Curved Mesh Generation and Mesh Refinement using Lagrangian Solid Mechanics

Per-Olof Persson[*]

University of California, Berkeley, Berkeley, CA 94720-3840, U.S.A.

Jaime Peraire[†]

Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

We propose a method for generating well-shaped curved unstructured meshes using a nonlinear elasticity analogy. The geometry of the domain to be meshed is represented as an elastic solid. The undeformed geometry is the initial mesh of linear triangular or tetrahedral elements. The external loading results from prescribing a boundary displacement to be that of the curved geometry, and the final configuration is determined by solving for the equilibrium configuration. The deformations are represented using piecewise polynomials within each element of the original mesh. When the mesh is sufficiently fine to resolve the solid deformation, this method guarantees non-intersecting elements even for highly distorted or anisotropic initial meshes. We describe the method and the solution procedures, and we show a number of examples of two and three dimensional simplex meshes with curved boundaries. We also demonstrate how to use the technique for local refinement of non-curved meshes in the presence of curved boundaries.

I.   Introduction

The use of high order methods for computational mechanics has been emphasized in recent years. This has been motivated by the need to tackle problems where high accuracy is a requirement, but also by the emergence of unstructured discontinuous Galerkin finite element methods[1] which appear to combine the right mix of geometric flexibility, stability, and accuracy.

The main advantages of unstructured mesh methods using tetrahedral elements derive from the availability of automatic mesh generators.[2–5] These generators are capable of producing meshes of linear tetrahedra automatically from CAD data and hence give a major advantage over other more user-intensive approaches. For high order methods, the linear elements are not adequate since they produce a planar faceted representation of the geometry which negates the advantages of using high order methods.

The generation of high order meshes which conform to the boundary geometry is not a well resolved issue. When small isotropic elements are used it is often possible to just deform the boundary of the elements which are in contact with the curved boundary. However, this simple strategy fails when either the elements are large, as desired in high order methods, or the required grids are highly anisotropic, such as in boundary layers. In these situations, it is not sufficient to simply deform the edges of the elements that are on the boundary, but it is necessary to propagate the mesh deformation into the domain interior.

Previous work on curved mesh generation include Refs. 6–8, where algorithms for curvilinear meshing are proposed. These methods identify mesh entities that produce invalid elements, and eliminate the problems by a combination of local mesh refinements, edge and face swaps, and node relocations. In Ref. 9, various solutions were proposed, including hybrid meshing using prism elements close to the curved boundary, and a curvature based refinement procedure.

In this paper, we propose an approach to deform an automatically generated triangular mesh into a curved boundary conforming mesh. We use a nonlinear elasticity analogy, where the geometry of the domain

---

[*]Assistant Professor, Department of Mathematics, University of California, Berkeley, Berkeley CA 94720-3840. E-mail: persson@berkeley.edu. AIAA Member.

[†]Professor, Department of Aeronautics and Astronautics, MIT, 77 Massachusetts Avenue 37-451, Cambridge, MA 02139. E-mail: peraire@mit.edu. AIAA Associate Fellow.
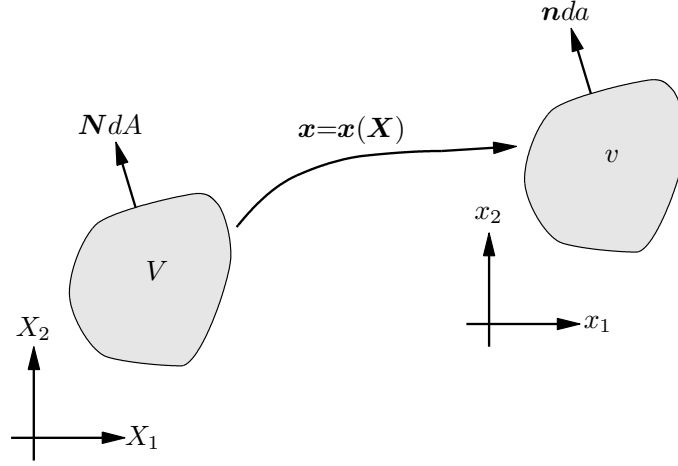
Figure 1. The mapping between the reference volume V and the physical volume v.

to be meshed is considered an elastic solid. By prescribing the boundary displacement to be that of the curved geometry, external loadings are obtained. The final configuration is determined by solving for the equilibrium configuration. The deformations are represented using continuous piecewise polynomials within each element of the original unstructured mesh. We describe the methods and show examples of successfully generated curved meshes involving highly curved boundaries, anisotropic boundary layers, and large number of elements.

II.   Problem Formulation

A.   Non-linear Lagrangian Formulation

We use a Lagrangian formulation,[10] where the solid configuration is defined by a mapping between the reference volume V and the physical volume v of the form $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{X})$, see Fig 1. Here, $\boldsymbol{x}$ is the equilibrium position vector of a point in the reference configuration with material coordinate $\boldsymbol{X}$. We differentiate $\boldsymbol{x}$ with respect to space to obtain the *deformation gradient tensor* $\boldsymbol{F}$:

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}. \tag{1}$$

Furthermore, we introduce the determinant $J = \det(\boldsymbol{F})$, which gives the ratio between a volume element in the reference and in the physical configuration. Conservation of mass requires that the density in the physical configuration ρ and in the reference configuration ρ_0 are related by $\rho J = \rho_0$. For a force equilibrium, we require that for for an arbitrary reference volume V,

$$\int_V \rho_0 \boldsymbol{b}\, dV + \int_{\partial V} \boldsymbol{t}\, dA = 0, \tag{2}$$

