National Aeronautics and Space Administration

Parallel Adaptive Mesh Refinement: Lessons and Experiences in Geophysical Modeling

Charles D. Norton

Jet Propulsion Laboratory
California Institute of Technology

2007 CIG AMR Workshop

October 24th, Boulder, Colorado

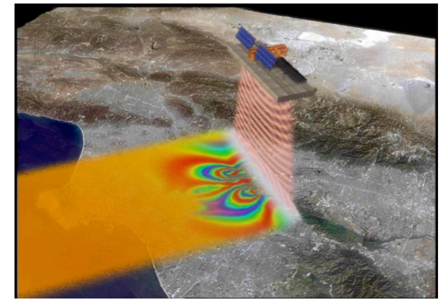
Contributors: Jay Parker, Greg Lyzenga, Margaret Glasscoe, Peggy Li, and Andrea Donnellan

JPL Clearance: CL#06-3092

Objectives of this AMR Development Project

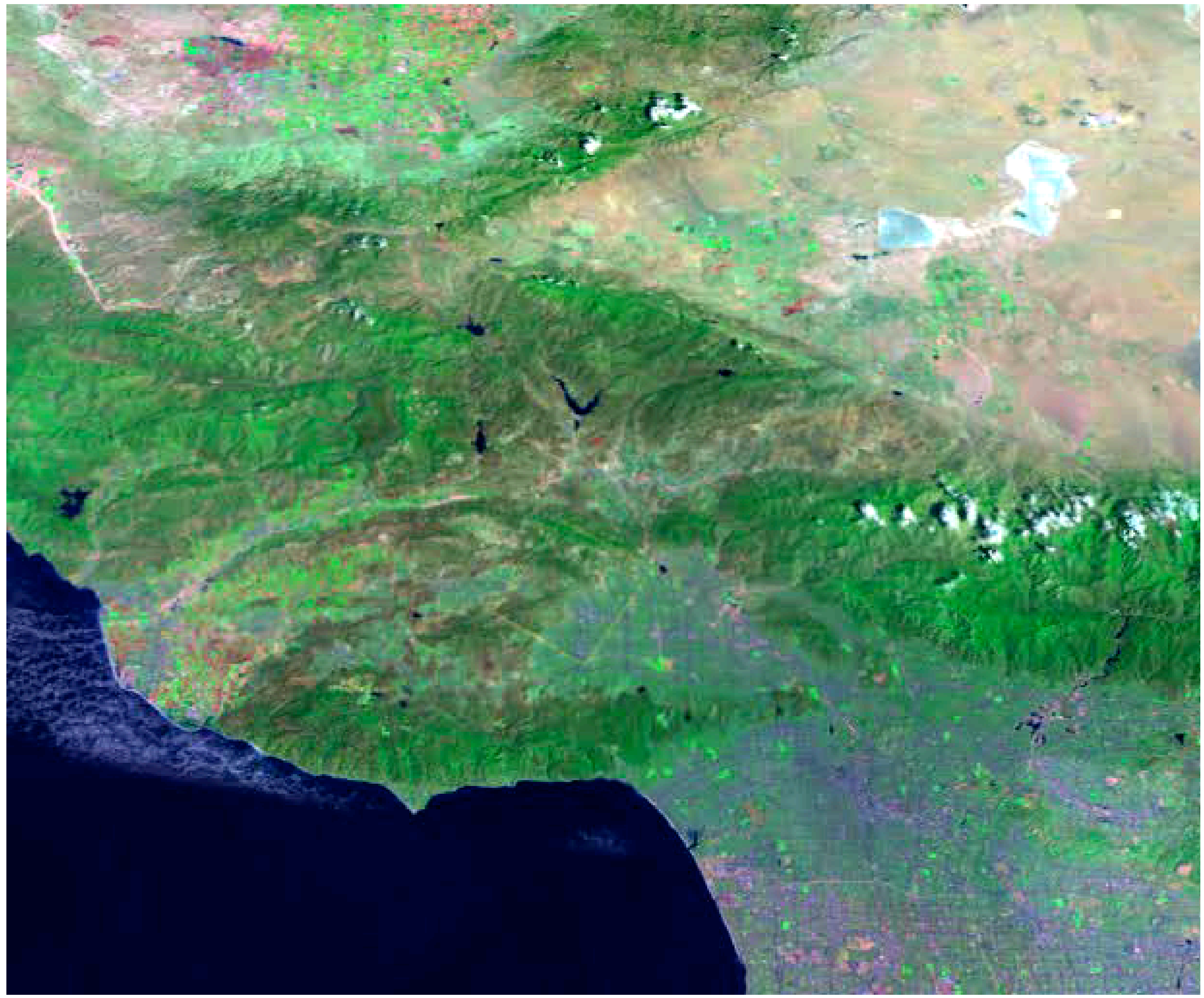
- **Purpose and Significance**
  - Develop and enhance software infrastructure supporting high performance modeling and data interpretation for potential NASA missions, such as DESDynI, for Earth surface deformation.
  - Adaptive FEM software supports such objectives for various Earth and space science applications.

- **Objectives and Topics**
  - Parallelize GeoFEST and explore application of solution-driven adaptive refinement for problems of geophysical interest.
  - Demonstrate portability among multiple systems.
  - Disseminate software to the scientific community.



500 Year Simulation of 1994 Northridge Earthquake and Postseismic Deformation

- **1994 Northridge Earthquake** simulation enabled by HPC
- **Deformation of Earth's surface** shown as InSAR fringes over 500 year timescale.
- **Buried Fault** is shown within yellow box.
- **Simulation region** is overlaid on Landsat surface data



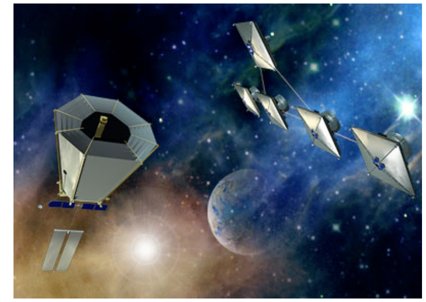
Parallel simulation showing In-Sar fringes of surface uplift and relaxation from Northridge earthquake fault event (5.6cm per InSAR fringe)
Quakesim Team: A. Donnellan, J. Parker, G. Lyzenga, C. Norton, M. Glasscoe, P. L

Modeling of Astrophysical Phenomena

Dan Spicer (GSFC), Maharaj Bhat (GSFC), Charles Norton, and John Lou

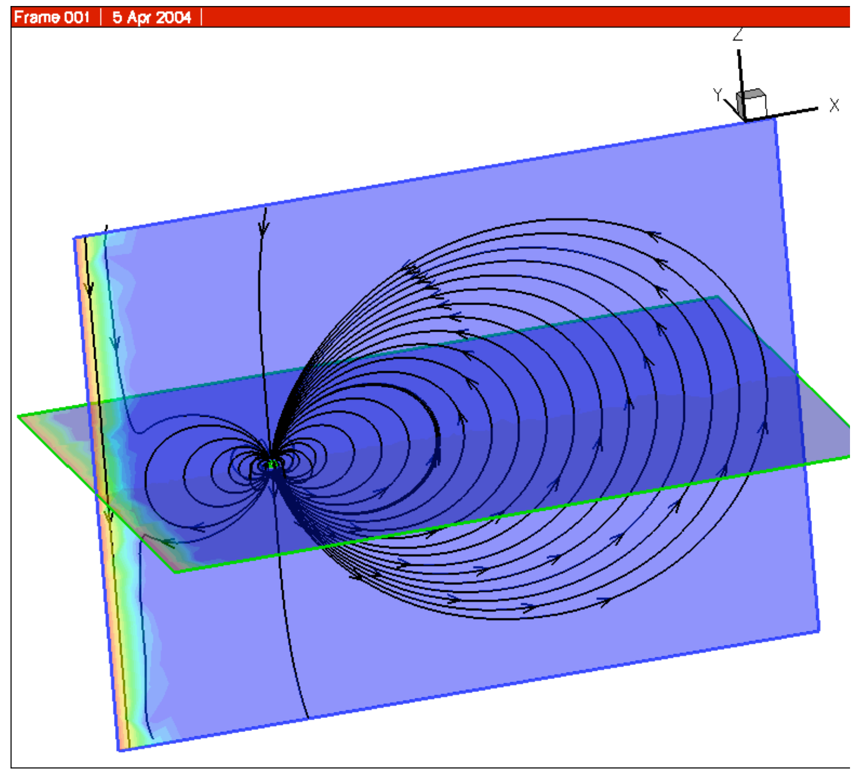
Motivation

NASA supports current and future missions aimed at observing and understanding the creation and evolution of space phenomena that will rely on simulations to model and interpret mission results



Benefits

Infrastructure tools allow one to achieve coupling of physics models to adaptive meshes for resolution of moving and curved boundary layers in flows

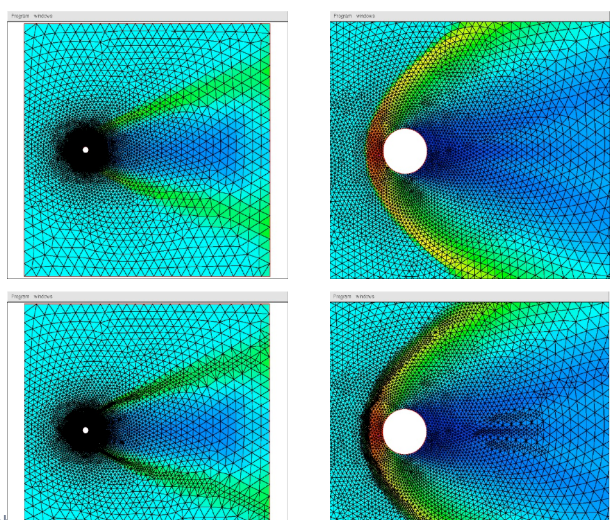


Impact

Modeling to interpret magnetospheric flows, dusty protostellar disks, star and planetary formation regions, and other scientifically significant space phenomena

2D and 3D Model of Magnetosphere

Density and magnetic field lines are shown. Many phenomena of interest have complex and moving boundary layers covering large spatial domains



Coupled Tools: APOLLO, PYRAMID

Why Adaptive Mesh Refinement?

