Isometric embedding of curvilinear meshes defined on Riemannian metric spaces

Philip Claude Caplan, Robert Haimes and Xevi Roca

Abstract An algorithm for isometrically embedding curvilinear meshes defined on Riemannian metric spaces into Euclidean spaces of sufficiently high dimension is presented. The method is derived from the Landmark-Isomap algorithm and a previous method for embedding straight-sided meshes. The former is used to decrease the computational complexity of the embedding problem, notably the dense shortest-path problem used to estimate geodesic lengths across the mesh domain as well as the dense eigenvalue decomposition needed to compute the codimension coordinates. A method for defining curvilinear meshes from straight-sided ones in a dimension-independent manner is also discussed. Examples in two- and three-dimensions for both analytic embeddings and analytic metric fields are used to evaluate the method.

1 Introduction

High-order discretisations of partial differential equations demand a curvilinear representation of the mesh geometry [1]. Within an adaptive framework for numerical simulations, the curvilinear metric-conforming mesh can either be generated natively or a posteriori from a straight-sided metric-conforming mesh. Recent interest in using isometric embeddings [3, 5–7, 14], founded on the Nash embedding theorem, offer the potential to generate curvilinear metric-conforming meshes in a native manner [4] by projecting high-order mesh vertices to the embedded mesh during the mesh generation stage. This work answers an important question related to the ability to natively generate curvilinear metric-conforming meshes: how to embed

Philip Claude Caplan and Robert Haimes
Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge 02139, USA
Xevi Roca
Computer Applications in Science and Engineering, Barcelona Supercomputing Center, 08034 Barcelona, Spain

a curvilinear mesh equipped with a Riemannian metric into a Euclidean space of sufficiently high dimension? In particular, an algorithm for embedding curvilinear meshes given an initial linear mesh and spatially continuous definition of a metric field is introduced. The algorithm is then evaluated by studying the effect of the geometric mesh order and ambient dimension of the Euclidean space on the isometry of the embedded curvilinear mesh. Examples drawn from analytic embeddings and analytic metrics in $2d$ and $3d$ are used to illustrate the method.

2 Methodology

This section describes the methodology used to embed curvilinear meshes, upon which a spatially continuous field of metric tensors is assumed. Before doing so, consider how a curvilinear mesh can be constructed from a linear one in a dimension-independent manner.

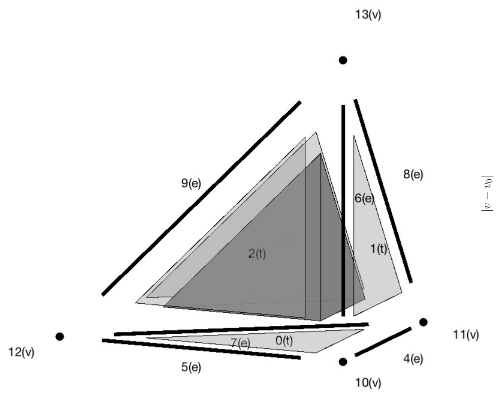
2.1 Construction of curvilinear meshes

In essence, this section describes how any high-order continuous Galerkin field can be defined from a straight-sided mesh but it is important to illustrate the mechanics of this procedure since, to our knowledge, this has not been covered in the literature for higher-dimensional ($d > 3$) simplicial meshes.

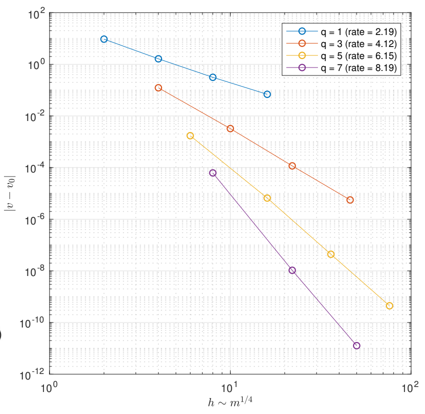
Given a straight-sided d-simplicial mesh $\mathscr{T} \in \mathbb{R}^d$, the first step consists of identifying all j-simplices $0 \leq j \leq d$ in $\mathscr{T}$ along with the set of d-simplices and local facet labels from which each facet is constructed. As an example, a tetrahedral mesh is first decomposed into its vertices, edges, triangles and tetrahedra along with the parent tetrahedra and local indices within each parent tetrahedron that these vertices, edges and triangles are constructed. Note the identification of the 0-simplices and d-simplices is trivial. All other j-simplices are identified by hashing a string representing the sorted integers describing the facet. Another option might be to represent each facet as $\sum_{i=0}^{|f|} f_i n_v^i$ where n_v is the number of vertices in $\mathscr{T}$, however, this risks overflowing the integer representation; as such, the string representation is preferred when hashing the facets upon construction. The labelling of this facet decomposition is illustrated in the sketch of Fig. 1a.