where $\boldsymbol{b}$ are the external volume forces per unit mass. The traction vector in the physical configuration $\boldsymbol{t}$ for an area element dA in the reference configuration is related to the unit normal $\boldsymbol{N}$ in the reference configuration by the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$:
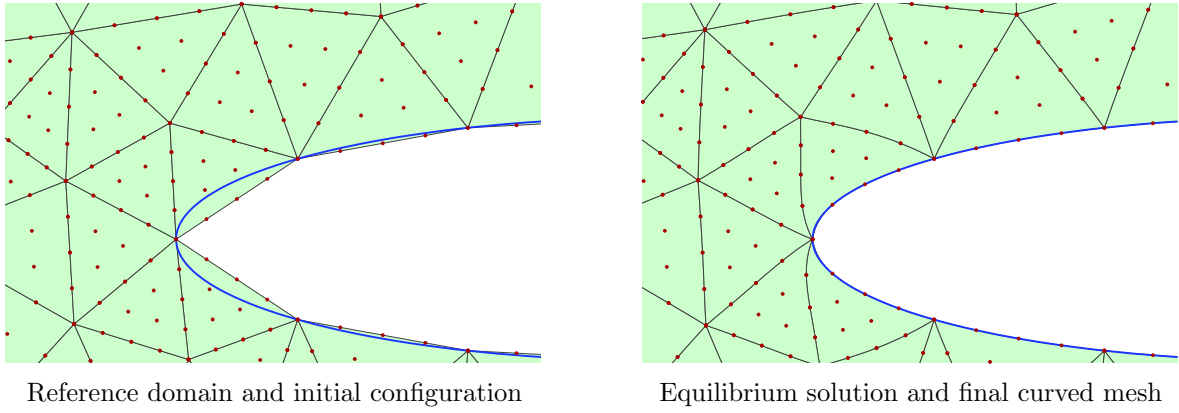
$$\boldsymbol{t} = \boldsymbol{P}\boldsymbol{N}. \tag{3}$$

Applying the divergence theorem to (2) and setting the external forces $\boldsymbol{b} = 0$ lead to the equilibrium condition

$$\nabla \cdot \boldsymbol{P} = 0, \tag{4}$$

where ∇ is the gradient operator in the reference space. We use a neo-Hookean constitutive model for $\boldsymbol{P}$ defined by[10]

$$\boldsymbol{P}(\boldsymbol{F}) = \mu((\boldsymbol{F}^T\boldsymbol{F})\boldsymbol{F}^{-T} - \boldsymbol{F}^{-T}) + \lambda(\ln J)\boldsymbol{F}^{-T}, \tag{5}$$

American Institute of Aeronautics and Astronautics

Reference domain and initial configuration   Equilibrium solution and final curved mesh

Figure 2. The solid mechanics approach to curved mesh generation. When the boundaries of the mesh are curved, the elements are deformed according to the equilibrium solution of a nonlinear elasticity problem.

with shear modulus μ and Lame constant λ. On the boundary ∂V of the domain we impose Dirichlet conditions:

$$\boldsymbol{x}(\boldsymbol{X}) = \boldsymbol{x}_{\mathrm{bnd}}(\boldsymbol{X}), \quad \boldsymbol{X} \in \partial V, \tag{6}$$

where $\boldsymbol{x}_{\mathrm{bnd}}(\boldsymbol{X})$ is a given prescribed position for the boundary of the physical domain.

B. Generating Curved Meshes using Solid Mechanics

In our curved mesh generation algorithm, we view the initial, straight-sided mesh as the undeformed configuration of a solid, see Fig. 2 (left). This is also the reference domain V. The external forces come from prescribing the position of the nodes on the boundary to be on the true curved geometry, resulting in the Dirichlet boundary conditions $\boldsymbol{x}_{\mathrm{bnd}}$ in (6). We solve (4) for an equilibrium configuration $\boldsymbol{x} = \boldsymbol{x}(X)$, and obtain the actual curved mesh as the deformed shape in the physical domain v, see Fig. 2, (right).

We note that our formulation requires Dirichlet conditions $\boldsymbol{x}_{\mathrm{bnd}}$ at all domain boundaries. In this work, these conditions are obtained with an element-wise local approach. For each element face located on a curved boundary, we either project the nodes to the true boundary or evaluate the node positions using mappings from a parameter space. For the three-dimensional surfaces, we apply some smoothing within each triangular element to improve the shapes.

In a more general setting, the boundary data would also have to be solved for using the elasticity analogy in order to avoid self-intersecting elements. We mention two possible solutions for this:

- A *bottom-up approach*, which is standard in most mesh generation algorithms, would treat the entities of the geometry in order of increasing dimension. The geometry vertices are first placed, then the curves are meshed, then the surfaces, and finally the volume. In this setting, it is natural to apply our elasticity technique individually at each level by solving for well-shaped elements in each parameter space. Distorted mapping would have to be compensated for by solving using a different metric, similar to the ones used for the actual mesh generation.[11]

  In practice, one would only apply the elasticity method on the surfaces and the volume, since the non-intersecting meshing of curves and vertices is straight-forward. This method also has the benefit that all subdomains are decoupled, resulting in higher performance and improved parallelization.

- Alternatively, one could relax the boundary conditions and allow for the boundary points to move along the true surface. The entire problem would then be solved using only one elasticity problem, which would determine node points inside the domain as well as on all the boundaries. This would likely produce better shaped curved elements, and the approach would be more consistent with the physical analogy of an elastic solid.

Finally, we note that the domain V can be chosen arbitrarily, which means that in particular only a subset of the elements could be used for the elasticity model. This can have important benefits, including

American Institute of Aeronautics and Astronautics

- Improved performance by only solving for elements that are likely to be highly curved. In our large three dimensional examples, we have used this technique and solve only for a layer of neighbors to the elements at the boundary.

- Non-curved elements have certain benefits when used with high-order finite element methods, since they produce a linear mapping between a reference element and the true element. This simplifies the assembly procedure, reduces the degrees of the integrals that have to be evaluated, and makes the elemental mass matrices scalar multiples of the reference mass matrix. By forcing a large number of elements to remain non-curved, the solver can take advantage of these properties.

