\xi^i)\, r(\xi^i),$$

where $r(\xi) = \nabla f(\xi)$ is the gradient and $J(\xi) = Hf(\xi)$ is the Hessian of the cost function. The gradient of the cost function is given as

$$\nabla f(\xi) = D\varphi(\xi) \cdot (\varphi(\xi) - x),$$

where

$$D\varphi(\xi) = \sum_a \nabla p_a(\xi) \otimes P_a.$$

The Hessian of the cost function can be written as

$$Hf(\boldsymbol{\xi}) = D^2\boldsymbol{\varphi}(\boldsymbol{\xi}) \cdot (\boldsymbol{\varphi}(\boldsymbol{\xi}) - \boldsymbol{x}) + D\boldsymbol{\varphi}(\boldsymbol{\xi}) \otimes D\boldsymbol{\varphi}(\boldsymbol{\xi}),$$

where

$$D^2\boldsymbol{\varphi}(\boldsymbol{\xi}) = \sum_a Hp_a(\boldsymbol{\xi}) \otimes \boldsymbol{P}_a.$$

As a starting point for Newton's method, we select $\boldsymbol{\xi}^0 = \sum_a w_a^P(\boldsymbol{x}) \, \boldsymbol{\xi}_a$; here, w_a^P are the local linear weights from the kth nearest neighbors of $\boldsymbol{x}$ belonging to P. The local weights are computed in the spirit of LLE [50, 51]. This procedure is very robust and fast. We denote the solution of the minimization problem, the preimage of the closest point projection of a point in the high-dimensional space, by $\boldsymbol{\xi} = \boldsymbol{\varphi}^{-1}(\pi(\boldsymbol{x}))$.

APPENDIX B: CONTROL POINTS BY A LEAST-SQUARE FIT TO THE SAMPLED MANIFOLD

Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$, $K > N$ be a good sampling of the manifold $\mathcal{M} \in \mathbb{R}^D$. Let us assume we have a reasonably good set of control points, for example, lying on the manifold, which we use to define the embedding and the *max-ent* basis functions. We wish to find the control points $\boldsymbol{P}_a$, $a = 1, \ldots, N$ such that the numerical surface best fits the data in an L_2 sense. For this purpose, we minimize the cost function

$$g(\boldsymbol{P}) = \frac{1}{2} \sum_{i=1}^{K} |I(\boldsymbol{x}_i) - \boldsymbol{x}_i|^2,$$

where $\boldsymbol{P}$ denotes a vector with the coordinates of all the control points. The reconstruction operator can be rewritten as

$$I(\boldsymbol{x}) = \sum_{\kappa \in \mathcal{N}_{\boldsymbol{x}}} \psi_\kappa(\boldsymbol{x}) \sum_{a \in \mathcal{J}_\kappa} p_a(\boldsymbol{\xi}_\kappa(\boldsymbol{x})) \, \boldsymbol{P}_a = \boldsymbol{M}\boldsymbol{P},$$

where $\boldsymbol{M} \in \mathbb{R}^{K \times N}$ is a sparse matrix. The unique solution to this linear least-squares problem follows from the sparse linear system of equations $\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{P} = \boldsymbol{M}^{\mathrm{T}}\boldsymbol{X}$, where $\boldsymbol{X}$ is a vector collecting all the coordinates of the sampling points.

APPENDIX C: OUT-OF-BALANCE FORCES AND TANGENT STIFFNESS MATRIX

We provide here expressions for the gradient and the Hessian of the potential energy. With a view on the implementation, we resort to Voigt's notation for symmetric tensors. To keep the notation clean, depending on the context, we ignore the subscript indicating the patch number, that is, for instance, we denote $\boldsymbol{\varphi}_{0\kappa}$, a reference configuration of the middle surface in the κth patch simply by $\boldsymbol{\varphi}_0$. By the chain rule, the gradient of the discrete internal energy is

$$\frac{\partial \Pi_{\mathrm{int}}^h}{\partial \boldsymbol{P}_a} = \sum_{\kappa=1}^{L} \int_{\mathscr{A}_\kappa} \left[\left(\boldsymbol{n}^{\mathrm{T}} \frac{\partial \boldsymbol{\varepsilon}}{\partial \boldsymbol{P}_a} + \boldsymbol{m}^{\mathrm{T}} \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{P}_a} \right) \bar{j}_0 \right]_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0) \, \mathrm{d}\xi^1 \mathrm{d}\xi^2,$$

where the subindex κ means that the expression between the brackets is computed with the local parametrization of the κth patch.