The next step consists of placing equispaced high-order nodes along the *interior* of each j-simplex $j \leq d$. Each j-simplex is then fully defined by identifying the local indices of the d-simplex lattice coordinates corresponding to every vertex. Should a boundary representation of the geometry be provided, vertices can be projected to the appropriate entities.

For visualization purposes, the list of triangles and edges can be obtained by extracting the bounding $2-$ and $1-$simplices of the d-triangulation obtained with,



(a) Facet decomposition of the reference tetrahedron. Parentheses indicate either a vertex (v), edge (e) or triangle (t). The only facet not visible is the triangle 1(t)).



(b) Convergence to the analytic volume of a curved domain for a $4d$ curvilinear mesh with various geometric mesh orders.

Fig. 1: Demonstration of facet decomposition in $3d$ and verification of curvilinear construction process in $4d$.

say, a Delaunay triangulation of the lattice coordinates of the curvilinear reference simplex.

As a demonstration of the d-dimensional curvilinear construction process, an initially straight-sided Kuhn-Freudenthal triangulation in $[0,1]^d$ [13] is augmented to various curvilinear mesh orders. The first two coordinates of the vertices in this curvilinear mesh are then mapped to polar coordinates, $0 \leq r \leq 1$, $0 \leq \theta \leq \pi$ with the remaining dimensions arbitrarily stretched to $l_d = d$. The volume of the mesh is computed by integrating over the elements, using high-order Lagrange basis functions from Burkardt [2] along with Grundmann-Moeller quadrature rules [9]. As seen in Fig. 1b, the volume of a four-dimensional mesh asymptotically approaches the analytic one of $\frac{1}{2}\pi \prod_{i=2}^{d} l_i$ with mesh size h at a rate of h^{q+1} where $h \sim m^{1/d}$ with m the total number of vertices. Standard curved meshing is focused on approximating two-dimensional and three-dimensional geometries. Herein we expose one of the first, if not the first, discussions on setting up curvilinear meshes for $4d$ problems. It is worth noting that the focus is on studying the influence of the metric field (and not the geometry) in the embedding of curvilinear meshes. As such, all geometries studied in the remainder of the work are straight-sided.

2.2 Embedding algorithm

Here, the mechanics of the algorithm used to embed a curvilinear mesh are presented. The only assumption is that a metric field is defined (or can be evaluated) at the high-order vertex locations. Recent work [4] employs a variant of the Isomap algorithm of Tenenbaum et al. [10] to embed straight-sided meshes along with a Riemannian metric to a higher-dimensional Euclidean space. This approach is advantageous because it reduces the embedding procedure to an eigenvalue decomposition of a centered distance matrix, however, the computational cost of directly applying this approach is restrictive due to the dense eigenvalue problem needed to compute the codimension coordinates, requiring an order of $\mathscr{O}(n^3)$ operations and $\mathscr{O}(n^2)$ where n is the number of vertices in the mesh. Further, the $\mathscr{O}(n^3)$ running time of Dijkstra's algorithm used to estimate the shortest distance between all pairs of vertices in the mesh is prohibitive. Even in the straight-sided case, this cost becomes intractable for large meshes which is further exacerbated by the presence of curvilinear vertices.

Additionally, it is unclear how curvilinear vertices are "connected" in the genereal case to ensure that the result of the shortest path algorithm is valid. To see this, consider the internal vertex of a $q = 3$ triangle. A triangulation of the reference simplex vertices may yield the connections between all curvilinear vertices to construct the graph for Dijkstra's algorithm but it biases the shortest paths as being the connections within the reference simplex.

For these two reasons, a sparse approach is sought which is equal in cost to that of embedding the straight-sided mesh. The methodology is based on the Landmark-Isomap algorithm of de Silva [8]. Of course, this sparse approach can be used to obtain embeddings of straight-sided meshes at a lower computational cost but that is left for future endeavours since the selection of landmarks is less trivial in that case. Other sparse approaches exist to achieve an isometric embedding at a lower computational cost, such as Locally Linear Embedding [17] or Semidefinite Embedding [19], however, Landmark-Isomap is preferred here due to its isometry preserving properties and its ability to work from the embedding of the straight-sided mesh.

Here, the landmarks are chosen to be the vertices of the straight-sided mesh. As such, the first step consists of embedding the vertices of the straight-sided mesh. The full computational approach to embed a curvilinear mesh is described in Algorithm 1. The number of vertices in the straight sided (q_1) mesh is denoted by n whereas the number of vertices in the high-order curvilinear (q) mesh is denoted by m.