III. Discretization and Solution Procedure

A. The Finite Element Formulation

To develop a finite element formulation for (4), we define the space of continuous piecewise polynomials of degree p:

$$V_h^p = \{\boldsymbol{v} \in [\mathcal{C}_0(\Omega)]^3 \mid \boldsymbol{v}|_K \in [\mathcal{P}_p(K)]^3 \ \forall K \in \mathcal{T}_h\}, \tag{7}$$

where the domain Ω is divided into elements $\mathcal{T}_h = \{K\}$, and $\mathcal{P}_p(K)$ is the space of polynomial functions of degree at most $p \geq 1$ on K. Furthermore, we define the subspaces of functions in V_h^p that satisfy the non-homogeneous Dirichlet boundary conditions:

$$V_{h,D}^p = \{\boldsymbol{v} \in V_h^p, \boldsymbol{v}|_{\partial V} = \boldsymbol{x}_{\text{bnd}}^p\}, \tag{8}$$

as well as the homogeneous Dirichlet boundary conditions:

$$V_{h,0}^p = \{\boldsymbol{v} \in V_h^p, \boldsymbol{v}|_{\partial V} = 0\}. \tag{9}$$

Here, $\boldsymbol{x}_{\text{bnd}}^p$ is a suitable projection of $\boldsymbol{x}_{\text{bnd}}$ onto the space of piecewise polynomials of order p defined over ∂V. By multiplying (4) by an arbitrary test function $\boldsymbol{z} \in V_{h,0}^p$, integrating over the domain V, and applying the divergence theorem, we obtain our finite element formulation: find $\boldsymbol{x}_h \in V_{h,D}^p$ such that for all $\boldsymbol{z} \in V_{h,0}^p$,

$$\int_V \boldsymbol{P}(\boldsymbol{F}(\boldsymbol{x}_h)) \cdot \nabla \boldsymbol{z} \, dV = 0. \tag{10}$$

The system of equations (10) is implemented using standard finite element techniques. The discrete solution vector $\bar{\boldsymbol{x}}$ and test function $\bar{\boldsymbol{z}}$ are represented at the nodes using nodal basis functions. The integrals are computed using high-order Gauss integration rules. The computed elemental residuals are assembled into a global discrete residual vector $\bar{\boldsymbol{r}}(\bar{\boldsymbol{x}})$, and a nonlinear system of equations

$$\bar{\boldsymbol{r}}(\bar{\boldsymbol{x}}) = 0 \tag{11}$$

is obtained. We solve this using a standard Newton method, where we start from an initial guess $\bar{\boldsymbol{x}}^{(0)}$ and iterate by

$$\bar{\boldsymbol{x}}^{(k+1)} = \bar{\boldsymbol{x}}^{(k)} - \boldsymbol{K}(\bar{\boldsymbol{x}}^{(k)})^{-1}\bar{\boldsymbol{r}}(\bar{\boldsymbol{x}}^{(k)}). \tag{12}$$

Here, the Jacobian matrix $\boldsymbol{K} = \partial \bar{\boldsymbol{r}}/\partial \bar{\boldsymbol{x}}$ is evaluated for each element and assembled into a global matrix, stored in the sparse compressed column format. The prescribed displacement at the boundary nodes is imposed by elimination of the corresponding variables from the system of equations.

The linear systems of equations that are obtained in the Newton iterations,

$$\boldsymbol{K}\Delta\bar{\boldsymbol{x}} = \boldsymbol{f} \tag{13}$$

are solved using a standard restarted GMRES algorithm,[12] preconditioned with a threshold incomplete LU factorization.[13,14] This provides sufficient performance for all our examples, which include fairly large problems. However, if required, a large variety of specialized solvers are available for these problems, including multigrid-based schemes and variants for parallel computers.[15]

American Institute of Aeronautics and Astronautics

B.   The Adaptive Newton-Krylov Solver

In simple cases, the nonlinear system of equations (11) can be solved using a standard Newton method as described above. However, in general more sophisticated methods are required. To see this, it is clear that in the first Newton iteration, the iterate $\boldsymbol{x}$ corresponds to the initial, straight-sided mesh, but with the Dirichlet boundary conditions imposed. This is equivalent to curving the boundaries but not the interior, which will generally lead to inverted elements and obvious numerical difficulties.

To obtain a robust global convergence in the non-linear solver, we use homotopy in a scalar variable α. We define a series of problems $\bar{\boldsymbol{r}}_\alpha(\bar{\boldsymbol{x}}_\alpha)$, parametrized by α, with Dirichlet boundary conditions

$$\boldsymbol{x}_{\mathrm{bnd}} = \alpha\boldsymbol{x}_0 + (1-\alpha)\boldsymbol{x}_{\mathrm{curved}}. \tag{14}$$

When $\alpha = 0$, the boundary conditions reduce to the initial locations $\boldsymbol{x}_0$, which are already satisfied by the initial configuration and require no solving. By slowly increasing α from 0 to 1, we approach the actual problem with boundary conditions $\boldsymbol{x}_{\mathrm{curved}}$. This results in a series of well-behaved problems that can be solved using regular Newton iterations.

In practice, the selection of the sequence of α-values can be automated in an adaptive way using simple control mechanisms. We control the increase $\Delta\alpha$ by monitoring the Newton convergence. If the method does not converge or inverted elements are found during the assembly process, then $\Delta\alpha$ is reduced. If the Newton method converges fast enough, $\Delta\alpha$ is increased. The final non-linear solution algorithm has the following form:

> **function $\boldsymbol{x} = AdaptiveNewton(\boldsymbol{x}_0)$**
> **Description**: *Solve $\boldsymbol{r}_\alpha(\boldsymbol{x}) = 0$ for $\alpha = 1$ by adaptive homotopy in α*
>
> $\boldsymbol{x} = \boldsymbol{x}_0$ *Initial configuration*
> $\alpha = 0,\ \Delta\alpha = 1$ *Start at 0, attempt full first step*
> **while** $\alpha < 1$
> $\boldsymbol{x}_{\mathrm{old}} = \boldsymbol{x},\ \alpha_{\mathrm{old}} = \alpha$ *Backup old solution*
> $\alpha = \min(\alpha + \Delta\alpha, 1)$ *Advance α*
>   Solve $\boldsymbol{r}_\alpha(\boldsymbol{x}) = 0$ using Newton's method   *Terminate early if NaNs or Infs*
>   **if** Newton convergence
> $\Delta\alpha = 2\Delta\alpha$ *Accept step, increase stepsize*
>   **else**
> $\boldsymbol{x} = \boldsymbol{x}_{\mathrm{old}},\ \alpha = \alpha_{\mathrm{old}}$ *Revert to old solution*
> $\Delta\alpha = \Delta\alpha/2$ *Decrease stepsize*
>   **if** $\Delta\alpha < \alpha_{\mathrm{min}}$
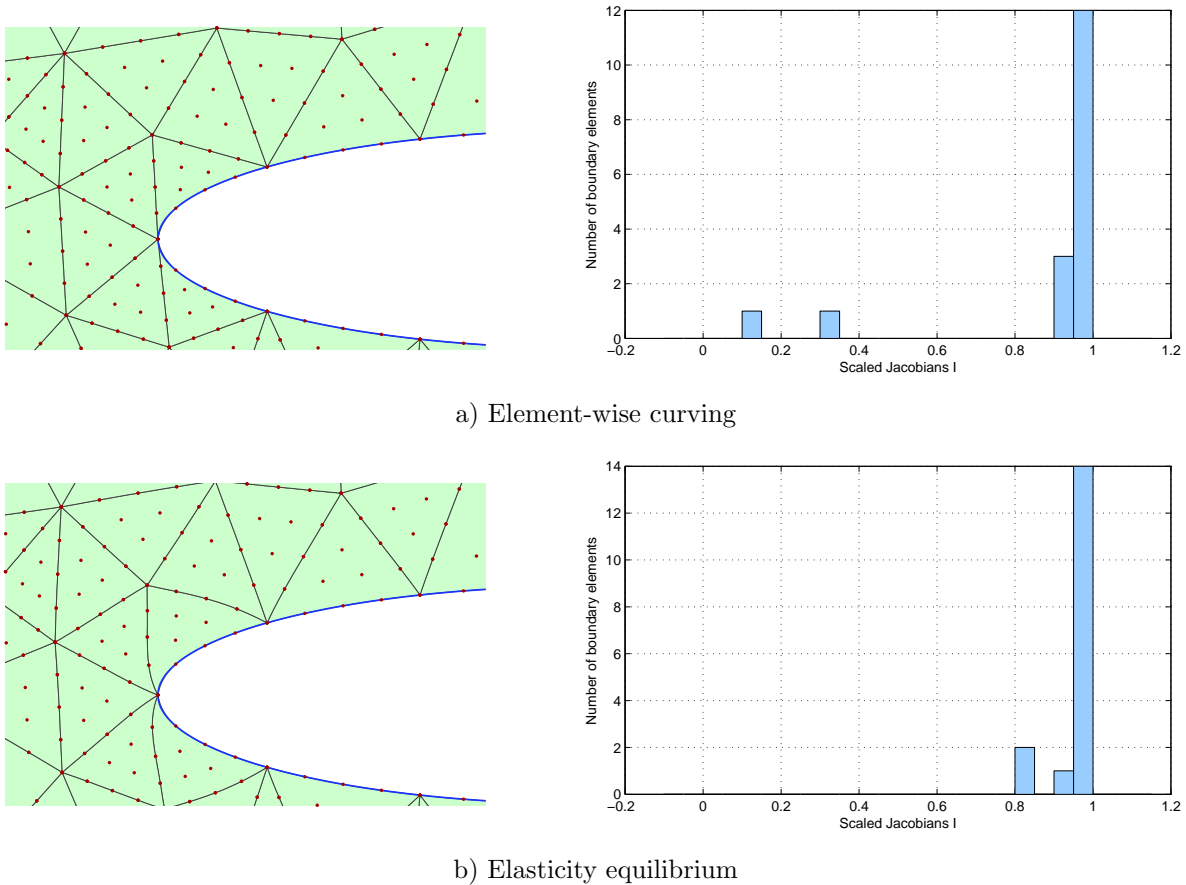>   Solver failed   *No further increase possible*

This simple strategy can certainly be improved upon, but it does contain the essential ingredients of adaptive parameter stepping based on feedback from the Newton convergence.

IV.   Results

A.   Curved Mesh Generation

We now show a number of two and three dimensional meshes with curved boundaries that are generated using the proposed technique. In all examples, we use polynomial approximations of degree $p = 4$ and set Poisson's ratio $\nu = 0.4$. To measure how much an element has been deformed, we use the scaled Jacobian measure[6]

$$I = \frac{\min_{\boldsymbol{X} \in \kappa} J(\boldsymbol{X})}{\max_{\boldsymbol{X} \in \kappa} J(\boldsymbol{X})} \tag{15}$$

American Institute of Aeronautics and Astronautics

a) Element-wise curving

b) Elasticity equilibrium

Figure 3. A simple example of isotropic two dimensional curved mesh generation. A simple element-wise algorithm generates non-inverting elements (top), but the elasticity equilibrium produces larger scaled Jacobians which means better shaped elements.

where $J(\boldsymbol{X}) = \det(\partial\boldsymbol{x}/\partial\boldsymbol{X})$ is the Jacobian of the mapping from the reference coordinate $\boldsymbol{X}$ to the physical coordinate $\boldsymbol{x}$. We note that $I \leq 1$, and $I = 1$ for a straight-sided simplex element (since $J(\boldsymbol{X})$ is then a constant).

Note that the definition of the scaled Jacobian I is independent of the *element quality* of the corresponding mesh element.[16] Even a poorly shaped element will have the optimal $I = 1$ if it is straight-sided. However, as I gets negative or small, the effect of the curving is that the element gets inverted or close to being degenerate. This typically decreases the quality of discretizations on the mesh and makes the corresponding systems of equations ill-conditioned.

1. Simple triangular mesh

Our first example is a simple isotropic mesh with well-shaped triangular elements. Fig. 3 shows a zoom-in around the curved boundary. Two strategies are compared – a simple element-wise curving of the boundary elements (top) and the elasticity equilibrium (bottom). Although all the scaled Jacobians I are positive for both approaches, they are much closer to 1 using the solid mechanics method.

2. Anisotropic triangular mesh

Next, we show an anisotropic mesh with element ratios as high as 100:1, see Fig. 4. Here it is clear that an element-wise approach would not work, but all of the elements in the anisotropic boundary layer mesh must be curved.

