Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2015) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

24th International Meshing Roundtable (IMR24)

Thin-Plate-Spline Curvilinear Meshing on a Calculus-of-Variations Framework

Shankar P. Sastry*[a], Vidhi Zala[a], Robert M. Kirby[a]

[a]*Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112, U.S.A.*

Abstract

High-order, curvilinear meshes have recently become popular due to their ability to conform to the geometry of the domain. Curvilinear meshes are generated by first constructing a straight-sided mesh and then curving the boundary elements (and, consequently, some of the interior edges and faces) to respect the geometry of the domain. The locations of the interior vertices can be viewed as an interpolation of a mapping function whose values at the boundary vertices (of the straight-sided mesh) are equal to the vertex locations on the curved domain. We solve this interpolation problem using radial basis functions (RBFs) by extending earlier algorithms that were developed for linear mesh deformation. An RBF interpolation technique using a biharmonic kernel is also called a thin plate spline. We analyze the resulting mapping function (the RBF interpolation) in a framework based on calculus of variations and provide a detailed explanation of the reasons the thin plate kernel RBF-based techniques have always yielded higher-quality meshes than other techniques. It is known that the thin plate kernel RBF interpolation minimizes the "bending energy" associated with a function, which depends on its second-order partial derivatives. We show that the minimization of the bending energy attempts to preserve the shape of an element after the transformation. Other techniques minimize either a functional (that depends on the first-order partial derivatives) that attempts to preserve the size of an element, or the bending energy in a smaller subspace of functions. Thus, our experimental results show that our algorithm generates higher-quality meshes than prior algorithms.

Peer-review under responsibility of organizing committee of the 24th International Meshing Roundtable (IMR24).

Keywords: Curvilinear mesh generation; Radial basis functions; Mesh deformation; Calculus of variations

1. Introduction

High-order finite element meshes have gained popularity in recent years due to their higher accuracy in numerical simulations than their linear counterparts and their superior convergence properties when they are used to solve partial differential equations (PDEs) with the finite element method. Since it is possible to curve the boundary elements and make them conform to the geometry of the domain with greater accuracy[1], several algorithms have been proposed in

* Shankar P. Sastry. Tel.: +1-979-220-4430
E-mail address: sastry@sci.utah.edu

[1] The greater accuracy in geometrical conformity comes at the cost of the interpolation error of the solution of the partial differential equations and the convergence of the solver.

Peer-review under responsibility of organizing committee of the 24th International Meshing Roundtable (IMR24).

the last decade to generate curvilinear, high-order meshes [11,26,28,32,33,41]. The generation of such meshes follows a template that first constructs a coarse, straight-sided mesh and then deforms the mesh to respect the geometry of the domain. The deformation (also called mesh morphing or mesh warping) can be viewed as a mapping function that moves the boundary vertices to their intended location and computes the location of the interior vertices to produce a mesh for which the size and shape of its elements is comparable to that of the original mesh. We develop a radial basis function (RBF) interpolation-based technique using the thin plate kernel to deform a straight-sided mesh into a curvilinear mesh. We analyze the mapping function that produces the curvilinear mesh and examine why it, in contrast with existing techniques, produces meshes whose elements bear a greater resemblance to the elements in the straight-sided mesh. This analysis leads to the development of a framework based on calculus of variations, which can provide a pathway for mesh deformation algorithms in the future.

Mesh deformation and curvilinear mesh generation are currently identical problemswith identical solutions. These solutions can be broadly divided into the following three classes: optimization [31,41], PDE [28,34], and interpolation-based techniques [8]. Both linear mesh deformation and curvilinear mesh generation have been solved using the first two classes. The RBF-based technique belongs to the last class, but it has been applied only for linear mesh deformation. It has also been shown to produce higher-quality meshes than other techniques (although, it was much slower for larger meshes) [37]. The last two classes can be viewed as a single class of interpolation techniques, where the positions of the interior mesh vertices are interpolated from the positions of the boundary vertices using either the solution of a PDE or an explicit interpolation technique. These algorithms are discussed in detail in Section 2.

We extend the RBF-based technique for curvilinear mesh generation in 2D using the thin plate kernel as it minimizes the "bending energy" of the mapping function[2]. We provide a brief overview of the RBF interpolation methods and their acceleration techniques in Section 3. We then analyze why the minimization of the bending energy is a more suitable choice than other choices that are minimized in the PDE-based methods. The solution of Laplace's equation, for instance, minimizes the Dirichlet energy. This analysis is extended to a framework that is based on calculus of variations to develop mesh deformation and curvilinear mesh generation algorithms in the future. The analysis and the framework are provided in Section 4.

From our preliminary numerical experiments to generate curvilinear meshes from the RBF interpolation-based technique, we visually observe that the quality of the final mesh elements (the size and shape) has a greater correlation with the original mesh than the thermoelasticity-based method [25] for an input geometry. These visuals and some quantitative results are presented in Section 5.

An avenue for future research includes the formulation of a functional whose optimization leads to the "perfect" mesh deformation algorithm. Other directions include the acceleration of RBF-based interpolation techniques specifically for mesh deformation and an RBF-based interpolation technique for boundary-layer curvilinear mesh generation with compactly supported kernels. These research directions are discussed in Section 6.

2. Related Work

As described above, curvilinear mesh generation and linear mesh deformation are identical problems and are carried out using similar methods, which are described in detail below. In these methods, an initial straight-sided mesh and the boundary deformation information are provided as the input. The output is obtained from one or a combination of these techniques.

2.1. Optimization-Based Techniques

We categorize algorithms that use the local mesh topology information (edge connections, etc.) to deform meshes under a single umbrella called optimization-based techniques because they typically involve local optimization of some objective function that improves the quality of the mesh. For linear meshes, Wicke et al. [42] carried out mesh deformation for elastoplastic simulations through dynamic meshing techniques that locally change the mesh topology to account for the loss of mesh quality due to vertex movement during the deformation. Sastry et al. [31] used a

[2] For 3D meshes, too, we recommend the triharmonic kernel for the same reason.

log-barrier optimization routine to dictate vertex movement and to improve the quality of a tangled mesh (due to the deformation) to obtain a valid mesh. Staten et al. [39] compare several algorithms that includes a mesh optimization-based algorithm. Their list of algorithms also includes the LBWARP algorithm by Shontz and Vavasis [34], which uses a weighted Laplacian smoothing framework. There are several techniques that use a torsional spring analogy to move the mesh vertices [9,10,43].