The algorithm up to line 19 essentially computes the embedding of the vertices of the straight-sided mesh. The remainder of the algorithm computes the distance from each curvilinear vertex to each landmark vertex (line 30) which is stored in the matrix $\mathbf{R}$. This is done by first identifying the set of landmark vertices ($\mathscr{N}$) in the set of elements ($\mathbf{K}$) referencing the i-th vertex. The distance from this vertex to all local landmarks in $\mathscr{N}$ is then computed using the provided metric field and stored in $\mathbf{p}$. To estimate the distance to all remaining landmark vertices, the local landmark

```
function $\mathscr{T}_d^N$ = embed ($\mathscr{T}_d, \mathbf{m}, N_0$);
Input  : $\mathscr{T}_d, \mathbf{m}, N_0$
Output: $\mathscr{T}_d^N$
$E \leftarrow$ getEdges($\mathscr{T}_d$);
for $e \in E$ do
    $\ell_e =$ metricLength($e, \mathbf{m}$);
    $l_e =$ euclideanLength($e$);
    $f_e = (\ell_e / l_e)^2$;
end
$f_{\max} \leftarrow \max(\{f_e\}) \forall e \in E$;
for $e \in E$ do
    $\mathbf{G}(e_0, e_1) \leftarrow \sqrt{f_{\max}\ell_e^2 - l_e^2}$ ; /* setup the graph adjacency matrix */
end
$\mathbf{d} =$ dijkstra($\mathbf{G}$) ; /* geodesic length between linear vertices */
$\mu \leftarrow$ mean($\mathbf{d}$) ;                          /* row-wise mean of $\mathbf{d}$ */
$\mathbf{D} = \mathbf{d}^2$ ;                             /* squared geodesic lengths */
$\mathbf{B} = -\frac{1}{2}\mathbf{JDJ}$ ;          /* where the centering matrix $\mathbf{J} = \mathbf{I} - \mathbf{11}^t/n$ */
$(\mathbf{Q}, \Lambda) =$ eig($\mathbf{B}$) ;                    /* eigendecomposition of $\mathbf{B}$ */
$\mathbf{V} = \mathbf{Q}_{N_0}$;
$\mathbf{L} = \Lambda_{N_0}$ ;     /* save $N_0$ largest eigenvalues and eigenvectors */
$\mathbf{R}_{n,m} = \mathbf{0}$ ;   /* initial pairwise distance matrix to landmarks */
for $i = 1, \ldots, m$ do
    $\mathbf{K} \leftarrow$ elementsWithVertex($i$); /* any $d$-simplex with vertex $i$ */
    $\mathscr{N} \leftarrow$ allLinearVertices($\mathbf{K}$);
    $\mathbf{p} \leftarrow \mathbf{0}$ ;        /* initial distance from $i$ to members of $\mathscr{N}$ */
    for $j \in \mathscr{N}$ do
        $\ell_{i,j} =$ metricLength($\{i, j\}, \mathbf{m}$) ;  /* distance between $i$ and $j$ */
        $l_{i,j} =$ euclideanLength($\{i, j\}$);
        $\mathbf{p}_j = \sqrt{f_{\max}\ell_{i,j}^2 - l_{i,j}^2}$;
    end
    for $j = 1, \ldots, n$ do
        $l = \arg\min_{\mathbf{p}} ||\mathbf{p} + \mathbf{d}(\mathbf{p}, j)||$ ; /* closest landmark to landmark $j$ */
        $\mathbf{R}(j, i) = ||\mathbf{p}_l + \mathbf{d}(\mathbf{p}_l, j)||$;
    end
    $\mathbf{L}^{\#} \leftarrow \mathbf{0}$ ;                    /* apply landmark-Isomap to $\mathbf{D}_{n,m}$ */
    for $i = 1, \ldots, N_0,\ j = 1, \ldots, n$ do
        $\mathbf{L}^{\#}(i, j) = \mathbf{V}(j, i) / \sqrt{\mathbf{L}(i, i)}$;
    end
    for $i = 1, \ldots, m$ do
        $\mathbf{b} = -\frac{1}{2}\left(\mathbf{R}(:, i)^2 - \mu\right)$;
        $\mathbf{u}_0 = \mathbf{L}^{\#}\mathbf{b}$ ;                        /* codimension coordinates */
        $\mathbf{u}_i = [\mathbf{x}_i, \mathbf{u}_0 / \sqrt{f_{\max}}]$ ;   /* embedding coordinates by lifting */
    end
end
```

Algorithm 1: Embedding algorithm

vertex l corresponding to the minimizer of $||\mathbf{p}+\mathbf{d}(\mathbf{p},j)||$ is used as the distance from vertex i to landmark j. Lines 33–39 are referred to as the distance-based triangulation portion of the landmark-Isomap algorithm [8]. The embedding coordinates computed in line 40 are modified similar to previous efforts [4] to achieve a one-to-one mapping between the original mesh and the embedded one. Note the same factor $f_{\max}$ is used (computed from the original distance matrix) to modify the local distances between the curvilinear vertices and the landmarks (line 27) since these are shorter than any local distance between landmarks.

