# 2005 Department Review

## Department 9211: Optimization and Uncertainty Estimation


# Patrick Knupp



**Sandia National Laboratories, 2005-3649P**

# Overview

Research Areas:

- Mesh Optimization Methods & Applications
- Code Verification & Method of Manufactured Solutions (MMS)

Projects:

- TSTT SciDAC Center
  - Terascale Simulation Tools & Technology,
  - Members: ANL, BNL, LLNL, ORNL, PNNL, RPI, SNL, Stonybrook
  - SNL PI, (Taugtes, Brewer, Kraftcheck)
  - Meshing & discretization technology for SciDAC applications,
  - Team: Glimm, Diachin, Fisher, Shephard
- MICS
  - Mathematical Research for Mesh Optimization & Improvement,
  - PI (Leung, van der Zee, de Sturler)
- Code Verification for ASC V&V,
  - Open questions in MMS related to Premo, Alegra, Presto
  - PI (Oberkampf, Koteras, LaBreche, Ober & Bond, Robinson, Trucano)

# Outline

I. Code Verification

II. Mathematical Research for Mesh Optimization & Improvement

III. Mesquite

IV. SLAC Shape Optimization

# Code Verification Research & Development

I. Measuring Progress in Code Verification

II. The Verification Software Toolkit

III. Open Questions in Method of Manufactured Solutions

# Code Verification Ressearch – I

## Measuring Progress in Code Verification

ASC V&V Program Office: " How does one measure and communicate progress in verification from the perspective of code development and application?" (justify to Congress what has been done & will be done with the money spent in ASC V&V)

I wrote a paper on measuring progress in code order-verification for JASON's Meeting

Main points of paper:

-Order-verification is a closed-ended process that can be measured,

-The benefits are (i) finds coding mistakes and (ii) demonstrates to others that code is verified

- Can measure progress in both (i) test-suite construction and (ii) test-suite execution

- Test suite execution: progress is measured by comparing <observed> order-of-accuracy vs. <formal>

- Test suite construction: progress is measured by the following milestones:
    - -Identification of codes' governing equations,
    - -Inventory of codes' discretization algorithms and formal order-of-accuracy,
    - -Construction of order-verification test-suite:
        - -Map from each test to governing equations, discretization algorithms, and physics capability,
        - -Identification of quantities for which order-of-accuracy will be observed,
    - -Coverage of all discretization algorithms,
    - -Creation of corresponding code input files

# Functional Requirements for a Verification Software Toolkit (VST)

Background

A major obstacle in doing both code and solution verification at SNL is the need for pre- and post-processing software for grid resolution studies.

Scope

The VST will be aimed primarily at uniform refinement of mesh and time-step studies for the purpose of evaluating solution error.

Order verification
Solution verification

Is not aimed at adaptive meshing (requires non-uniform refinement)

Developer Buy-in is Critical (Memo)

Suggested Approach:
Collaboration with DSTK

Impact: If such software were available, grid resolution studies would be less daunting and thus encouraged.

# Code Verification Research - III

## Resolution of Open Questions in the Method of Manufactured Solutions

Example: Collaboration with Bond & Ober on Premo code verification.

Two Papers:

1. *Bond, Knupp, Ober, "A Manufactured Solution for Verifying CFD Boundary Conditions," AIAA 2004-2629* (Verification of Euler equations with slip & supersonic BC)

2. *Bond, Knupp, Ober, "A Manufactured Solution for Verifying CFD Boundary Conditions, Part II," AIAA 2005-0088* (Verification of Navier-Stokes equations with no-slip (adiabatic & Isothermal) and outflow (supersonic, subsonic, mixed))

# Code Verification Research - III

Presentation:

MMS Case Study for Trilabs V&V meeting (LLNL): I emphasized the MMS process instead of the final results, e.g., the bugs and algorithmic issues that were uncovered during the course of verification.

http:\\scico.sandia.gov/premo/premo_new/html_src/Presentations.html

Impact of Premo work:

Open question B2:

"Can we extend the list of boundary condition types to which MMS can confidently be applied?"

"Example: Can MMS be applied to verifying a mixed supersonic/subsonic outflow condition?"

YES !!! (I knew that would be the answer, but now others are convinced.)

Next open question: Applicability of MMS to verification of codes capable of discontinuous solutions.

# Mathematical Research for Mesh Optimization

I. Elaboration of the Target-Matrix Paradigm

II. Other Work

# Elaboration of the Target-Matrix Paradigm

## Unifying Theme:

Develop a complete theory of mesh optimization (since none currently exists) to guide implementation of Mesquite framework and algorithms. Has turned out to be a mathematically rich theory.

## Publications in Progress:

-Part I: Basic Theory (mesh corners, matrices, corner metrics, objective functions, weighted Laplace)

-Part II: Properties of the Objective Function (symmetry relations, replication, gradients, Hessians, weighted Winslow)

-Part III: Corner Metrics (corner metrics, matrix sets, stationary points, local convexity, deforming meshes)

-Part IV: Target Matrix Construction (matrix QR factorization, automatic schemes, mesh rezoning)

- Part V: Adaptivity (local relaxation, equivalence, compatibility conditions, aspect ratio control, adaptivity)

Presented Part I at the SIAM CS&E (Orlando)

Submit paper on Part I this summer.

