Poster #5   Patrick Knupp, PI

Mesh Quality Optimization via the Target-matrix Paradigm

Enabling automatic mesh generation, along with mesh updating for more accurate & efficient simulations.

Motivation

❖ Automatic Initial-Mesh Post-Processing:
- *Quality Improvement: Shape, Size, Orientation*
- *Removal of Inverted Elements*
- *Triangle, Tetrahedral, Quadrilateral, Hexahedral Meshes*
- *Hybrid, Polygonal Meshes, High-order Node Meshes*

❖ Application-embedded Mesh Updating:
- *Adaptive Mesh R-Refinement and Local Swapping*
- *Updating Meshes on Deforming Domains*
- *Mesh Rezone in Arbitrary Lagrange-Eulerian Simulations*

• Target-matrix Paradigm: do all of the above as special cases of a unified theory of mesh optimization
- *Each canonical smoothing algorithm is encapsulated within a wrapper for non-expert users*
- *Rapid deployment of custom-build smoothers*
- *Allows use of maintainable software*
- *Mesh quality optimization is a cross-cutting technology*

Target-matrix Paradigm Implemented in Mesquite Library

Low-level API Implements Key Objects in Target-matrix Paradigm:

- Finite Element Mappings,
- Active Jacobian Matrices
- Target Calculator
- Local TMP Quality Metrics, along with Analytic Gradients & Hessians
- Power-mean and Hierarchical Objective Functions
- TMP-based Local Relaxation Solvers

Special Cases Implemented as Wrappers

Mesh Quality Improvement Toolkit

Mesquite doubles as both an application service and a research platform (so need to balance flexibility vs. efficiency)

Mesquite is limited to node-movement, but compliments other interoperable ITAPS tools: (e.g., swap, refine, geometry, visualization)

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

Target-matrix Paradigm Enables Rapid Delivery of Custom-Built Smoothers

•**The Application:** Computational Mechanics Simulation Code with Explicit Time-Stepping.

•**The Mesh Quality Issue:** Generated mesh contained needlessly small edge-lengths that overly constrained the time-step size. Calculation therefore ran too slowly.

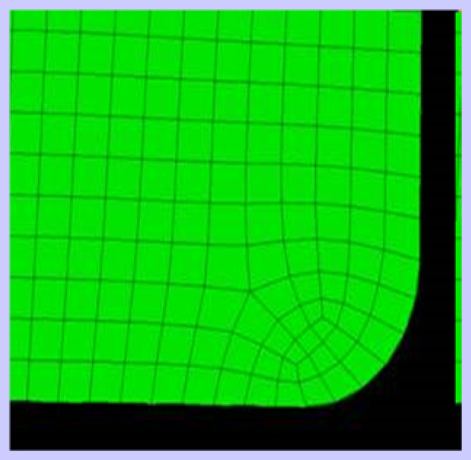
•**The Customers' Question:** Can Mesquite re-locate the mesh vertices so that the small edge-lengths are eliminated?

•**The Research Question:** Can we devise a suitable Target-construction algorithm within TMP to solve this problem?

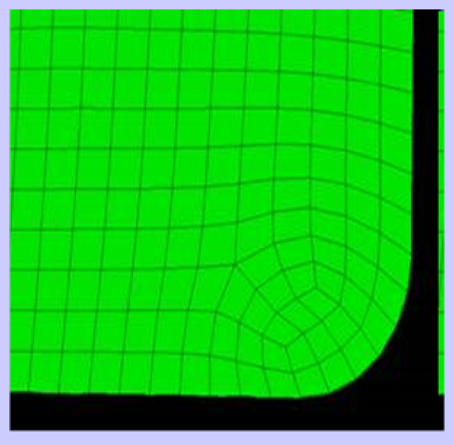
•**Observations:** 1) if all the mesh edge-lengths were the same, then none of them would be needlessly small.
2) Element shapes should ideally be squares since that is the goal of the Paving Algorithm

*__Conclusions:__ 1) A Shape+Size metric is needed (equal-area squares have equal edge-lengths)
2) The Target-matrix should be W = L * I, where L = average edge lenth in initial mesh, and I is the identity matrix (represents the shape of a square element).

•**Results:** 1) A new, concrete Target-construction algorithm was implemented in Mesquite, based on the above.
2) The Proposed Algorithm Met the Customers' Needs (see figure).
3) Because the theory and software were largely in place already, it took only 2 weeks to deliver the finished product, even though this particular smoother is new.