For curvilinear mesh generation, Shephard's group [23,24,27,32] developed several techniques that construct a Bézier polynomial-based representation of the curvilinear boundary and suitably curve some of the interior elements and modify the local connectivity to obtain a valid mesh. Remacle's group developed a log-barrier technique [14,41] (independently from [31]) that generates a valid mesh by maximizing the minimum Jacobian of high-order elements in the mesh. Similar techniques were also used by Gargallo-Pieró et al. [11–13].

2.2. PDE-Based Techniques

PDE-based techniques construct a mapping function based on the solution of a PDE that transforms a mesh from the initial domain to the final domain. The locations of the boundary nodes are treated as Dirichlet boundary conditions, and the locations of the interior nodes are obtained from the solution of the PDE. Naturally, the choice of the elliptic PDEs used for this purpose gives rise to several techniques.

For linear meshes, the solution of the Laplace's equation was one of the first mappings used to deform meshes. Baker [2] and Shontz and Vavasis [34] were the first to carry out extensive research on this technique. Shontz and Vavasis carried out further research [35] to suggest modifications to the method to improve the results. We will revisit their suggestions in Section 4 and explain how our framework generalizes the modifications they proposed. Instead of Laplace's equation, the biharmonic PDE [18] has also been used to deform meshes. The solution to the biharmonic equation is also the solution to the PDE that models linear elasticity with the Dirichlet boundary condition. This technique has been extended to consider PDEs that model hyperelastic [36] and superelastic [30] materials.

For curvilinear mesh generation, PDEs that model linear elasticity [1,14,19,40] have been used for various applications. Persson and Peraire [28] extended the technique to model the mesh as a nonlinearly elastic material. In the latest work, thermoelastic equations were used by Moxey et al. [25].

In many of these techniques, meshes were progressively deformed over many steps to obtain a valid, high-quality mesh. In nonlinear elasticity-based techniques, additional variables such as the material stiffness and thermal stresses were iteratively modified to obtain a valid mesh in case the early attempts failed.

2.3. Interpolation-Based Technique

Interpolation-based techniques have been employed only for linear mesh deformation algorithms. For instance, Staten et al. [39] developed the simplex-linear transformation algorithm, which carries out a linear interpolation of mesh vertices after coarsening the mesh. Other forms of interpolation techniques for scattered data such as kriging and inverse distance weighting have not yet been used for mesh deformation. RBF interpolation techniques, however, have been used to interpolate the positions of the interior mesh vertices [8,20,22,37,38]. Such techniques have consistently shown better deformation than that by other techniques described above. Thus, we extend the RBF interpolation-based technique to curvilinear meshes in this paper. Besides the straightforward extension, we analyze why such algorithm have always produced higher-quality meshes than other algorithms and develop a framework based on calculus of variations to aid in the development of future algorithms.

3. Background

We provide a brief background on the RBF interpolation techniques in this section. A naive implementation of the RBF interpolation technique, whose time complexity is $O(n^3)$, scales very poorly. There are acceleration techniques that scales very well, whose complexity is $O(n \log n)$. Readers who are familiar with the literature may skip this section.

3.1. Radial Basis Function Interpolation

Consider the following scattered data interpolation problem. Given $y_i = f(x_i)$ for $i \in 1, 2, ..., n$, find a function that interpolates the data. The vector x_i may be of any dimension. The RBF interpolation technique computes the function by assuming that it is of the form

$$f(x) = \sum_{i=1}^{n} \omega_i \phi(\|x - x_i\|_2),$$

where ω_i are the weights that need to be computed and $\phi(\cdot)$ is a radial basis kernel whose value depends only on the length of the vector, not its direction. The kernel may be defined using any function, but some functions are preferred over others due to certain desirable properties. Some examples of such kernels include

- the Gaussian kernel: $\exp(-\varepsilon r)^2$,
- the multiquadratic kernel: $\sqrt{(1 + (\varepsilon r)^2)}$,
- the inversequadratic kernel: $\frac{1}{1+(\varepsilon r)^2}$, and
- the polyharmonic spline: r^k, for $k \in \{1, 3, 5, ...\}$ and $r^k log r$, for $k \in \{2, 4, 6...\}$,

where r is the length of a vector and ε is some scaling factor. It can be inferred from the function definitions that some of these functions have "global" support, i.e, their value increases as r increases, whereas others have "compact" support, i.e., their value decreases as r increases. Also, when $k = 2$ for RBF interpolation using polyharmonic splines, it is also called a thin plate spline.

The weights ω_i are computed by solving a system of linear equations with n equations and n variables obtained by substituting x_i for x and y_i for $f(x)$ in the interpolating function above. For globally supported kernels, as n or the domain of the scattered data points increases, the conditioning of the matrix deteriorates. For compactly supported kernels, the condition number also increases, but gradually. In both cases, it is advisable to use iterative solvers rather than direct solvers for two reasons. For kernels with global support, an ill-conditioned system results in large errors in the output when a direct solver is used. For kernels with compact support, preconditioned iterative solvers converge to a solution much faster than direct solvers. Globally supported kernels, however, extrapolate the data very well outside the domain of the input data. Although mesh deformation is an interpolation problem, the scattered data is provided only at the boundary of the mesh. A point in the interior of the mesh may be far from the boundary, and the interpolation of the function values at those locations may become close to zero if kernels with a compact support are used. Thus, for large domains, either (a) a kernel with global support or (b) a kernel with compact support, but with a large scaling factor must be used. Note that the advantages of compactly supported kernels are progressively lost as the scaling factor is slowly increased.

An additional polynomial term may be added to the RBF interpolation technique so that functions that are already close to being a polynomial are easily reconstructed. We use a linear polynomial in our implementation because the locations of the vertices in the undeformed mesh correspond to the simplest linear function, $f(x) = x$, in each axis and also because it results in the "smoothest" interpolant, i.e., An integral of a function of the second-degree partial derivatives is minimized. Its importance is discussed in Section 4. To account for the additional terms due to the polynomial in the interpolating function, the following equations are added to the linear system:

$$\sum_{i=1}^{n} \alpha_j p_j(x_i) = 0,$$

for $j \in \{1, 2, ..., m\}$, where $p_j(\cdot)$ is a j^{th} basis polynomial and α_j is its multiplicative coefficient.