- **AMR simultaneously improves** solution quality, time to solution, and computer memory requirements when compared to generating/running on a globally fine mesh.

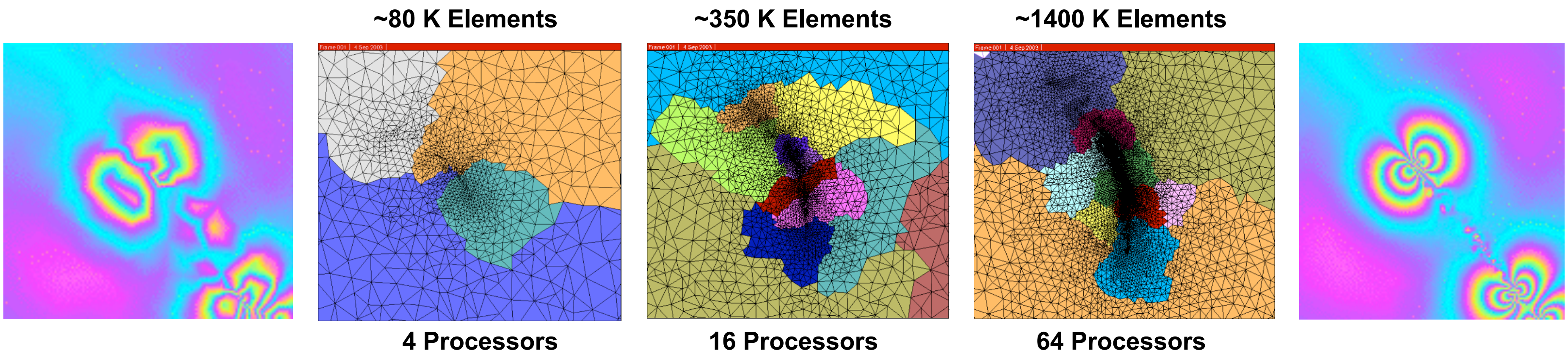


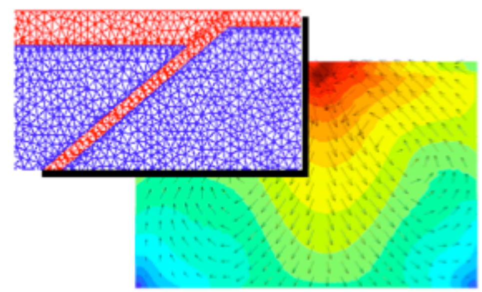
Illustration of AMR showing improvement in surface displacement solution quality with mesh density (Landers faulted mesh solved with GeoFEST/PYRAMID).

- **Future proposed NASA missions** will require support for large-scale adaptive numerical methods using AMR to model observations.
- **AMR is applied across disciplines**, but has seen the greatest success in computation fluid dynamics for predictive simulation of complex flows around complex structures.
- **Enables high resolution** simulation of interacting fault systems.

GeoFEST: Geophysical Finite Element Simulation Tool

- **What is GeoFEST**
  - Finite element solution of static elastic or time varying viscoelastic stress and strain problems
  - Unstructured 3D meshes and material variations.
  - Fault dislocations and geophysical sources.
  - Supports parallel computing, grid generation, web portal, visualization (via Quakesim).
- **GeoFEST Applications**
  - Prompt and long-term deformation effects of earthquakes.
  - Physics of fault mechanics and stress transfer.
  - Analysis/simulation of plate tectonic driving forces and material properties.

GEOFEST V. 4.5



GEOPHYSICAL FINITE ELEMENT SIMULATION TOOL

User's Guide

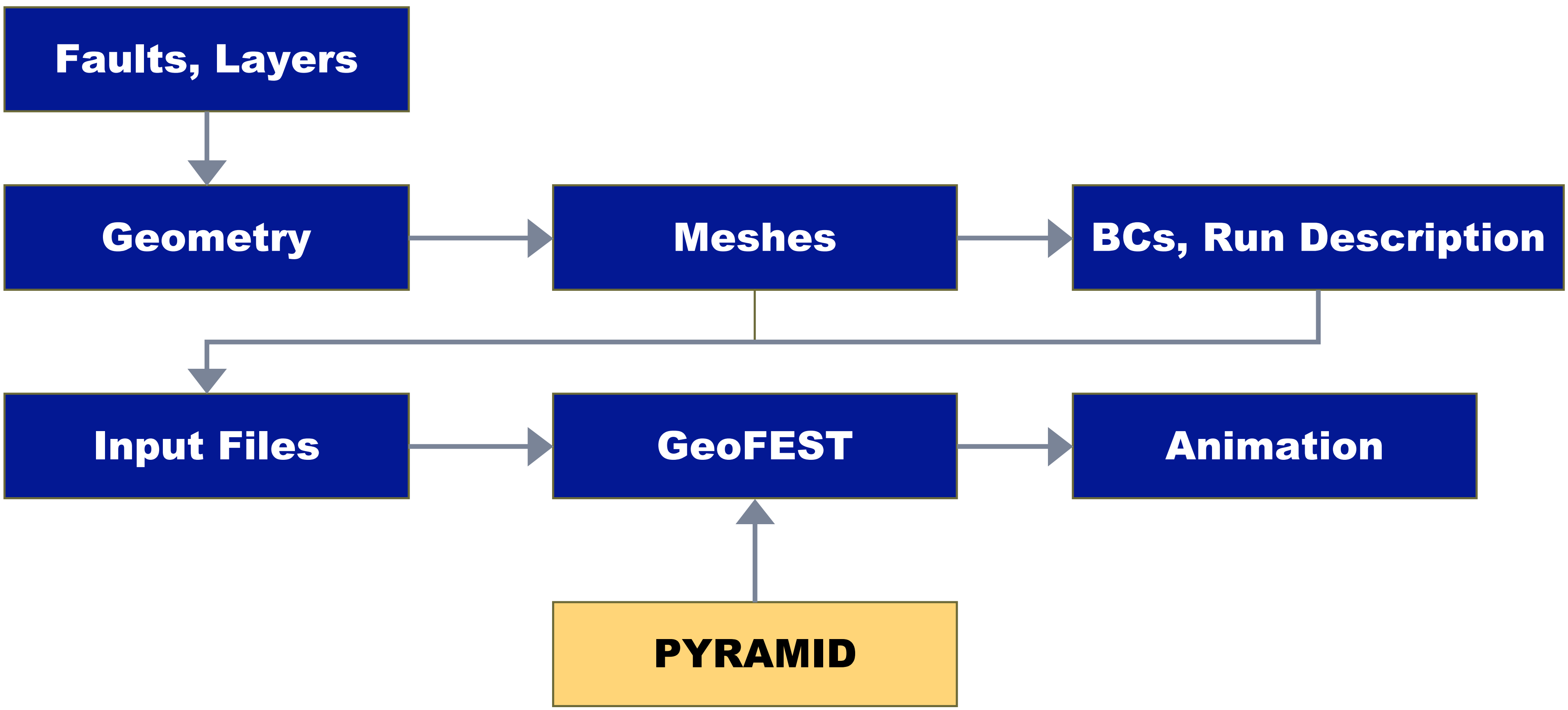
rev 5: 04/01/04

Andrea Donnellan (Andrea.Donnellan@jpl.nasa.gov)
Greg Lyzenga (Gregory.A.Lyzenga@jpl.nasa.gov)
Jay Parker (Jay.W.Parker@jpl.nasa.gov)
Charles Norton (Charles.Norton@jpl.nasa.gov)
Maggi Glasscoe (Maggi.Glasscoe@jpl.nasa.gov)
Teresa Baker (Teresa.S.Baker@jpl.nasa.gov)

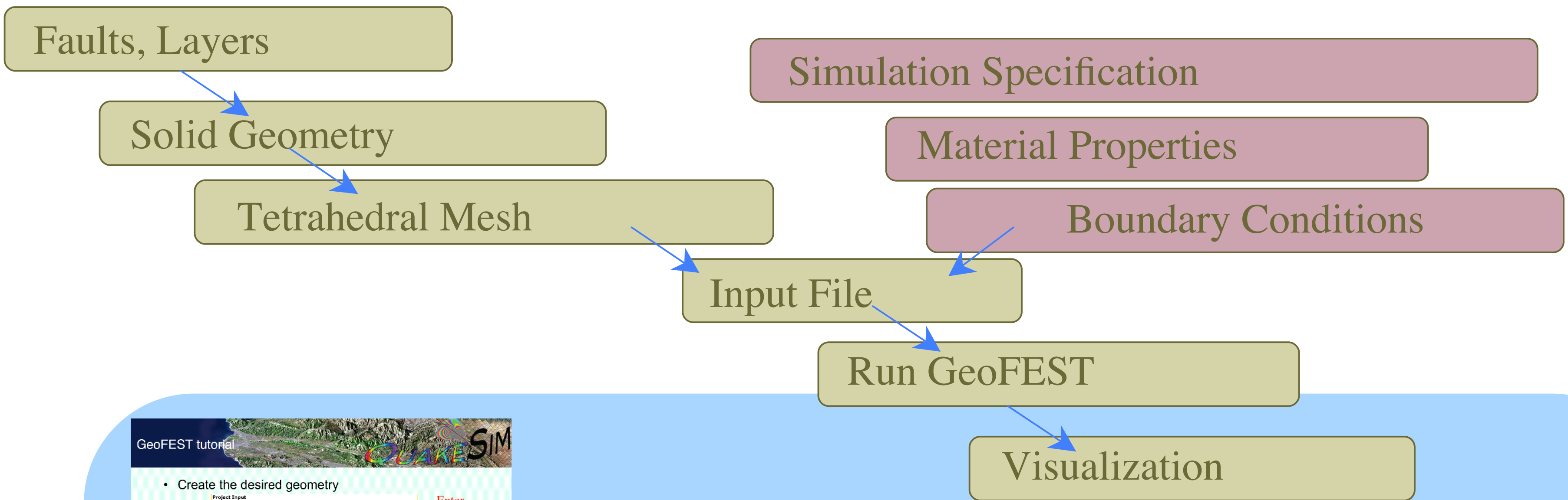
Available at http://openchannelfoundation.org

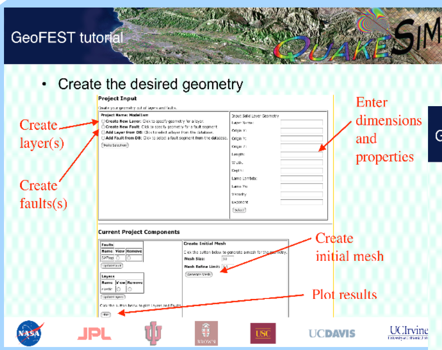
GeoFEST Run Process

- **The Web Portal or Desktop Tools** can drive this process.