Initial Mesh



Optimized Mesh

Mathematical Description of the Target-matrix Paradigm

1. Objective Function

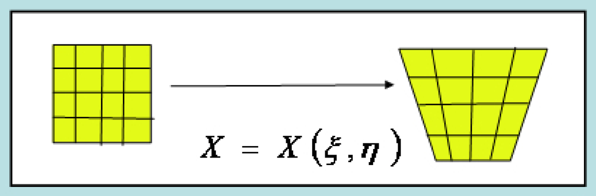
$$F(x) = \frac{1}{N_s} \sum_e \sum_k c_k^e \mu(T_k^e)$$

Measure of Global Mesh Quality

$$M(p,x) = \left\{ \frac{1}{N_s} \sum_e \sum_k \left[c_k^e \mu(T_k^e) \right]^p \right\}^{1/p}$$

= Power Mean

2. Element Mappings & Sample Points



Master Element   Physical Element

Sample Point: $X_k = X(\xi_k, \eta_k)$

3. Active Jacobian Matrix

$$A(\Xi) = \frac{\partial X}{\partial \Xi}$$
$$A_k = A(\Xi_k)$$

Active Jacobian Matrix is the Jacobian of the Map at a Sample Point within an Element.

The active matrix is a function of the vertex coordinates in the active mesh.

Contains Shape, Size, and Orientation Information at the Given Sample Point.

4. Target Matrix

For each Sample Point there exists a pair of active & target matrices (A and W). Target matrices represent the desired Jacobians in the optimal mesh.

The Target-matrix is automatically constructed prior to the optimization phase, based on the initial mesh, a reference mesh, and/or a priori application information.

For each application goal (e.g., shape improvement, alignment, adaptivity, ALE), there is a distinct Target-matrix construction algorithm.

Targets can be used either to create a quality not present in the initial mesh or to preserve a quality that is present.

5. Scaled Active Matrix

$$T \equiv A W^{-1}$$

Control Shape+Size+Orientation: Active Jacobian matches entire Target-matrix, that is, when A=W or T=I.

Control Shape+Size: Active Jacobian matches non-orientation part of Target-matrix, that is, when A=RW or T=R (with R an arbitrary rotation matrix).

Control Shape: Active Jacobian matches only shape portion of Target-matrix, that is, when A=sRW or T=sR (with s an arbitrary positive scalar and R an arbitrary rotation matrix).

6. Local TMP Quality Metrics

$$\mu = \mu(T)$$
$$\mu_k = \mu(T_k)$$

Shape+Size+Orientation Metric:
$$\mu(T) = \|T - I\|_F^2$$

Shape+Size Metric:
$$\mu(T) = \|T\|_F^2 - 2\sqrt{\|T\|^2 + 2\tau} + 2$$

Shape Metric:
$$\mu(T) = \|T\|_F^2 - 2\tau$$

7. Scalar Tradeoff Coefficients

Tradeoff coefficients allow us to emphasize quality in some locations in the mesh over other locations via Equidistribution.

Equidistribution Principle: $c_k \mu_k = cnst$ (for all k)

Large coefficient means small metric value (better quality).

Can also be used to perform tradeoffs between multiple quality improvement goals.

Tradeoff coefficients automatically constructed prior to the optimization phase, based on spatial position, local quality, and/or a priori application data.

8. Prior Direct Optimization Methods

Comparison to Target-matrix Paradigm:

$$F = \sum_{i=1}^{M+1} \sum_{j=1}^{N} (L_h)^2_{i,j} + \sum_{i=1}^{M} \sum_{j=1}^{N+1} (L_v)^2_{i,j}$$

- Methods do not use sample points or element mappings; therefore they have no mechanism for controlling quality within elements.
- Are not formulated directly in terms of Jacobian matrices; therefore they cannot avail themselves of matrix theory.
- Do not use Target-matrices and thus have only one goal (e.g., only shape improvement, smoothness, etc.).
- Do not have SSO, SS, and Sh metrics; therefore, there exists a plethora of quality metrics, not always well-formulated.
- Do not use Tradeoff Coefficients; cannot vary emphasis on quality from one location to the next.