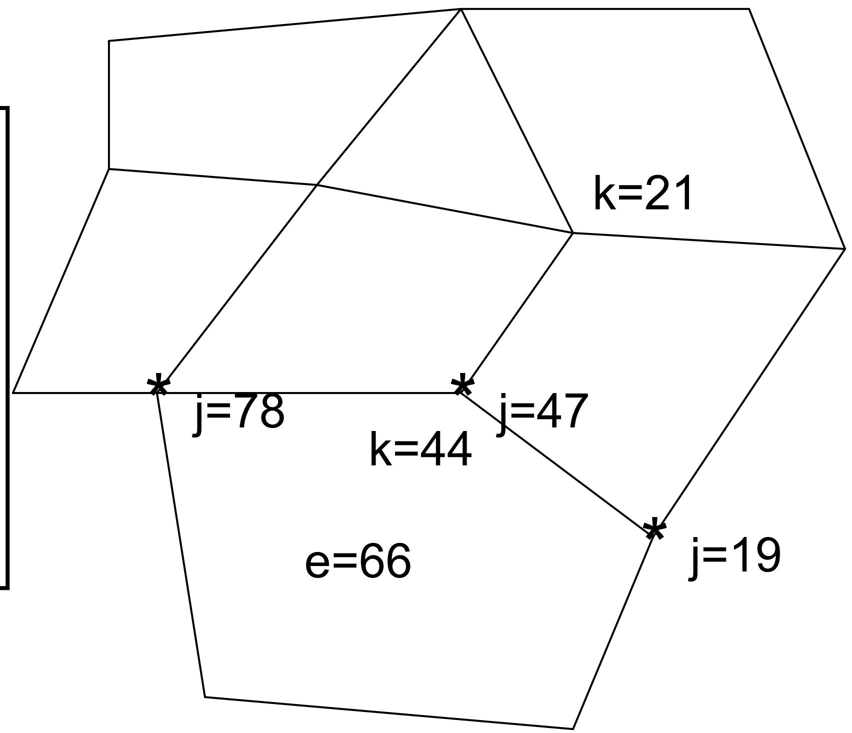
# Mesh Connectivity

Global Indices

e:Elements

j:Vertices

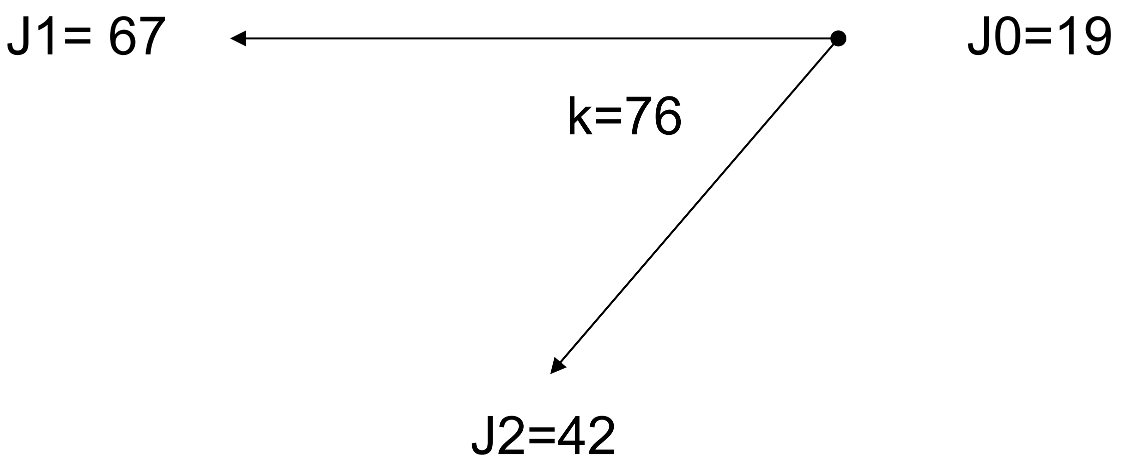
k:Corners




One-to-one map: (e,j) ←→ k

# Mesh Corners

Given global corner index k, we assume there exists maps from k to global vertex indices J0=J0(k), J1=J1(k), J2=J2(k) that define a *Mesh Corner*.




Ordering of J0, J1, J2 determines *orientation* of corner.

Mesh vertices have *coordinates*: e.g., (x67,y67).
Coordinates are those of the *computational* mesh (the one being optimized)

# Mesh Corner Matrices

$$A_k = \begin{pmatrix} x_{J1} - x_{J0} & x_{J2} - x_{J0} \\ y_{J1} - y_{J0} & y_{J2} - y_{J0} \end{pmatrix}$$

The corner matrix is a Jacobian-like object which contains shape, size, and orientation information about the mesh corner.

Note that the corner matrix is *small* (2x2 or 3x3).

The *units* of the corner matrices: Length.

The set of Corner Matrices in the computational Mesh:

$$\{A_k\}, \qquad k = 1,2,\ldots,K$$

# Why Matrices?

- Can use mathematical results for matrices to analyze & guide Target-matrix formulation
- Control all the first-order properties of the mesh
  - Size: determinant
  - Shape: condition number
  - Orientation: trace

Frobenius Matrix Norm: $$\|A\| = \sqrt{trace(A^t A)}$$

# The Target Matrix

Assume that for every corner matrix there is an associated target matrix.

$$A_k \leftrightarrow W_k$$

W is same size as A (2x2 or 3x3) and has units of Length.

The target matrix describes the *desired* size, shape, and orientation of the mesh corner.

The goal is to make (via optimization) the set of computational corner matrices
as close to the targets as possible.

The concept of the optimal mesh is relative to set of target matrices, not an absolute.
Permits application specific definitions of optimal.

## Significant of the Targets:

Can control mesh shape, size, and orientation via a single objective function instead of having to use combinations of objective functions.

# Weighted Corner Matrices

$$T_k = A_k W_k^{-1}$$

Weighted Corner Matrices are Dimensionless

The computational corner matrix equals the target matrix
if and only if the weighted corner matrix is the Identity matrix

If A is close to W, then T is close to I, so $|T| \sim \sqrt{n}$
(numerical problem is well-scaled).

If the weighted corner matrix is orthogonal, then the corner
matrix is proportional to the target, via an orthogonal transformation.