American Institute of Aeronautics and Astronautics

Original mesh	Curved mesh
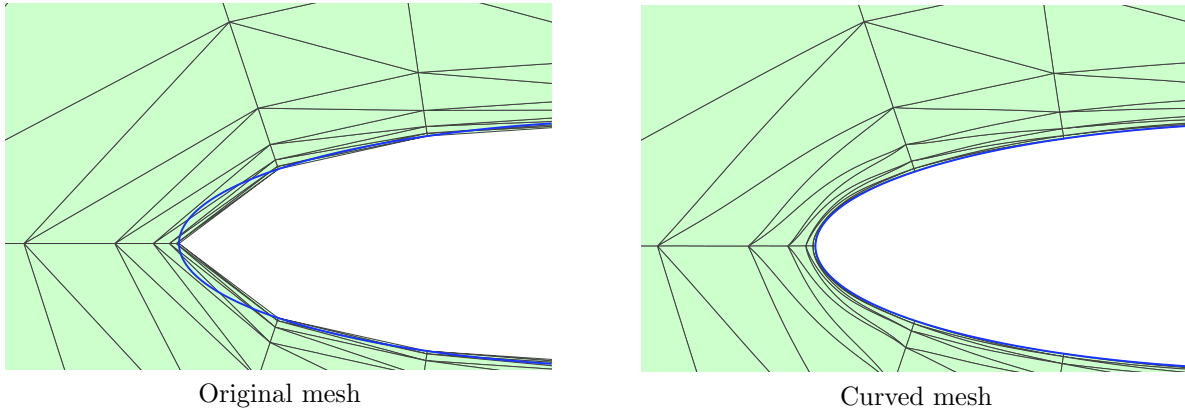
Figure 4. Anisotropic curved mesh generation. Note that the curved boundaries must affect a large layer of elements inside the domain in order to avoid inverted elements.

3. Tetrahedral mesh around a sphere

In Fig. 5, a tetrahedral mesh of the exterior of a sphere is shown. The volume mesh was generated using a standard Delaunay refinement method, and although the tetrahedra are well-shaped, a simple element-wise curving would not work. With our elasticity method, well-shaped curved elements are produced. We also show the convergence history of the adaptive Newton solver, showing that it required 4 steps to obtain the final mesh and highlighting the benefits of using an adaptive approach to select the sequence of homotopy parameters α.

4. Tetrahedral mesh of cylindrical component

In Fig. 6, an unstructured tetrahedral mesh of a cylindrical component is shown. We note that the method handles thin structures, even with only a single layer of elements, and it produces well-shaped curved elements.

5. Tetrahedral mesh of Falcon aircraft

In our final example we consider a complex, unstructured, real-world mesh as shown in Fig. 7. The mesh is relatively coarse, although for high-order methods it is a rather realistic inviscid mesh. We note that the highly distorted elements are located around edges of high curvature, but the elasticity equilibrium mesh does not have any inverted elements. The adaptive Newton solver required only 7 steps to reach the true curved boundary at $\alpha = 1$.

B. Local Mesh Refinement for Curved Meshes

As another application of our technique, we illustrate the use of the mesh deformation approach to perform local refinement of unstructured meshes in the presence of curved boundaries. Note that we are now considering regular linear elements, and apply the elasticity method with piecewise linear approximating functions. The approach is illustrated in Fig. 8. An original mesh is refined using standard local mesh refinement techniques,[17] without consideration of the true, curved boundary. The newly created nodes located on the boundary are then constrained to the true boundary, and a well-shaped non-inverting mesh is solved for using the solid equilibrium.

The procedure is straight-forward and it requires no additional developments for the refinement, since it is performed on a polygonal or polyhedral geometry. It is even capable of refining unstructured anisotropic meshes, without requiring unnecessary refinements along the boundaries. Standard mesh smoothing algorithms can of course be applied on the final mesh, which is boundary conforming and well-shaped from using the elasticity approach.

American Institute of Aeronautics and Astronautics

a) Original mesh

b) Curved mesh

c) Convergence of the adaptive Newton solver

d) Scaled element Jacobians (boundary elements)
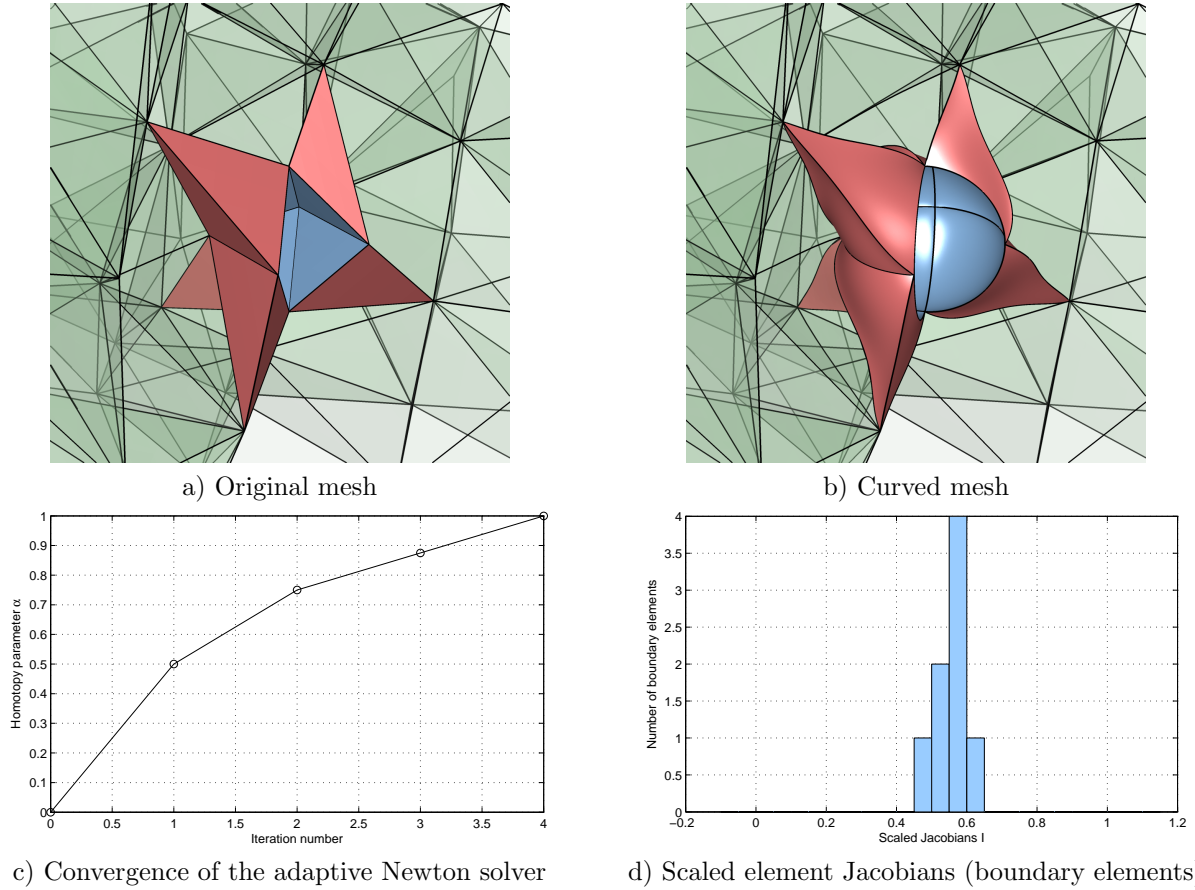
Figure 5. Curved mesh of the exterior of a sphere, using polynomials of degree $p = 4$. Even for this well-shaped isotropic mesh, it is essential to allow for curved interior edges to avoid inverted elements.