3.2. Acceleration Techniques

Clearly, a naive implementation of the above technique leads to poor scaling of the algorithm. For an RBF-based method to be a viable option, acceleration techniques, which improve the expected runtime from $O(n^3)$ to $O(n \log n)$, where n is the number of points on the boundary of a mesh, should be employed. Note that the evaluation of the function at any point also takes $O(n)$ time (after the weights, ω_i, have been computed). The evaluation should also

be reduced to $O(\log n)$ for fast computation of the RBF interpolant. These reductions, however, come at the cost of accuracy of the evaluated function, which may affect the quality of the mesh. Thus, they should be used cautiously.

The linear system constructed from the equations above is solved using Krylov subspace methods, which involve repeated matrix-vector multiplications. Such evaluations may be accelerated through fast multipole methods (FMMs) for RBF interpolations [15,16]. The FMMs recursively decompose the domain into octrees such that it becomes possible to group the points that are close together in the domain. During the matrix-vector multiplications, for all the points in the dataset, their distance to a given point and their associated weights are multiplied independently [17] and added together. It is possible to approximate the sum of the products by hierarchically considering the groups of points in the octree decomposition. This technique was originally used to evaluate the value of the interpolating function after the weights are accurately computed [15,16].

Moreover, the linear system may be preconditioned using a matrix constructed by a subset of the scattered data points whose RBF interpolation approximates the RBF interpolation for all points in the dataset. Such a subset of points may be constructed using an octree structure such that the points are dispersed over the domain [3–5,17]. These techniques have not been implemented yet, and we leave the analysis of the technique for realisitic domains as future work.

4. The Algorithm and the Analysis

We provide a straightforward algorithm for curvilinear mesh generation below. We then analyze the functionals that are minimized by some of the methods described in Section 2. Based on the analysis, some possible extensions are recommended for the RBF interpolation-based algorithms.

4.1. RBF-Based Curvilinear Mesh Generation Algorithm

In the early mesh deformation algorithms such as Laplace's equation or biharmonic equation-based methods, the respective equations were solved independently for each dimension. For the elasticity-based algorithms, the equations are inherently dependent along all the dimensions, so the mapping functions for each of the dimensions are computed simultaneously. In the RBF-based algorithm, we compute the mapping functions independently for each dimension. The mapping function values for the boundary nodes are set based on the required curvature at the boundary. The interior vertices are moved to the locations dictated by the interpolated values.

We implemented this algorithm in C++ using the standard template library. The linear conjugate gradient (CG) solver was solve the system of linear equation that arise from the formulation. In our formulation, we added a linear polynomial term to the RBF interpolant. Since the resulting system is symmetric and positive definite, the CG solver was sufficient for our purpose. In theory, the CG solver converges in n iteration if there are n equations, but due to floating point issues, it may need more. We run the solver for $5n$ iterations and use the solution from the step for which the residual error is minimum.

The results, provided in Section 5, show that the shape of the deformed elements respects their original shape more "closely" than they do for the thermoelasticity-based equations. Similar results have been published in a number of papers for RBF-based mesh deformation algorithms. In the rest of this section, we provide possible explanations for the results and improve the RBF-based algorithm based on the analysis.

4.2. Calculus-of-Variations Analysis of Mesh Deformation

The solution of PDEs as well as the interpolation of scattered data minimize the value of *functionals*, which map a set of functions to a set of real numbers, under a set of certain constraints on the functions that vary for each problem. The study of optimization of functionals is called calculus of variations. We study the functionals that are minimized and explain why the RBF-based techniques are most likely than other techniques to produce high-quality meshes.

4.2.1. Laplace's Equation

Early algorithms for mesh deformation used the solution of Laplace's equation mainly because of its simplicity in both the application of boundary conditions and implementation. In addition, it minimized the following functional,

also known as the Dirichlet energy:

$$\iint_\Omega \frac{1}{2} \|\nabla f\|^2 \, dA = \frac{1}{2} \iint_\Omega \left(\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right) dA,$$

where Ω is the domain. The functional contains only the first-order partial derivatives, which can be intuitively seen as controlling the affine translation matrix for an infinitesimal element in the context of mesh deformation. We will provide a Taylor series-based analysis in Section 4.3 to mathematically and geometrically illustrate the intuition.

4.2.2. Biharmonic Equations

In addition to the Dirichlet boundary conditions associated with the movement of boundary vertices, the biharmonic equation requires boundary conditions that may constrain the gradients, the Hessian, or the third-order partial derivatives. These conditions provide additional flexibility (over the Laplacian equation) to control the mapping function for the mesh deformation. Helenbrook [18] provides an analysis of possible boundary conditions for the problem and the contexts in which the conditions are applicable. The solution of the biharmonic equation minimizes the following functional subject to the boundary conditions:

$$\iint_\Omega \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) dA.$$

This functional is also called the "bending enery", and it contains only second-order partial derivative terms for the mapping function. Intuitively, it controls the location of a vertex with respect to another vertex in the domain. If the location of the first vertex is known, the location of the second vertex is constrained to lie within an rectangle, whose center is at a location that is a function of the first-order partial derivatives, and the length and the bredth is a function of the bounds on the second-order partial derivatives. Section 4.3 provides more details.