# Functions of Matrices

| | |
|---|---|
| Set of Real, Square Matrices | $M_n(\Re)$ |
| B is a real, square matrix | $B \in M_n(\Re)$ |
| mu is a continuous, scalar function of B | $\mu = \mu(B) \in \Re$ |
| Examples are determinant, trace, and norm | $\det(B)$, $trace(B)$, $\lVert B \rVert^2$ |
| Domain of Mu is S, a subset of the real, square matrices | $S \subset M_n(\Re)$ |
| Range of Mu on S. | $(\mu_m, \mu_M)$ |

# Mesh Corner Metrics

Local measures of mesh corner properties:

$$\mu = \mu(T) = \mu\left(A\,W^{-1}\right) = \tilde{\mu}(A)$$

$$\mu_k = \mu(T_k) = \tilde{\mu}(A_k) = \tilde{\mu}_k$$

# Barrier Metrics

We use barrier metrics to keep valid meshes from becoming invalid.

Nu is continuous metric with range:

$$\nu_m < c < \nu_M$$

(Nu,c) forms a partition of the set of Real, square matrices into three sets:

$$M_n^{\nu,c-} = \{B \in M_n \mid \nu(B) < c\}$$

$$M_n^{\nu,c} = \{B \in M_n \mid \nu(B) = c\}$$

$$M_n^{\nu,c+} = \{B \in M_n \mid \nu(B) > c\}$$

B-tilde is in the middle set.

$$\tilde{B} \in M_n^{\nu,c}$$

The metric Mu has a barrier on the middle set if Mu is continuous on c+ and if:

$$\lim_{|B-\tilde{B}| \to 0} \mu(B) = \pm\infty$$

# The Weighted Power-Mean

p is a non-zero scalar,

$$| p | > 0$$

Given positive constants c

$$\{c_n\}, n = 1,2,\ldots N$$

The weighted p-Mean of Sigma's >0

$$\{\sigma_n\}, n = 1,2,\ldots N$$

Is:

$$m_{p,N}\{\sigma_n\} = \left( \frac{\sum_{n=1}^{N} \left(c_n \sigma_n\right)^p}{\sum_{n=1}^{N} c_n^p} \right)^{1/p}$$

# Properties of the Power-Mean

Mean is order-independent. Further,

$$m_{p,N}\{\beta\ \sigma_n\} = \beta\, m_{p,N}\{\sigma_n\}$$ when $\beta > 0$

$$m_{p,N}\{\gamma\} = \gamma$$

$$\sigma_{\min} \le m_{p,N}\{\sigma_n\} \le \sigma_{\max}$$ for all p

$$m_{p,N}\{\sigma_n\} \le m_{q,N}\{\sigma_n\}$$ when p ≤q

$$(L+N)\,m_{p,L+N}^{p}\left(\{\rho_\ell\}\cup\{\sigma_n\}\right) = L\, m_{p,L}^{p}\{\rho_\ell\} + N\, m_{p,N}^{p}\{\sigma_n\}$$

# The Mesh Metric

Global assembly of the set of mesh corner metrics is done via the p-Mean:

$$m_{p,K}\{\mu(T_k)\} = \left( \frac{\sum_{k=1}^{K} \left(c_k \mu_k\right)^p}{\sum_{k=1}^{K} c_k^p} \right)^{1/p}$$

For |p|>0, requires the corner metrics to be positive.

The c's are constants supplied in advance to permit variable emphasis on the corners.

Methods for automatic construction of c's are under study.

# Element Metrics

Definition of Element metric
In terms of weighted p-Mean
of element corner metrics

$$\varepsilon_{e,p,\mu} = \left( \frac{\sum_{k_e} \left( c_{k_e} \mu_{k_e} \right)^p}{\sum_{k_e} c_{k_e}^p} \right)^{1/p}$$

Mesh Metric in terms of weighted
p-Mean of all corner metrics

$$m_{p,\mu} = \left( \frac{\sum_{k} \left( c_{k} \mu_{k} \right)^p}{\sum_{k} c_{k}^p} \right)^{1/p}$$

Mesh Metric in terms of
weighted p-mean of element metrics

$$m_{p,\mu} = \left( \frac{\sum_{e} \left( d_{e} \varepsilon_{e,p,\mu} \right)^p}{\sum_{e} d_{e}^p} \right)^{1/p}$$

$$d_e^p = \sum_{k_e} c_{k_e}^p$$

# Objective Functions

Objective Function is a function of mesh vertex coordinates.

We define the objective function to be the mesh-metric raised to the p-power.

$$F_{p,\mu}\{x_j\} = \left(m_{p,\mu}\{T_k\}\right)^p$$

By raising to the p-power, we avoid a dense Hessian matrix.

# Minimize or Maximize?

We restrict our mesh optimization problems to two basic cases:

I. Range of mu is $[a, \infty)$ where $a \geq 0$.

Assume F has at least one minimum.
F will be *minimized*, with $p>0$.
Linear average ($p=1$) will be frequently used.

II. Range of mu is $(0,b]$ where $0<b<\infty$.

Assume F has at least one maximum.
F will be *maximized*, with $p<0$.
Harmonic average ($p=-1$) will be frequently used.

Should cover most problems of interest.

# Admissible Set

Boundary vertices are not included in objective function because we assume their coordinates are fixed.

Let S be the set of all meshes with the same connectivity.
If both M and M* are meshes in S, then we write M~M*
(an equivalence relation).

If the initial mesh to be optimized is M, then
the admissible set is

$$\chi_M = \left\{ M^* \mid M^* \sim M \right\}$$

# Optimization of F

In general, F will not be convex.

If F is convex and the initial mesh is M, the optimal mesh is:

$$M_{opt} = \underset{M^* \in \left( \chi_M \cap M_n^{\nu,c+} \right)}{\arg\min} F_{p,\mu}\left(M^*\right)$$

(or arg max).

If F is not convex, we will seek a local minimum (or max).