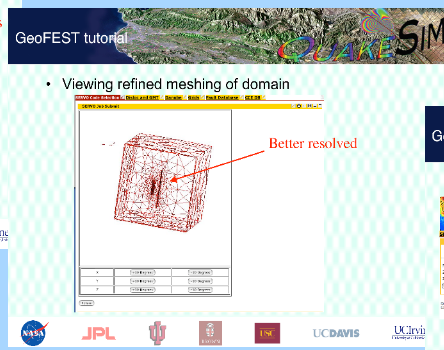


GeoFEST Problem Definition via the Portal

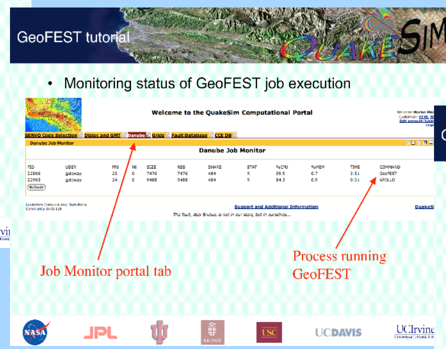




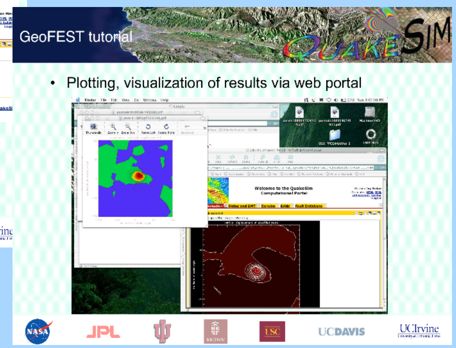
Setting geometry (QuakeTables fault database)



Mesh generation



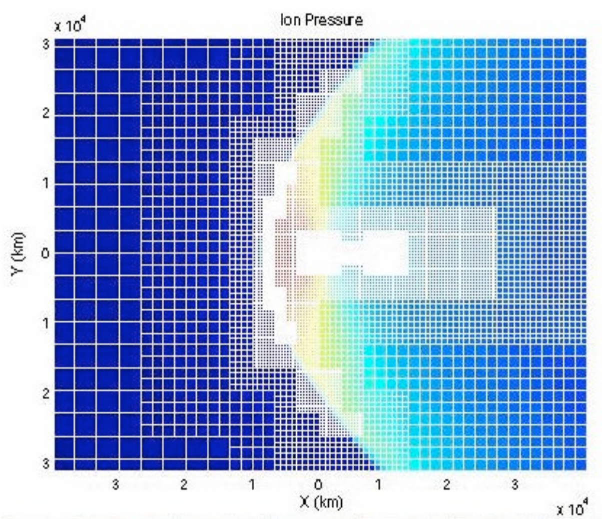
GeoFEST simulation, job submission and control



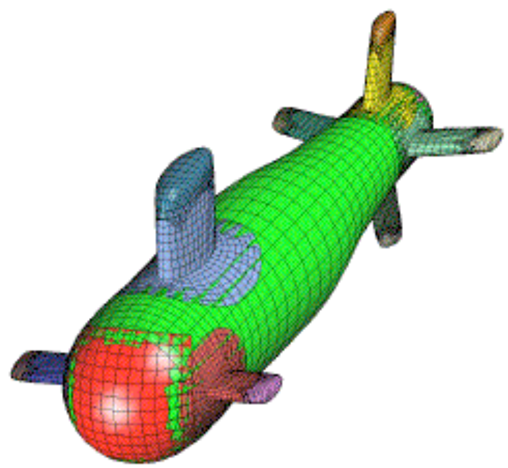
Quick-view visualization

What is the State-of-the-Art for Parallel AMR?

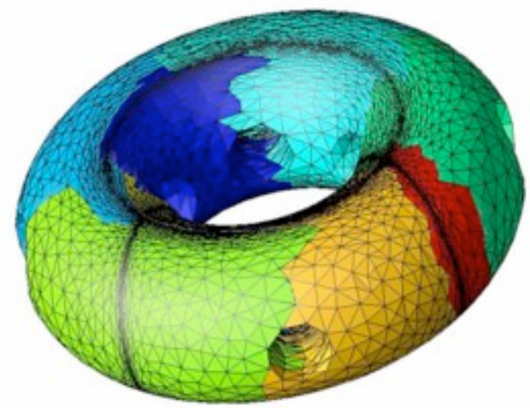
- **Structured AMR** is well established for rectilinear block-adaptive geometries.
- **Unstructured AMR** has been applied both for arbitrary geometries and where structured methods have also be used.
- **Composite/Overlapping AMR** allows "patching" of structured grids to gain benefits of both approaches (with additional complexities in mesh generation)



Structured AMR: Cometary atmospheric pressure field using PARAMESH. M. Benna et. al., GSFC.



Composite AMR: Submarine mesh gridded using Overture. W. Henshaw, et. al. LLNL

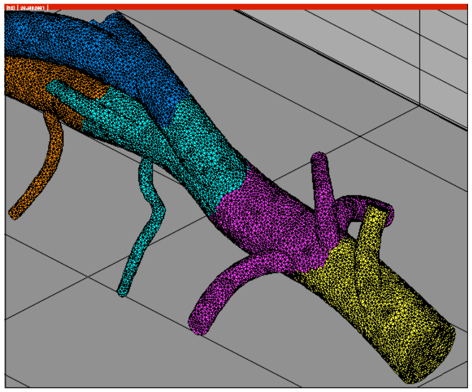


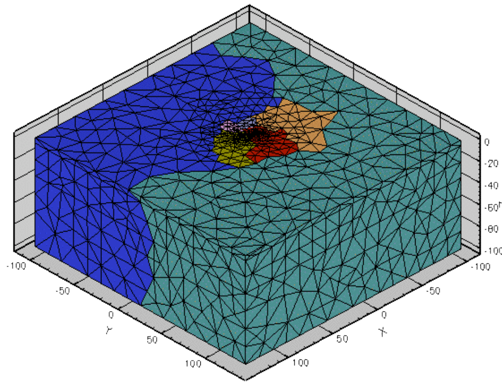
Unstructured AMR: Torus with 4 holes gridded using FMDB. E. Seol and M. Shepard, Rensselaer.

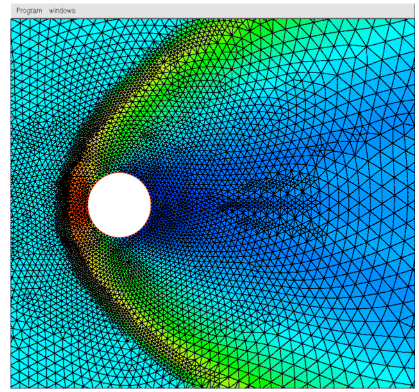
- Only a handful of tools exist for parallel unstructured AMR, but many good sequential tools exist

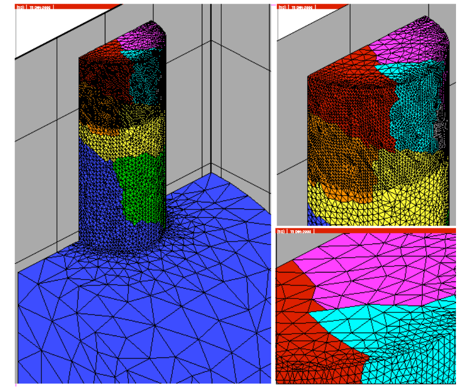
Overview of PYRAMID Parallel AMR Library

Modern... Simple... Efficient... Scalable...









Task Objective

Development of a Fortran object-based software library supporting parallel unstructured adaptive mesh refinement for large-scale scientific & engineering modeling applications.

NASA Programmatic Relevance

- Large scale modeling and simulation applications with complex geometry including support of various NASA science teams.

