Mesquite Mesh Optimization Toolkit

Brian Miller, Lori Diachin

Computation Directorate, LLNL

LLNL-PRES-492592

Why worry about mesh quality?

- Mesh quality impacts both:
  - Solution Efficiency
    - Iteration count may grow as a function of minimum angle
    - Number of degrees of freedom will impact time to solution
  - Solution Accuracy
    - Truncation errors usually are a function of mesh geometry.
    - Some applications will have unique sensitivities to geometry.

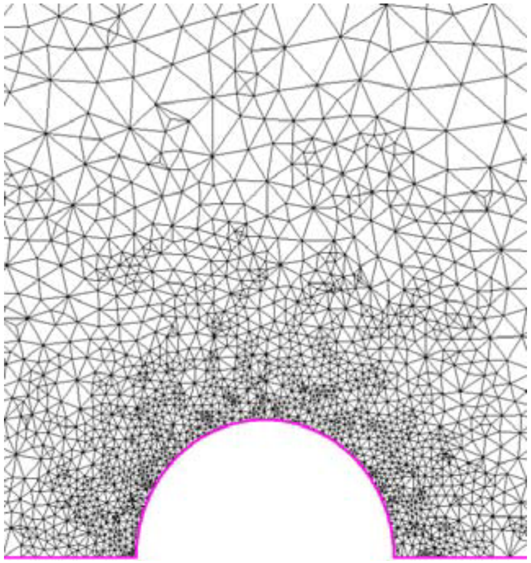
The *application* really defines what is a 'good' mesh.

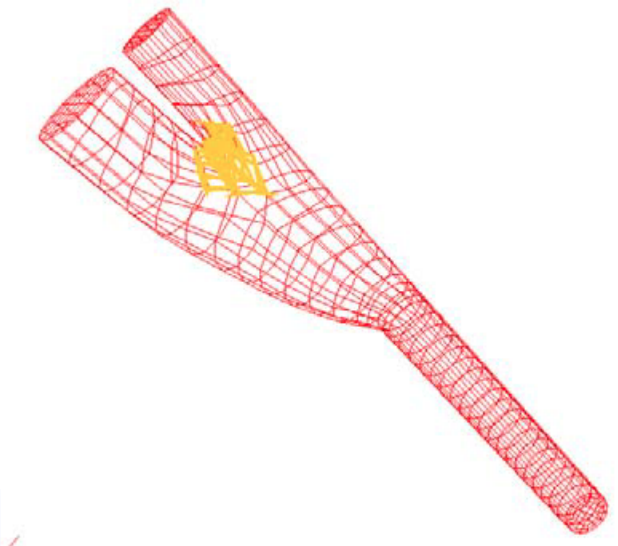
Mesh quality can impact time to solution

•Arteriovenous Graft Turbulent Flow Simulation (Knupp, Fischer 2000)
•Compute maximum shear stress with high order spectral methods
Poorly-shaped Elements Increase CG Solver Iterations

•Mesh Optimized by Condition Number
•reduced maximum number of solver iterations from 169 to 150
•reduced the average condition number from 18.06 to 15.46 (about a 17% savings).

Four hours of applications solver time was traded for 19 minutes of mesh smoothing time.





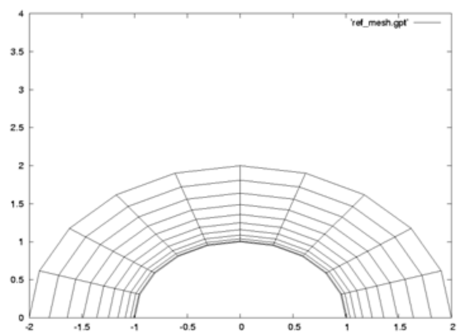
•Compressible flow test (Freitag, Ollivier-Gooch, 1998)

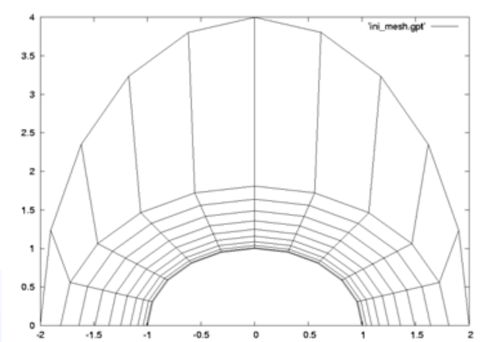
•Mesh optimized with Active Set solver
•Improved convergence rate by 25%

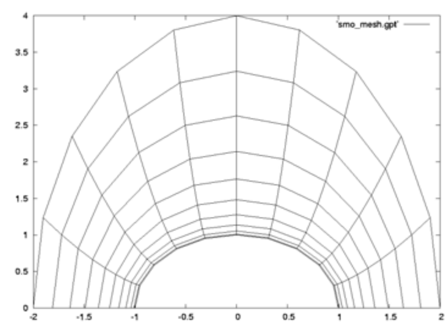
Cost less than one multigrid iteration.

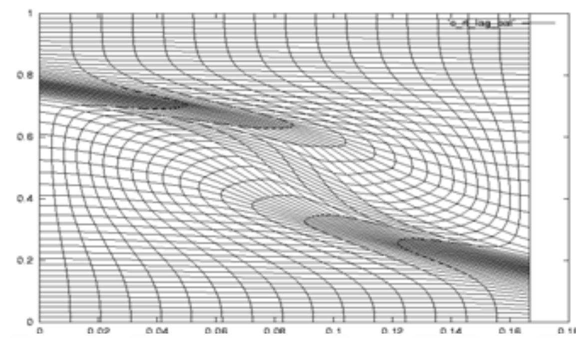
Mesh alignment can improve simulation results

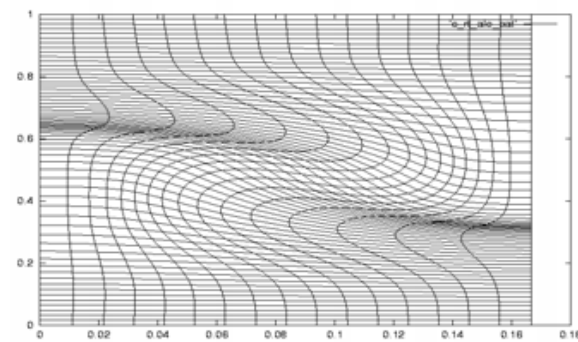
- •Aligning vertices or elements to a vector field
- •Improving ALE mesh quality while preserving some flow characteristics
- •Adapting mesh to a deforming boundary





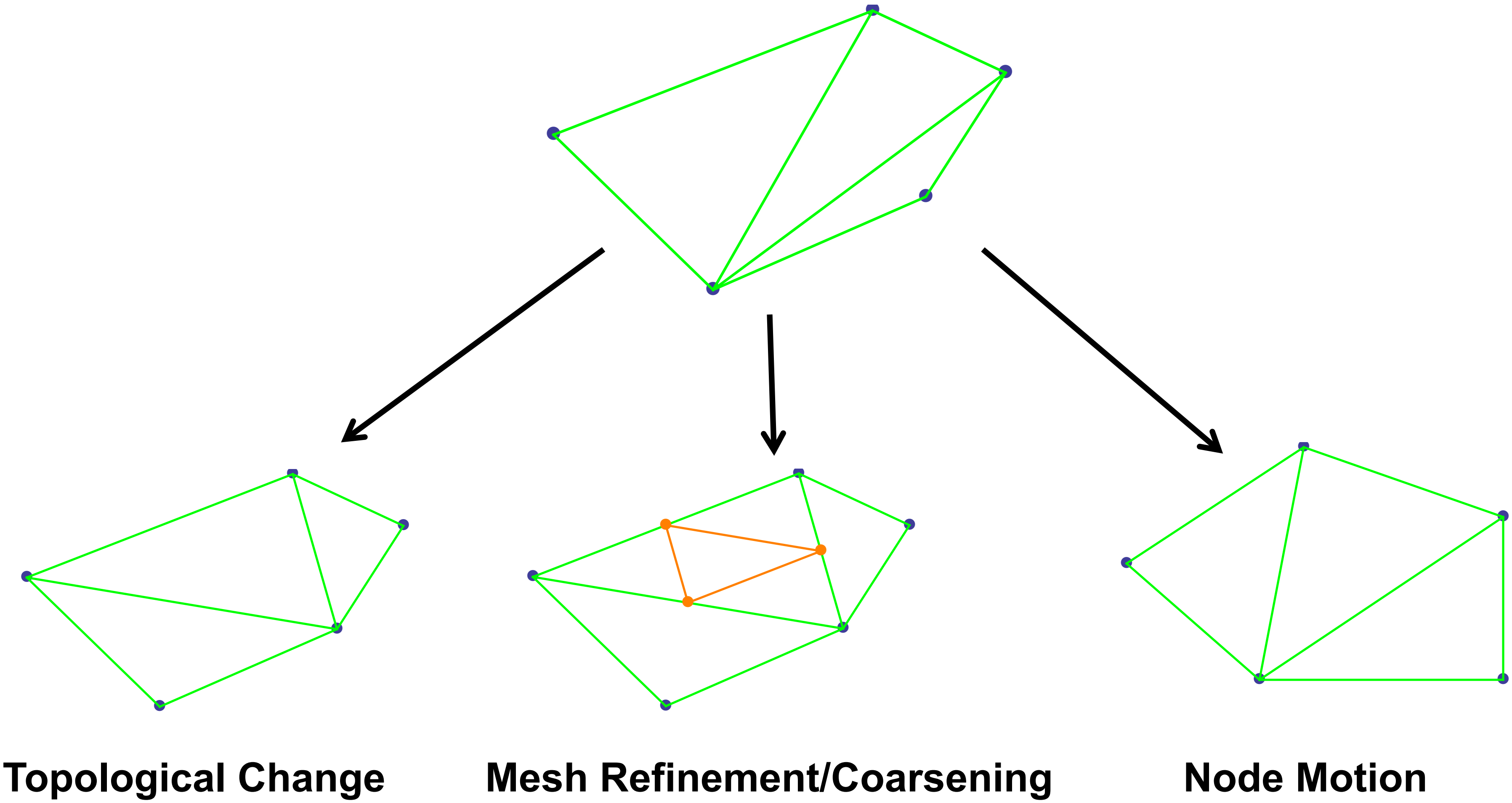






Knupp,
Shashkov,
Garimella, (2000)

Methods of mesh modification



Smoothing versus Optimization

"Smoothing is a procedure for improving mesh quality via a node-movement strategy in which a non-linear system of equations is solved."