V.   Conclusions

We have presented a new technique for generation of curved meshes based on a nonlinear elasticity analogy. The method is highly resistant to inverted elements and typically produces very well-shaped elements. Another benefit is that it curves the original mesh, without introducing new nodes or modifying the mesh topology. Finally, we showed how the same framework can be used for the problem of locally refining an unstructured mesh close to curved boundaries.

We have applied our technique on many problems with excellent results, and future work include the development of the techniques mentioned in section II for obtaining the boundary data, and possibly improving the performance of the solver with parallelization and other solution algorithms.

References

[1]Cockburn, B. and Shu, C.-W., "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems," *J. Sci. Comput.*, Vol. 16, No. 3, 2001, pp. 173–261.

[2]Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations," *J. Comput. Phys.*, Vol. 72, No. 2, 1987, pp. 449–466.

[3]Ruppert, J., "A Delaunay refinement algorithm for quality 2-dimensional mesh generation," *J. Algorithms*, Vol. 18, No. 3, 1995, pp. 548–585.

[4]Shewchuk, J. R., "Delaunay refinement algorithms for triangular mesh generation," *Comput. Geom.*, Vol. 22, No. 1-3, 2002, pp. 21–74.

[5]Owen, S. J., "A Survey of Unstructured Mesh Generation Technology," *Proceedings of the 7th International Meshing Roundtable*, Sandia Nat. Lab., 1998, pp. 239–267.

[6]Dey, S., O'Bara, R., and Shephard, M., "Curvilinear mesh generation in 3D," *Comput. Aided Geom. Design*, Vol. 33, 2001, pp. 199–209.

a) Original mesh

b) Curved mesh

c) Curved mesh, split view

d) Scaled element Jacobians
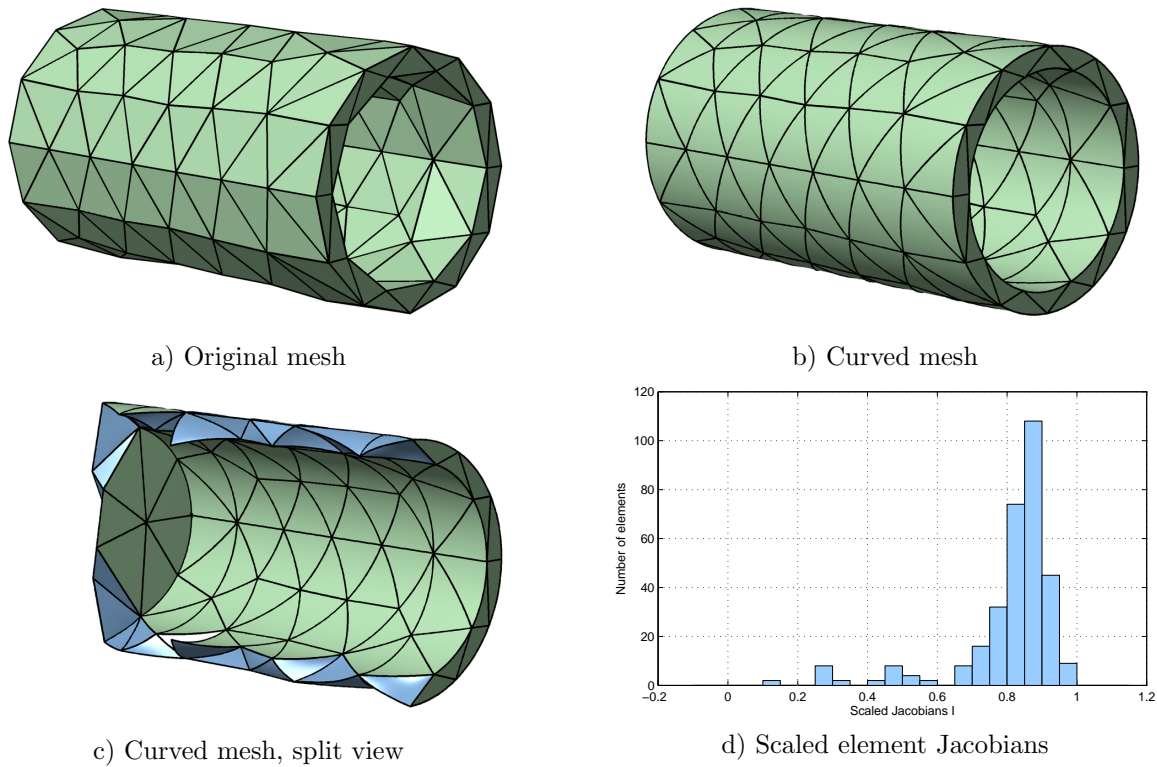
Figure 6. Curved mesh of a cylindrical component, with polynomial degrees $p = 4$. The method easily handles thin structures, even with only a single element across the thickness.

[7]et al., X. L., "Automatic p-version mesh generation for curved domains," *Eng. Comput.*, Vol. 20, No. 3, 2004, pp. 273–285.

[8]Shephard, M. S., Flaherty, J. E., Jansen, K. E., Li, X., Luo, X., Chevaugeon, N., Remacle, J.-F., Beall, M. W., and O'Bara, R. M., "Adaptive mesh generation for curved domains," *Appl. Numer. Math.*, Vol. 52, No. 2-3, 2005, pp. 251–271.