Design Approach

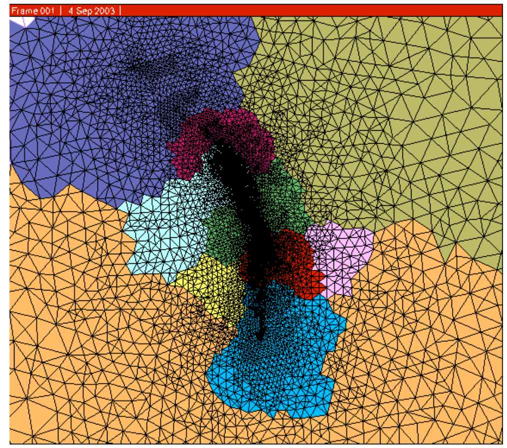
- Efficient object-based design in Fortran 90/95 and MPI.
- Automatic mesh quality control, dynamic load balancing, mesh migration, partitioning, integrated mathematics and data accessibility routines, easy solver integration.
- Scalable to hundreds of processors and millions of elements using triangles (2D) and tetrahedra (3D).
- Ease of use with development driven by application needs.
- Only refinement is officially supported at this time, but a experimental coarsening capability exists

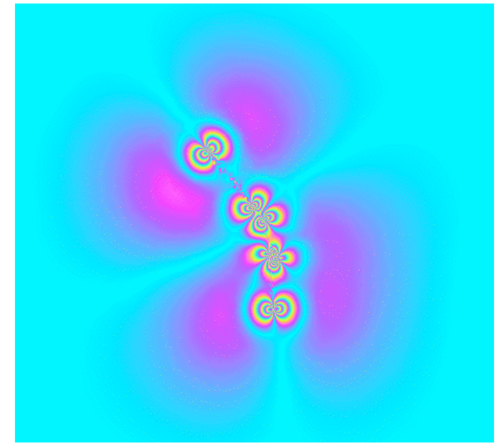
Relevant Application Areas

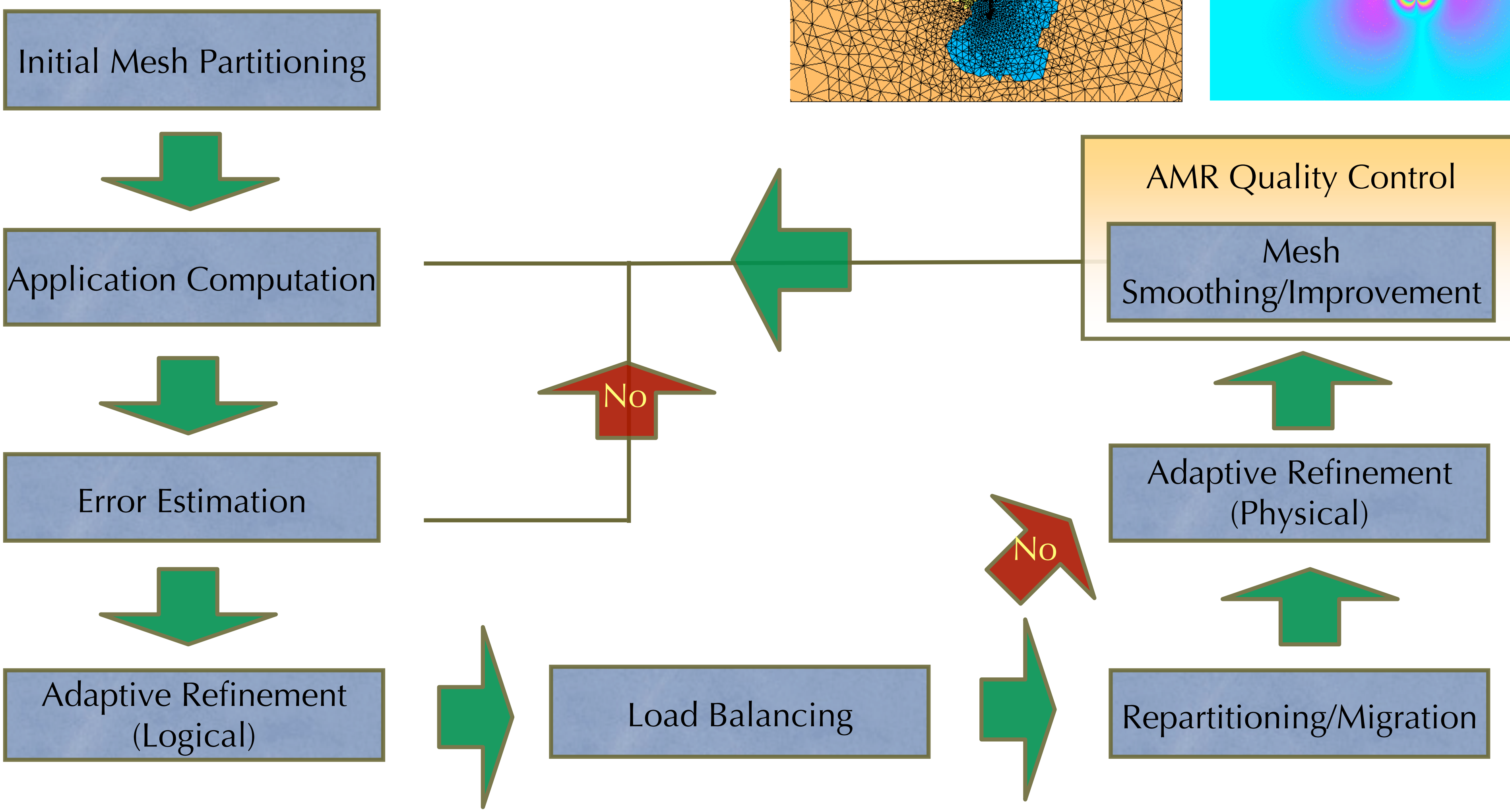
- Structural modeling and engineering mechanics for Earth and Space science applications.
- Fluid mechanics and gas dynamics.
- Solid Earth active tectonics simulation models.
- Design modeling of microwave active devices.
- Fast mesh generation from high quality coarse meshes.

PYRAMID Parallel AMR Process

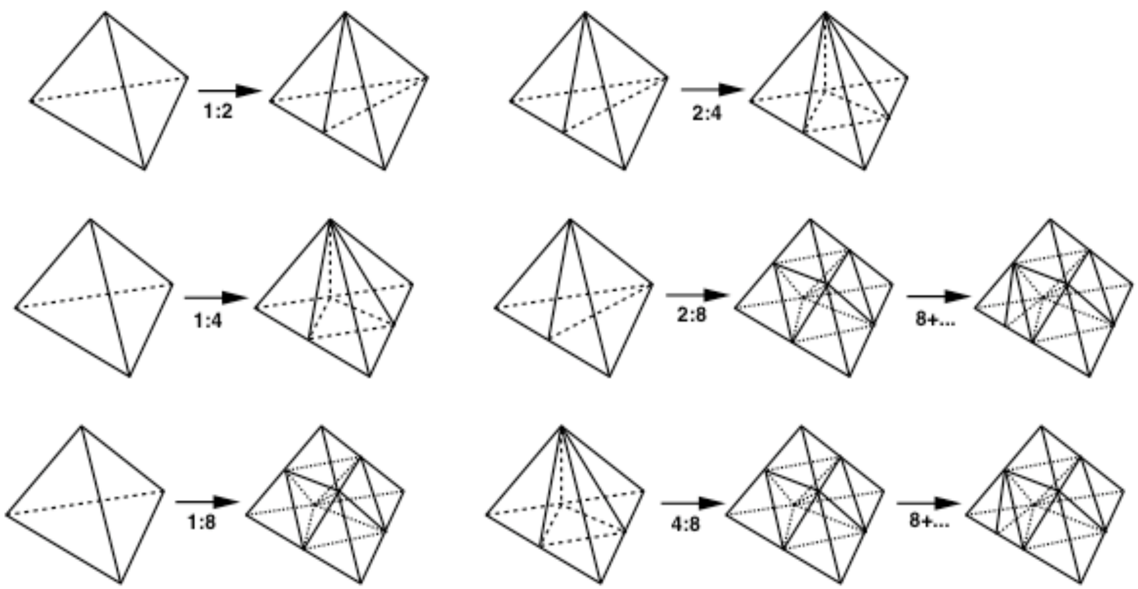
- Partitioning, Load Balancing, Adaptive Refinement, Mesh Migration, Quality Control



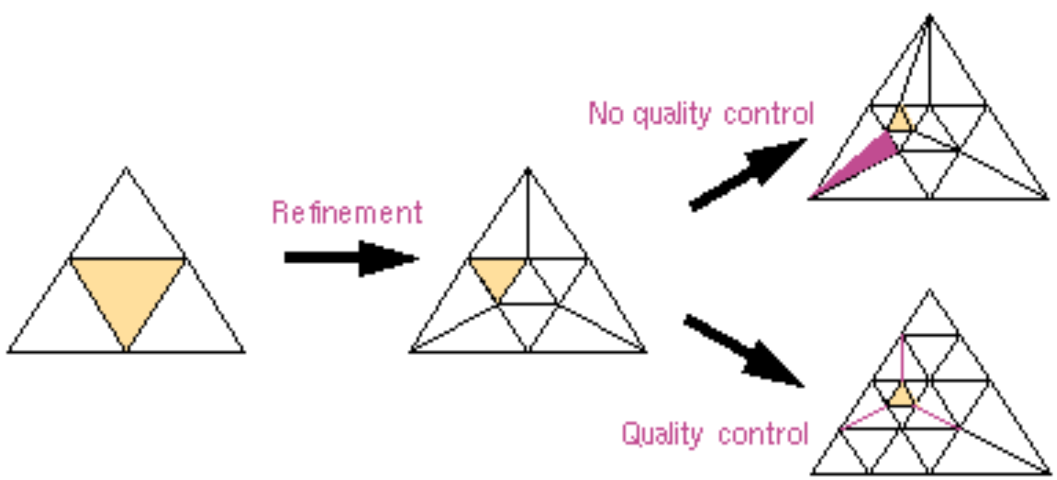




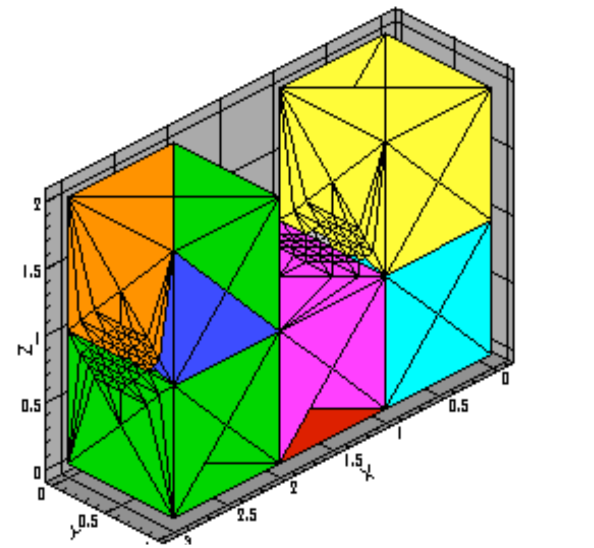
Element Refinement



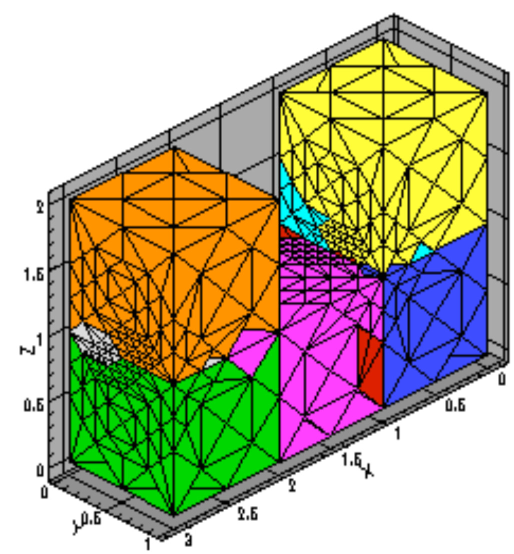
Refinement Patterns help ensure mesh consistency (no hanging nodes)