Discrete equation:

$$\mathbf{f}(\ldots, \mathbf{x}_i, \ldots) = 0$$

Picard Iteration:

$$\mathbf{x}_i^{n+1} = \mathbf{g}(\ldots, \mathbf{x}_i^n, \ldots)$$

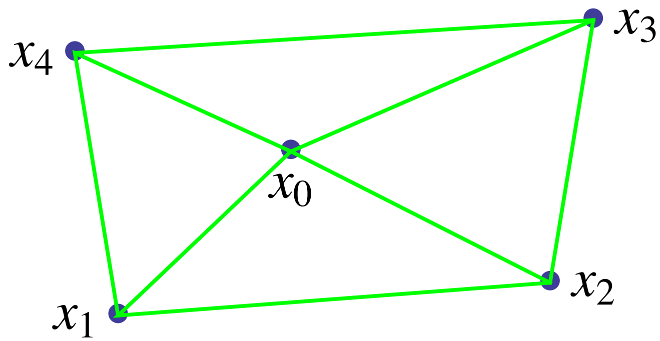
"Mesh quality optimization is the process of changing nodal positions to find the extremae of some scalar objective function that measures one or more aspects of mesh quality."

Minimization problem:

$$\mathcal{F} = \frac{1}{N} \sum_{i=1}^{N} \mu(\ldots, \mathbf{x}_i, \ldots)$$

Connection between Optimization & Smoothing: For unconstrained optimization, extremae of the objective function occur where the gradient is zero. Setting the gradient of the objective function to zero yields a non-linear set of equations that can result in a smoothing scheme.

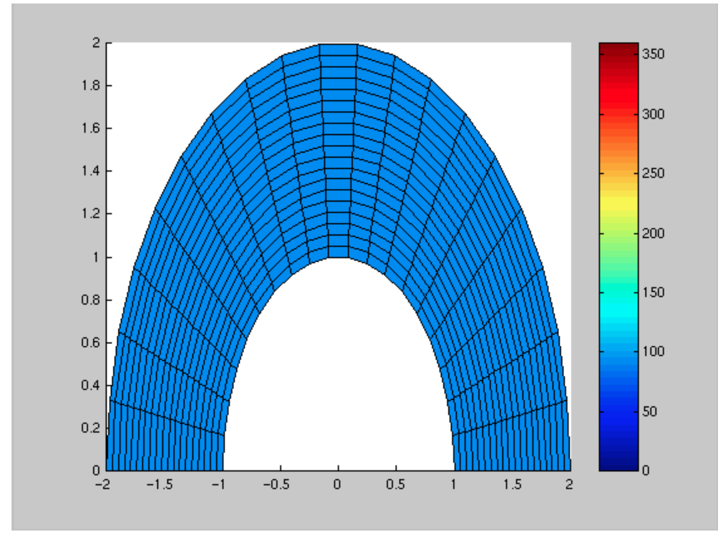
Laplace smoothing details

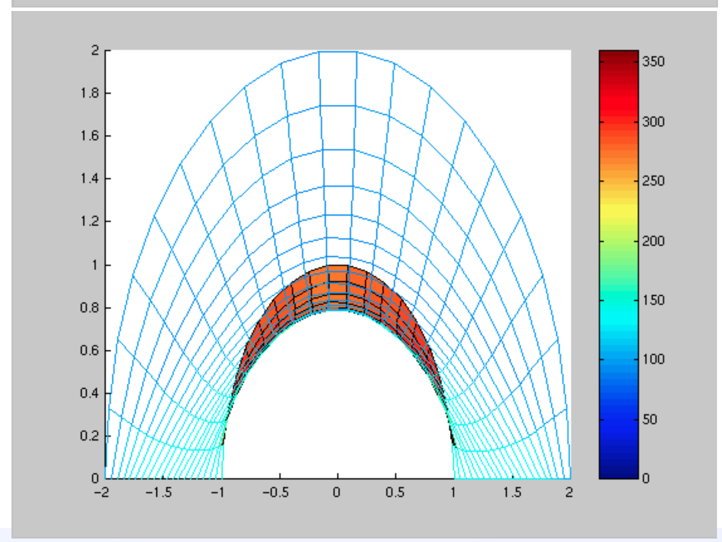


$$\mathbf{x}_0^{n+1} = \frac{1}{N}\sum_{i=1}^{n} \mathbf{x}_i^n$$

- Improvement not guaranteed
- Unmodified use can tangle mesh

- Easy to implement
- Can be extended in various ways to improve robustness





Mesquite is based on sound mathematical principles

- Mesh quality improvement posed as an optimization problem
- Element Quality: $q_i(\mathbf{x}), i = 1, \ldots, n_s$
  - A function of the vertex locations
  - Can be vertex or element based
- Mesh quality objective function

$$F(\mathbf{x}) = f(q_i(\mathbf{x})), i = 1, \ldots, n_s$$

  - A function of element quality metrics defined on some subset of the free vertices
- Optimization problem

$$\min F(x)$$

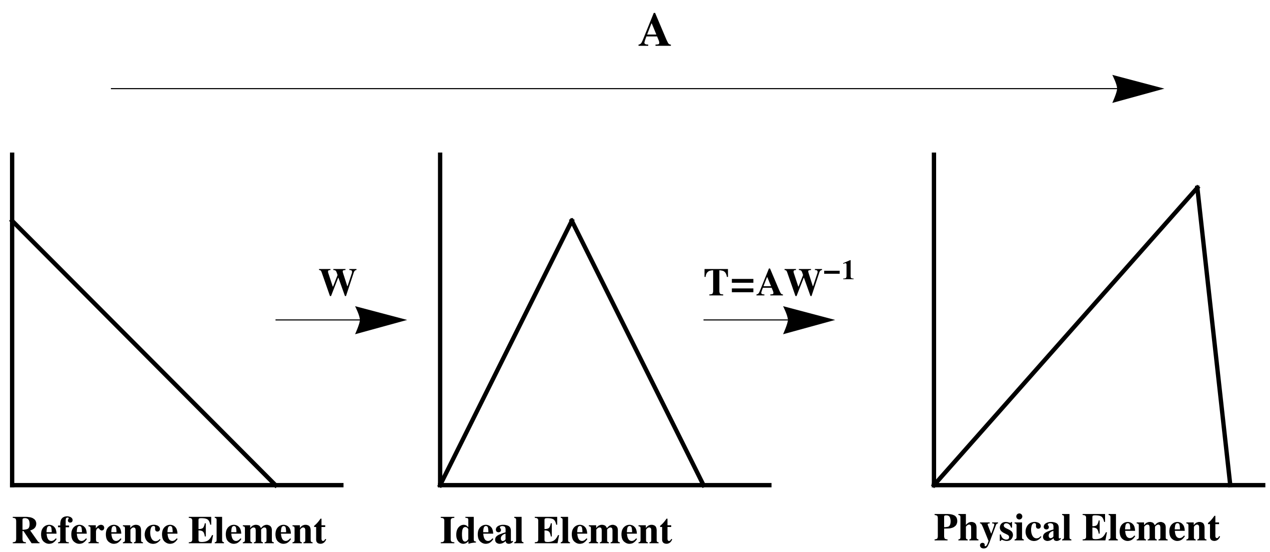
Element shape, s_i
0 = degenerate
1 = ideal

$F = 1/n \sum 1/s_i$

min F

Target Matrix Paradigm (TMP)

Element characteristics are well represented by Jacobian matrices.



- A (Active Matrix) maps index to physical space.
- W (Target Matrix) maps ideal to reference element.
- T then maps ideal element to physical element.

TMP Metric Construction

- Target metrics can be constructed which preserve size, shape, orientation or combinations of these qualities.

$$\mu(\vec{x}) = |T|^2 - 2 \cdot \det(T)$$

2D Shape Metric

$$\mu(\vec{x}) = |T - I|^2$$

2D Shape+Size+Orient (SSO) Metric

- Barrier versions of the metrics prevent element inversion.

$$\mu(\vec{x}) = \frac{|T|^2}{2 \cdot \det(T)} - 1$$

2D Shape Barrier Metric

$$\mu(\vec{x}) = \frac{|T - I|^2}{2 \cdot \det(T)}$$

2D SSO Barrier Metric

The Minimization Problem

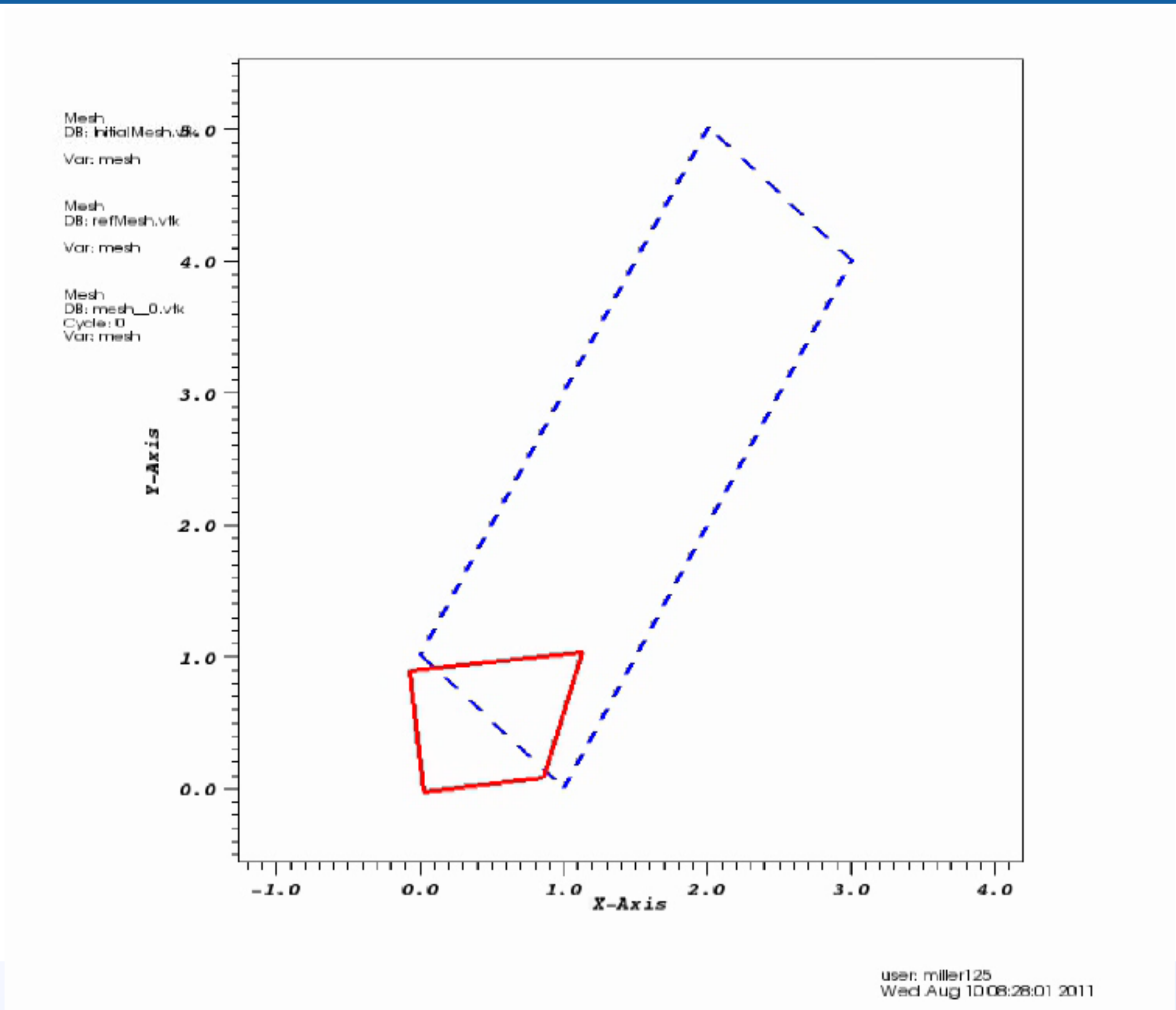
The Function to minimize:

$$F(\mathbf{x}) = \left[\frac{1}{N} \sum_{k=1}^{N} \left(c_k \cdot \mu(T_k(\vec{x_k})) \right)^p \right]^{\frac{1}{p}}$$

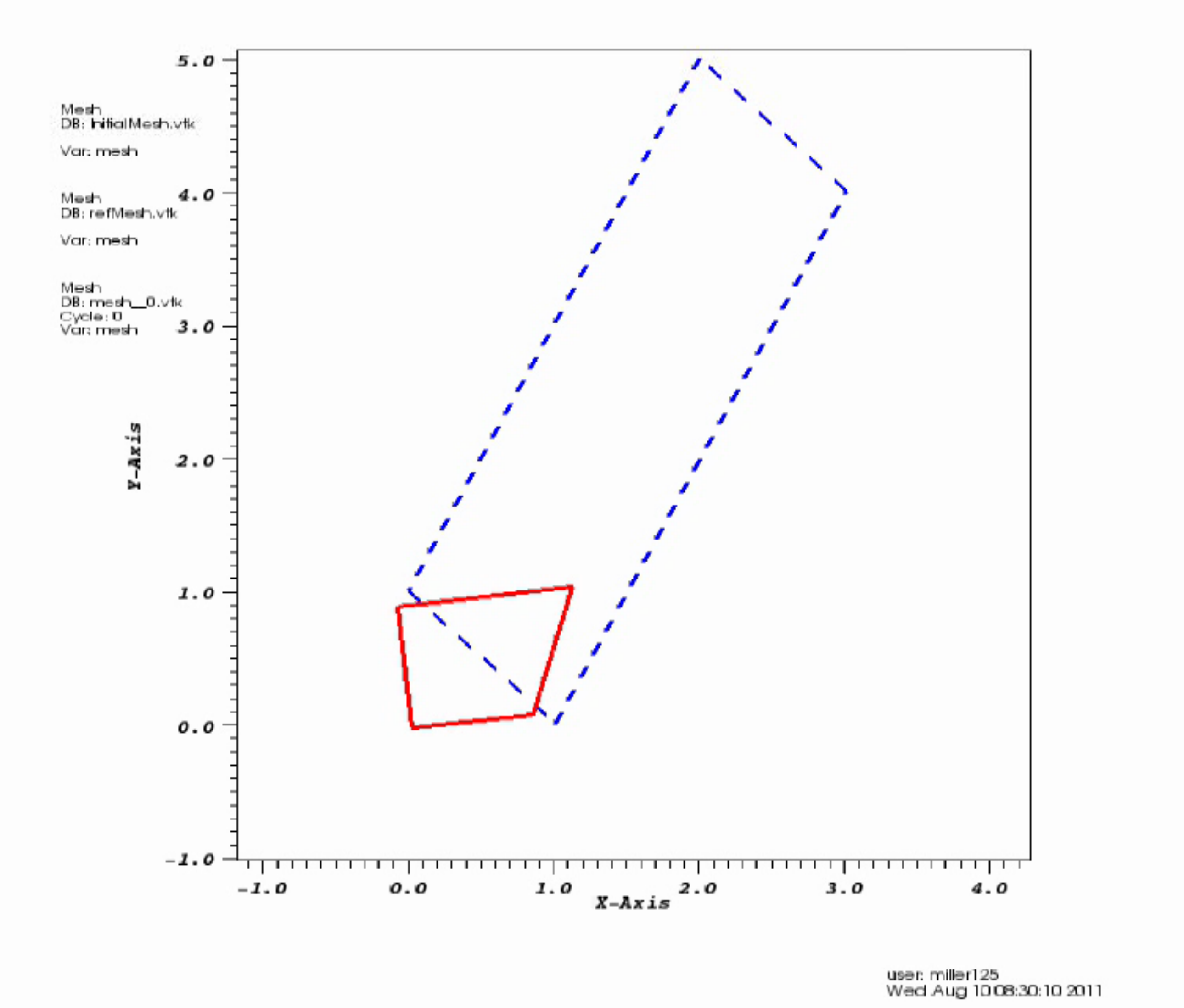
- Is a function of all free vertex coordinates.
- Is convex for some choices of metric.
- Solver can use analytic gradients and Hessians.

Essential problem is specification of Target Matrices and quality metric.

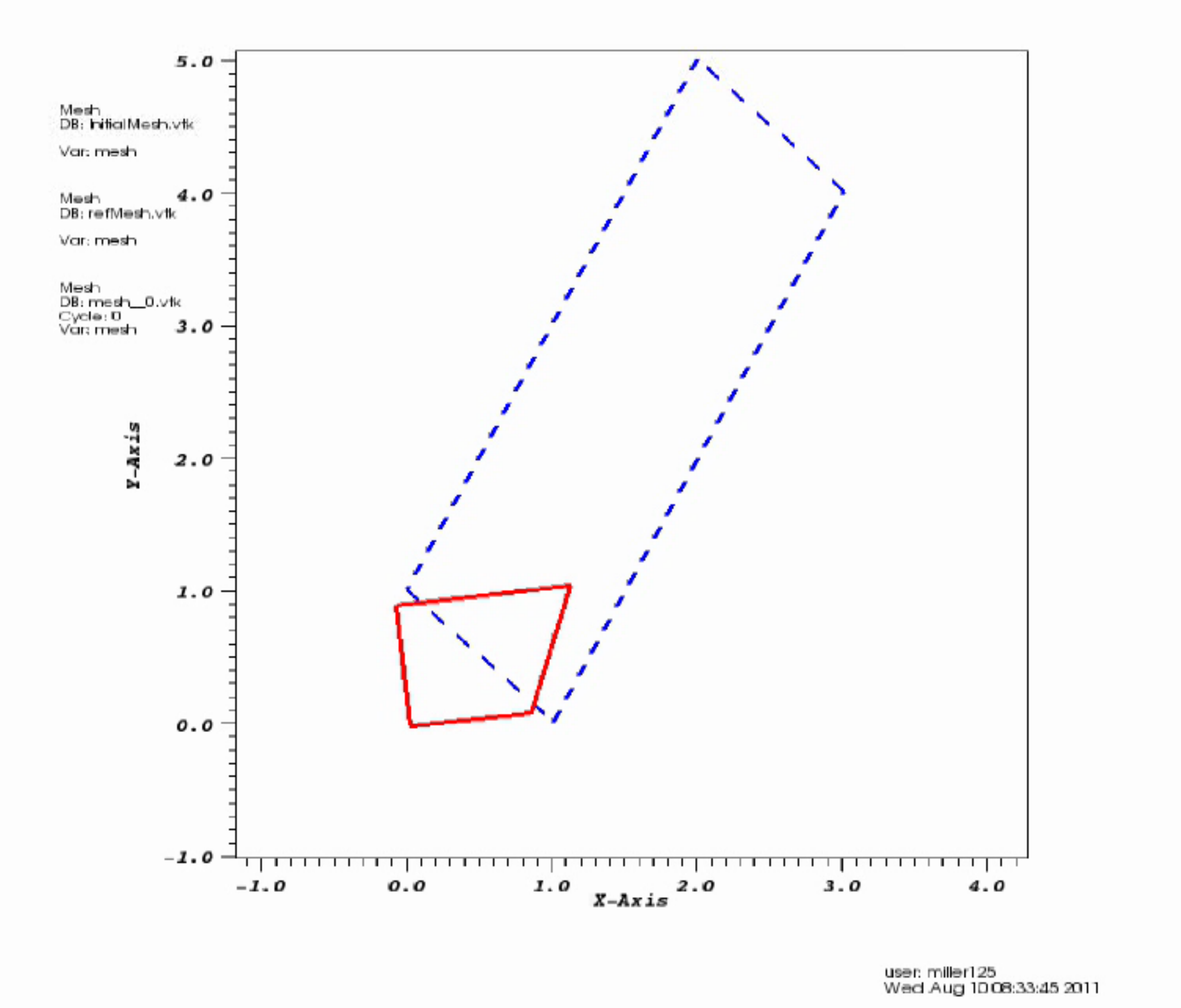
Single Quad Reference Mesh Example – Shape Metric