# An example Metric

A mesh 'replication' metric:

$$\mu(T) = \frac{1}{2}\left\| T - I \right\|^2$$

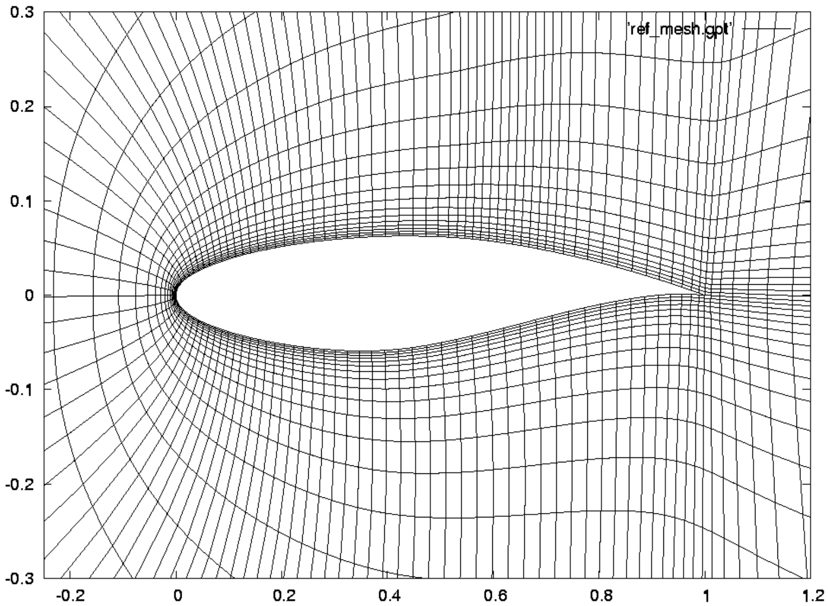
If T=I, then A=W, so if F=0 is attained, then all the mesh corners will match their targets.

We believe F corresponding to this metric is convex.

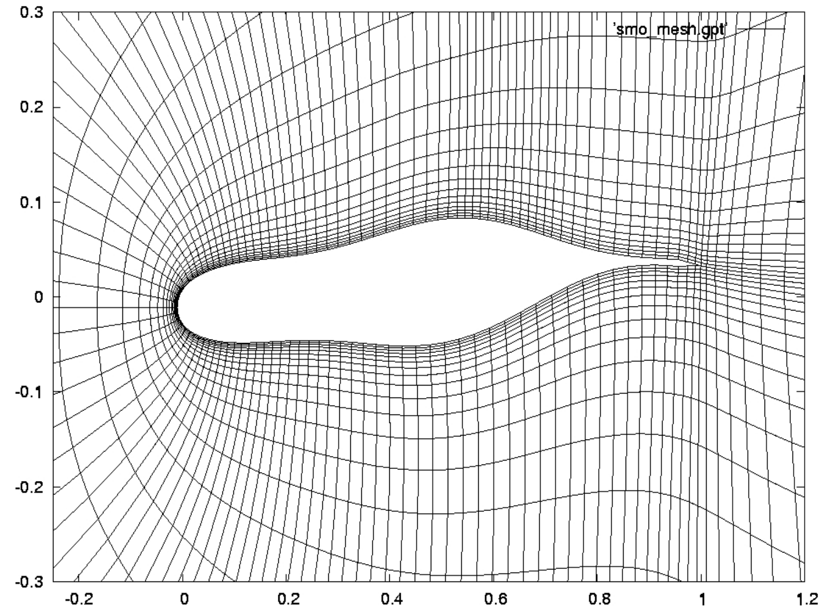
It has no barrier against folding. So use

$$\mu(T) = \frac{1}{2}\left\| T^{-1} - I \right\|^2$$

# Application: Deforming Domain Smoother




Mesh on Undeformed Geometry




Mesh on Deformed Geometry

Target Matrices computed from ‘good’ mesh on Left.

Optimized objective function using ‘Replication’ metric to obtain ‘good’ mesh on Right.

# Related Mesh Optimization Research

Tri & Tet Swapping via Integer Program (Leung) – working with primal mesh

LDRD: Optimization of Mesh Transformations (Tautges) – working with dual mesh

Two Papers: (local vs global)

Diachin, Knupp, Munson, Shontz, "A Comparison of Inexact Newton and Coordinate Descent Mesh Optimization Techniques," 13th IMR, p243-254, 2004.

Diachin, Knupp, Munson, Shontz, "A Comparison of Two Optimization Methods for Mesh Quality Improvement," IJNME Special Issue (submitted)

Submitted 50 page MICS academic-style proposal – "R-adaptive Mesh Quality Improvement using the Target-Matrix Paradigm," TriLab Proposal (Knupp, Diachin, Munson)

CSRI Summer Students

-van der Zee – target matrix

-Ricco – meshing and design optimization for 9211

Will work this summer with visiting prof. Eric de Sturler (UIUC) on interactions between mesh & solvers

# Mesquite = Mesh Quality Improvement Toolkit
(a software library)

Purpose:
Serves both as a research platform and a mechanism for delivering break-through smoothers to applications codes. Not a mesh generator, but a mesh post-processor.

Approach:
Improve computational meshes by (a) vertex-movement and/or (b) swapping via optimization

Injection of MICS Mesh Optimization Theory into Mesquite: - Target Matrix Concepts guide code development

What's New?:
- Addition of tags for storing data on the application (Kraftcheck)
- Performance (accessing move-to-owner); Kraftcheck
- Local Feasible Newton Solver (Munson)
- Target-based local relaxation smoothers (Brewer)
- Termination Criteria (Brewer, Kraftcheck)
- Interface (upgrading to TSTT spec); Kraftcheck
- Quality Assessor (quality reporting function)
- Surface Mesh Smoothing from CAD geometry (Munson, Kraftcheck)
- Smoothing for Pyramid & Wedge Elements (Brewer, Kraftcheck, Munson)
- Parallel Drivers (HMC clinic) for SLAC, Alegra, CSAR

# Mesquite Status

Development currently funded exclusively by TSTT (main customers are SciDAC apps)

Mesquite serves as a major component in testing the TSTT interface spec

- Works with TSTT interfaces: MOAB, AOMD (Interoperability)

Version 0.90 on 9200 Website (LGPL); currently at 0.96; will release 1.0 this summer.