The derivatives of the membrane and bending strain tensors with respect to the ath control point can be expressed in terms of the nonlinear membrane and bending strain–displacement matrices, $\boldsymbol{M}^a$ and $\boldsymbol{B}^a$, respectively, as

$$\frac{\partial \boldsymbol{\varepsilon}}{\partial \boldsymbol{P}_a} = \boldsymbol{M}^a, \quad \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{P}_a} = \boldsymbol{B}^a.$$

We express the strain–displacement matrices by introducing auxiliary vectors

$$M_{ij}^a = M_i^a \cdot e_j \text{ and } B_{ij}^a = B_i^a \cdot e_j,$$

which can be written as

$$M_\alpha^a = p_{a,\alpha}\, \boldsymbol{\varphi}_{,\alpha},$$
$$M_3^a = p_{a,2}\, \boldsymbol{\varphi}_{,1} + p_{a,1}\, \boldsymbol{\varphi}_{,2},$$
$$B_\alpha^a = -p_{a,\alpha\alpha}\, t$$
$$\quad + \bar{j}^{-1} \left[(\boldsymbol{\varphi}_{,\alpha\alpha} \times \boldsymbol{\varphi}_{,2}) p_{a,1} + (\boldsymbol{\varphi}_{,1} \times \boldsymbol{\varphi}_{,\alpha\alpha}) p_{a,2} \right]$$
$$\quad + \bar{j}^{-1} (t \cdot \boldsymbol{\varphi}_{,\alpha\alpha}) \left[(\boldsymbol{\varphi}_{,2} \times t) p_{a,1} + (t \times \boldsymbol{\varphi}_{,1}) p_{a,2} \right],$$
$$B_3^a = -2 p_{a,12}\, t$$
$$\quad + 2\, \bar{j}^{-1} \left[(\boldsymbol{\varphi}_{,12} \times \boldsymbol{\varphi}_{,2}) p_{a,1} + (\boldsymbol{\varphi}_{,1} \times \boldsymbol{\varphi}_{,12}) p_{a,2} \right]$$
$$\quad + 2\, \bar{j}^{-1} (t \cdot \boldsymbol{\varphi}_{,12}) \left[(\boldsymbol{\varphi}_{,2} \times t) p_{a,1} + (t \times \boldsymbol{\varphi}_{,1}) p_{a,2} \right].$$

By e_j, we denote the canonical basis vectors of $\mathbb{R}^3$. Note that repeated indices in the expressions for M_α^a and B_α^a do not imply summation.

For the tangent stiffness matrix, ignoring follower loads, we compute the second-order partial derivatives of the internal potential energy with respect to the control point positions as

$$\frac{\partial^2 \Pi_{\text{int}}^h}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b} = \frac{1}{2} \sum_{\kappa=1}^{L} \int_{\mathscr{A}_\kappa} \left[\frac{\partial^2 (\boldsymbol{\varepsilon} \cdot \boldsymbol{n} + \boldsymbol{\rho} \cdot \boldsymbol{m})}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b} \bar{j}_0 \right]_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0)\, \mathrm{d}\xi^1 \mathrm{d}\xi^2,$$

where

$$\frac{\partial^2 (\boldsymbol{\varepsilon} \cdot \boldsymbol{n} + \boldsymbol{\rho} \cdot \boldsymbol{m})}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b} = \boldsymbol{n}^{\text{T}} \frac{\partial^2 \boldsymbol{\varepsilon}}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b} + h\, \boldsymbol{M}^{a\text{T}} \boldsymbol{C} \boldsymbol{M}^b + \boldsymbol{m}^{\text{T}} \frac{\partial^2 \boldsymbol{\rho}}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b} + \frac{h^3}{12} \boldsymbol{B}^{a\text{T}} \boldsymbol{C} \boldsymbol{B}^b,$$

and $\boldsymbol{C}$ denotes the Voigt representation of $C^{\alpha\beta\gamma\delta}$. The second derivatives of the membrane and bending strain tensors can be computed as