4.2.3. Linear Elasticity

The solution of the PDE that model linear elasticity minimizes the following functional:

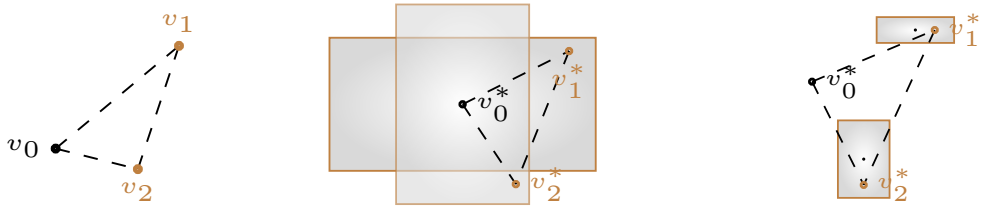
$$\iint_\Omega \left(\lambda (\nabla . f)^2 + \mu \varepsilon_{xy}^2(f) \right) dA, \text{ where } \varepsilon_{xy}(f) = \frac{1}{2} \left(\frac{\partial f_x}{\partial y} + \frac{\partial f_y}{\partial x} \right),$$

f is a vector-valued function $[f_x; f_y]$, and λ and μ are the Lamé parameters, which are material-specific values. It can be shown that the solution of the linear elasticity equations also satisfy the biharmonic equations, but the additional boundary conditions needed to obtain a unique solution to the biharmonic equations are functions of the Lamé parameters. Thus, the solution of the linear elasticity equations also minimizes the energy functional for the biharmonic equations for a smaller subspace of functionals. One can argue that the biharmonic equation-based method to deform meshes is a generalization of the linear elasticity-based method, but note that elasticity-based methods provide greater control on the mapping function as the Lamé parameters can be modified at specific parts of the domain where the technique is likely to produce inverted elements.

We do not provide the energy functionals for nonlinear elasticity equations [28] or thermoelasticity equations [25] as they are tedious to write down. For the purpose of analysis, an intuitive understanding is sufficient. For nonlinear elasticity, the energy functionals are functions of first-order partial derivatives of the mapping raised to fractional powers based on any relevant model. The energy functional minimized by the thermoelasticity equation contains an additional temperature-based term that is designed to prevent inversion of vulnerable elements based on the minimum determinant of their Jacobian matrix.

4.2.4. Thin Plate Spline

RBF interpolation with biharmonic kernels (also known as the thin plate splines) minimize the same functional as the biharmonic equations above (the bending energy). The difference is that thin plate splines are not constrained by the additional boundary conditions as in the case of the biharmonic equations. Thus, the energy functional is optimized over a larger space of functions. As a consequence of the minimization of the energy functional, we prove the following lemma, which was also proved for Laplace's equation-based mesh deformation by Shontz and Vavasis [34].



(a) an undeformed element (b) gradient-based range (c) Hessian-based range

Fig. 1: The undeformed element is shown on the left. We may be provided with a bound on the first- or second-order partial derivatives of the mapping function. If the mapping of one of the vertices v_0 is known, the range of the locations of the other vertices can be computed using the Taylor-series approximation. As shown in the central figure, if the bounds of the first-order partial derivatives are provided, the other two vertices are constrained to be within their respective rectangles. If the bounds on the second-order partial derivatives are provided (and the first-order derivatives are known), the vertices are constrained to be within their respective rectangles as seen in the figure on the right. The dimensions of all regions are proportional to the relevant partial derivatives and the length of the edges in the undeformed triangle.

Lemma 1. *If the boundary vertices are affinely transformed, the interior vertices are also affinely transformed by the thin plate-spline mesh deformation algorithm.*

Proof. For an affine transformation, the bending energy of the transformation vanishes because the second-order partial derivatives vanish, but for any other transformation function that respects the affine transformation of the boundary vertices, the bending energy is positive because the second-order partial derivatives no longer vanish. As thin plate splines minimize the bending energy, the interior vertices are also affinely transformed. □

We can also prove the lemma above by noting that the linear system arising from the data points is nonsingular. Thus, it has a unique solution. If the linear system arises from an affine transformation of the data points, only the weights associated with the linear polynomial basis functions are sufficient to interpolate the function, and the weight associated with all data points is zero. Therefore, even the interior vertices are affinely perturbed.

Since the same energy functional is minimized by the biharmonic equations (under certain constraints), this technique can also be viewed as a generalization of the biharmonic equation-based technique to deform meshes. Note that RBFs have been used to solve PDEs using Kansa's technique [7,21] with the advent of "meshfree" techniques. In such techniques, the solutions of the PDEs are treated as a constrained interpolation problem, where the constraints are imposed by the PDEs themselves. In this sense, the RBF-based mesh deformation and curvilinear mesh generation technique can be viewed as a generalization of the PDE-based techniques. Thus, this is a clean-slate approach (devoid of any constraints) in which additional constraints can be added (explained in Section 4.4) when necessary to obtain untangled, high-quality meshes.

4.3. Taylor Series-Based Analysis of Element Inversion

Shontz and Vavasis [34] carried out a Taylor series-based analysis of the possibility of element inversion when a mesh is being deformed using the solution to Laplace's equation. We generalize their analysis in the calculus-of-variations framework for any algorithm that computes a mapping from the source domain to the target domain.

Consider a triangle whose vertices are located at $\vec{v_0} = [x_0, y_0]$, $\vec{v_1} = [x_1, y_1]$, and $\vec{v_2} = [x_2, y_2]$ (see Fig. 1a). Let the mapping function be denoted by u_x and u_y for $x-$ and $y-$axis, respectively. Let the first vertex be mapped to $\vec{v_0^*} = (x_0^*, y_0^*)$. Let $\vec{v_1} - \vec{v_0} = \vec{v_{01}} = (\Delta x_1, \Delta y_1)$ and $\vec{v_2} - \vec{v_0} = \vec{v_{02}} = (\Delta x_2, \Delta y_2)$. In the rest of the section, we use the Taylor series expansion to bound the locations of the mesh vertices.

Since the solutions of Laplace's equation and the linear elasticity equations impose a bound on the first-order partial derivatives, we may assume that $|\partial u_x/\partial x| \leq c_{xx}$, $|\partial u_x/\partial y| \leq c_{xy}$, $|\partial u_y/\partial x| \leq c_{yx}$, and $|\partial u_y/\partial y| \leq c_{yy}$. Then, the new locations of two other vertices, $\vec{v_1^*} = [x_1^*, y_1^*]$ and $\vec{v_2^*} = [x_2^*, y_2^*]$, are within a rectangle centered around $\vec{v_0^*}$, i.e, $x_1^* \leq x_0 + c_{xx}\Delta x_1 + c_{xy}\Delta y_1$ and $y_1^* \leq y_0 + c_{yx}\Delta x_1 + c_{yy}\Delta y_1$, with similar equations for the lower bound of x_1 and y_1. Similar bounds may also be obtained for $\vec{v_2^*}$. Fig. 1b provides a geometrical illustration of these constraints.