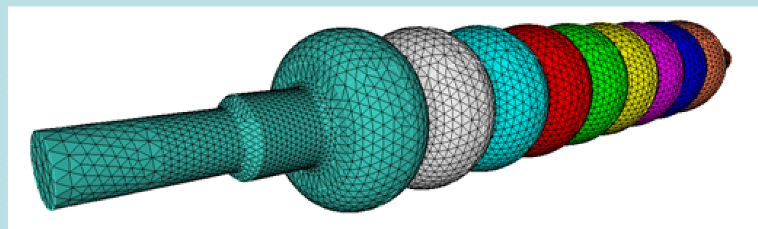
9. TMP Reports

1. "Formulation of a Target-matrix Paradigm for Mesh Optimization," SAND2006-2730J.
2. "Local 2D Metrics for Mesh Optimization," SAND2006-7382J.
3. "Analysis of 2D Rotational-invariant, Non-barrier Metrics," SAND2008-8219P.
4. "Measuring Quality Within Mesh Elements," SAND2009-3081J.
5. "Target-Matrix Construction Algorithms," SAND2009-7003P.
6. "Surface Mesh Optimization," in progress.
7. "Tradeoff Coefficient & Binary Metric Construction Algorithms," in progress.
8. "Relaxation Solvers for TMP Quality Metrics," in progress.

Plus various TMP-related journal papers on Anisotropic Mesh Adaptivity, Updating Meshes on Deforming Domains, Label-invariant Mesh Quality Metrics, and other.

Digging Deeper: A result from Quality Metrics Paper (Ref. 3)

Proposition 1. Let T be any real 2x2 matrix and define ψ^2 = |T|^2 + 2 det(T). Then ψ >= 0, with ψ=0 if and only if T is a scaled flip.
Proposition 2. R = [T + adj(T)^t]/ψ(T) is a rotation matrix.
Proposition 3. Let μ(T)=|T|^2 – 2 ψ(T)+2. Then mu is non-negative.
Proposition 4. μ(T) is globally minimized if and only if T is a rotation.
Proposition 5. μ is differentiable on the set of 2x2 matrices which are not scaled flips.
Proposition 6. The stationary points of μ are rotation matrices.
Proposition 7. μ can be expressed as |T-R|^2.
Therefore, μ satisfies the properties needed for a well-posed Shape+Size metric in the Target-matrix Paradigm.



Design optimization for wave-guide cavity requires mesh updating

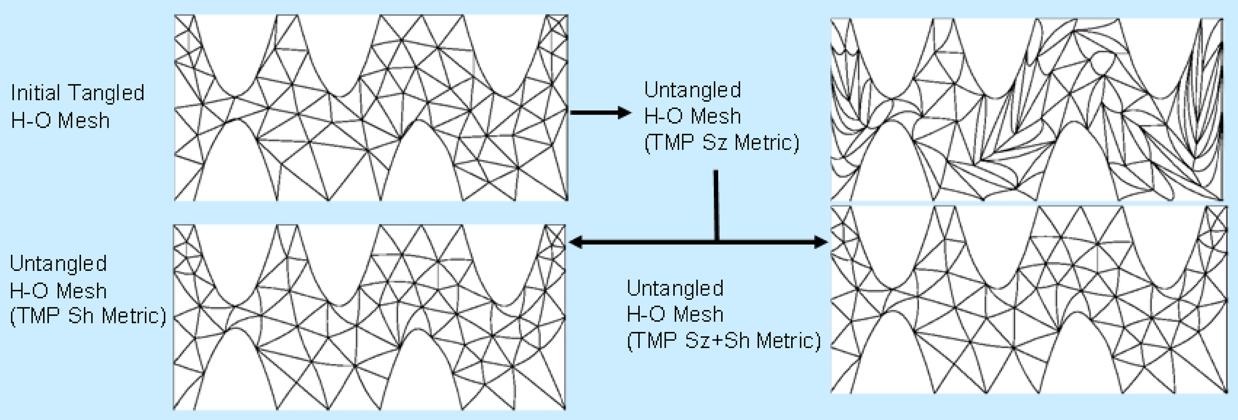
On-Going Research & Development of Target-Matrix Paradigm

Optimizing Quality of High-Order-Node Meshes

The Issue: To create finite element meshes with high-order nodes, most mesh generators first create linear elements and then add the high-order nodes later. After the high-order nodes are created, they must be "snapped" to the geometry to achieve high accuracy. Snapping often creates poor quality or even inverted elements.