$$\frac{\partial^2 \boldsymbol{\varepsilon}}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b} = \begin{pmatrix} p_{a,1}\, p_{b,1} \\ p_{a,2}\, p_{b,2} \\ p_{a,1}\, p_{b,2} + p_{a,2}\, p_{b,1} \end{pmatrix} \otimes \boldsymbol{I},$$

and

$$\boldsymbol{m}^{\text{T}} \frac{\partial^2 \boldsymbol{\rho}}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b} = -\bar{j}^{-1} \left[\frac{\partial \bar{j}}{\partial \boldsymbol{P}_a} \otimes \boldsymbol{f}_\rho^b + \boldsymbol{f}_\rho^a \otimes \frac{\partial \bar{j}}{\partial \boldsymbol{P}_b} \right] + \bar{j}^{-1} \left(\boldsymbol{m}^{\text{T}} H \boldsymbol{\varphi}\, t \right) \frac{\partial^2 \bar{j}}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b}$$
$$\quad - \bar{j}^{-1} \left(\boldsymbol{m} \cdot H p_a \right) [\boldsymbol{w}_b]_\times - \bar{j}^{-1} \left(\boldsymbol{m} \cdot H p_b \right) [\boldsymbol{w}_a]_\times^{\text{T}}$$
$$\quad - \bar{j}^{-1} (p_{a,2}\, p_{b,1} - p_{a,1}\, p_{b,2}) \left[(\boldsymbol{m}^{\text{T}} H \boldsymbol{\varphi})^{\text{T}} \right]_\times.$$

In the preceding equation, $\boldsymbol{f}_\rho^a = [\boldsymbol{m}^{\text{T}} \boldsymbol{B}^a]^{\text{T}}$, $[\boldsymbol{v}]_\times$ denotes the skew-symmetric matrix

$$[\boldsymbol{v}]_\times \overset{\text{def}}{=} \begin{pmatrix} 0 & -\boldsymbol{v} \cdot e_3 & \boldsymbol{v} \cdot e_2 \\ \boldsymbol{v} \cdot e_3 & 0 & -\boldsymbol{v} \cdot e_1 \\ -\boldsymbol{v} \cdot e_2 & \boldsymbol{v} \cdot e_1 & 0 \end{pmatrix},$$

and we introduce

$$\boldsymbol{w}_a = -p_{a,1}\, \boldsymbol{\varphi}_{,2} + p_{a,2}\, \boldsymbol{\varphi}_{,1}$$

and

$$H\varphi = \begin{pmatrix} \varphi_{,11} \cdot e_1 & \varphi_{,11} \cdot e_2 & \varphi_{,11} \cdot e_3 \\ \varphi_{,22} \cdot e_1 & \varphi_{,22} \cdot e_2 & \varphi_{,22} \cdot e_3 \\ 2\varphi_{,12} \cdot e_1 & 2\varphi_{,12} \cdot e_2 & 2\varphi_{,12} \cdot e_3 \end{pmatrix} \quad Hp_a = \begin{pmatrix} p_{a,11} \\ p_{a,22} \\ 2p_{a,12} \end{pmatrix},$$

to keep the notation compact.

The first-order and second-order partial derivatives of the Jacobian are

$$\frac{\partial \bar{\jmath}}{\partial P_a} = p_{a,1}(\varphi_{,2} \times t) + p_{a,2}(t \times \varphi_{,1})$$

and

$$\frac{\partial^2 \bar{\jmath}}{\partial P_a \partial P_b} = -\bar{\jmath}^{-1}\frac{\partial \bar{\jmath}}{\partial P_a} \otimes \frac{\partial \bar{\jmath}}{\partial P_b} + \bar{\jmath}^{-1}[w_a]_\times^{\mathrm{T}}[w_b]_\times + (-p_{a,1}\, p_{b,2} + p_{a,2}\, p_{b,1})[t]_\times.$$