Quality Control Algorithm shown in 2D

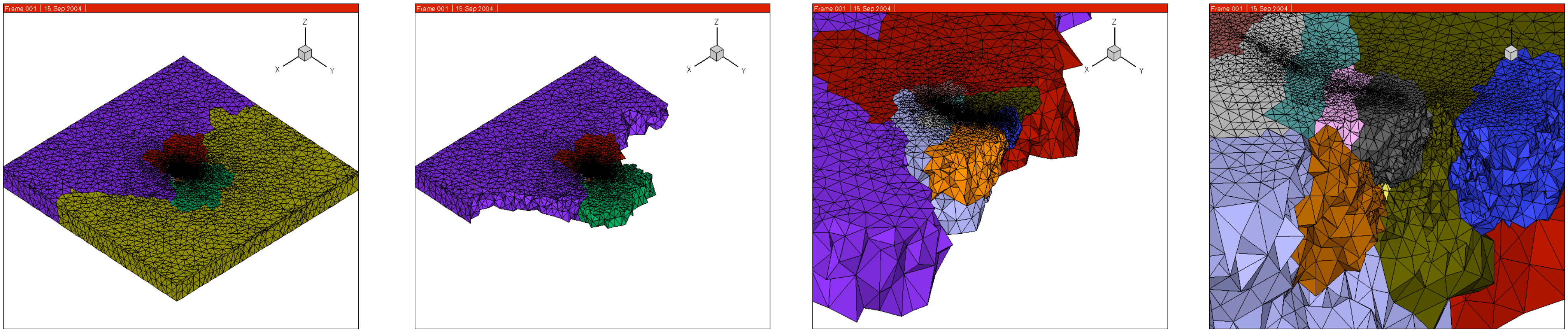


No Quality Control gives poor aspect ratios

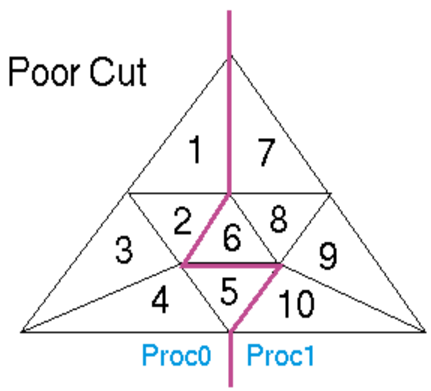


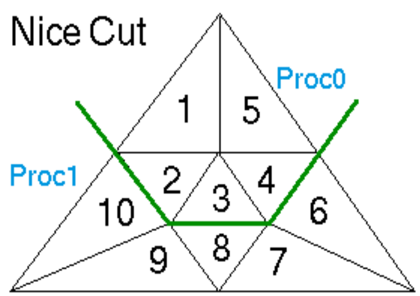
Quality Control maintains element geometry better

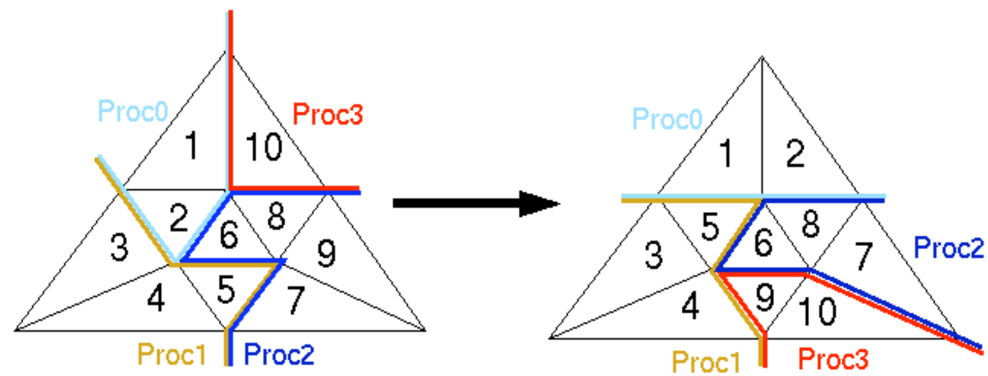
Mesh Migration, Partitioning, and Load Balancing



Element-based Partitioning achieves balanced load, connected components, and reasonable edge-cut quality. Coarse elements are weighted, migrated, then refined to load-balanced partitions.







Graph Partitioning represents an important aspect of minimizing communication at boundaries.

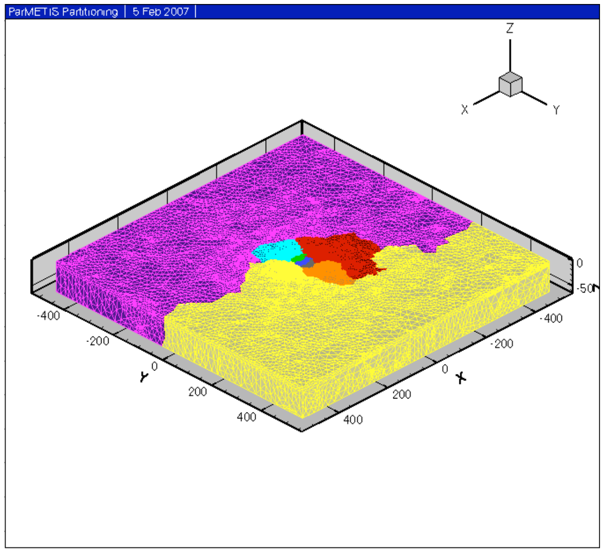
Mesh Migration Algorithms must handle irregular redistribution of mesh components efficiently and correctly

ParMetis is used for graph partitioning, but we are currently transitioning to **Zoltan**.

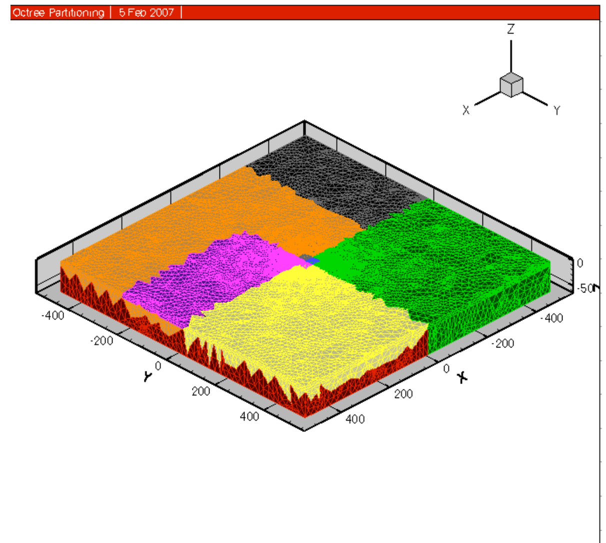
Examining Multiple Partitioning Schemes

- **Multiple partitioning schemes** impact the performance of the solvers and overall code based on issues such as communication optimization for specific problems under study.