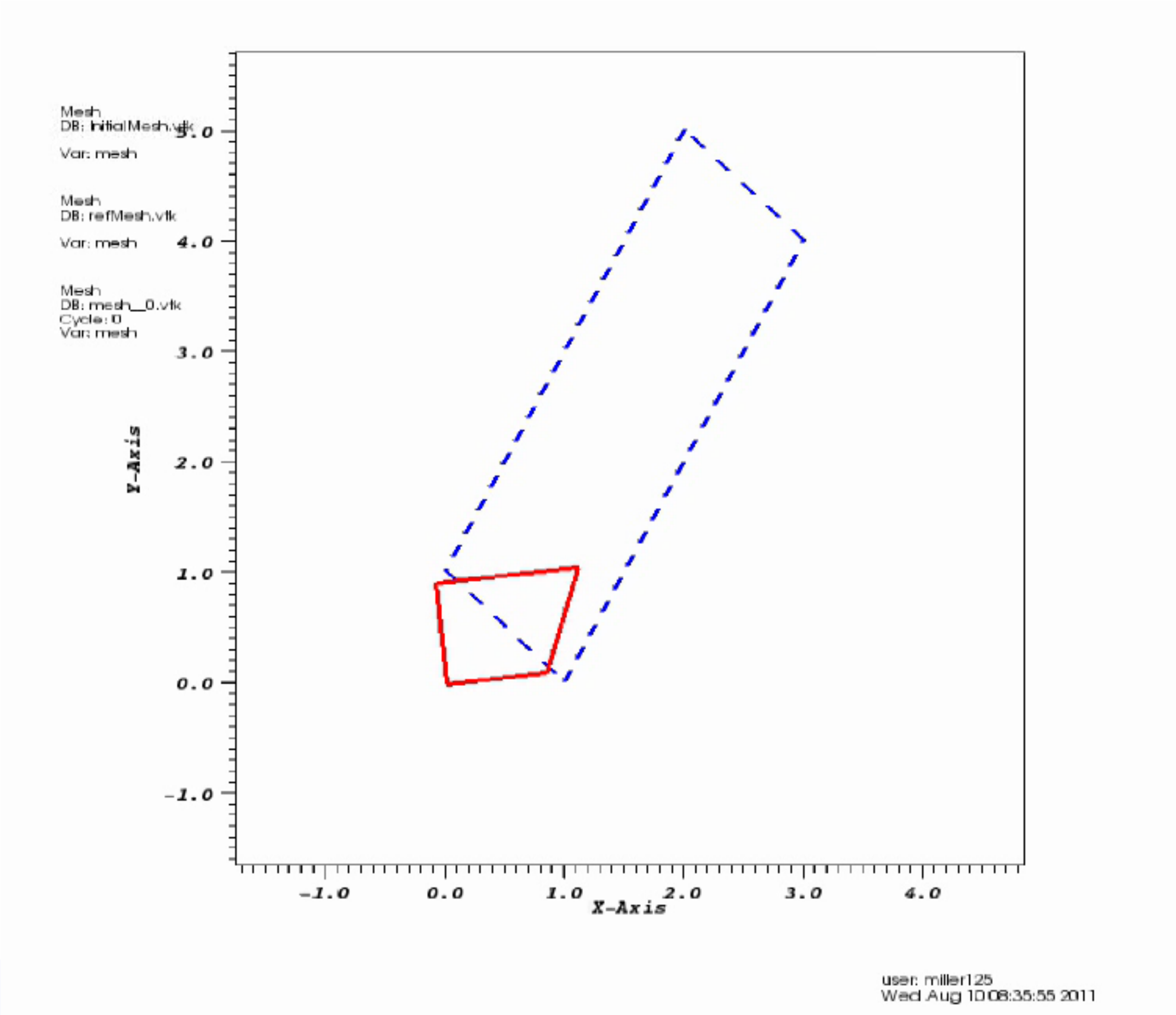
Single Quad Reference Mesh Example – Shape+Size Metric



Single Quad Reference Mesh Example – Shape+Orientation Metric



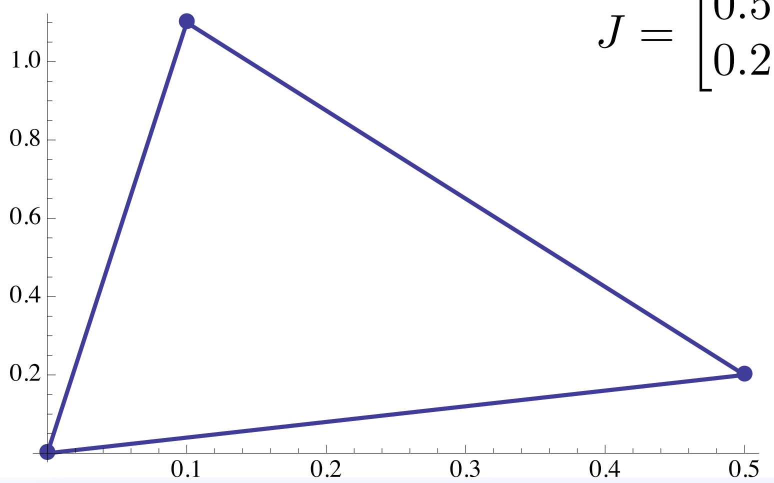
Single Quad Reference Mesh Example – Shape+Size+Orient Metric



LVQD Factorization

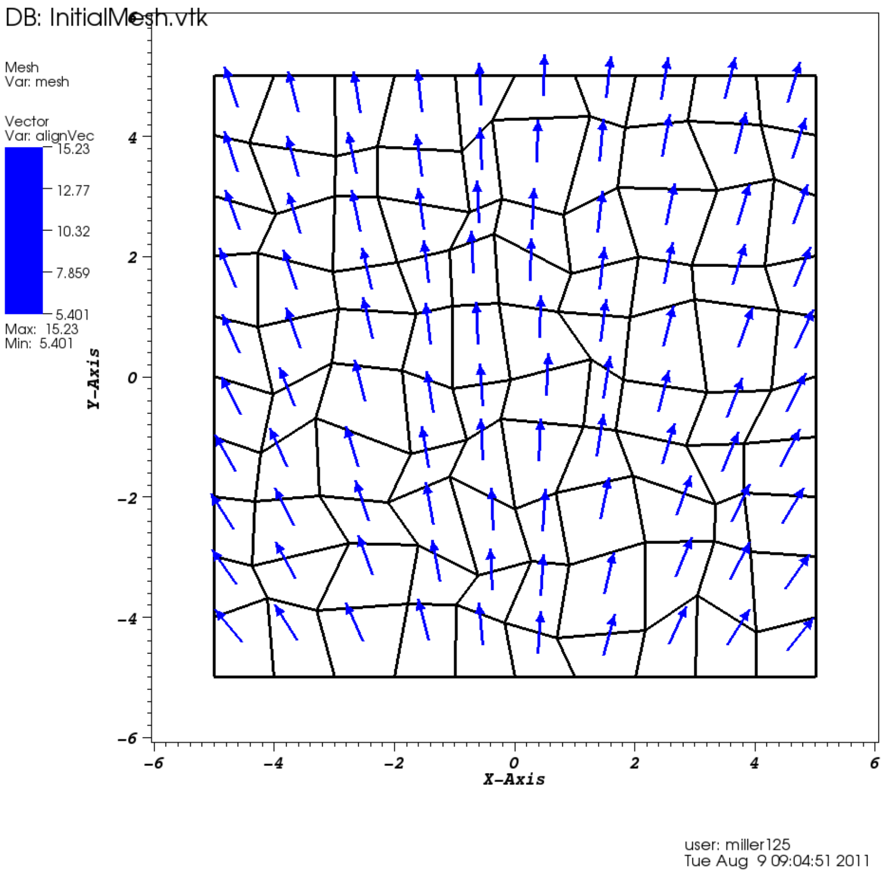
Each Jacobian matrix can be factored into the useful form