To conclude, we give the explicit first and second derivates of the normal contracted with an auxiliary vector, $v \in \mathbb{R}^3$:

$$\frac{\partial t}{\partial P_a} = -\bar{\jmath}^{-1}\left(t \otimes \frac{\partial \bar{\jmath}}{\partial P_a} - [w_a]_\times\right),$$

$$\left(v^{\mathrm{T}}\frac{\partial t}{\partial P_a}\right)^{\mathrm{T}} = -\bar{\jmath}^{-1}\left[(v \cdot t)\frac{\partial \bar{\jmath}}{\partial P_a} - \left(p_{a,1}(\varphi_{,2} \times v) + p_{a,2}(v \times \varphi_{,1})\right)\right],$$

and

$$v^{\mathrm{T}}\frac{\partial^2 t}{\partial P_a \partial P_b} = -\bar{\jmath}^{-1}\left[(v \cdot t)\frac{\partial^2 \bar{\jmath}}{\partial P_a \partial P_b} + \frac{\partial \bar{\jmath}}{\partial P_a} \otimes \left(v^{\mathrm{T}}\frac{\partial t}{\partial P_b}\right)^{\mathrm{T}} + \left(v^{\mathrm{T}}\frac{\partial t}{\partial P_a}\right)^{\mathrm{T}} \otimes \frac{\partial \bar{\jmath}}{\partial P_b}\right]$$
$$+ \bar{\jmath}^{-1}(-p_{a,1}\, p_{b,2} + p_{a,2}\, p_{b,1})[v]_\times.$$

These expressions are needed, for example, to impose essential boundary conditions of rotation (Appendix D).

APPENDIX D: ESSENTIAL BOUNDARY CONDITIONS

We describe here the numerical constraints needed to impose the essential boundary conditions, for both displacements and rotations. We describe the variational formulation with Lagrange multipliers and the matrices needed in the augmented Lagrangian scheme.

Let us consider first the integral of a function f over the lateral boundary surface $\partial \mathscr{S}_0$ of a thin-shell object $\mathscr{S}_0$ ($\partial \mathscr{S}_0$ excludes the body boundary surfaces parallel to the middle surface). Assuming the function does not change through the thickness, we have

$$\int_{\partial \mathscr{S}_0} f \, dS_0 = \int_{\partial \Omega_0} f \left(\int_{-\frac{h}{2}}^{\frac{h}{2}} \frac{\left|\frac{\partial \Phi_0}{\partial \xi} \times \frac{\partial \Phi_0}{\partial t}\right|}{\left|\varphi_{0,t}\right|} d\xi \right) d\ell_0,$$

where t is a tangent coordinate along the boundary curve $\partial \mathscr{A}$. By introducing $\partial \Phi_0 / \partial t = \varphi_{0,t} + \xi\, t_{0,t}$, we obtain

$$\int_{-\frac{h}{2}}^{\frac{h}{2}} \frac{\left|t_0 \times \frac{\partial \Phi_0}{\partial t}\right|}{\left|\varphi_{0,t}\right|} d\xi = h \frac{\left|t_0 \times \varphi_{0,t}\right|}{\left|\varphi_{0,t}\right|}.$$

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

$$
\int_{\partial \mathscr{S}_0} f \, \mathrm{d}S_0 = \int_{\partial \Omega_0} h \, f \, \frac{|\boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t}|}{|\boldsymbol{\varphi}_{0,t}|} \, \mathrm{d}\ell_0
$$

$$
= \sum_{\kappa=1}^{L} \int_{\partial \mathscr{A}_\kappa} \left[h \left(f \circ \boldsymbol{\varphi}_0 \right) |\boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t}| \right]_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0) \, \mathrm{d}\ell_\xi .
$$

Here, subindex κ means that the expression between the brackets is computed with the local parametrization of the κth patch.

Displacement constraints on a curve

Let $\boldsymbol{\lambda}_u$ be the Lagrange multipliers field associated to the displacement constraints $\boldsymbol{\varphi} = \bar{\boldsymbol{\varphi}}$ on $\partial \Omega^u$. We discretize the Lagrange multipliers as $\boldsymbol{\lambda}_u = \sum_i N_i(\xi) \, \boldsymbol{\Lambda}_i^u$, where N_i are the standard piecewise linear basis functions defined from the boundary nodes. With the PU, the displacement constraints can be expressed variationally as

$$
0 = \int_{\Omega_0^u} h \, \boldsymbol{\lambda}_u \cdot (\boldsymbol{\varphi} - \bar{\boldsymbol{\varphi}}) \frac{|\boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t}|}{|\boldsymbol{\varphi}_{0,t}|} \, \mathrm{d}\ell_0 = \sum_{\kappa=1}^{L} \int_{\partial \mathscr{A}_\kappa^u} \left\{ h \, \boldsymbol{\lambda}_u \cdot (\boldsymbol{\varphi} - \bar{\boldsymbol{\varphi}}) \, |\boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t}| \right\}_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0) \, \mathrm{d}\ell_\xi ,
$$