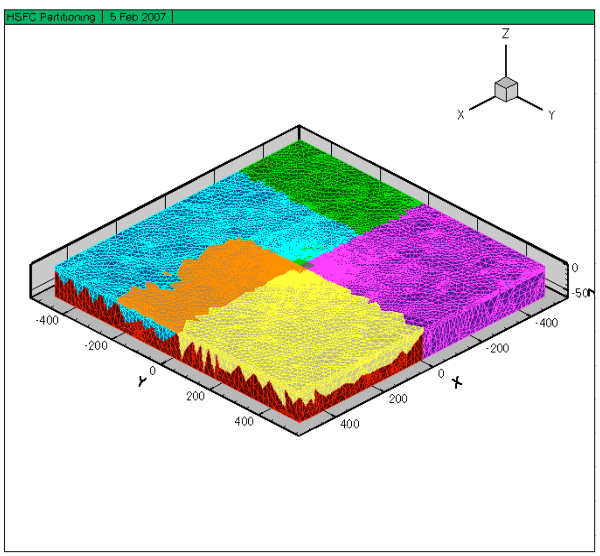
ParMetis



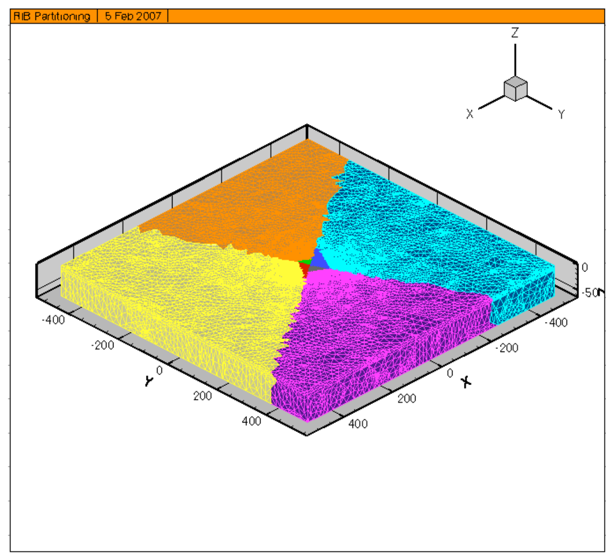
OctTree



Space Filling Curve

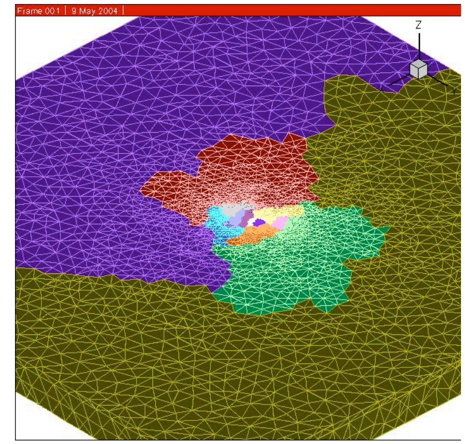


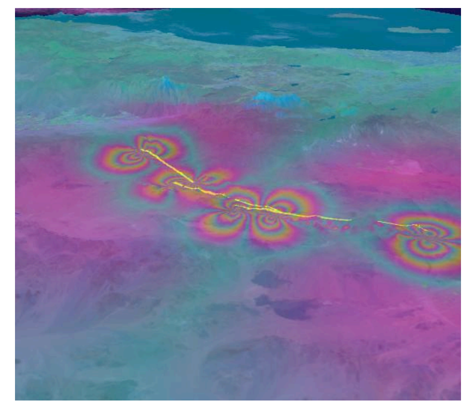
Recursive Inertial Bisction

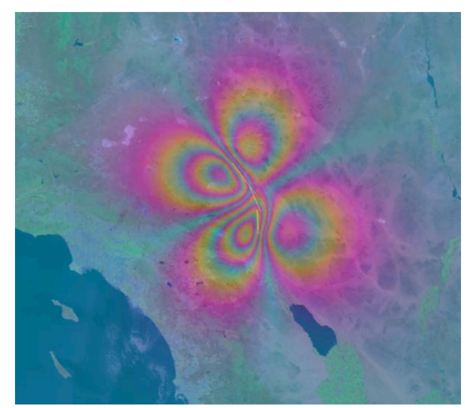


Overview of Using AMR in GeoFEST

- **Software Library Approach**
  - GeoFEST performs operations to modify the "state" of Pyramid objects and relies on Pyramid for information and operations related to parallel processing
- **PreProcessing**
  - Run GFMeshParse on GeoFEST input to create Pyramid input
- **Processing**
  - Specify # of field variables to track on each mesh component
  - Distribute GeoFEST input data using Pyramid mesh partitioning
  - Solve on the coarse mesh
  - Apply per-element error estimation criteria (strain energy)
  - Perform AMR (logical refinement, load balancing, physical refinement)
  - Update GeoFEST data structures and interpolate to new field variables
  - Solve on the refined mesh
- **Post Processing**
  - Visualize results by transformations to TecPlot format







Sample Pyramid Command

- **PAMR_GET_PARTITION_NODES**
- **Interface**
  - function PAMR_GET_PARTITION_NODES(this) RESULT(this)
- **Arguments**
  - type (mesh), intent(in) :: this
  - Integer, dimension(this%loc_boundary_nodes) :: terms
- **Description**
  - Returns, in a one-dimensional array, the global_ids of nodes on the partition boundary.
- **Notices**
  - Use PAMR_NODE_PARTITION_COUNT to get the size of the array to allocate for the result of this function.

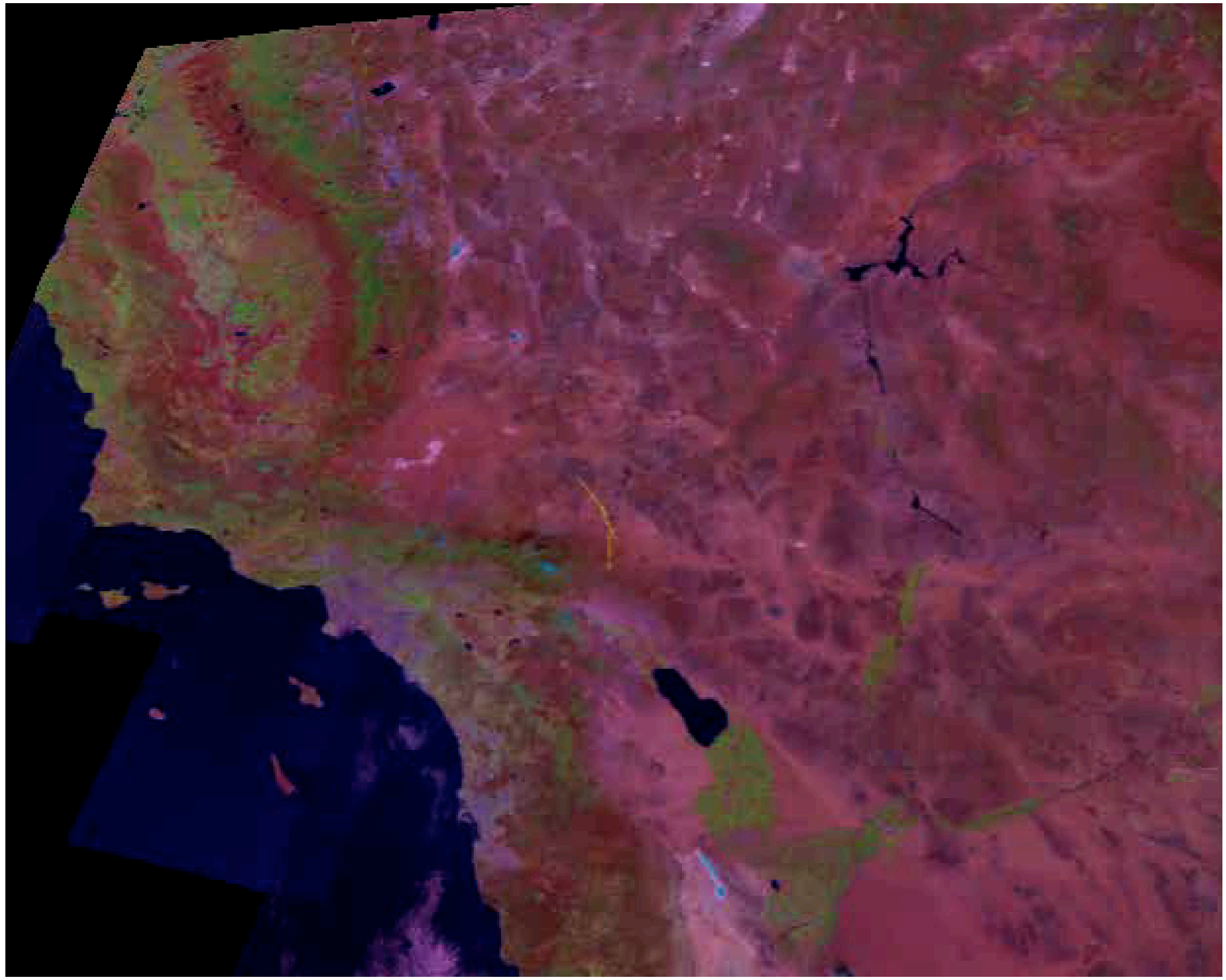
- **Most Pyramid Commands** operate on a "mesh" type/object as the primary mechanism to modify and or access mesh components.
- **Mesh index locations** can be returned as well to map directly into GeoFEST storage.
- **Over 100 Commands Exist** but really only a handful are needed to be productive.

Notional PYRAMID Call Structure

```
PROGRAM pyramid_example
USE pyramid_module
     type (mesh), dimension(10) :: meshes
     real, dimension(:), allocatable :: node_terms, element_terms
     integer, dimesion(:), allocatable :: refine_elems
     call PAMR_INIT( meshes )
     call PAMR_DEFINE_MESH_TERMS( meshes, num_node_terms = 10, num_element_terms = 5 )
     call PAMR_LOAD_MESH( meshes(1), “Meshes/input_mesh.dat” )
     call PAMR_REPARTITION( meshes(1) )
     call PAMR_SET_MESH_NODE_TERMS( meshes(1), node_terms )
     do I = 1, refine_level
          call PAMR_MARK_REFINEMENT( meshes(1), meshes(2), refine_elems )
          call PAMR_LOGICAL_AMR( meshes(1) ) ; call PAMR_REPARTITION( meshes(1) )
          call PAMR_PHYSICAL_AMR( meshes(1), meshes(2) )
          call PAMR_ELEMENT_COUNT( meshes(2) )
     end do
     call PAMR_FINALIZE( .mpi_active = .true. )
END PROGRAM pyramid_example
```