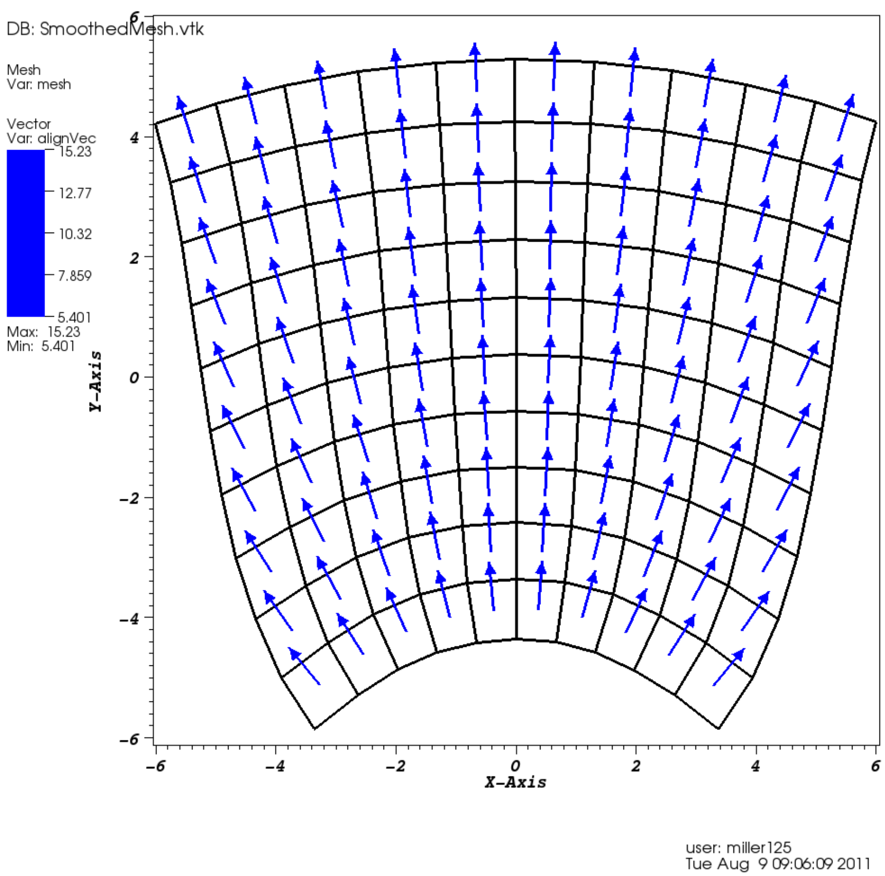
$$J = \lambda V Q D$$



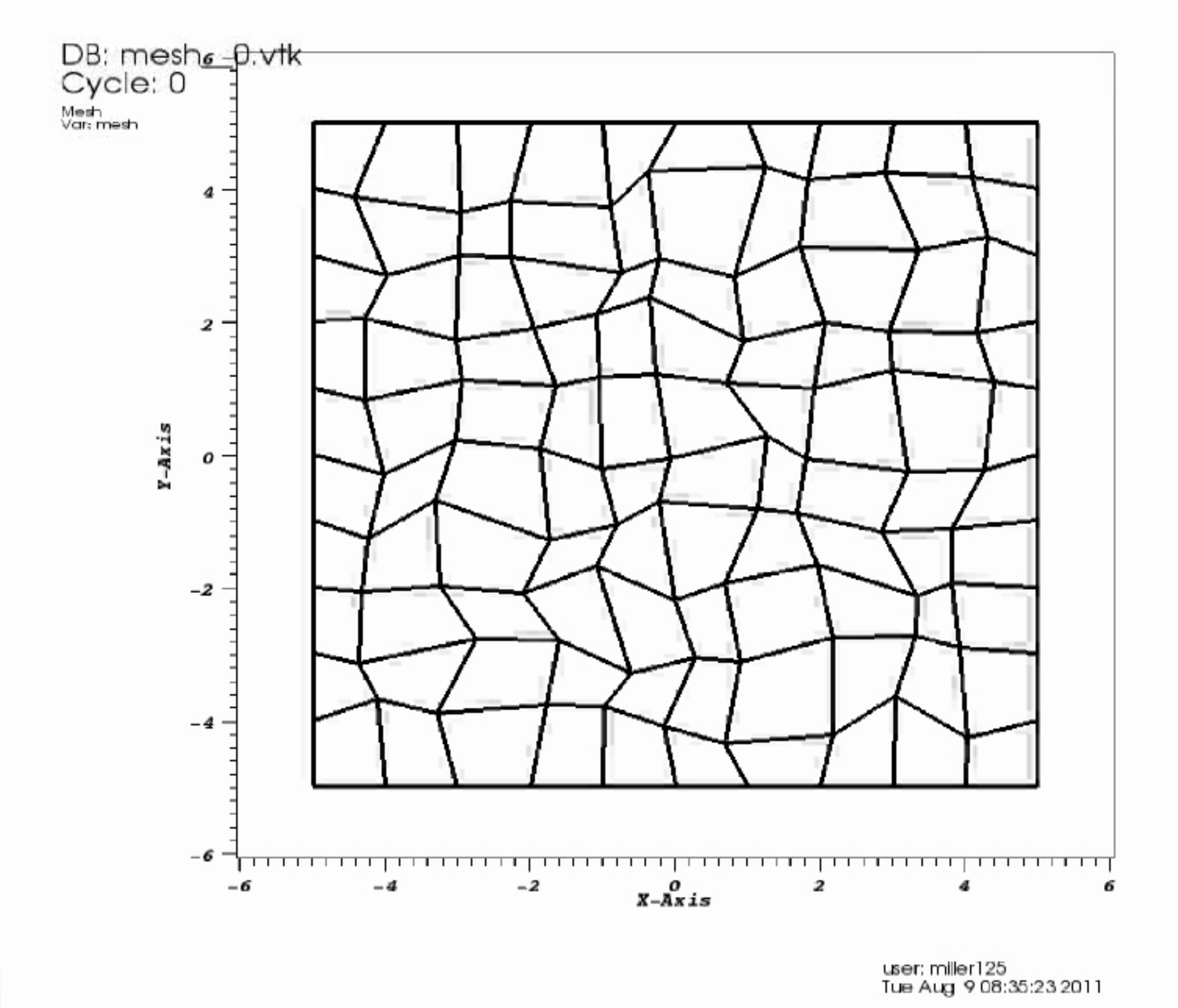
$$J = \begin{bmatrix} 0.5 & 0.1 \\ 0.2 & 1.1 \end{bmatrix} \approx 0.54 \cdot \begin{bmatrix} 0.93 & -0.37 \\ 0.37 & 0.93 \end{bmatrix} \begin{bmatrix} 1 & 0.45 \\ 0 & 0.89 \end{bmatrix} \begin{bmatrix} 1 & 0.0 \\ 0 & 2.05 \end{bmatrix}$$

Vector Alignment using LVQD Factorization





Vector Alignment using LVQD Factorization

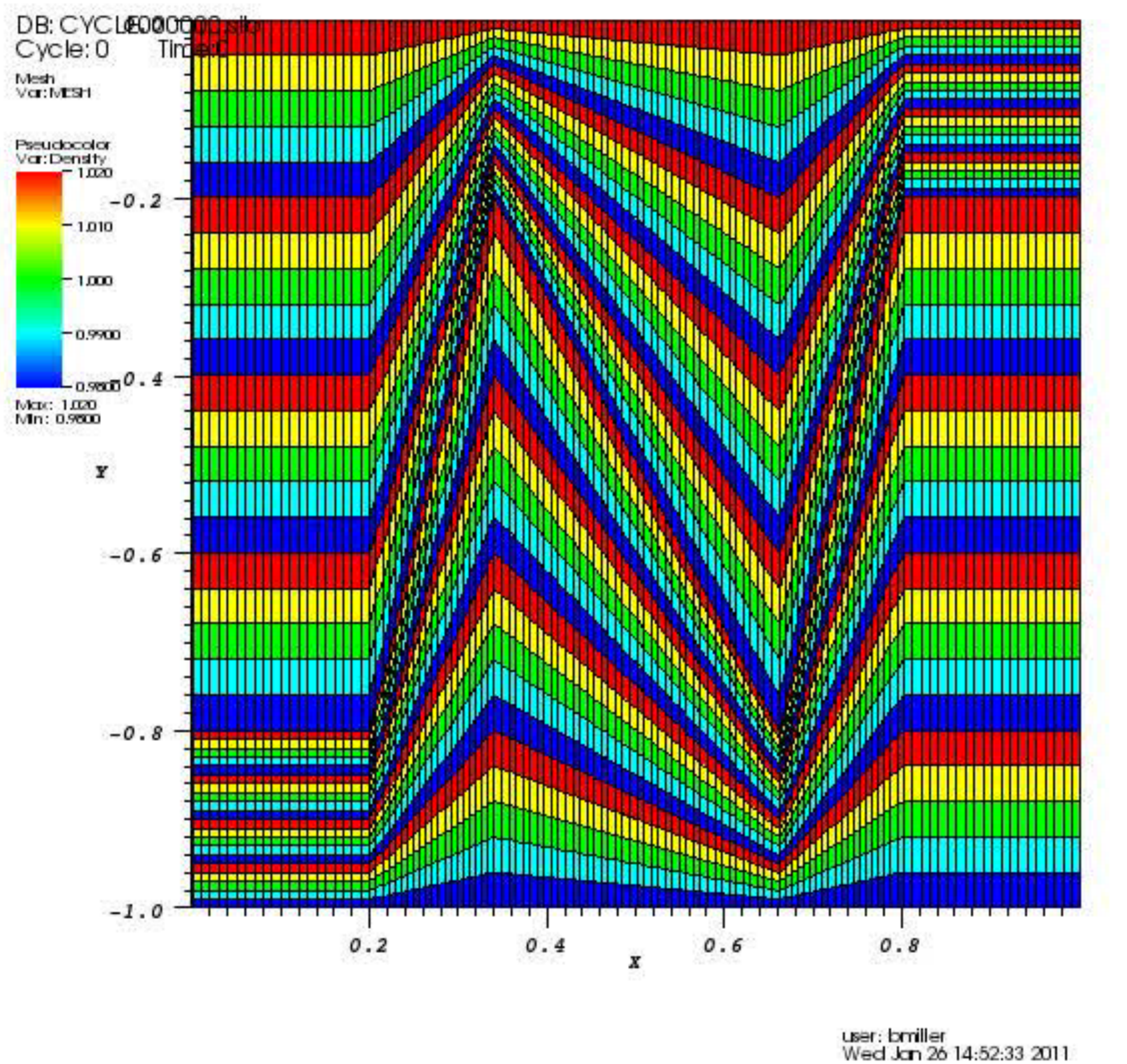


Objectives of Rezoning for Advection Based ALE Methods

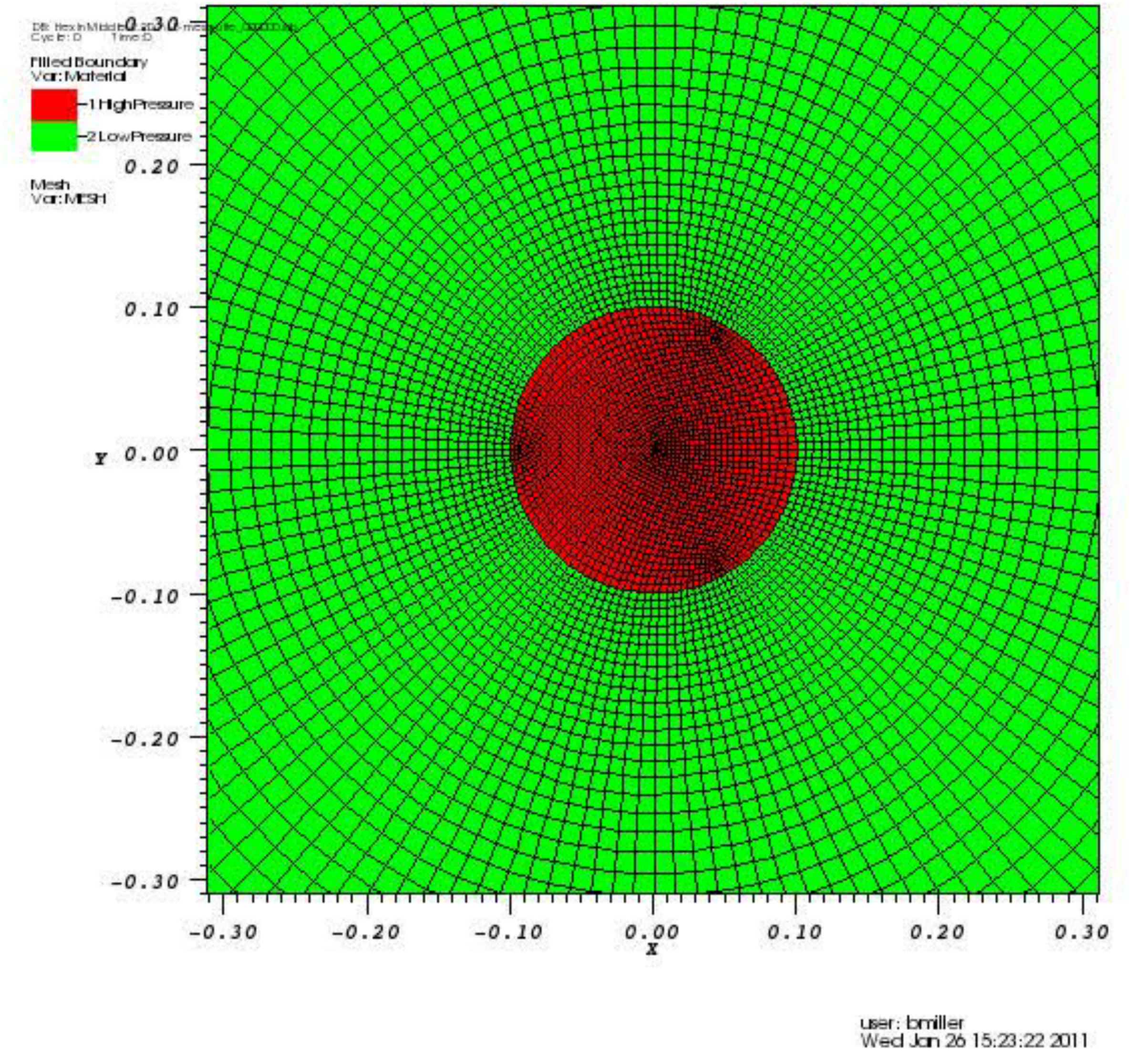
- Minimize diffusion errors of remap phase
- Improve quality of deformed elements in the new mesh
- Rezoned elements ‘tuned’ to the underlying physics
- Resolve material interfaces for as long as possible.