where $\partial \mathscr{A}_\kappa^u = \boldsymbol{\varphi}_{0\kappa}^{-1}(\partial \Omega^u \cap \mathrm{supp}(\psi_\kappa))$, for all Lagrange multipliers. Recalling their discretization, we can write the constraint in matrix form as $\boldsymbol{C}_u(P) = \boldsymbol{0}$, where

$$
\boldsymbol{C}_i^u(P) = \sum_{\kappa=1}^{L} \int_{\partial \mathscr{A}_\kappa^u} \left\{ h \, N_i \, (\boldsymbol{\varphi} - \bar{\boldsymbol{\varphi}}) \, |\boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t}| \right\}_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0) \, \mathrm{d}\ell_\xi .
$$

These constraints are linear, with

$$
\frac{\partial \boldsymbol{C}_i^u}{\partial \boldsymbol{P}_a} = \left(\sum_{\kappa=1}^{L} \int_{\partial \mathscr{A}_\kappa^u} \left\{ h \, N_i \, p_a \, |\boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t}| \right\}_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0) \, \mathrm{d}\ell_\xi \right) \boldsymbol{I} ,
$$

and $\partial^2 \boldsymbol{C}_i^u / \partial \boldsymbol{P}_a \partial \boldsymbol{P}_b = 0$.

Rotation constraints on a curve

Let $\boldsymbol{\tau}_0 = \boldsymbol{\varphi}_{0,t} / |\boldsymbol{\varphi}_{0,t}|$ be a unit vector tangent to the boundary curve of the middle surface Ω_0 satisfying $\boldsymbol{t}_0 \cdot \boldsymbol{\tau}_0 = 0$. The rotation boundary conditions take the form $\boldsymbol{v}_0 \cdot \boldsymbol{t} = \bar{g}_\theta$ on $\partial \Omega^\theta$, where $\boldsymbol{v}_0 = \boldsymbol{t}_0 \times \boldsymbol{\tau}_0$ is the outward tangent vector to the boundary curve. With λ_θ denoting the Lagrange multiplier field associated with this constraint, the variational statement of the constraint is

$$
0 = \int_{\partial \Omega_0^\theta} h \, \lambda_\theta \, (\boldsymbol{v}_0 \cdot \boldsymbol{t} - \bar{g}_\theta) \frac{|\boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t}|}{|\boldsymbol{\varphi}_{0,t}|} \, \mathrm{d}\ell_0 = \sum_{\kappa=1}^{L} \int_{\partial \mathscr{A}_\kappa^\theta} \left\{ h \, \lambda_\theta \, (\boldsymbol{v}_0 \cdot \boldsymbol{t} - \bar{g}_\theta) \, |\boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t}| \right\}_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0) \, \mathrm{d}\ell_\xi ,
$$

for all Lagrange multipliers. With the discrete representation of the Lagrange multipliers $\lambda_\theta = \sum_i N_i(\xi) \, \Lambda_i^\theta$, the discrete constraints become

$$
0 = C_i^\theta(P) = \sum_{\kappa=1}^{L} \int_{\partial \mathscr{A}_\kappa^\theta} \left\{ h \, N_i \, (\boldsymbol{v}_0 \cdot \boldsymbol{t} - \bar{g}_\theta) \, |\boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t}| \right\}_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0) \, \mathrm{d}\ell_\xi .
$$