500 Year Simulation of Landers Earthquake and Postseismic Deformation

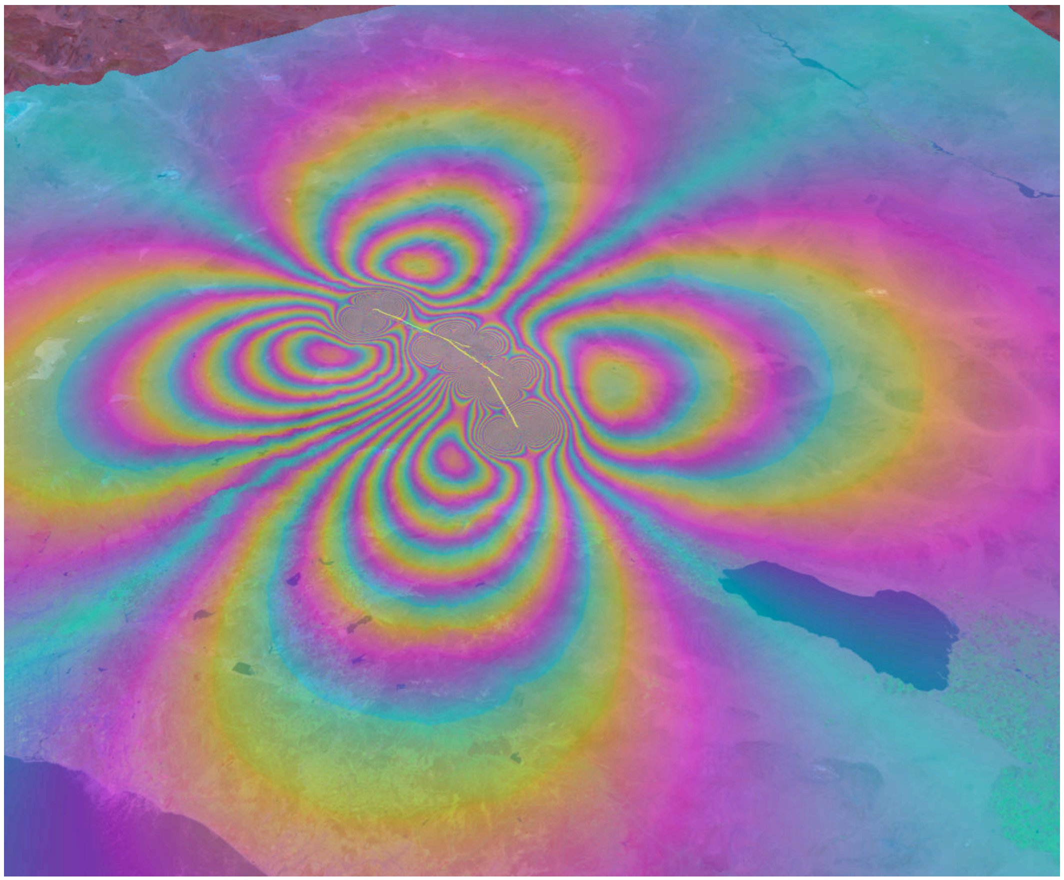
- **Deformation of Earth's surface** shown as InSAR fringes over 500 year timescale.
- **3 Faults** shown as yellow segments
- **Simulation region** is overlaid on Landsat surface data
- **Parallel AMR applied to improve solution from this 1.4 M element case** (shown next)



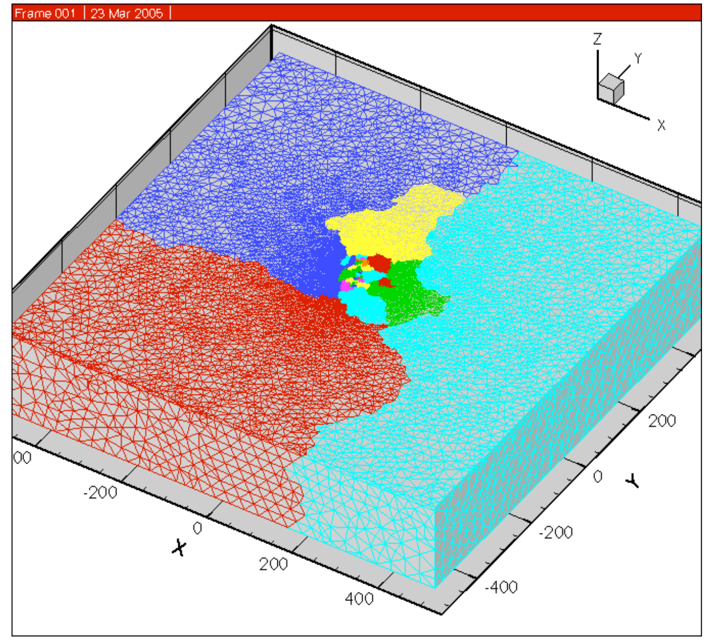
Parallel simulation showing In-Sar fringes of surface uplift and relaxation from Landers earthquake fault event (5.6cm per InSAR fringe)
Quakesim Team: A. Donnellan, J. Parker, G. Lyzenga, C. Norton, M. Glasscoe, P. L

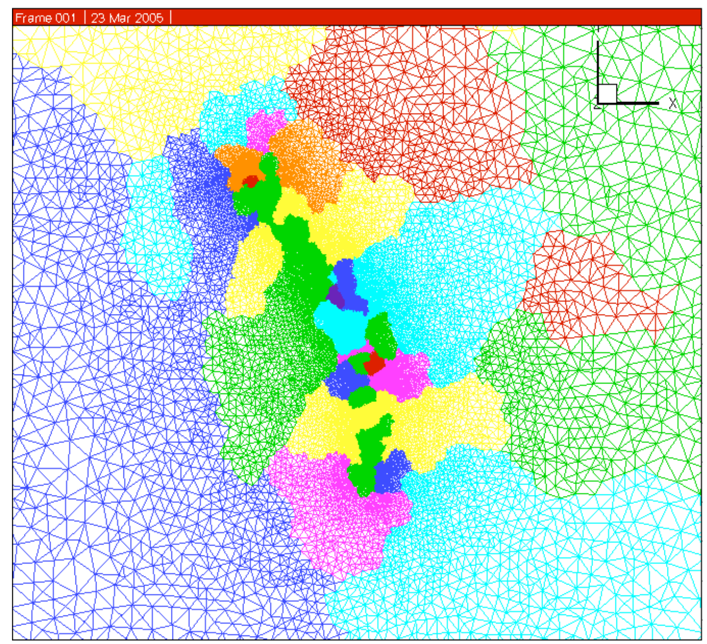
Solution Driven AMR for Landers Simulation

Views of Parallel AMR applied to form ~16 M element mesh that could not be generated using GuiVisco sequentially



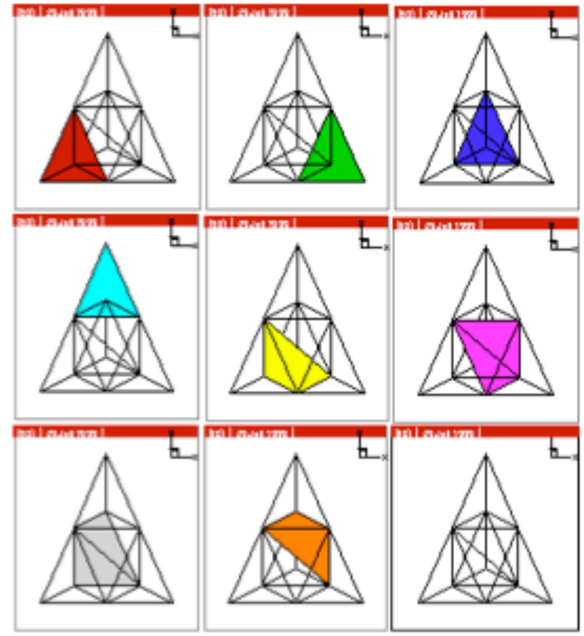
GeoFEST Simulated Surface Displacement from coseismic Landers model. Viscoelastic phase (not shown) run for 500 year simulation on ~500 procs over ~12 hours where AMR processing is negligible.

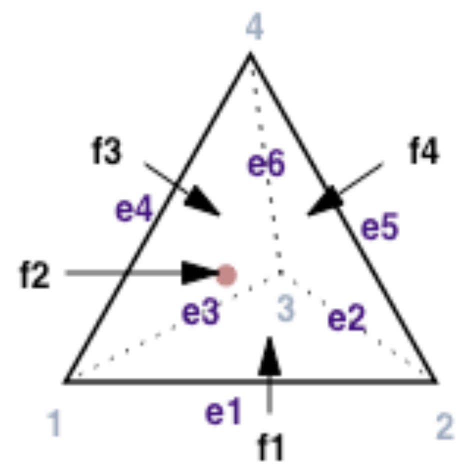




Some Points on Software Development

- **On Interfacing Between C and Fortran 90**
  - An adaptor library allows C and Fortran 90 to reference pointers, but current version requires compiler dependent knowledge of pointer representations (a compiler independent strategy is in testing)
- **On Arrays, Not Lists, For Mesh Data Structure Hierarchies**
  - Well optimized, support collective operations, efficient block-memory allocations and usage
- **Memory Usage vs. Implementation Trade-offs Exist**
  - Libraries and legacy applications often keep their own mesh representations
  - New applications can/should utilize library storage services
- **Risk vs. Reward**
  - The computational overhead introduced by AMR is generally small for time-dependent simulations
  - Refinement schemes (smoothing/quality-control) trade-offs
- **Visualization/Animation**
  - Complex, and special tools are almost always required for large simulations.





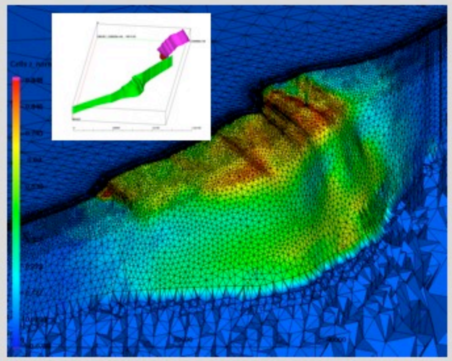
Elastic Solutions with AMR on CFM using Quakesim

National Aeronautics and Space Administration

GeoFEST elastic calculation using LaGriT-meshed CFM fault

Jay Parker (JPL/Caltech), Carl W. Gable (LANL), Gregory Lyzenga (JPL/Caltech), Charles Norton, (JPL/Caltech)

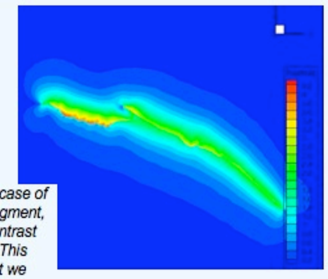
Abstract: Recent meshing techniques using the LaGriT system have enabled conforming tetraheral volumetric finite element domains that contain SCEC Community Fault Model (CFM) meshes consisting of triangular faceted surfaces, with minor adjustments of the fault geometry to improve computational mesh quality. We present very early results of computing elastic solutions using the QuakeSim finite element software, GeoFEST. A three-layer elastic model containing the San Andreas fault (from about Wrightwood to the Salton Sea) with an imposed 5 m slip is used to compute the change in stress on the nearby Cucamonga and Sierra Madre faults. The domain is 1000x1000x1000 km in extent, containing nearly 2 million linear elements representing deformation at over 3 hundred thousand node points. We anticipate this technique for including CFM fault models in LaGriT meshes will be used for large-scale simulations of the Southern California fault system, including earthquake interactions and geodetic interpretation.