As in previous work [4], the algorithm is sensitive to the codimension of the ambient Euclidean space (N_0) which corresponds to the number of eigenvalues and eigenvectors retained from the decomposition of the centered disparity matrix (line 16). The next section studies the effect of N_0 on the isometry achieved by the algorithm.

It is worth mentioning that the current work simply focuses on developing an algorithm to isometrically embed curvilinear meshes with a metric field into a Euclidean space. The current focus is not on the validity of the embedded mesh but is the subject of future work.

3 Numerical studies

Here the developed algorithm is applied using various analytic metric fields with different geometric orders of the mesh (q) and different embedding dimensions. The procedure used to evaluate the embedding approach is outlined below:

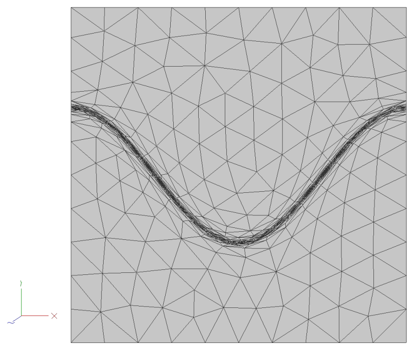
1. Obtain a straight-sided metric-conforming mesh using the software `fefloa` [15, 16] which is based on a unique approach to perform a variety of local mesh operations to achieve a straight-sided metric-conforming mesh,
2. Construct the curvilinear mesh from the straight-sided one using the method of Section 2.1,
3. Sweep over embedding dimensions:
   - Embed the curvilinear mesh using the algorithm of Section 2.2 into a Euclidean space of dimension $N = d + N_0$,
   - Compute the edge lengths of the embedded mesh. A Lagrange basis for the high-order basis functions is used to compute these lengths. The edge lengths are reported as a histogram in the following section. Since the mesh produced by `fefloa` is metric-conforming, these edges should be of unit length in the embedding space.

3.1 Analytic embeddings

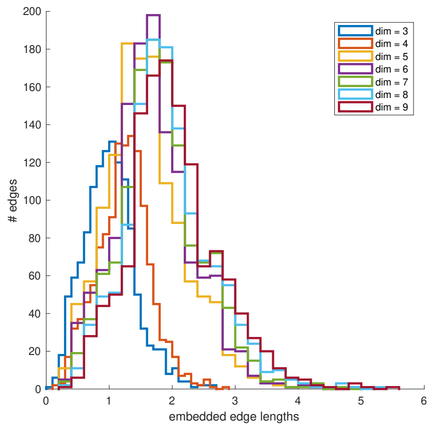
The first case studied is defined by the embedding into a Euclidean space of unit codimension:

$$\mathbf{u}(\mathbf{x}) = [\mathbf{x}, 5\tanh(10y - 20\cos(\pi x/5) - 50)], \quad \mathbf{x} = [0,10]^2. \tag{1}$$

The metric is then $\mathbf{m}(\mathbf{x}) = \nabla\mathbf{u}\cdot\nabla\mathbf{u}$. The linear mesh produced by `fefloa` is shown in Fig. 2a and the edge length histograms produced by embedding this straight-sided mesh into Euclidean spaces of various dimensions is shown in Fig. 2b. The best isometry is achieved for a unit codimensional space which is expected since the analytic embedding is defined in three-dimensional space. The edge lengths tend to increase when the dimension of the ambient Euclidean space is increased. The sacrifice in *local* edge length isometry is due to the fact that longer geodesics are better matched by the greater number of eigenvalues taken in the decomposition of the centered disparity matrix ($\mathbf{B}$ in Algorithm 1).



(a) Metric conforming straight-sided mesh for the embedding of Eq. 1.

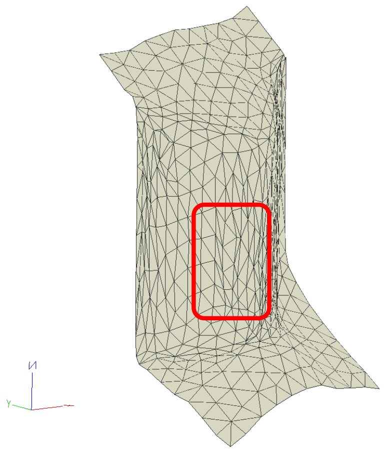


(b) Edge length histogram for edges produced by embedding the straight-sided mesh conforming to the embedding of Eq. 1 into Euclidean spaces of various dimensions.

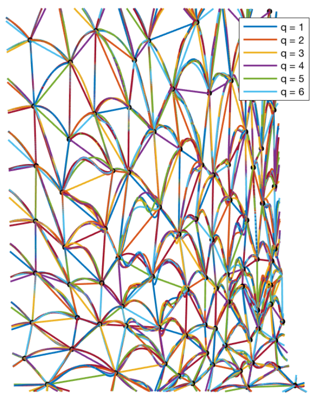
Fig. 2: Edge length histogram for edges produced by embedding the straight-sided mesh conforming to the embedding of Eq. 1 into Euclidean spaces of various dimensions.