Mesquite Linked to Cubit (0.80), Overture (0.90), Sierra (0.90), Alegra (0.95), NWGrid (0.95), DDRV (0.95)

Additional Users: UIUC Rocket Center (0.96), CEA (France) (0.90)

Many external non-Lab requests: France, Italy, Portugal, China, Canada

- 80 downloads since Sept 04 with no advertising,
- Suggests strong interest/need for smoothing.

# SciDAC Project:
# Terascale Simulation Tools & Technology (TSTT)

## Knupp = SNL Principle Investigator

TSTT Center serves as supplier of meshing and discretization technology to the SciDAC applications groups (climate, accelerators, fusion, biology)

Why should Sandia care about TSTT?

TSTT Applications provide Leverage for SNL Applications of Mesquite, Moab, DDRV, Zoltan

Deforming Meshes for Design Optimization:

SLAC = one of SciDAC's most prominent applications and is a major customer of TSTT/Sandia geometry & meshing technology. This work makes TSTT and Sandia look good.

SLAC Short term plan: deliver to accelerator designers an optimized design for an ILC cavity by end of summer. SNL plays key role here.

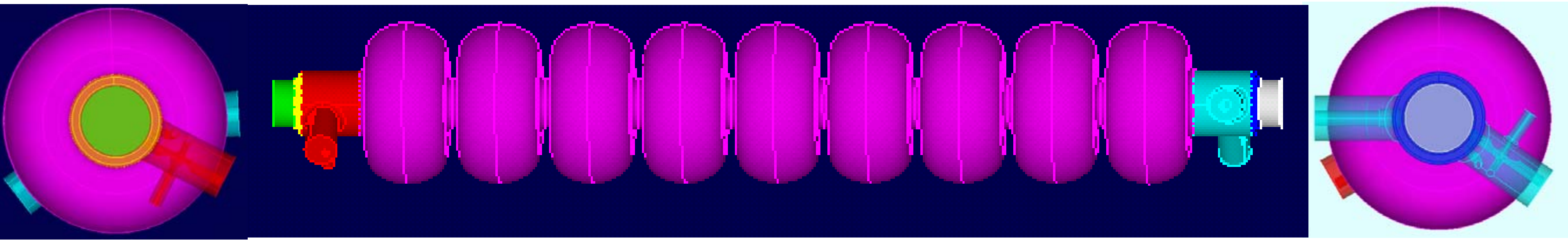
SLAC Long term plan: routinely perform design optimization studies on accelerator & electromagnetic hardware. SNL will play important role in getting there.

Other TSTT Applications of Mesquite:

- Hybrid Mesh Smoothing (Overture, LLNL) (Cubit – Hex/Tet)
- Deforming Meshes for Design Optimization (SLAC - K. Ko w/Ghattas) (9211, GOMA)
- Smoothing of Adapted Meshes (M. Shepard, RPI) (Sierra)
- Boundary Layer Mesh Optimization (P. Fisher, ANL) (Cubit)
- Smoothing of biological meshes (H. Trease, PNNL) (Sandia bio-meshes)

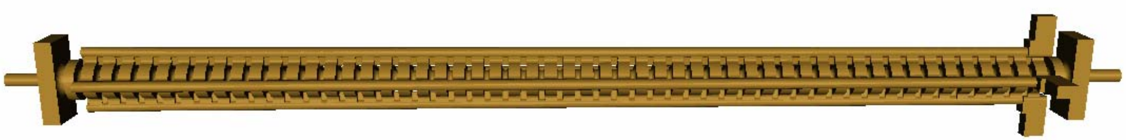
# Shape Optimization for Accelerator Structures

- Numerical modeling has replaced trial and error prototyping approach
- SciDAC adds advances that increase fidelity, speed, and accuracy:



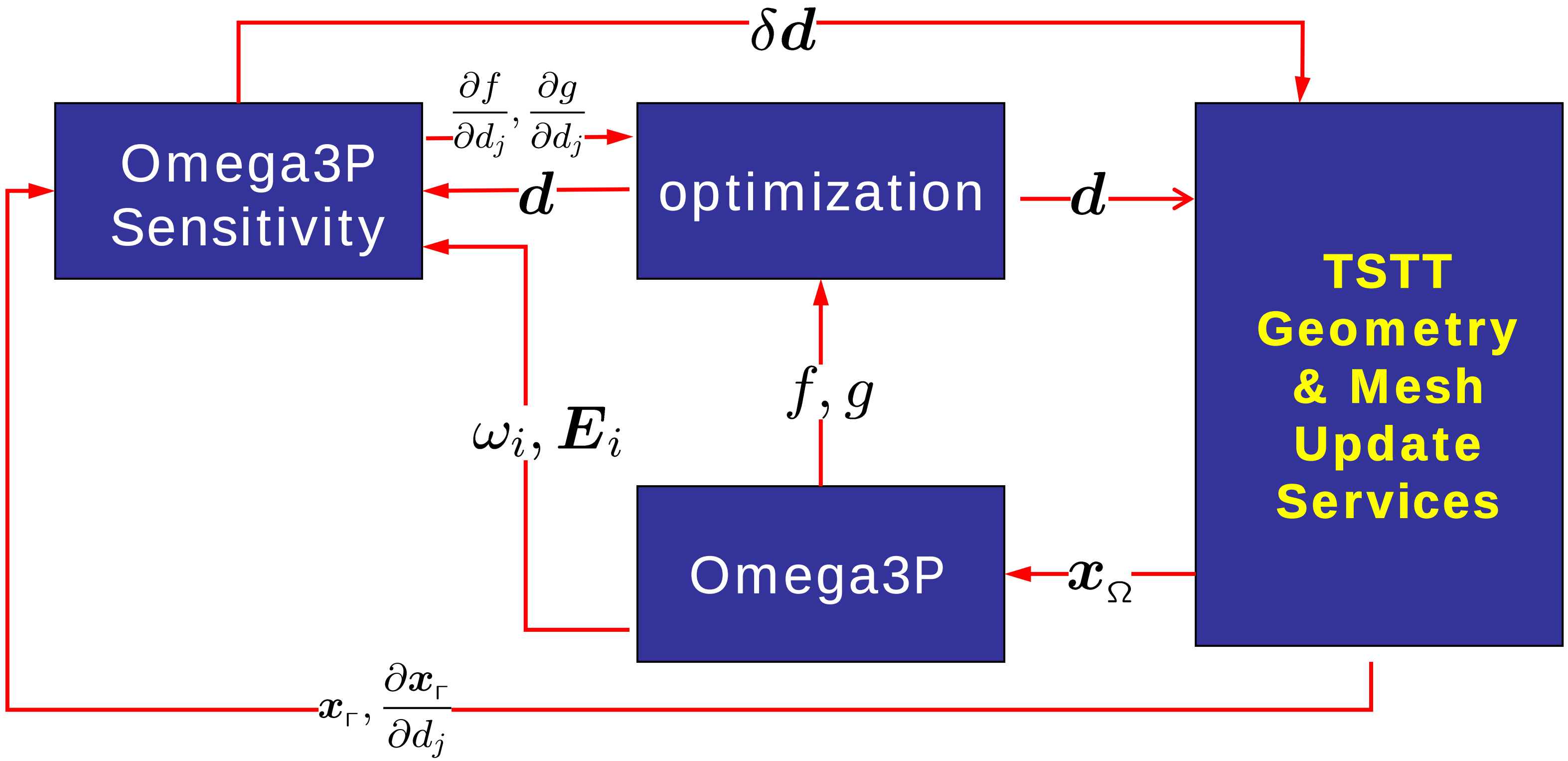
***ILC Cavity Design***

- Next generation accelerators have complex cavities that require shape optimization for improved performance and reduced cost
- AST/TOPS/TSTT are collaborating to develop an automated capability to accelerate this otherwise manual process