These constraints are nonlinear. For the augmented Lagrangian implementation, we need

$$\frac{\partial C_i^\theta}{\partial \boldsymbol{P}_a} = \sum_{\kappa=1}^{L} \int_{\partial \mathscr{A}_\kappa^\theta} \left\{ h\ N_i\ \left(\boldsymbol{v}_0^{\mathrm{T}} \frac{\partial \boldsymbol{t}}{\partial \boldsymbol{P}_a} \right)^{\mathrm{T}} \left| \boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t} \right| \right\}_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0)\, \mathrm{d}\ell_\xi$$

and

$$\frac{\partial^2 C_i^\theta}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b} = \sum_{\kappa=1}^{L} \int_{\partial \mathscr{A}_\kappa^\theta} \left\{ h\ N_i\ \left(\boldsymbol{v}_0^{\mathrm{T}} \frac{\partial^2 \boldsymbol{t}}{\partial \boldsymbol{P}_a \partial \boldsymbol{P}_b} \right) \left| \boldsymbol{t}_0 \times \boldsymbol{\varphi}_{0,t} \right| \right\}_\kappa (\psi_\kappa \circ \boldsymbol{\varphi}_0)\, \mathrm{d}\ell_\xi.$$

Imposed displacement on a point

Suppose we want to constrain a point $\boldsymbol{x}_0 \in \Omega_0$ to be at location $\boldsymbol{x}_1$ in the deformed configuration. By defining $\boldsymbol{\xi}_0 = \boldsymbol{\varphi}_0^{-1}(\boldsymbol{x}_0)$, the preimage of this point by the reference configuration of the middle surface, we can write this constraint as

$$\boldsymbol{0} = \boldsymbol{C}^p(P) = \boldsymbol{\varphi}(\boldsymbol{\xi}_0) - \boldsymbol{x}_1.$$

APPENDIX E: AUGMENTED LAGRANGIAN SOLUTION METHOD

The equilibrium solutions of the nonlinear constrained minimization problem described in Section 4.4 are stationary points of the Lagrangian

$$\mathcal{L}(P, \boldsymbol{\Lambda}) = \Pi^h(P) - \boldsymbol{\Lambda}^{\mathrm{T}} \boldsymbol{C}(P),$$

where $\boldsymbol{C}$ collects all the discrete constraints of the previous sections and $\boldsymbol{\Lambda}$ collects the corresponding Lagrange multipliers (reaction forces and torques). The Kuhn–Tucker optimality conditions, $\partial_P \mathcal{L} = 0$, $\partial_{\boldsymbol{\Lambda}} \mathcal{L} = 0$, may be solved with Newton's method, yet this approach may lead to unstable equilibria and avoid physically relevant buckled stable solutions.

A robust strategy that guarantees stable equilibria is based on the augmented Lagrangian method, which combines the standard Lagrangian with penalties. This method retains the exactness of the Lagrange multipliers method and the minimization principle of penalty methods. The minimization is performed iteratively on the control points only for frozen Lagrange multipliers, which are updated explicitly (see [78, 79] for further details). The augmented Lagrangian is

$$\mathcal{L}_A(P, \boldsymbol{\Lambda}) = \Pi^h(P) - \boldsymbol{\Lambda}^{\mathrm{T}} \boldsymbol{C}(P) + \frac{1}{2\mu} \boldsymbol{C}(P)^{\mathrm{T}} \boldsymbol{C}(P),$$

where μ is the penalty parameter. We solve the problem in two stages. First, we find an approximate minimizer with a slow, robust method and a coarse tolerance. For this, in the augmented Lagrangian inner minimization loop, we adopt the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm. Then, the minimizer is refined by resorting to Newton's method with line search.

ACKNOWLEDGEMENTS

REFERENCES

1. 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.
2. Bucalem M, Bathe J. Higher-order MITC general shell elements. *International Journal for Numerical Methods in Engineering* 1993; **36**(21):3729–3754.

3. 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.

4. 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.

5. 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**(12):2039–2072.

6. 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.

7. 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.

8. Cottrell J, Hughes T, Bazilevs Y. *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, Ltd.: Hoboken, NJ, 2009.

9. Cottrell J, Reali A, Bazilevs Y, Hughes T. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(41–43):5257–5296.

10. Kiendl J, Bletzinger KU, Linhard J, Wüchner R. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(49–52):3902–3914.

11. Bazilevs Y, Calo V, Cottrell J, Evans J, Hughes T, Lipton S, Scott M, Sederberg T. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(5–8):229–263.

12. 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.

13. 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.

14. Wells G, Dung N. Geometrically nonlinear formulation for thin shells without rotation degrees of freedom. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**(35–36):3370–3380.

15. 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.

16. Noels L. A discontinuous Galerkin formulation of non-linear Kirchhoff–Love shells. *International Journal for Numerical Methods in Engineering* 2009; **78**(3):296–323.

17. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. In *SIGGRAPH '92: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*. ACM: New York, NY, 1992; 71–78.

18. Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva C. Point set surfaces. In *VIS '01: Proceedings of the Conference on Visualization '01*. IEEE Computer Society: Washington, DC, 2001; 21–28.

19. Pauly M. Point primitives for interactive modeling and processing of 3D geometry. *PhD Thesis*, Federal Institute of Technology (ETH) of Zurich, 2003.

20. Levin D. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, Brunnett G, Hamann B, Mueller H, Linsen L (eds). Springer-Verlag: Heidelberg, 2003; 37–49.

21. Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H. Multi-level partition of unity implicits. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)* 2003; **22**:463–470.

22. Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva C. Computing and rendering point set surfaces. *Transactions on Visualization and Computer Graphics* 2003; **9**(1):3–15.

23. Amenta N, Kil Y. Defining point-set surfaces. *ACM Transactions on Graphics* 2004; **23**(3):264–270.

24. Alexa M, Gross M, Pauly M, Pfister H, Stamminger M, Zwicker M. Point-based computer graphics. *SIGGRAPH* 2004 Course Notes 2004.

25. 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.

26. 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**(2):151–172.

27. 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**(5):524–548.

28. Millán D, Rosolen A, Arroyo M. Thin shell analysis from scattered points with maximum-entropy approximants. *International Journal for Numerical Methods in Engineering* 2011; **85**(6):723–751.

29. 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.

30. 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**(13):2167–2202.