The embedded linear mesh and curvilinear mesh edges are shown in Figs. 3a and 3b, respectively. As indicated by the better isometry induced by the greater number of unit edge lengths of the histogram in Fig. 4a, it that seems increasing the geometric order of the meshes improves the representation of the metric in three-dimensional space. However, observe the oscillatory behaviour of these edges in

Fig. 3b. Though better isometry seems to be achieved, the resulting embedding may not produce valid elements in the embedding space. Ongoing work consists of evaluating the validity of the embedded mesh with a variant of the algorithm of Johnen et al. in higher-dimensional spaces [11, 12]. Though the best isometry is achieved in three-dimensional space, the edge length histogram produced across various curvilinear mesh orders is given for a five-dimensional embedding in Fig. 4b. As stated earlier, the local edge lengths are longer than they should be, likely a result of better representing the longer geodesics across the domain with the greater number of eigenvalues retained.



(a) Embedded straight-sided mesh produced with the metric field derived from Eq. 1.



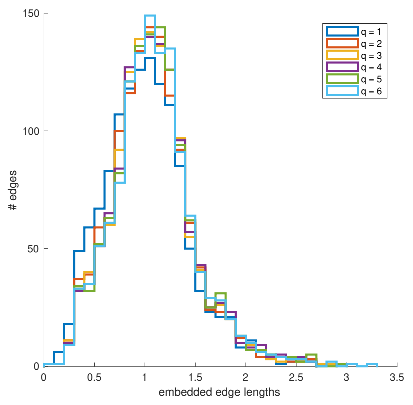
(b) Close-up (of the section in the red box of the left figure) of the curvilinear mesh edges produced by the metric field derived from Eq. 1.

Fig. 3: Straight-sided and curvilinear meshes produced by embedding the mesh conforming to the metric derived from the embedding of Eq. 1.

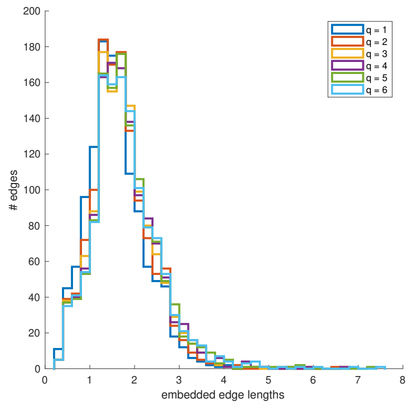
A similar procedure is used to study a three-dimensional case, whereby an analytic embedding into four-dimensional space is defined by

$$\mathbf{u}(\mathbf{x}) = \left[\mathbf{x}, 10\tanh(10||\mathbf{x}||^2 - 0.75^2)\right], \quad \mathbf{x} = [0,1]^3. \tag{2}$$

Unlike its two-dimensional counterpart, this case achieves the best isometry when embedded into a five-dimensional Euclidean space; see Fig. 5b. It is interesting, however, that isometry can be recovered in four dimensions by using a high-order embedding as demonstrated in the edge lengths of Fig. 6a. For higher-dimensional embeddings, however, the lower order embeddings achieve better isometry since the higher-order ones generate longer edge lengths, likely due to approximating the longer geodesics across the domain (Fig. 6b).

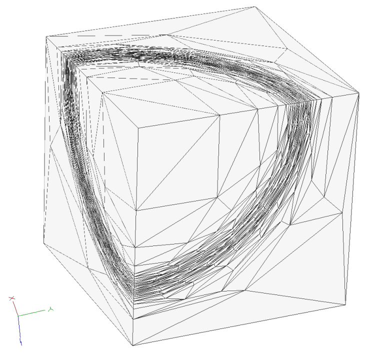


(a) Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 1 into three-dimensional space.

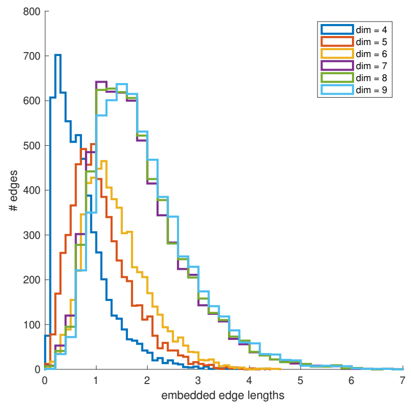


(b) Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 1 into five-dimensional space.

Fig. 4: Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 1 into three- and five-dimensional space.

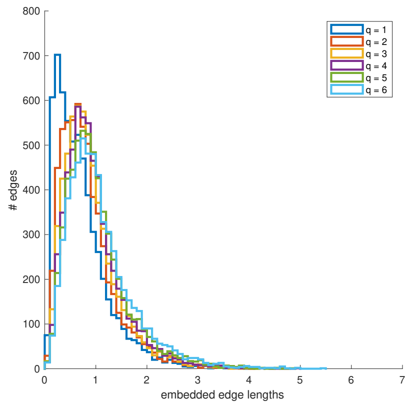


(a) Metric conforming straight-sided mesh for the embedding of Eq. 2.