The solution of the biharmonic equation and the RBF interpolation both limit the second-order partial derivative. Thus, we may assume $|\partial^2 u_x/\partial x^2| \leq k_{xx}$, $|\partial^2 u_x/\partial y^2| \leq k_{yy}$, and $|\partial^2 u_x/\partial x \partial y| \leq k_{xy}$. Let us also assume that

$|\partial u_x/\partial x|_{x=x_0,y=y_0} = a_x$ and $|\partial u_x/\partial y|_{x=x_0,y=y_0} = a_y$. Then, the location of x_1^* is within

$$x_0^* + a_x\Delta x_1 + a_y\Delta y_1 \pm \frac{1}{2!}\left(k_{xx}\Delta x_1^2 + k_{yy}\Delta y_1^2 + k_{xy}\Delta x_1\Delta y_1\right).$$

Similarly, the location of y_1^* can also be constrained. These constraints imply that the location of the two vertices is constrained to be within the two rectangles shown in Fig 1c. The centers of the rectangles are the first-order approximate locations of the vertices under consideration.

Clearly, the equations and figures indicate that the gradients of the mapping functions can control the scaling of the mapped triangle but not its shape. On the other hand, when the second-order partial derivatives are bounded, the shape of the triangle is also bounded to a great extent. As a result, when the second-order derivatives are bounded, the likelihood of element inversion is minimized. In fact, if the quality of an element is good, the mapping function in that region can tolerate a large bound on the second-order partial derivatives.

Shontz and Vavasis [34] derived a set of sufficient conditions that ensure that no elements in the mesh get inverted. The condition is an inequality that bounds the ratio of the first- and second-order partial derivatives of the mapping function as a function of the shape and the size of an element. In this paper, we focus on how limiting the second-order partial derivatives is more important than limiting the first-order partial derivatives, which is a consequence of the result from Shontz and Vavasis [34]. In fact, we ought to limit the ratio of second-order partial derivatives to the first-order partial derivatives to develop an "ideal" mesh deformation algorithm. We leave that as future work.

4.4. A Framework for Mesh Deformation

In this subsection, we analyze how RBF-based algorithms should be modified to make them more robust. We also compare and contrast our modifications with the analogous modifications in other mesh deformation and curvilinear mesh generation techniques.

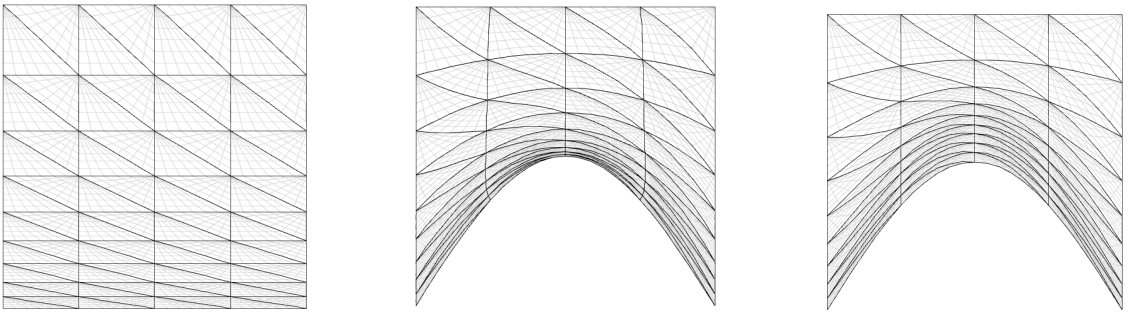
Shontz and Vavasis [35] identified the following three causes of element inversion during mesh deformation: (a) large boundary transformation, which results in unfavorably directed gradients in the mapping function; (b) large second-order partial derivatives, which result in poor-quality or large elements becoming inverted; or (c) the difference between the numerical and the analytical solution of the equations that compute the mapping function.

For type (a) element inversion, a common solution is to circumvent the issue by carrying out the deformation in multiple steps. This solution has been used in almost all techniques for mesh deformation. We recommend the same solution as it is very effective. For curvilinear mesh generation, this cause of element inversion is rather uncommon because the deformations are themselves quite small.

For type (b) element inversion, Shontz and Vavasis [35] recommend refining the mesh. This strategy works because the length of an edge in the mapped triangle is asymptotically proportional to the length of the edge in the initial triangle (assuming the first-order partial derivatives of the mapping functions are bounded), but the uncertainty in the position of the vertex (the dimensions of the rectangle) is proportional to the square of the length of the edge in the initial triangle. Geometrically, as the length of the edge of the initial triangle reduces, the dimensions of the rectangle reduce much faster than the length of the edge in the mapped triangle[3]. Thus, if the elements are small enough, they will not be inverted after the transformation. In the elasticity-based techniques, physical quantities such as stiffness constants or the temperature are locally modified in each subsequent iteration to indirectly address this problem of a large second-order derivative. In our approach, it is possible to add constraints on the first- and the second-order partial derivatives at any point in the domain. Thus, we recommend adding more equations to the linear system to mitigate the problem as it does not refine the mesh.

Type (c) element inversion is typically not seen in any of the techniques because numerical techniques are very accurate. In our technique, faster, approximate algorithms may be used to compute the transformation. If the solution is not accurate enough, a technique to adaptively evaluate more-accurate mapping functions in certain parts of the region should be developed.

[3] It is a consequence of the Taylor series-based analysis presented above.



(a) the undeformed mesh (b) thermoelasticity-based technique (c) RBF-based technique

Fig. 2: The initial straight-sided, high-order, undeformed mesh (left) and curvilinear meshes generating using thermoelasticity equation (center) and RBF interpolation-based (right) techniques. The high-order elements are shown using dark lines, and the interior quadrature points are shown using light lines. The domain of the initial mesh ranges from (0, 0) to (1, 1). The meshes were deformed such that the vertices on the bottom boundary lie of the curve $y = 0.5 \sin \pi x$. The meshes were deformed in a single step.

5. Numerical Experiments

In this section, we present the results from the implementation of the RBF-based (thin plate spline) curvilinear mesh generation technique. We mainly focus on how the placement of vertices by our method attempts to preserve the shape of the initial mesh. We compare our method against the thermoelasticity-based technique [25] that was implemented in Nektar++ [6]. We use the same material parameters as given in [25]. The results may vary for other parameters. Note that there are no parameters to set for the thin plate spline RBF interpolation technique. This is an additional advantage of the technique.