[9]Sherwin, S. J. and Peiró, J., "Mesh generation in curvilinear domains using high-order elements," *Internat. J. Numer. Methods Engrg.*, Vol. 53, No. 1, 2002, pp. 207–223.

[10]Bonet, J. and Wood, R. D., *Nonlinear continuum mechanics for finite element analysis*, Cambridge University Press, Cambridge, 2nd ed., 2008.

[11]Borouchaki, H., George, P. L., Hecht, F., Laug, P., and Saltel, E., "Delaunay mesh generation governed by metric specifications. I. Algorithms," *Finite Elem. Anal. Des.*, Vol. 25, No. 1-2, 1997, pp. 61–83.

[12]Saad, Y. and Schultz, M. H., "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, Vol. 7, No. 3, 1986, pp. 856–869.

[13]Meijerink, J. A. and van der Vorst, H. A., "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix," *Math. Comp.*, Vol. 31, No. 137, 1977, pp. 148–162.

[14]Saad, Y., *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2nd ed., 2003.
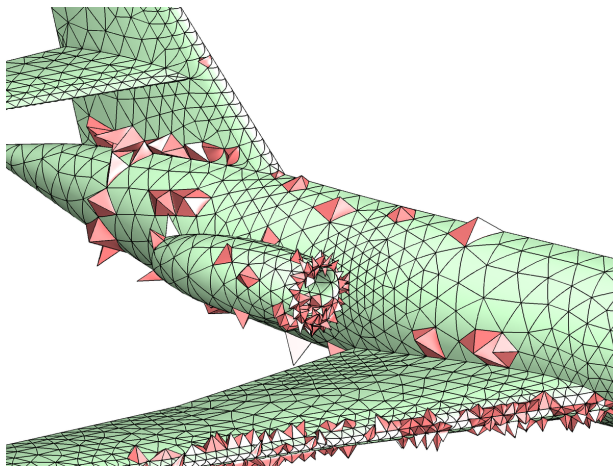
[15]Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., "PETSc home page," www.mcs.anl.gov/petsc.

[16]Field, D. A., "Qualitative measures for initial meshes," *Internat. J. Numer. Methods Engrg.*, Vol. 47, 2000, pp. 887–906.
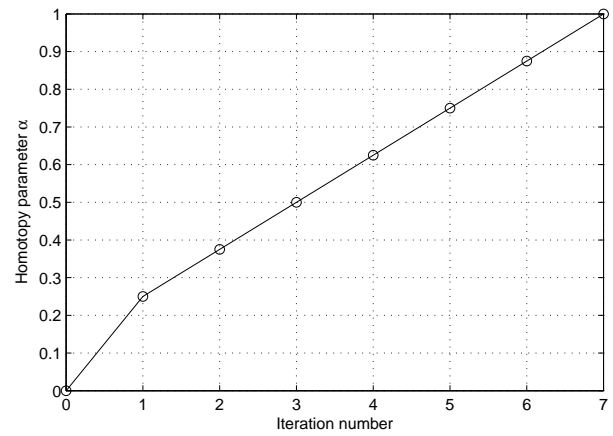
[17]Rivara, M.-C., "Mesh refinement processes based on the generalized bisection of simplices," *SIAM J. Numer. Anal.*, Vol. 21, No. 3, 1984, pp. 604–613.
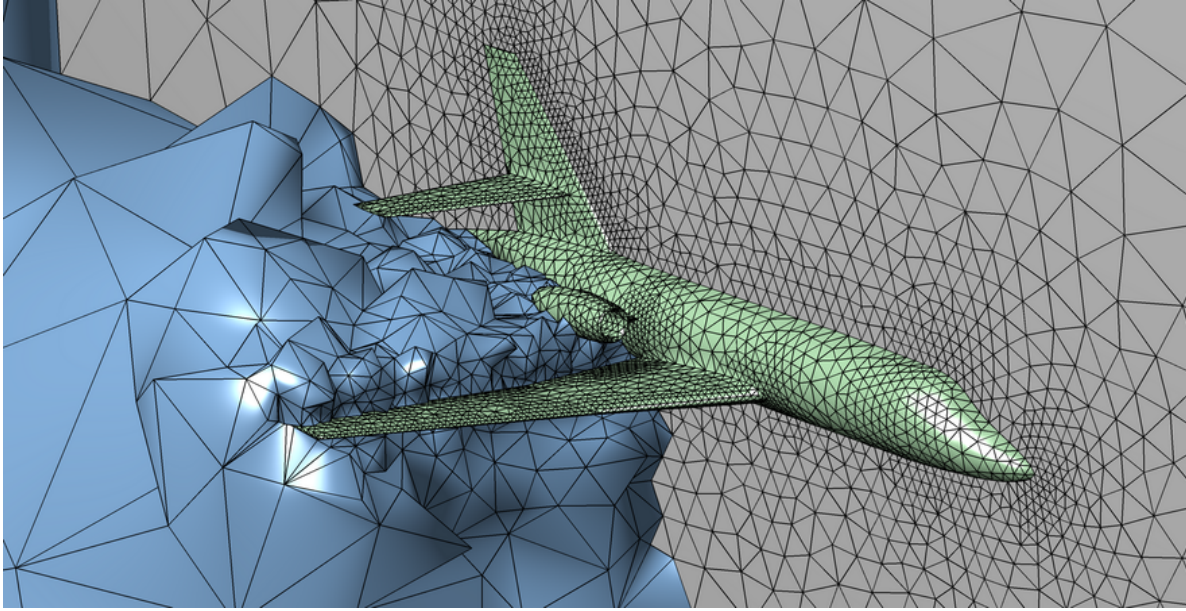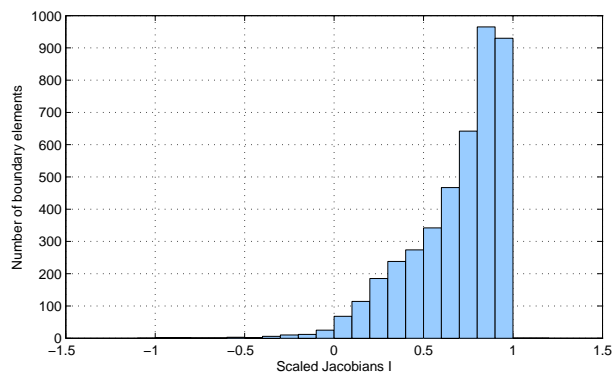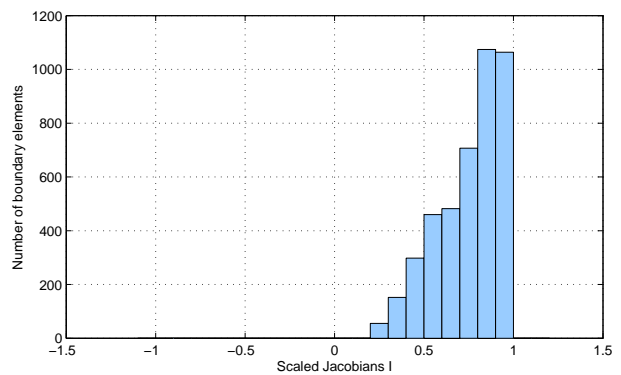
a) Original mesh