The Standard Solution: Local Mesh Modification (node insertion, local swapping & smoothing toolbox)

Research Question: Can this issue be addressed more effectively if the Target-paradigm is included in the toolbox?

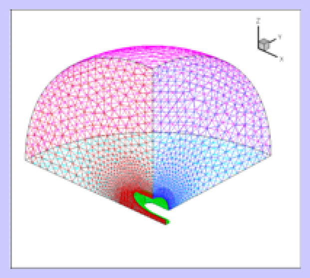


Sliver-removal for a viscous CFD Mesh

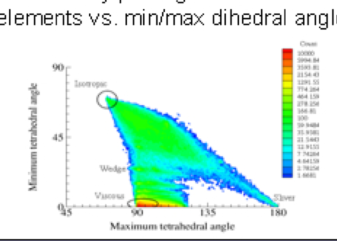
The Problem: Given an all-tet CFD mesh, use mesh optimization to remove the sliver elements without disturbing the viscous boundary tetrahedral layer.

Approach: Use Binary TMP Metric and Tradeoff Coefficients to Improve Shape in Far-field, while Preserving the Boundary Layer.

(Figures courtesy Jan-Renee Carlson, NASA-Langley)

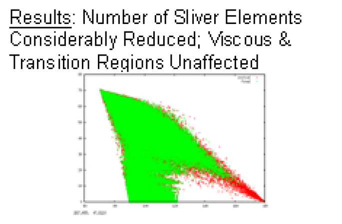


Viscous, Isotropic, and Sliver regions identified by plotting number of elements vs. min/max dihedral angle.

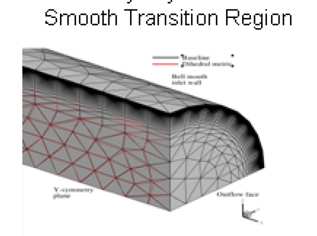


Tradeoff Coefficient Model Based on Dihedral Angle Plot.

Results: Number of Sliver Elements Considerably Reduced; Viscous & Transition Regions Unaffected



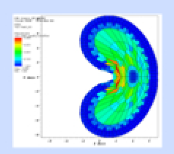
Boundary Layer Preserved Smooth Transition Region



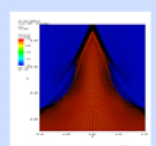
Mesh Rezoning for Arbitrary Lagrange-Eulerian Methods

- *Kull, ALE3D, Alegra contain mesh rezoning algorithms to prevent mesh tangling and improve quality.*
- *A good algorithm allows code to run to completion.*
- *A good algorithm does not move nodes very far, so that the re-map step is accurate.*
- *A good algorithm has been a "Holy Grail" for many years.*
- *Research Question: can Mesh Optimization Methods based on TMP help?*

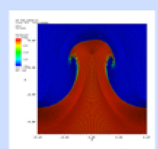
A new approach uses TMP Tradeoff Coefficients to create a Binary Metric that improves Shape where needed and preserves the Lagrange Mesh elsewhere. (Jury still out).



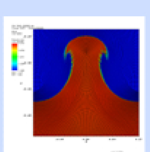
TMP Barrier Shape Metric



Lagrange



Winslow



Ref. Jacobian

Figures courtesy Brian Miller, LLNL

Local Relaxation Solvers for TMP Quality Metrics

Minimize F(x) via Block Coordinate Descent (like Gauss-Seidel).
Examples: Laplace smoothing, Length-weighted Laplace, Many others.
Easy to implement, fast if accurate solutions not needed.
Research Goal: Apply this approach to TMP Objective Functions.

$$\mu(T) = |T|^2 - 2\sqrt{|T|^2 + 2\tau} + 2$$
$$X_{i+1} = X_i - H_i^{-1} G_i$$
$$G = \sum_{m=1}^{M} \{(T-R)s\}_m$$
$$H = \left(\sum_{m=1}^{M} \left(1 - \frac{1}{\psi_m}\right) |s|_m^2\right)$$

For each free mesh vertex:
  - create a local patch
  - fix the patch boundary nodes
  Update the free vertex position:
    x-new = x-old + a-old * grad F
  End Inner Iteration
End Outer Iteration

<u>Challenges</u>:

- Derive the update rules for each TMP metric,
- Ensure the rules result in convergent iteration.
- Rule depends on the selected mapping.
- Operation counts & timing-tests.