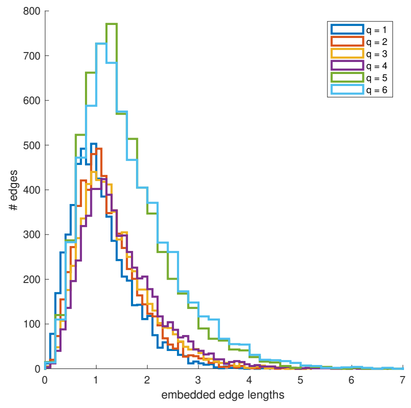


(b) Edge length histogram for edges produced by embedding the straight-sided mesh conforming to the embedding of Eq. 2 into Euclidean spaces of various dimensions.

Fig. 5: Edge length histogram for edges produced by embedding the straight-sided mesh conforming to the embedding of Eq. 2 into Euclidean spaces of various dimensions.



(a) Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 2 into four-dimensional space.



(b) Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 2 into six-dimensional space.

Fig. 6: Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 2 into four- and six-dimensional space.

3.2 *Analytic metric fields*

Now the method is studied with analytic metric fields. In particular, the metrics defined by

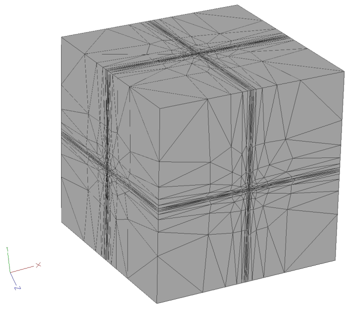
$$\mathbf{m}(\mathbf{x}) = \begin{bmatrix} \frac{1}{(|x-0.5|+0.0025)^2} & 0 & 0 \\ & \frac{1}{(|y-0.5|+0.0025)^2} & 0 \\ & & \frac{1}{(|z-0.5|+0.0025)^2} \end{bmatrix}, \quad \mathbf{x} \in [0,1]^3 \qquad (3)$$

and

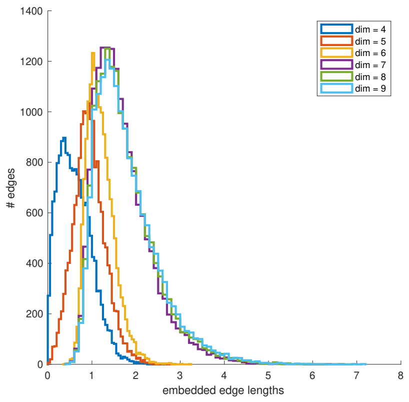
$$\mathbf{m}(\mathbf{x}) = \begin{bmatrix} \frac{\cos^2\theta}{h_x^2} + \frac{\sin^2\theta}{h_y^2} & \left(\frac{1}{h_x^2} - \frac{1}{h_y^2}\right)\cos\theta\sin\theta & 0 \\ & \frac{\sin^2\theta}{h_x^2} + \frac{\cos^2\theta}{h_y^2} & 0 \\ & & 4 \end{bmatrix}, \quad \mathbf{x} \in [0,1]^3 \qquad (4)$$

where $h_x = \min(0.005\cdot 5^a, 0.5)$, $h_y = \min(0.1\cdot 2^a, 0.5)$ $(a = 10|0.75 - \sqrt{x^2+y^2}|$ and $\theta = \arctan(x,y))$ [18], are used to create straight-sided metric-conforming meshes with `fefloa` (Figs. 7a and 9a). The embedding algorithm is applied as in the last section, however, the dimension of the ambient Euclidean space cannot be predicted a priori. As such, a sweep over dimensions is used to embed the straight-sided mesh; the histogram of edge lengths resulting from this sweep is shown in Figs. 7b and 9b.

The metric of Eq. 3 appears to be best represented in five- and six-dimensional Euclidean spaces. As such, these spaces are used to now sweep over the geometric orders of the mesh, the histograms of which are shown in Figs. 8a and 8b. Little difference is observed across the different mesh orders in five-dimensional space but isometry seems to degrade with increased mesh order in six-dimensions since it appears that edges become longer for reasons stated earlier. It is further surprising that only five dimensions are needed to represent this metric field isometrically because of the three directional variations in the metric, implying three codimensions would be needed. Further theoretical work is needed to understand this result.