Challenge is to balance these sometimes competing objectives in a sensible fashion!

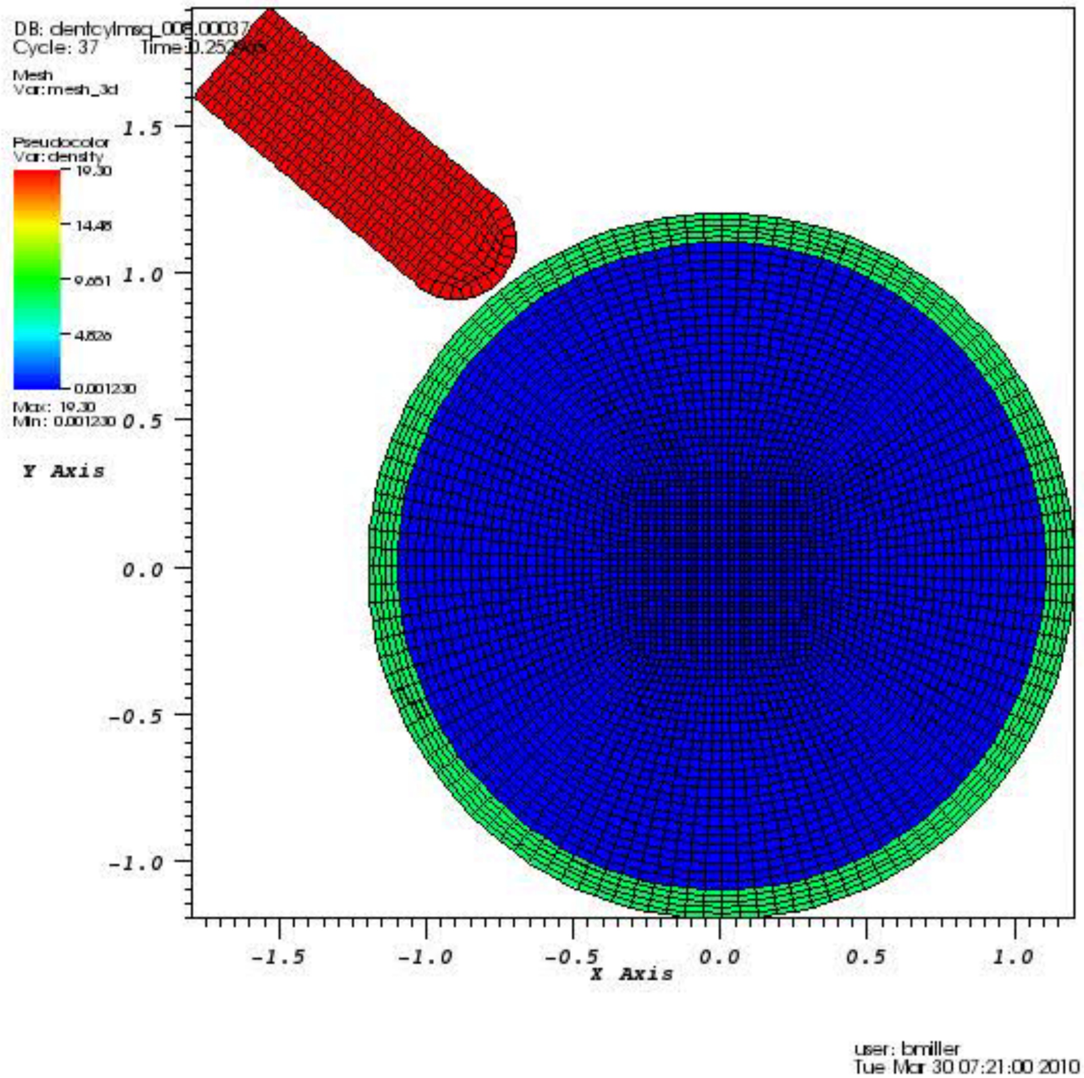
Over Advection Unit Test



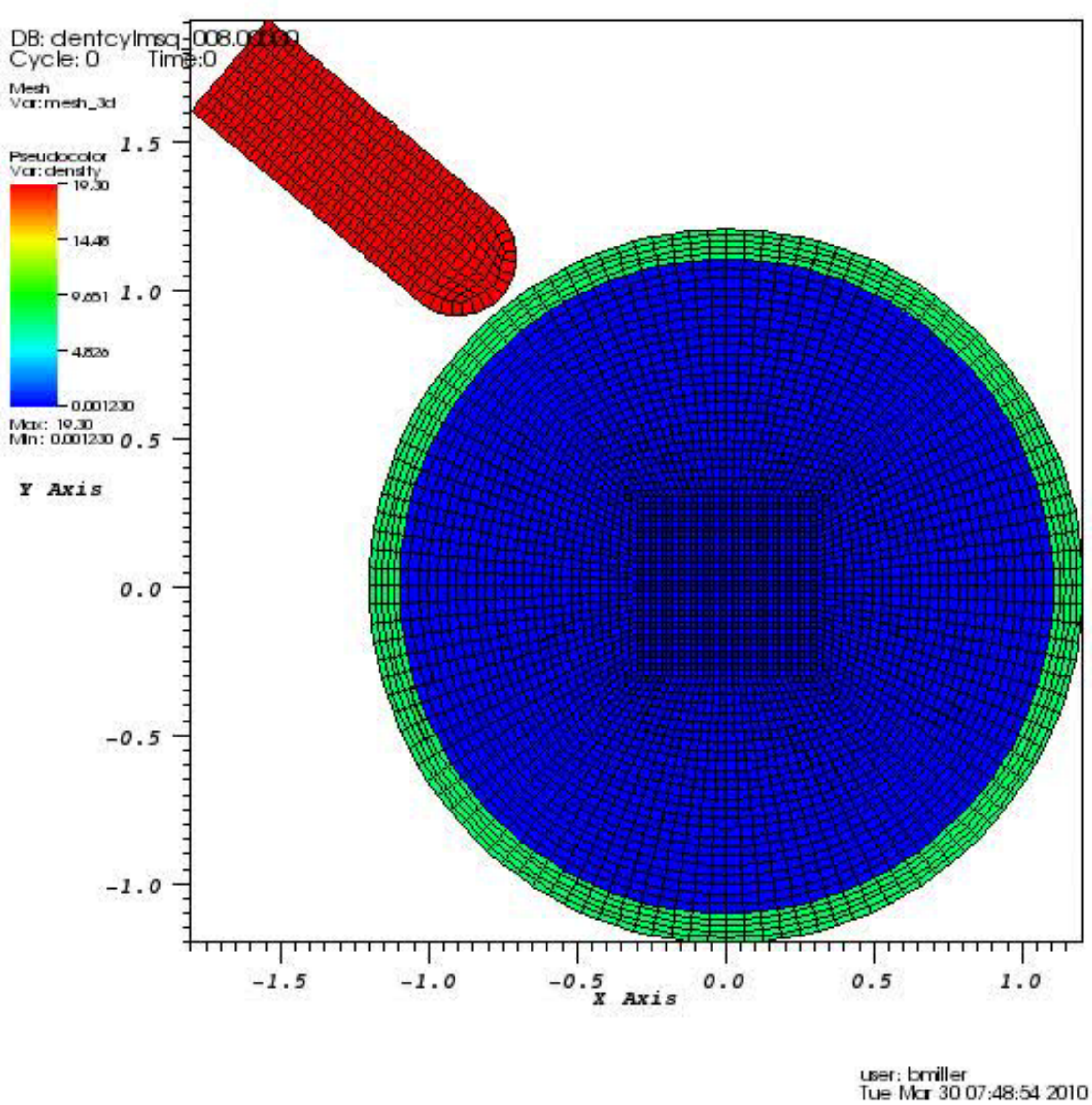
MagBlastAle Run With Boundary Smoothing



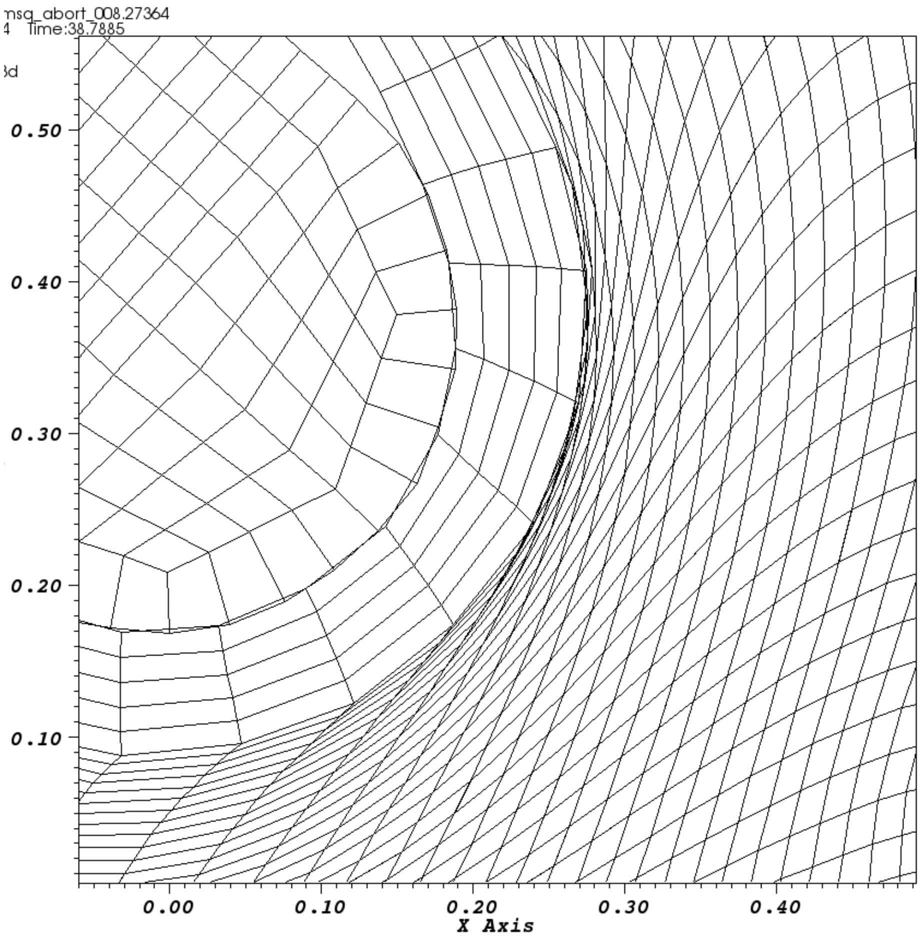
ALE3D Dented Cylinder problem (Equipotential Smoothing)