b) Curved mesh, elements with $I < 0.5$

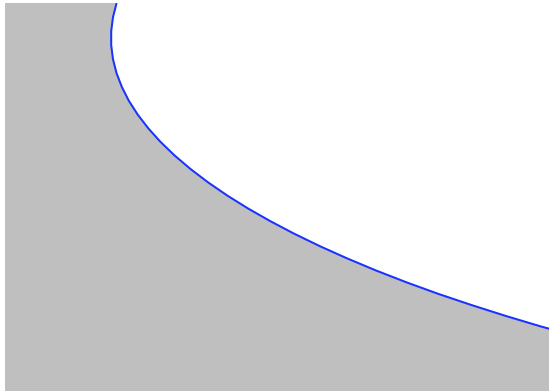
c) Convergence of the adaptive Newton solver

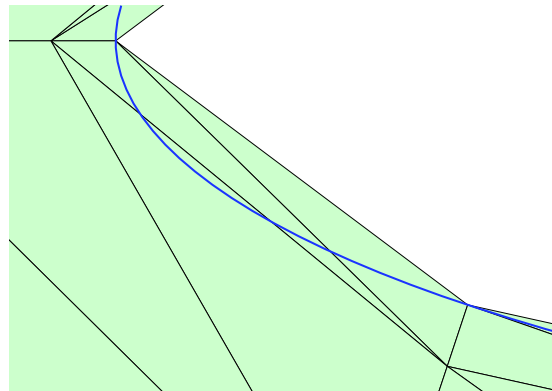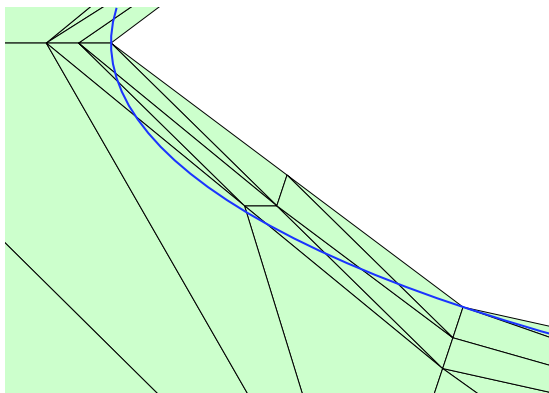d) Scaled Jacobians, local curving only

e) Scaled Jacobians, elasticity equilibrium

Figure 7. Curved mesh of a Falcon aircraft. Note that with only local curving of the boundary elements, many elements invert ($I < 0$).
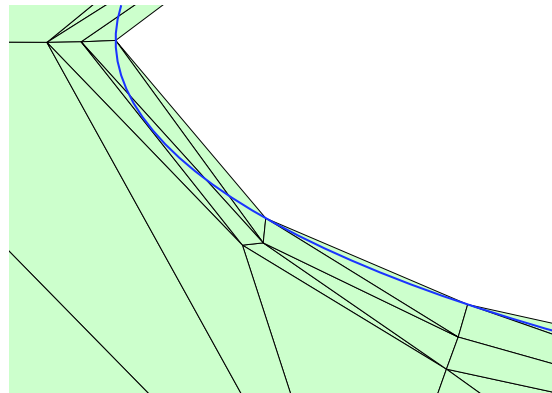
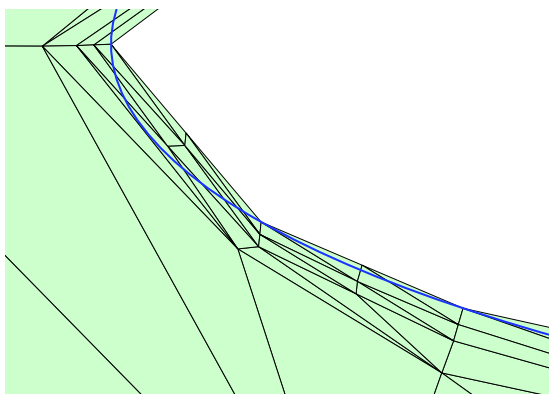American Institute of Aeronautics and Astronautics
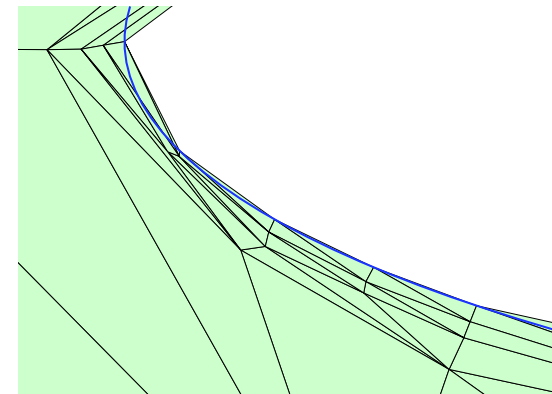
Geometry

Initial mesh

Local refinement of non-curved mesh

Curved boundary and equilibrium solution

Local refinement of non-curved mesh

Curved boundary and equilibrium solution

Figure 8. The solid mechanics approach to local mesh generation with curved boundaries. The interior nodes are determined by the equilibrium solution of a nonlinear elasticity problem, which avoids inverted elements at the curved boundaries.