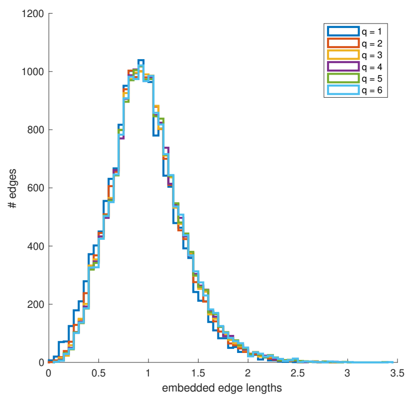
(a) Metric conforming straight-sided mesh for the embedding of Eq. 3.



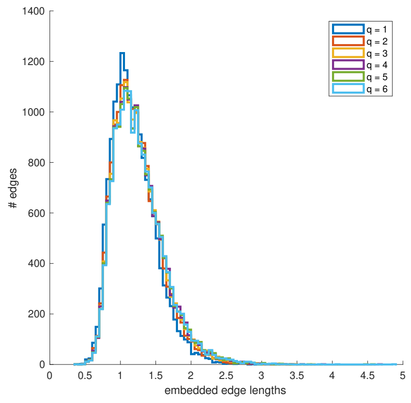
(b) Edge length histogram for edges produced by embedding the straight-sided mesh conforming to the embedding of Eq. 3 into Euclidean spaces of various dimensions.

Fig. 7: Edge length histogram for edges produced by embedding the straight-sided mesh conforming to the embedding of Eq. 3 into Euclidean spaces of various dimensions.

The metric of Eq. 4 is best represented in four- and five-dimensional space as observed in the edge length histogram of Fig. 9b. Isometry slightly improves with increasing curvilinear mesh order in four dimensions but little difference is observed in five dimensions (see Figs.10a and 10b). This further supports the finding that when the ambient dimension of the Euclidean space is not high enough to achieve an isometric embedding of a straight-sided mesh, better isometry can be achieved by embedding a curvilinear representation of the mesh instead. However, when the straight-sided mesh is well represented by an embedding of its vertices into a suitable Euclidean space, then little difference is observed when increasing the geometric order of the embedded mesh.

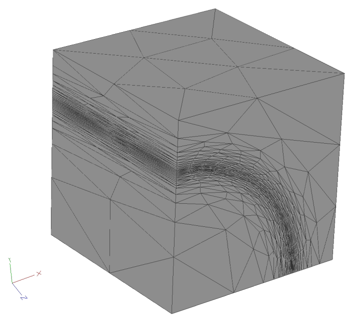


(a) Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 3 into five-dimensional space.

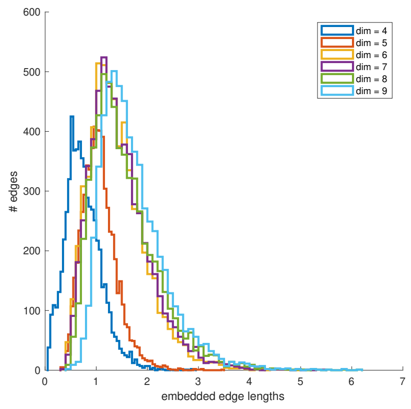


(b) Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 3 into six-dimensional space.

Fig. 8: Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 3 into five- and six-dimensional space.

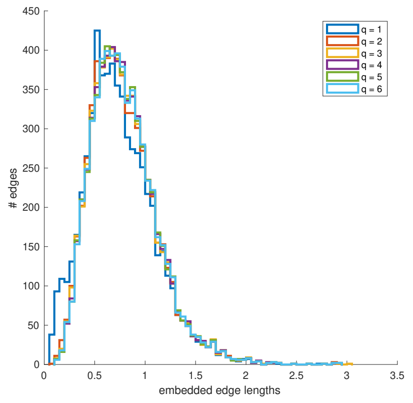


(a) Metric conforming straight-sided mesh for the embedding of Eq. 4.

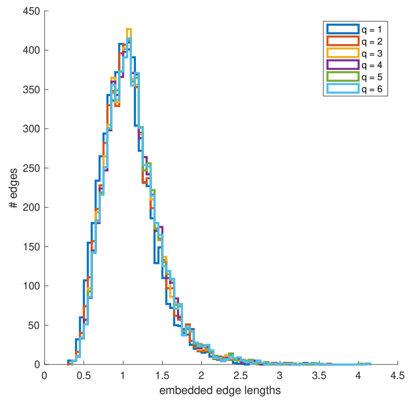


(b) Edge length histogram for edges produced by embedding the straight-sided mesh conforming to the embedding of Eq. 4 into Euclidean spaces of various dimensions.

Fig. 9: Edge length histogram for edges produced by embedding the straight-sided mesh conforming to the embedding of Eq. 4 into Euclidean spaces of various dimensions.



(a) Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 4 into four-dimensional space.



(b) Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 4 into five-dimensional space.

Fig. 10: Edge length histogram for edges produced by embedding curvilinear meshes for the metric conforming to the embedding of Eq. 4 into four- and five-dimensional space.