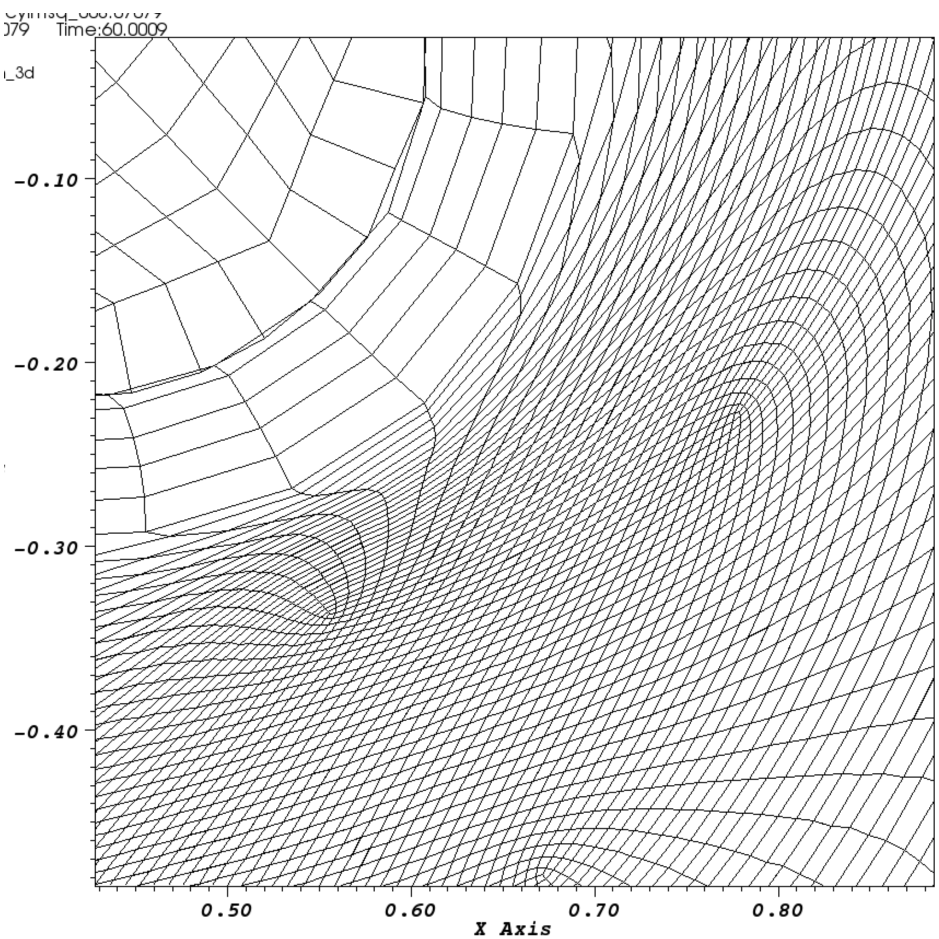


ALE3D Dented Cylinder (Mesquite Shape Optimization)



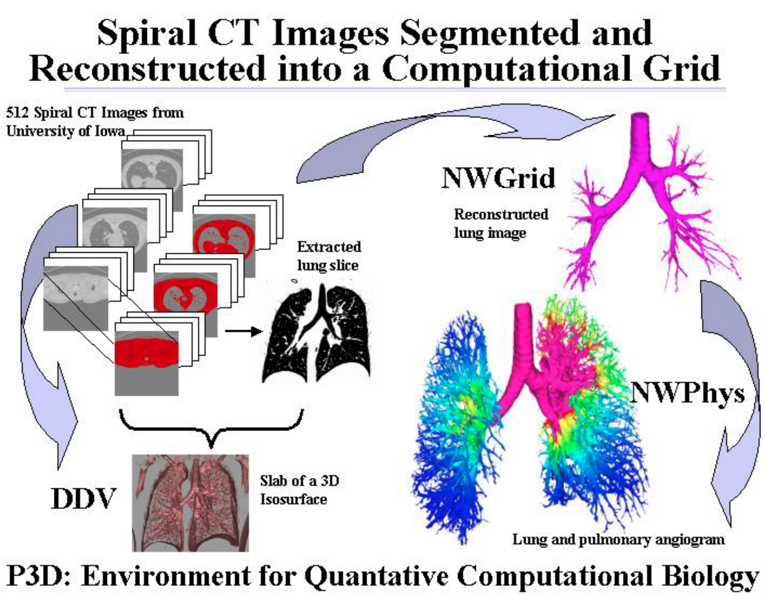
Close-up of Dented Cylinder Final Cycle Meshes

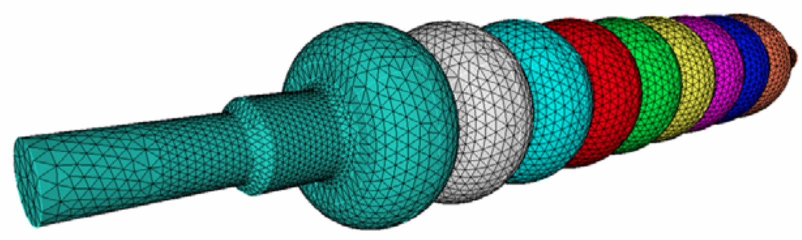




Our Mesh Quality Improvement Work has Impacted Many DOE Applications

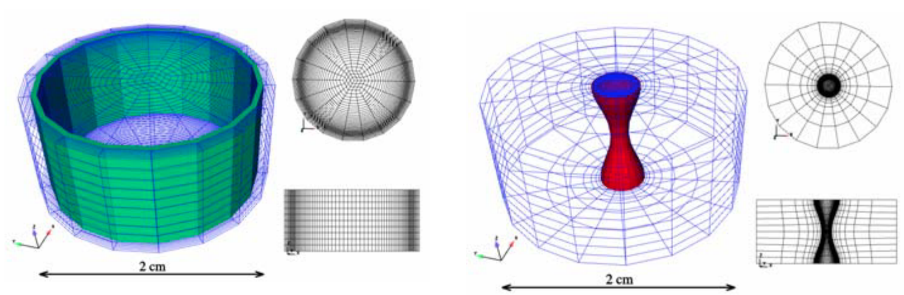
Application: **Computational Biology: CT images converted into a computational grid**
Challenge: Highly complex geometries
Impact: Mesquite enables PNNL to create good quality meshes for computational biology.





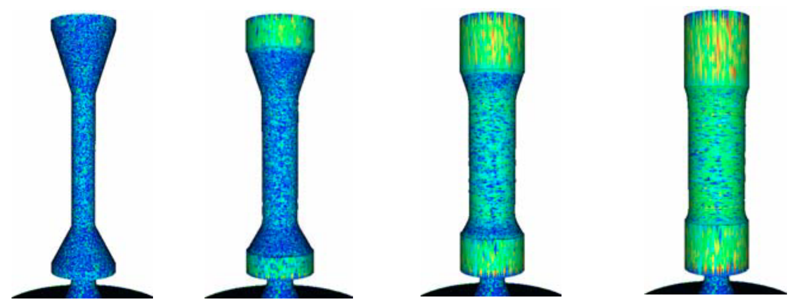
Application: **Shape optimization for accelerator cavities to minimize losses**
Challenge: Rapidly and smoothly update the mesh to conform to trial geometries
Impact: Used the deforming mesh metric to prototype geometry & mesh update model for potential use in SLAC accelerator design studies.

Our Mesh Quality Improvement Work has Impacted Many DOE Applications

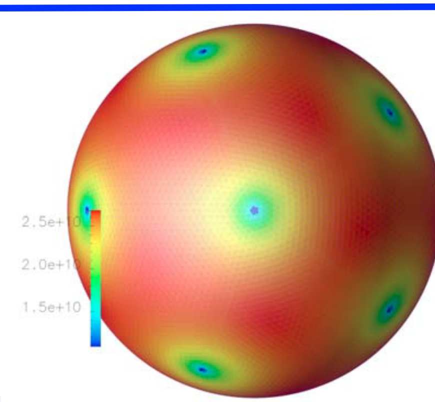


Application: **Plasma implosion using ALE methods**
Challenge: Maintain good mesh quality and biasing during deformation of plasma.
Impact: Prior to use of Mesquite, this calculation could not be performed by Alegra due to ineffective mesh rezoning algorithm.

Application: **Burn of rocket propellants in a time-deforming domain**
Challenge: Maintain good tetrahedral element shape quality as domain deforms
Impact: Condition number smoother (through ShapeImprovementWrapper) enabled many burn simulations at CSAR/UIUC.