5.1. Experimental Setup

Since previous studies have shown the effectiveness of the RBF-based mesh deformation algorithms in generating high-quality meshes, in this paper, we focus only on the differences in the vertex placement of the RBF interpolation-based technique and thermoelasticity-based technique. Our results show that the RBF-based method can handle larger deformation without producing an invalid mesh[4]. We show the results from a single-step and a multi-step deformation of the initial straight-sided mesh.

Our numerical experiments are described using Fig. 2. We start with an undeformed square mesh and move the vertices on its bottom boundary to lie on a sinusoidal curve $y = \alpha \sin(\pi x)$. The initial sixth-order mesh is shown in Fig. 2a. The mesh contains both isotropic and anisotropic elements of different areas, with 72 triangular elements.

We perform the following two experiments to study the effectiveness of the curvilinear mesh generation techniques: (a) vary α to determine how much deformation the two techniques can handle, and (b) compare the meshes after one- and two-step deformation is carried out for the same α.

5.2. Results

For $\alpha = 0.5$, the results of the screenshot of the whole domain are shown in Fig. 2. The high-order elements are shown using dark lines, and the interior quadrature points are shown using light lines. We curve the bottom boundary of the domain, and the resulting curvature results in greater deformation of the elements in the bottom half of the domain. Since the elements in the bottom half of the domain are highly anisotropic, this deformation is more challenging than the deformation of the top boundary. The differences in the two techniques can be clearly seen in the figure. The thermoelasticity-based technique was successful in generating a valid mesh, but the elements near the boundary are very skinny. The RBF-based technique has preserved the thickness of the boundary layer of the

[4] Triangular elements with a negative determinant of the Jacobian matrix at some point in the triangle are considered invalid.

(a) thermoelasticity-based technique (b) RBF-based technique

Fig. 3: Zoomed-in views of the curvilinear meshes obtained from the thermoelasticity equations (left) and the RBF-based interpolation (right). The meshes were deformed such that the vertices on the bottom boundary lie on the curve $y = 0.5 \sin \pi x$. The meshes were deformed in a single step.

(a) thermoelasticity-based technique (b) RBF-based technique

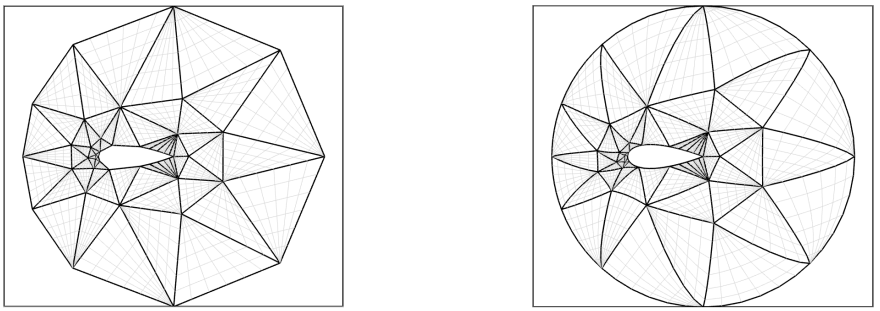
Fig. 4: Zoomed-in views of the curvilinear meshes obtained from the thermoelasticity equations (left) and the RBF-based interpolation (right). The meshes were deformed such that the vertices on the bottom boundary lie on the curve $y = 0.5 \sin \pi x$. The meshes were deformed in two steps.

triangles at the bottom of the domain. Another difference is in the movement of vertices along $x-$axis. Since the thermoelasticity-based technique uses material properties such as the Poisson's ratio, which relates the deformation along all axes, we see that the vertices have moved significantly along the $x-$axis near the bottom boundary. On the other hand, the vertex movement is independent along each axis for the RBF-based technique, so the vertices have not moved along the $x-$axis. We consistently observe this behavior in the rest of the paper. We have also provided the screenshots of the meshes that are zoomed-in near the maxima of the sinusoidal boundary (see Fig. 3). In the rest of the paper, we provide the screenshots of the mesh that are zoomed-in near the same region.

In a separate experiment, we were able deform the meshes until $\alpha = 0.55$ and $\alpha = 0.65$ for the thermoelasticity and RBF-based techniques, respectively, in a single step. The results have not been visualized because the minimum determinant of the Jacobian matrix was too small in this experiment.

Fig. 4 shows the curvilinear meshes for $\alpha = 0.5$ as in the previous experiment, but the meshes were obtained in two steps. The meshes were first deformed corresponding to $\alpha = 0.25$ and then further deformed to obtain the final mesh. Although the meshes appear identical (to images in Fig. 3), the vertex positions result in thicker elements in the two-step process than in the one-step process. A closer observation of the ratio of the lengths of the vertical edges in the two images for the thermoelasticity-based technique reveals that the two-step process does provide favorable vertex positions, i.e., the size and the shape of the element is similar to that of the original element. The difference for the RBF-based technique is too small to be visualized. We also observed that the changes in the positions of the vertices become insignificant after more than five steps are used to obtain the final mesh.

In our next experiment, we held the number of steps at five. We then generated curvilinear meshes with as much deformation as possible. We found that the value of α at which the thermoelasticity-based technique fails to generate a valid mesh is around 0.6. For the RBF-based technique, however, the maximum value of alpha is around 0.8.



(a) straight-sided mesh (b) RBF-based technique

Fig. 5: Curvilinear meshes generated from an initial straight-sided mesh using the RBF-based technique for the air foil domain. The thermoelasticity-based technique fails to provide a valid mesh for this domain. The technique returns an element with a negative valued for the determinant of the Jacobian within an element near the bottom left boundary of the air foil.

We analyzed the scaled Jacobian values (the ratio of the minimum and the maximum values of the determinant of the Jacobian within an element) for the elements that gets inverted due to the deformation using the two techniques. The thermoelasticity-based technique returns a mesh with the smallest scaled Jacobian value for the lower, center-left element in the bottom row for large deformations. The RBF-based technique returns a mesh with the smallest scaled Jacobian value for the upper, central-right element in the fifth row of elements from the top. We leave the investigation of the reasons behind these results for future work.