SLAC

# Omega3P Design Optimization Cycle




$\boldsymbol{x}_\Gamma$ : surface grid

$\boldsymbol{x}_\Omega$ : volume grid

$f$: objective

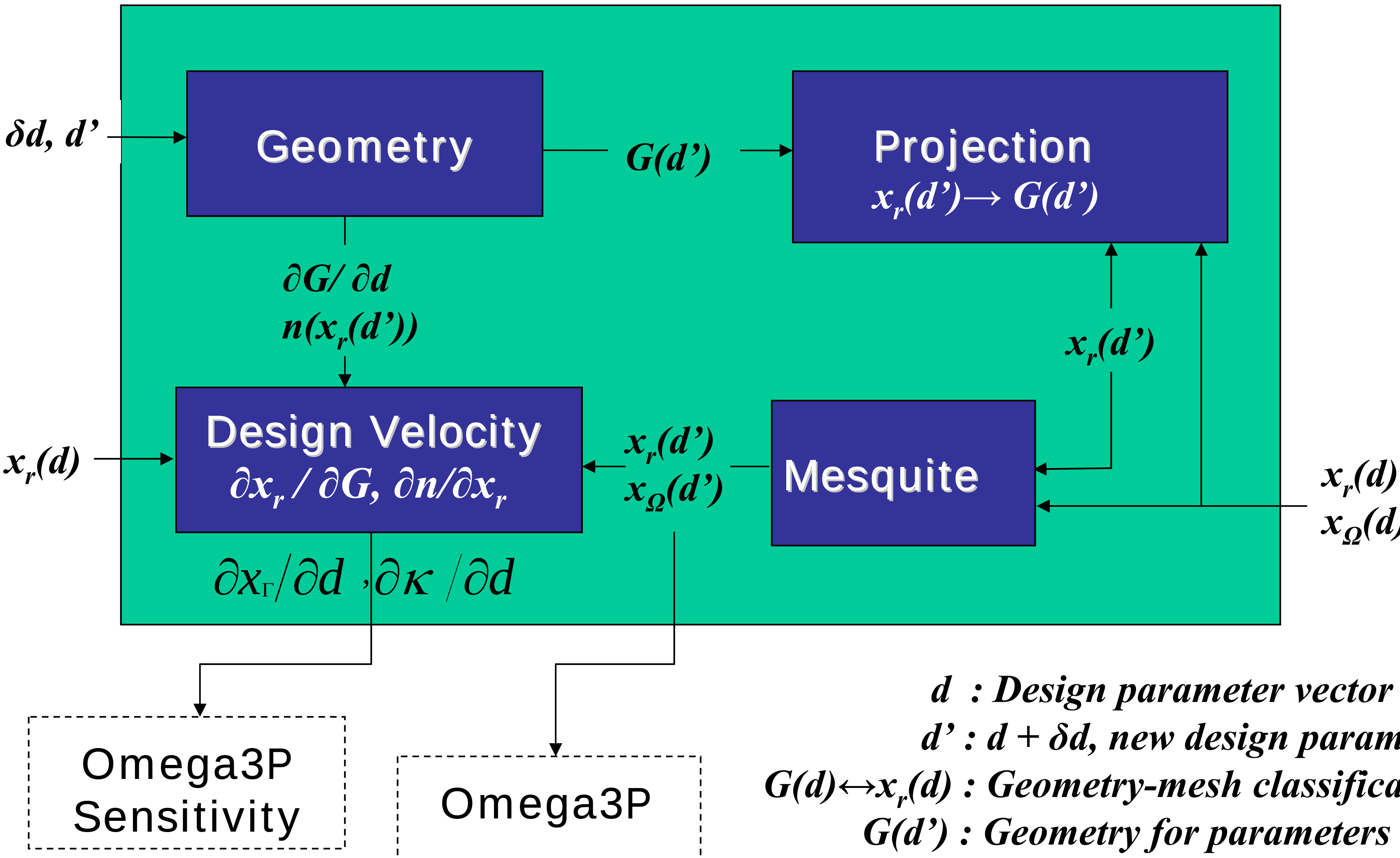
g: constraint

$\omega_i, \boldsymbol{E}_i$ : eigenpairs

$\boldsymbol{d}$: design vector

$\delta \boldsymbol{d}$: perturbed design

# TSTT Geometry & Mesh Update Services (DDRV)




*$d$ : Design parameter vector*

*$d'$ : $d + \delta d$, new design parameters*

*$G(d) \leftrightarrow x_r(d)$ : Geometry-mesh classification*

*$G(d')$ : Geometry for parameters $d'$*

*$x_r(d')$, $x_\Omega(d')$ : Mesh for new iteration*

# SNL Meshing Technology for Design Optimization

Tested Mesquite in DDRV: (Knupp, Kraftcheck)
- Added new capabilities,
- Tested robustness and performance on real accelerator cavities/waveguides

Goal of smoothing is to make mesh on deformed geometry as close as possible to mesh on un-derformed geometry, e.g., preserve biasing (for continuity of derivatives),

Uses deforming mesh smoother developed as part of target-matrix research,

Challenges:
- Movement of mesh boundary nodes onto (new) deformed geometry,
- Untangling inverted mesh & maintaining that,
- Surface mesh smoothing (normals, Hessian),
- Be robust under large deformations,
- Performance (smooth 30 surfaces with 80000 triangles and 1 volume with 1 million tets in less than 4 minutes)

Impact: DDRV will be installed within Dakota Framework this summer (S. Ricco, S. Browne, P. Knupp, T. Tautges)

# Mesquite & Alegra

Goal: Revitalize ALE option in Alegra via improved mesh smoothing algorithms (Voth, Brewer, Knupp)

FY05:

•Stand-alone tests using Mesquite (target-matrix rezone algorithm) outside of Alegra (example next slide) give better meshes than Tipton smoother,

•Mesquite version 0.92 linked to Alegra & Mesquite drivers written within Alegra

•Starting to test rezone algorithm on Alegra applications,

•Performance is currently an issue (local smoothing too slow),

•Linking Mesquite version 0.95 which has better performance,

•Testing new idea for single-stage rezoning scheme - derives naturally from Target-Matrix Paradigm
(less expensive, easier to implement)

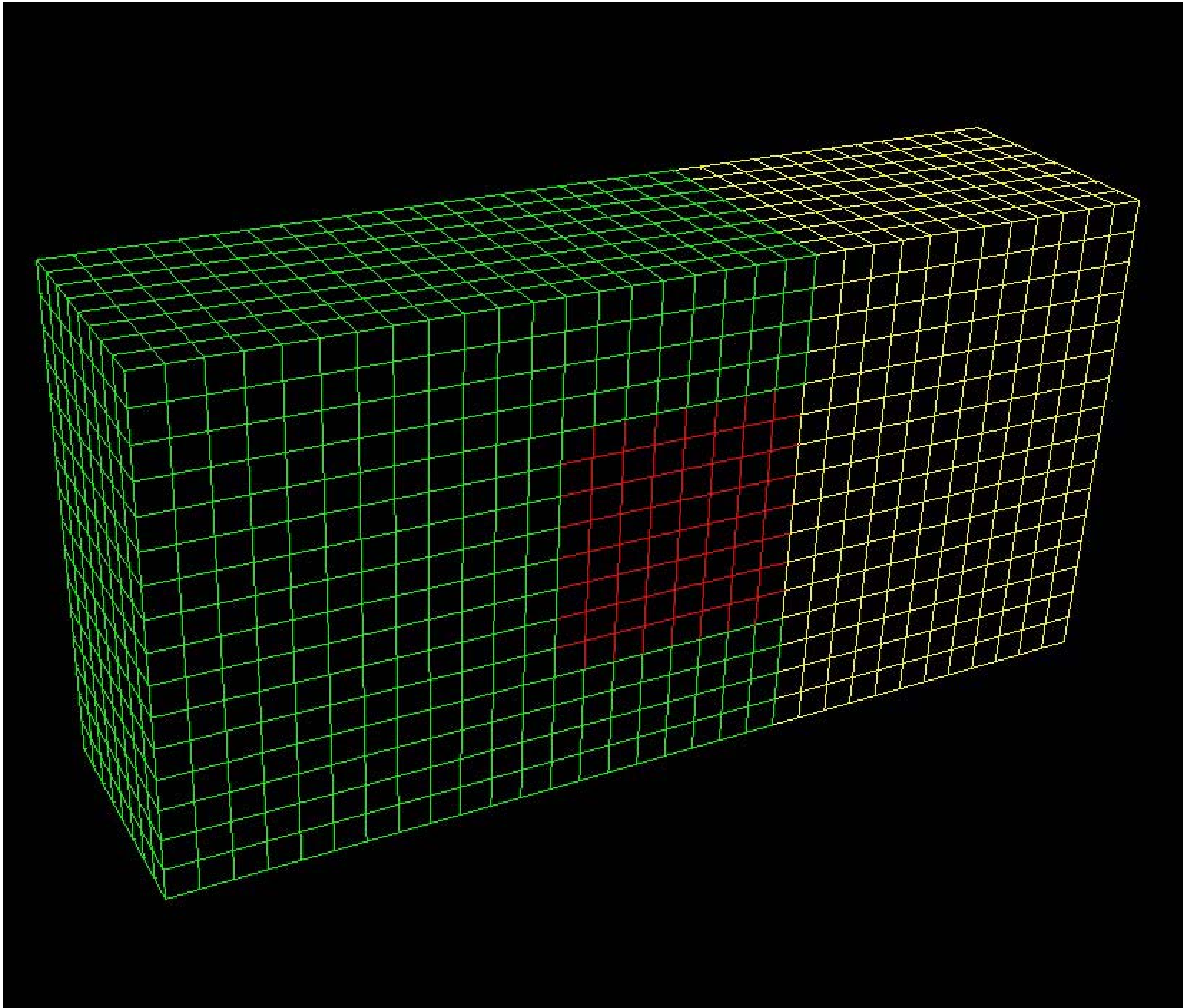
FY06 (if funded)