31. Pearson K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine* 1901; **2**(6):559–572.

32. Hotelling H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 1933; **24**(7):498–520.

33. Karhunen K. Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientiarum Fennicae* 1946; **34**:1–7.

34. Loève M. *Probability Theory.* University series in higher mathematics, Van Nostrand: New Jersey, 1955.

35. Lorenz E. Empirical orthogonal functions and statistical weather prediction. *Statistical Forecasting Project, Scientific Report 1*, MIT, Department of Meteorology, Cambridge, MA, 1956.
36. Lumley JL. The structure of inhomogeneous turbulent flows. In *Atmospheric Turbulence and Radio Propagation*, Yaglom AM, Tatarski VI (eds). Nauka: Moscow, 1967; 166–178.
37. Jolliffe IT. *Principal Component Analysis*, 2nd edn, Springer Series in Statistics, Springer: New York, NY, 2002.
38. Jain AK, Duin R, Mao J. Statistical pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2000; **22**(1):4–37.
39. Zhang Z, Zha H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal of Scientific Computing* 2005; **26**(1):313–338.
40. Lall S, Krysl P, Marsden J. Structure-preserving model reduction for mechanical systems. *Physica D* 2003; **184**:304–318.
41. Niroomandi S, Alfaro I, Cueto E, Chinesta F. Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering* 2010; **81**(9):1180–1206.
42. Torgerson W. Multidimensional scaling: I. Theory and method. *Psychometrika* 1952; **17**:401–419.
43. Cox TF, Cox MAA. *Multidimensional Scaling*, 2nd edn. Chapman & Hall: Boca Raton, FL, 2001.
44. Lee J, Verleysen M. *Nonlinear Dimensionality Reduction*, Information Science and Statistics. Springer: New York, NY, 2007.
45. Sammon JW. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers* 1969; **18**:401–409.
46. Hérault J, Oliva A, Guérin-Dugué A. Scene categorisation by curvilinear component analysis of low frequency spectra. *Proceedings of the 5th European Symposium on Artificial Neural Networks*, Bruges, Belgium, 1997; 91–96.
47. Hérault J, Jausions-Picaud C, Guérin-Dugué A. Curvilinear component analysis for high-dimensional data representation: I. Theoretical aspects and practical use in the presence of noise. In *Engineering Applications of Bio-Inspired Artificial Neural Networks*, Vol. 1607, Mira J, Sánchez-Andrés J (eds), Lecture Notes in Computer Science. Springer: Berlin/Heidelberg, 1999; 625–634.
48. van der Maaten LJP, Postma EO, van den Herik HJ. Dimensionality reduction: a comparative review. *Technical Report TiCC-TR 2009-005*, Tilburg University, 2009.
49. Tenenbaum J, de Silva V, Langford J. A global geometric framework for nonlinear dimensionality reduction. *Science* 2000; **290**(5500):2319–2323.
50. Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. *Science* 2000; **290**(5500):2323–2326.
51. Roweis S, Saul L. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research* 2003; **4**:119–155.
52. de Silva V, Tenenbaum J. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems*, Vol. 15. MIT Press: Cambridge, MA, 2003; 705–712.
53. Choi H, Choi S. Robust kernel Isomap. *Pattern Recognition* 2007; **40**(3):853–862.
54. Donoho D, Grimes C. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences* 2003; **100**(10):5591–5596.
55. Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 2003; **15**:1373–1396.
56. Chang H, Yeung DY. Robust locally linear embedding. *Pattern Recognition* 2006; **39**(6):1053–1065.
57. Zhang Z, Wang J. MLLE: modified locally linear embedding using multiple weights. In *Advances in Neural Information Processing Systems*, Vol. 19, Schölkopf B, Platt J, Hoffman T (eds). MIT Press: Cambridge, MA, 2007; 1593–1600.
58. Gámez AJ, Zhou CS, Timmermann A, Kurths J. Nonlinear dimensionality reduction in climate data. *Nonlinear Processes in Geophysics* 2004; **11**:393–398.
59. Das P, Moll M, Stamati H, Kavraki L, Clementi C. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences* 2006; **103**(26):9885–9890.
60. Brown WM, Martin S, Pollock SN, Coutsias EA, Watson JP. Algorithmic dimensionality reduction for molecular structure analysis. *The Journal of Chemical Physics* 2008; **129**(6):064118.
61. Vanderplas J, Connolly A. Reducing the dimensionality of data: locally linear embedding of Sloan galaxy spectra. *The Astronomical Journal* 2009; **138**(5):1365–1379.
62. do Carmo MP. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.
63. de Ridder D, Duin R. Locally linear embedding for classification. *Technical Report PH-2002-0*, Department of Imaging Science & Technology, Delft University of Technology, Delft, 2002.
64. Grassberger P, Procaccia I. Measuring the strangeness of strange attractors. *Physica D* 1983; **9**(1–2):189–208.
65. Karypis G, Kumar V. METIS: Unstructured graph partitioning and sparse matrix ordering system, version 4.0, 1998. Available from: http://www.cs.umn.edu/~metis [Accessed on 2011].
66. Sha F, Saul L. Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proceedings of the 22nd International Conference on Machine Learning,* ICML '05. ACM: New York, NY, 2005; 784–791.
67. Mount D, Arya S. ANN: a library for approximate nearest neighbor searching, version 1.1, 2010. Available from: http://www.cs.umd.edu/~mount/ANN [Accessed on 2011].
68. 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.

69. Ciarlet PG. *Mathematical Elasticity: Theory of Shells*, Vol. III: North-Holland, 2000.
70. Sze KY, Liu XH, Lo SH. Popular benchmark problems for geometric nonlinear analysis of shells. *Journal Finite Elements in Analysis and Design* 2004; **40**:1551–1569.
71. Millán D. Geometrically nonlinear meshfree thin-shell analysis in the context of Kirchhoff–Love theory, 2011. Available from: http://www.youtube.com/RDanielMillan [Accessed on 2011].
72. 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.
73. Turk G, Levoy M. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques,* SIGGRAPH '94. ACM: New York, NY, 1994; 311–318.
74. Rosolen A, Millán D, Arroyo M. Second order convex maximum entropy approximants with applications to high order PDE. *International Journal for Numerical Methods in Engineering,* under revision*, 2012.
75. Bompadre A, Perotti L, Cyron C, Ortiz M. Convergent meshfree approximation schemes of arbitrary order and smoothness. *Computer Methods in Applied Mechanics and Engineering* 2012:221–222:83–183.
76. Cirak F, Long Q. Subdivision shells with exact boundary control and non-manifold geometry. *International Journal for Numerical Methods in Engineering* 2011; **88**(9):897–923.
77. Rosolen A, Arroyo M. Blending isogeometric analysis and maximum entropy meshfree approximants. *In preparation*, 2012.
78. Nocedal J, Wright S. *Numerical Optimization*. Springer: New York, 1999.
79. Conn A, Gould N, Toint P. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis* 1991; **28**(2):545–572.