We carried out curvilinear mesh generation for an air foil domain as show in Fig. 5 using the RBF-based technique. The initial straight-sided mesh was obtained from Distmesh [29]. The thermoelasticity-based technique fails to provide a valid mesh because the mesh is too coarse for the deformation it is undergoing. This illustrates that the technique is suitable for a real-world examples and and can handle deformation in both axes.

These experiments show that the RBF-based technique can handle greater deformation then the thermoelasticity-based technique. We must mention that the RBF technique is slower for large meshes. Specifically, our implementation's running time is $O(n^3)$. The robust nature of the techniques comes at the cost of the complexity of the algorithm. The results, by themselves, are not novel as many other papers have reported similar results in the context of deformation of linear meshes. We leave the development of an $O(n \log n)$ implementaion as future work.

6. Conclusions and Future Work

The main contribution of the paper is a framework for mesh deformation and curvilinear mesh generation based on calculus of variations. We have shown how RBF interpolation with the thin plate kernel is close to meeting the objectives of a mapping function based on our framework. Clearly, developing an "ideal" algorithm for mesh deformation based on the framework is a pending task. We plan to explore numerical methods for problem in calculus of variations and develop a more robust algorithm for mesh deformation. The mapping from the algorithm should minimize the ratio of a second-order partial derivative and a first-order partial derivative at all points, especially near poor-quality elements.

In addition, in order to get the RBF-based algorithm to scale for large meshes, preconditioners and approximation schemes must be considered. Implementation of these techniques should result in more successful mesh deformation algorithms. As mentioned before, their application may result in suboptimal locations of the mesh vertices as the locations of the vertices are not approximations of the RBF interpolant. Thus, additional techniques must be developed to adaptively obtain greater accuracy in the solution.

Also, for curvilinear mesh generation, typically, only a few layers of element near the boundary are curved. Thus, compactly supported kernels might be sufficient for this purpose. The development of algorithms for this specific purpose is also an interesting direction of research. In application requiring boundary layer elements, the vertices near the boundary have to be moved in the direction of the boundary normal. This may be achieved using generalized Hermite interpolation techniques. We will study and develop these techniques in the near future.

Acknowledgment

The work of the first author was supported in part by the NIH/NIGMS Center for Integrative Biomedical Computing grant 2P41 RR0112553-12 and a grant from the ExxonMobil corporation. The first and second authors were supported in part by the FA9550-15-C-0027 grant. The third author was support in part by the ARO W911NF1210375 grant (Program Manager: Dr. Mike Coyle). The authors would like to thank David Moxey and Michel Turner from Imperial College, London for their help with the code in Nektar++. The authors would like to thank Per-olof Persson from University of California Berkeley for discussions connected to this paper. The authors would also like to thank Christine Pickett, an editor at the Scientific Computing and Imaging Institute at the University of Utah, for proofreading and finding numerous typos and errors in one of the drafts of the paper.

References

[1] R. Abgrall, C. Dobrzynski, and A. Froehly. A method for computing curved 2D and 3D meshes via the linear elasticity analogy: preliminary results. Research Report RR-8061, INRIA, Sept. 2012.

[2] T. J. Baker. Mesh movement and metamorphosis. *Engineering with Computers*, 18(3):188–198, 2002.

[3] R. Beatson, J. Cherrie, and C. Mouat. Fast fitting of radial basis functions: Methods based on preconditioned gmres iteration. *Advances in Computational Mathematics*, 11(2-3):253–270, 1999.

[4] R. K. Beatson, W. A. Light, and S. Billings. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM Journal of Scientific Computing*, 22(5):1717–1740, May 2000.

[5] R. K. Beatson and M. J. D. Powell. An iterative method for thin plate spline interpolation that employs approximations to lagrange functions. In *Numerical Analysis*, pages 17–39, 1994.

[6] C. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. D. Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. Kirby, and S. Sherwin. Nektar++: An open-source spectral/hp element framework. *Computer Physics Communications*, 192(0):205 – 219, 2015.

[7] W. Chen. New rbf collocation methods and kernel rbf with applications. In M. Griebel and M. Schweitzer, editors, *Meshfree Methods for Partial Differential Equations*, volume 26 of *Lecture Notes in Computational Science and Engineering*, pages 75–86. Springer Berlin Heidelberg, 2003.

[8] A. de Boer, M. van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers & Structures*, 85(11-14):784 – 795, 2007. Fourth {MIT} Conference on Computational Fluid and Solid Mechanics.

[9] C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structures*, 80(3-4):305 – 316, 2002.

[10] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163(14):231 – 245, 1998.

[11] A. Gargallo-Peiro, X. Roca, J. Peraire, and J. Sarrate. Defining quality measures for mesh optimization on parameterized cad surfaces. In X. Jiao and J.-C. Weill, editors, *Proceedings of the 21st International Meshing Roundtable*, pages 85–102. Springer Berlin Heidelberg, 2013.

[12] A. Gargallo-Peiro, X. Roca, J. Peraire, and J. Sarrate. Defining quality measures for validation and generation of high-order tetrahedral meshes. In J. Sarrate and M. Staten, editors, *Proceedings of the 22nd International Meshing Roundtable*, pages 109–126. Springer International Publishing, 2014.

[13] A. Gargallo-Peiro, X. Roca, and J. Sarrate. A surface mesh smoothing and untangling method independent of the cad parameterization. *Computational Mechanics*, 53(4):587–609, 2014.

[14] C. Geuzaine, A. Johnen, J. Lambrechts, J.-F. Remacle, and T. Toulorge. The generation of valid curvilinear meshes. In N. Kroll, C. Hirsch, F. Bassi, C. Johnston, and K. Hillewaert, editors, *IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach*, volume 128 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 15–39. Springer International Publishing, 2015.

[15] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, Dec. 1987.

[16] N. A. Gumerov and R. Duraiswami. Fast multipole method for the biharmonic equation in three dimensions. *Journal of Computational Physics*, 215(1):363 – 383, 2006.

[17] N. A. Gumerov and R. Duraiswami. Fast radial basis function interpolation via preconditioned krylov iteration. *SIAM Journal of Scientific Computing*, 29(5):1876–1899, Sept. 2007.

[18] B. Helenbrook. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, 56(7):1007–1021, 2003. cited By 64.