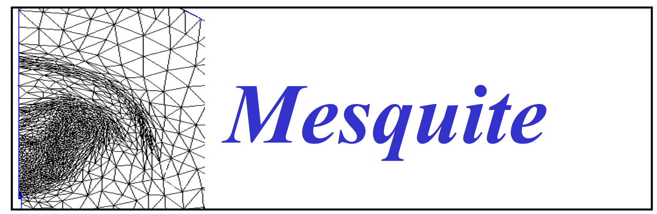


Application: **Climate: studies of finite volume discretization methods**
Challenge: Create many different high quality geodesic meshes on a sphere
Impact: Used many different mesh optimization methods to enable researcher Todd Ringler (CSU) to compare accuracy of discretization method on different meshes for climate calculations. CVT



Mesquite software infrastructure overview

- Mesquite uses metrics and optimization solvers
  - Untangle, smooth, size, shape metrics, anisotropic smoothing
  - Simple (Laplace) to complex (optimization) smoothers
    - Feasible Newton techniques, Active set solvers
  - Single-vertex and all-vertex solvers
- Combined solver approaches
  - Increase effectiveness and efficiency



- Efficient to run
  - Kernels written with inline functions and array-based access
  - Light-weight mesh data structure
- Open source
  - Developed primarily with ITAPS funding
  - Available http://www.cs.sandia.gov/optimization/knupp/Mesquite.html
  - Downloaded over 500 times in three years: most DOE labs, other govt. labs, 43 universities, and 30 private companies world-wide.

Mesquite is designed for use in a wide variety of applications

- Mesh Types
  - Structured, Unstructured, Hybrid
  - 2D, 3D
- Element Types
  - Triangular, Tetrahedral, Quadrilateral, Hexahedral, Pyramidal, Prisms currently
  - Polyhedral easily added
  - Linear and 2nd order elements
- Customizable
  - User-defined metrics, objective functions, and algorithms which can take advantage of existing Mesquite algorithms
- Callable as a library
  - Mesh and geometry information obtained through simple accessor functions

Integrating Mesquite into User Code – Wrapper interface

- Subclass Mesquite::Mesh and implement ~15 get/set functions.
- Alternatively, use ITAPS iMesh/iMeshP interface.
- Use of simple wrapper interface example:

```
MsqError err;  /* error state */

/* mesh construction derived Mesh or iMesh */
myMesh *mesh = new myMesh(…);

/* set mesh state here required for iMesh */

ShapeImprover theMeshSmoother;

theMeshSmoother.run_instructions( mesh, err );

if (err) cout << err << endl;
```

Integrating Mesquite into User Code – Low level interface

```
MeshImpl mesh;
mesh.read_vtk( input_file, err );

PlanarDomain geom( PlanarDomain::XY );

IdealShapeTarget W;                              /* TargetCalculator */
TShapeSizeOrientB2 target_metric;                /* Tmetric */
TQualityMetric mu( &W, &target_metric );         /* QualityMetric */

PMeanPTemplate objective_function( 2.0, &mu);    /* ObjectiveFunction */
FeasibleNewton solver( &objective_function );    /* QualityImprover */

TerminationCriterion outer, inner;

solver.set_inner_termination_criterion( &inner );
solver.set_outer_termination_criterion( &outer );

solver.use_global_patch();
outer.add_iteration_limit( 1 );
inner.add_relative_successive_improvement(1e-5);

InstructionQueue q;
q.set_master_quality_improver( &solver, err );
q.run_instructions( &mesh, &geom, err );
```

User Customization

- Users can insert their own algorithms without recompiling Mesquite:
  - Inherit from TargetCalculator, ObjectiveFunction or QualityMetric
- User-defined metrics/objective functions can take advantage of existing Mesquite algorithms
- Provides a platform for new research in mesh improvement algorithms
- Provides a platform for comparative studies

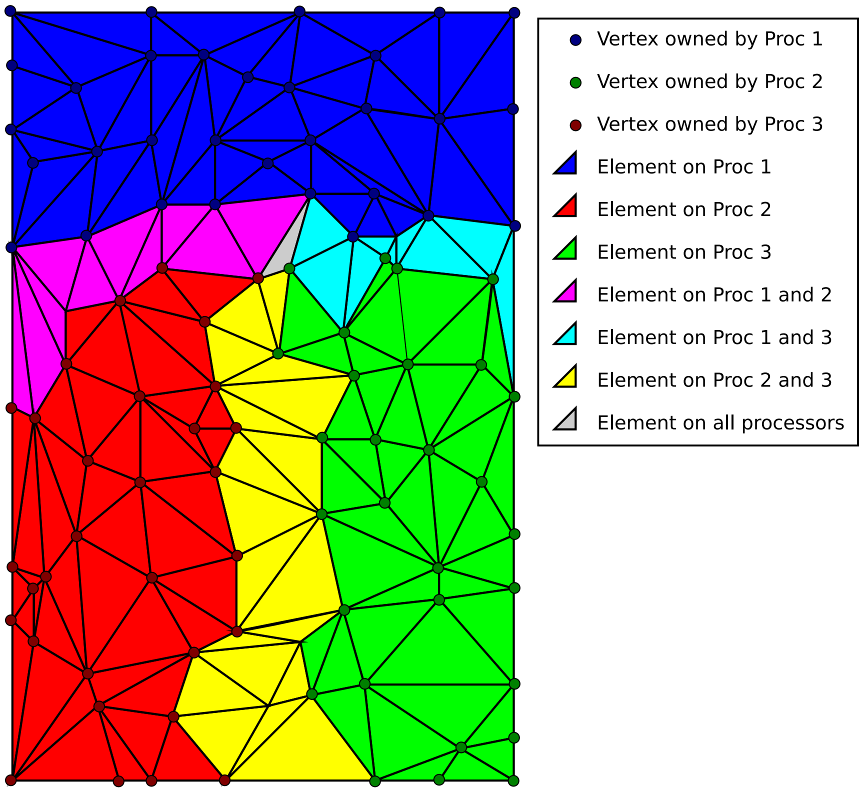
Parallel Algorithm

Idea from [Freitag, Jones, Plassmann, 1998]

- Treat partition boundary vertices separately

- Don't smooth at once, use Independent Sets
  - compute independent set
  - smooth
  - communicate
  - repeat until all partition boundary vertices Smoothed

- Number of independent sets for bounded degree graph with n=|V| is O(log(n)/loglog(n)) using a PRAM model with O(n) processors

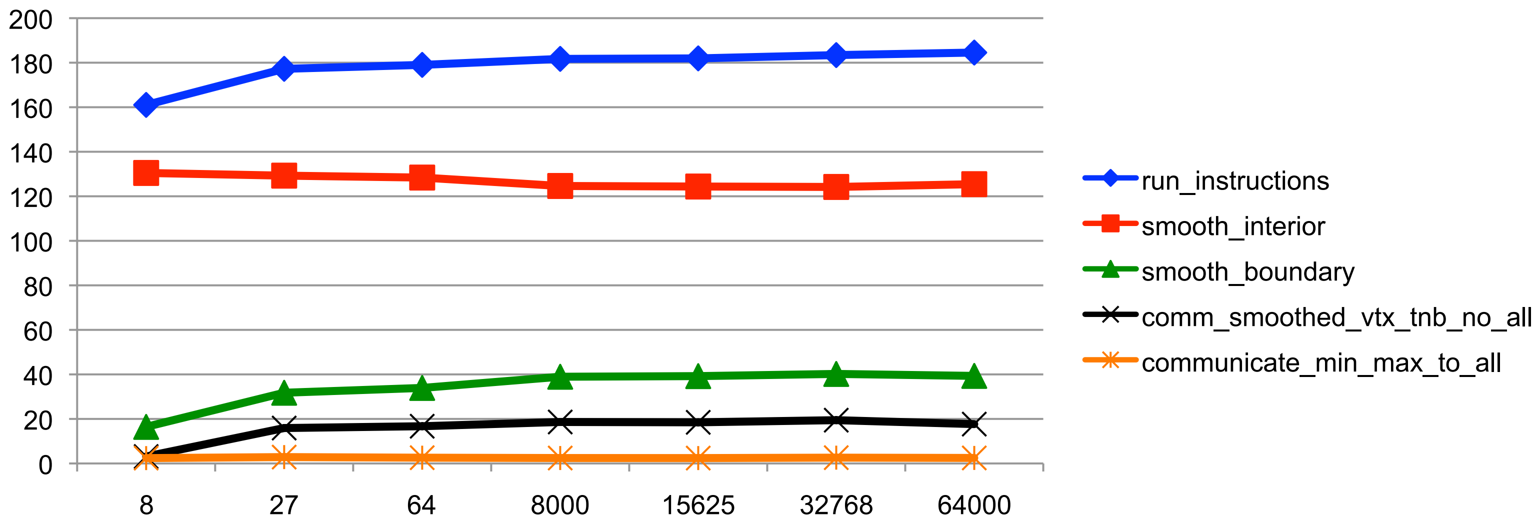


Parallelism Support

```
/* Create serial mesh as before */

/* global id and processor ownership must be set either in constructor or in function calls */
ParallelMeshImpl* mesquiteParallelMesh = new ParallelMeshImpl((Mesquite::Mesh*)&mesh,
                                                              "GLOBAL_ID", "PROCESSOR_ID");
ParallelHelper* helper = new ParallelHelperImpl();

helper->set_parallel_mesh(mesquiteParallelMesh);
helper->set_communicator((size_t)(MPI_COMM_WORLD));
helper->set_communication_model(0, err);
helper->set_generate_random_numbers(2, err);

mesquiteParallelMesh->set_parallel_helper(helper);

/* snipped same setup as serial case */

q.run_instructions( (Mesquite::ParallelMesh*) mesquiteParallelMesh, &geom, err );
```

Weak Scalability of Mesquite

- Large runs completed on LLNL BG/L and BG/P machines



Contact information

Software available:

- http://www.cs.sandia.gov/optimization/knupp/Mesquite.html
- http://www.itaps-scidac.gov

Contact Information:

Patrick Knupp
Optimization and Uncertainty Estimation Department
MS 1111
Sandia National Laboratories
Albuquerque, NM 87185
pknupp@sandia.gov

Lori Freitag Diachin
Center for Applied Scientific Computing
L-561
Lawrence Livermore National Laboratory
Livermore, CA 94551
diachin2@llnl.gov

Brian Miller
Center for Applied Scientific Computing
L-556
Lawrence Livermore National Laboratory
Livermore, CA 94551
miller125@llnl.gov

Acknowledgements

The Mesquite team (past and present)

- Michael Brewer (SNL)
- Lori Diachin (LLNL)
- Patrick Knupp (SNL)
- Jason Kraftcheck (Wisconsin)
- Thomas Leurent (ANL)
- Darryl Melander (SNL)
- Brian Miller (LLNL)
- Martin Isenburg (LLNL)

Funding:

SciDAC

Scientific Discovery through Advanced Computing

Auspices and Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Disclaimer