•Enhanced performance,

•Parallel Mesquite Implementation

•Implement in Mesquite extensions of basic rezone algorithm
(e.g., to handle interface meshes & boundary node movement)

•Further testing on important ALE applications



# Publications

1. *Bond, Knupp, Ober, "A Manufactured Solution for Verifying CFD Boundary Conditions," AIAA 2004-2629*

2. *Bond, Knupp, Ober, "A Manufactured Solution for Verifying CFD Boundary Conditions, Part II," AIAA 2005-0088*

3. Diachin, Knupp, Munson, Shontz, "A Comparison of Inexact Newton and Coordinate Descent Mesh Optimization Techniques," 13th IMR, p243-254, 2004.

4. Diachin, Knupp, Munson, Shontz, "A Comparison of Two Optimization Methods for Mesh Quality Improvement," Engr. with computers, Special Meshing Issue (submitted)

SIAM Review: Invited

V.D. Liseikin, "A computational differential geometry approach to grid generation," Springer 2003.

# Service

Short Course:
"Mesh Optimization," 13th Intl. Meshing Roundtable, Williamsburg VA, Sept 04.

Organized SIAM minisymposium, CS&E Orlando, "Mesh Optimization & Improvement"

U Texas Austin:
Contract Monitor for Graham Carey Meshing Research,

- Adaptive Modeling & Meshing,
- Design-Through-Analysis,
- Error Indicators,
- Mesh Smoothing

Reviewed approximately 12 papers on Meshing & code verification

Reviewer for:

- Intl. J. Num. Meth. Engr.,
- Finite Elements in Design & Analysis,
- Comp Methods in Applied Mechanics & Engr,
- J. Comp. Phys., and many others
- SIAM J. Scientific Computing,
- Engineering w/Computers,
- CRC Press,
- Meshing Roundtable

# Functional Requirements for a Verification Software Toolkit

## Summary of Requirements

- VST Drivers
  - ASC requirements for V&V
  - Promotion of verification efforts
  - Dependent on user buy-in
  - VST addresses grid & temporal resolution studies

- VST Non-Requirements
  - Does not address solution-adaptive meshing
  - Is not a mechanism for performing regression testing
  - Does not need to be linked to the application at runtime
  - Does not include symbol manipulation capabilities (will not calculate analytic expressions for source terms)
  - Will not determine the governing equations, exact solutions, and formal order of accuracy
  - Will not design comprehensive test suites

# Functional Requirements for a Verification Software Toolkit

Summary of Requirements (continued)

- VST High-Level Requirements
  - Pre- and post-process physics code I/O
  - Useful to wide set of SNL code groups
  - VST Input: initial mesh and code input file
  - MMS/MES analytic function library
  - Determines observed order-of-accuracy for code verification
  - Determine numerical error for both code & solution verification
  - Provide tables and data for plotting
  - Provide interpolation between grids
  - Provide several methods for estimating error for solution verification
  - Archiving of processing steps for SQE and regression testing
  - Perform calculations on mesh variables and derived variables
  - Work with both ExodusII and SAF file formats

# Technical Details – An example

Want a corner metric such that global mininuzer is A=RW, i.e., T=R.    R = arbitrary rotation matrix
Obvious metric,  |T' T – I|^2 has T=R as global minimizer.  However, it is not convex because the metric also allows T=F (flip).

Let M_n= set of n x n matrices on the real numbers, M_n(+) = the subset whose determinant is positive.
M_n(o) = { T in M_n | T' T = I } = set of n x n real orthogonal matrices
M_n(o+) = { T in M_n(o) | det(T)=1 } = set of n x n real rotation matrices

Want a corner metric:
(i) Whose domain is M_n, (no barrier)
(ii)Is bounded below on its domain
(iii) has at least one global minimizer
(iv) all of whose global minimizers belong to M_2(o+)
(v) every stationary point is a global minimizer
{vi} is locally convex in (x,y)  (i.e., positive definite Hessian)

mu_2^(o+) = ( |T' T|^2 – 4 det(T) + 2 ) / |T|^2   satisfies these properties (convexity not proved yet)

Metrics which work for n=2 often don't work for n=3, so must do separate analyses

Working on metrics in 4 basic sets:  M_n(i+), M_n(si+), M_n(o+), and M_n(so+)
Each of these sets has both a barrier and non-barrier form which must be analyzed.
A number of promising new metrics have been revealed by this type of analysis, including a condition number metric which behaves well as both volume and edge lengths go to zero.

# Parallel Drivers for Mesquite

Goal: Mesquite is a serial toolkit. Make it easy for applications to use Mesquite on parallel platforms

Who needs parallel drivers for Mesquite: Alegra, SLAC, CSAR Rocket Center

Harvey Mudd Clinic
-4 students & professor O'Neill,
-Knupp, liason,
-HMC wrote driver code to enable Mesquite to
  -(1) smooth faster via parallel computation,
  -(2) avoid mesh repartitioning & recombination
-Used LAM/MPI, HMC parallel cluster (16 processors), Zoltan
-Wrote 4 parallel algorithms (with and without cacheing & latency hiding):
  -local asynchronous,
  -local synchronous,
  -hybrid local/global synchronous
-Preliminary performance analysis (strong & weak scaling, efficiency)
-Delivered final report, driver code

Parallel drivers for Mesquite in FY06:
•Develop improved global hybrid synchronous algorithm suitable for SLAC application,
•Implement Mesquite wrapper that permits parallel smoothing

# Problems with Serial Smoothing