[19] F. Hindenlang, T. Bolemann, and C.-D. Munz. Mesh curving techniques for high order discontinuous galerkin simulations. In N. Kroll, C. Hirsch, F. Bassi, C. Johnston, and K. Hillewaert, editors, *IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach*, volume 128 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 133–152. Springer International Publishing, 2015.

[20] S. Jakobsson and O. Amoignon. Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers & Fluids*, 36(6):1119 – 1136, 2007.

[21] E. Kansa. Multiquadricsa scattered data approximation scheme with applications to computational fluid-dynamicsii solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers & Mathematics with Applications*, 19(89):147 – 161, 1990.

[22] Y. Liu, Z. Guo, and J. Liu. RBFs-MSA Hybrid Method for Mesh Deformation. *Chinese Journal of Aeronautics*, 25(4):500 – 507, 2012.

[23] Q. Lu, M. S. Shephard, S. Tendulkar, and M. W. Beall. Parallel mesh adaptation for high-order finite element methods with curved element geometry. *Engineering with Computers*, 30(2):271–286, 2014.

[24] X.-J. Luo, M. S. Shephard, R. M. O'Bara, R. Nastasia, and M. W. Beall. Automatic p-version mesh generation for curved domains. *Engineering with Computers*, 20(3):273–285, 2004.

[25] D. Moxey, D. Ekelschot, U. Keskin, S. Sherwin, and J. Peiró. A thermo-elastic analogy for high-order curvilinear meshing with control of mesh validity and quality. *Procedia Engineering*, 82(0):127 – 135, 2014. 23rd International Meshing Roundtable (IMR23).

[26] D. Moxey, M. Green, S. Sherwin, and J. Peiró. An isoparametric approach to high-order curvilinear boundary-layer meshing. *Computer Methods in Applied Mechanics and Engineering*, 283:636 – 650, 2015.

[27] A. Ovcharenko, K. Chitale, O. Sahni, K. E. Jansen, and M. S. Shephard. Parallel adaptive boundary layer meshing for cfd analysis. In X. Jiao and J.-C. Weill, editors, *Proceedings of the 21st International Meshing Roundtable*, pages 437–455. Springer Berlin Heidelberg, 2013.

[28] P. Persson and J. Peraire. Curved mesh generation and mesh refinement using lagrangian solid mechanics. In *AIAA Aerospace Sciences Meeting*, pages 1–11, 2009.

[29] P.-O. Persson and G. Strang. A simple mesh generator in MATLAB. *SIAM Review*, pages 329–345, 2004.

[30] S. P. Sastry, J. Kim, S. M. Shontz, B. A. Craven, F. C. Lynch, K. B. Manning, and T. Panitanarak. Patient-specific model generation and simulation for pre-operative surgical guidance for pulmonary embolism treatment. In Y. J. Zhang, editor, *Image-Based Geometric Modeling and Mesh Generation*, volume 3 of *Lecture Notes in Computational Vision and Biomechanics*, pages 223–249. Springer Netherlands, 2013.

[31] S. P. Sastry, S. M. Shontz, and S. A. Vavasis. A log-barrier method for mesh quality improvement and untangling. *Engineering with Computers*, 30(3):315–329, July 2014.

[32] M. S. Shephard, J. E. Flaherty, K. E. Jansen, X. Li, X. Luo, N. Chevaugeon, J.-F. Remacle, M. W. Beall, and R. M. O'Bara. Adaptive mesh generation for curved domains. *Applied Numerical Mathematics*, 52(23):251 – 271, 2005.

[33] S. J. Sherwin and J. Peiró. Mesh generation in curvilinear domains using high-order elements. *International Journal for Numerical Methods in Engineering*, 53(1):207–223, 2002.

[34] S. M. Shontz and S. A. Vavasis. A linear weighted laplacian smoothing framework for warping tetrahedral meshes. In *Proceedings of the 16th International Meshing Roundtable*, pages 147–158, 2004.

[35] S. M. Shontz and S. A. Vavasis. Analysis of and workarounds for element reversal for a finite element-based algorithm for warping triangular and tetrahedral meshes. *BIT Numerical Mathematics*, 50(4):863–884, 2010.

[36] S. M. Shontz and S. A. Vavasis. A robust solution procedure for hyperelastic solids with large boundary deformation. *Engineering with Computers*, 28(2):135–147, Apr. 2012.

[37] D. Sieger, S. Menzel, and M. Botsch. High quality mesh morphing using triharmonic radial basis functions. In X. Jiao and J.-C. Weill, editors, *Proceedings of the 21st International Meshing Roundtable*, pages 1–15. Springer Berlin Heidelberg, 2013.

[38] D. Sieger, S. Menzel, and M. Botsch. Constrained space deformation for design optimization. In *23rd International Meshing Roundtable (IMR23)*, volume 82, pages 114 – 126, 2014.

[39] M. L. Staten, S. J. Owen, S. M. Shontz, A. G. Salinger, and T. S. Coffey. A comparison of mesh morphing methods for 3D shape optimization. In W. R. Quadros, editor, *Proceedings of the 20th International Meshing Roundtable*, pages 293–311. Springer Berlin Heidelberg, 2012.

[40] P. Szaltys, J. Majewski, S. Gepner, and J. Rokicki. High-order 3D anisotropic hybrid mesh generation for high-reynolds number flows. In N. Kroll, C. Hirsch, F. Bassi, C. Johnston, and K. Hillewaert, editors, *IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach*, volume 128 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 79–100. Springer International Publishing, 2015.

[41] T. Toulorge, C. Geuzaine, J.-F. Remacle, and J. Lambrechts. Robust untangling of curvilinear meshes. *Journal of Computational Physics*, 254(0):8 – 26, 2013.

[42] M. Wicke, D. Ritchie, B. M. Klingner, S. Burke, J. R. Shewchuk, and J. F. O'Brien. Dynamic local remeshing for elastoplastic simulation. *ACM Transactions on Graphics*, 29(4):49:1–11, July 2010. Proceedings of ACM SIGGRAPH 2010, Los Angles, CA.

[43] D. Zeng and C. R. Ethier. A semi-torsional spring analogy model for updating unstructured meshes in 3D moving domains. *Finite Elements in Analysis and Design*, 41(11-12):1118 – 1139, 2005.