4 Conclusions and future work

This work introduced an algorithm for embedding curvilinear meshes into Euclidean spaces. The first step consisted of defining a curvilinear mesh from a metric-conforming straight-sided one. The next step consisted of embedding the straight sided mesh into Euclidean spaces of various dimensions and analyzing which dimension closely approximates the local geodesic lengths. Curvilinear meshes of varying geometric order were then embedded to the chosen Euclidean spaces. The findings suggest that when the dimension of the Euclidean space is not high enough, then isometry can be improved by increasing the geometric order of the mesh. However, when the embedding of the straight-sided mesh is sufficiently isometric, then little difference is observed in the produced edge lengths across geometric mesh orders.

Future work consists in using the produced embeddings to directly generate curvilinear metric-conforming meshes. A local approach for doing so, similar to the cavity-based approach of Loseille [15, 16] is attractive due to its ability to enlarge initially invalid cavities when inserting high-order vertices.

Acknowledgements

The authors wish to thank the members of the Aerospace Computational Design Labarotory at MIT for their encouragement throughout this project and the research staff at the Barcelona Supercomputing Center for hosting the first author while much of Section 2.1 was developed. This work was funded by the CAPS project, AFRL Contract FA8050-14-C-2472: "CAPS: Computational Aircraft Prototype Syntheses" with Dean Bryson as the Technical Monitor. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 715546. The work of Xevi Roca has been partially supported by the Spanish Ministerio de Economía y Competitividad and the Generalitat de Catalunya under grant agreements RYC-2015-01633 and 2017 SGR 1731, respectively.

References

1. F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.*, 138(2):251–285, 1997.
2. John Burkardt. The finite element basis for simplices in arbitrary dimensions. Technical report, Florida State University, Department of Scientific Computing, 2013.
3. Guillermo D. Cañas and Steven J. Gortler. Surface remeshing in arbitrary codimensions. *Vis. Comput.*, 22(9):885–895, September 2006.
4. Philip Claude Caplan, Robert Haimes, David L. Darmofal, and Marshall C. Galbraith. Anisotropic geometry-conforming d-simplicial meshing via isometric embeddings. In *Proceedings of the 26th International Meshing Roundtable*. Springer Berlin Heidelberg, 2017.
5. Franco Dassi, Patricio Farrell, and Hang Si. An anisoptropic surface remeshing strategy combining higher dimensional embedding with radial basis functions. *Procedia Engineering*, 163:72 – 83, 2016. 25th International Meshing Roundtable.
6. Franco Dassi, Simona Perotto, Hang Si, and Timo Streckenbach. A priori anisotropic mesh adaptation driven by a higher dimensional embedding. *Computer-Aided Design*, 85:111 – 122, 2017. 24th International Meshing Roundtable Special Issue: Advances in Mesh Generation.
7. Franco Dassi, Hang Si, Simona Perotto, and Timo Streckenbach. Anisotropic finite element mesh adaptation via higher dimensional embedding. *Procedia Engineering*, 124:265 – 277, 2015.
8. Vin De Silva and Joshua B Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Technical report, Stanford University, 2004.
9. A. Grundmann and H. M. Moller. Invariant Integration Formulas for the n-Simplex by Combinatorial Methods. *SIAM Journal on Numerical Analysis*, 15:282–290, April 1978.
10. V. de Silva J. B. Tenenbaum and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.
11. A. Johnen, C. Geuzaine, T. Toulorge, and J.-F. Remacle. Efficient computation of the minimum of shape quality measures on curvilinear finite elements. *Procedia Engineering*, 163:328 – 339, 2016. 25th International Meshing Roundtable.
12. Amaury Johnen, Jean-Francois Remacle, and Christophe Geuzaine. Geometrical validity of curvilinear finite elements. *J. Comput. Phys.*, 233(0):359–372, 2013.
13. H. W. Kuhn. Simplicial Approximation of Fixed Points. *Proceedings of the National Academy of Science*, 61:1238–1242, December 1968.
14. Bruno Lévy and Nicolas Bonneel. Variational anisotropic surface meshing with Voronoi parallel linear enumeration. In *Proceedings of the International Meshing Roundtable*, 2012.

15. Adrien Loseille. Metric-orthogonal anisotropic mesh generation. *Procedia Engineering*, 82:403 – 415, 2014.
16. Adrien Loseille and Rainald Löhner. Cavity-Based Operators for Mesh Adaptation. In American Institute of Aeronautics and Astronautics, editors, *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition.* American Institute of Aeronautics and Astronautics, 2013.
17. Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
18. Christos Tsolakis, Fotios Drakopoulos, and Nikos Chrisochoides. Sequential metric-based adaptive mesh generation. In *2018 Modeling, Simulation, and Visualization Student Capstone Conference*, Suffolk, VA, April 2018.
19. K.Q. Weinberger, B. D. Packer, and L. K. Saul. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In Z. Ghahramani and R. Cowell, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Barbados, West Indies, 2005.