Oblique view (from east) of initial mesh (saf_ver03) of San Andreas fault near San Bernardino, cut away to reveal fault complexity. Color indicates cosine of local dip. Fault definition is from SCEC Community Fault Model. Inset shows San Andreas, Cucamonga and Sierra Madre fault models

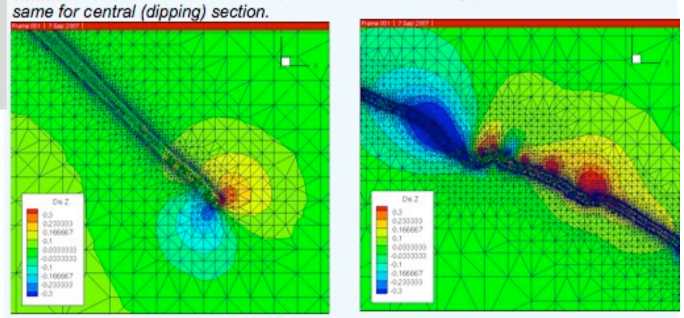
Modeling stress transfer with GeoFEST

Imposed slip on the San Andreas produces changes in stress on the other faults in the model. GeoFEST computes the full stress tensor in each element of the mesh. This simulation is the first step toward elastic and viscoelastic stress transfer calculations using the CFM.



GeoFEST Displacement Magnitude for artificial case of 5 m strike slip displacement on San Andreas segment, Cucamonga Fault and Sierra Madre Fault (in contrast to normal case with slip on San Andreas only). This shows our fault geometry and demonstrates that we have created split nodes at the correct locations.

Below left: GeoFEST vertical motion at southern tip (near Salton Sea) of San Andreas with imposed 5m strike-slip. Right: same for central (dipping) section.



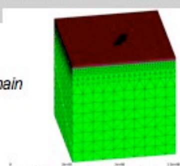
SCEC Community Fault Model

The Community Fault Model is a high-fidelity representation of the known aspects of the Southern California fault network. Each fault geometry is represented by a connected set of triangular facets, typically sub-km at the surface but much larger at depth where observations are limited.

Importing CFM faults into a finite element mesh with LaGriT

Direct insertion of a faceted CFM mesh into a finite element domain raises difficult issues of resolution, reconnection, and volumetric mesh quality. Techniques have been developed at Los Alamos that allow insertion of CFM faults into a background mesh, with moderate quality-based adjustments to the geometry. Currently this requires some hand tuning and iteration. A mesh was created with three CFM faults (Sierra Madre, Cucamonga, and the southern San Andreas), in a domain 1000x1000x1000 km in size. The large space around the active faults allows for neglect of boundary effects, and is inexpensive due to using very large elements away from the faults.

Initial LaGriT meshed domain

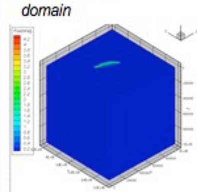


Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA

Creating a GeoFEST input file

GeoFEST is a quasistatic stress-displacement finite element system. LaGriT does not produce complete GeoFEST input files, but is used to create compatible lists of node coordinates, tetrahedral element connectivity, lists of fault nodes and their normal vector, and lists of nodes on each face of the box-shaped meshed domain. For calculations shown here, the top is a free surface, the bottom and sides are fixed. A scripting method is used to create a GeoFEST input that forces a 5m strike slip dislocation on the San Andreas fault, creating changes in stress on the other faults in the box. Fault slip is kinematically imposed upon split nodes. The material below 15 km is given twice the elastic modulus of that above 15 km.

GeoFEST solution (displacement magnitude) on surface of full domain



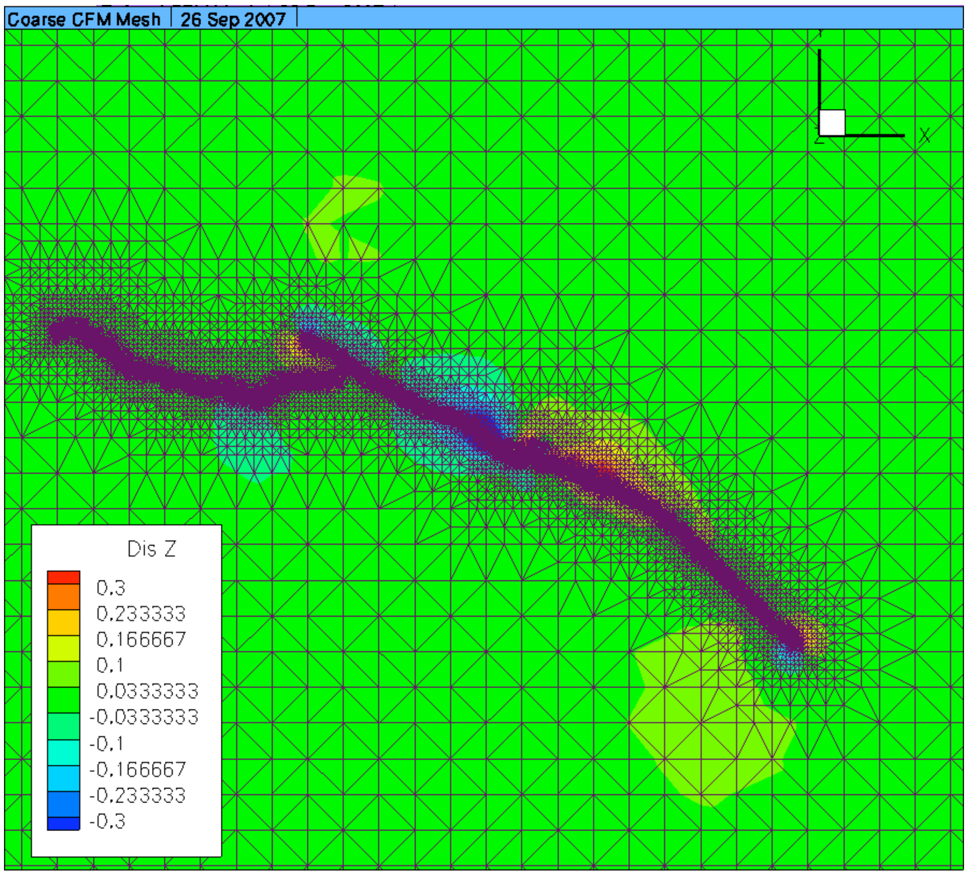
Computational Details

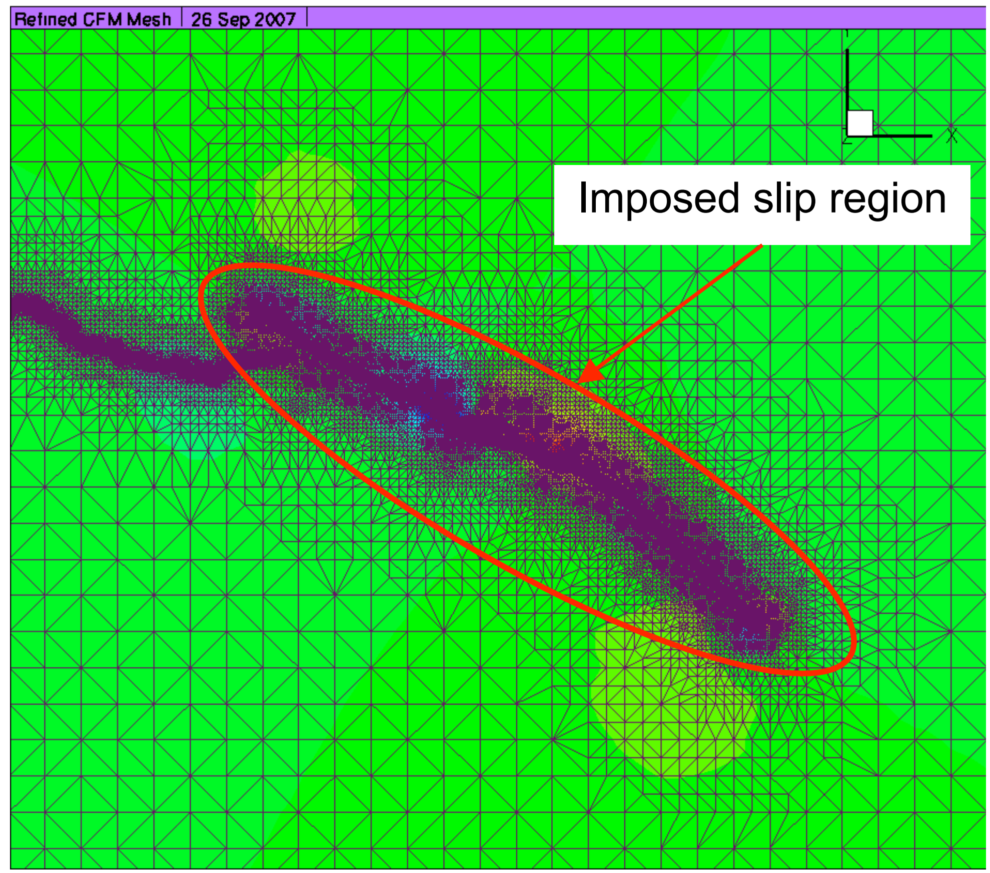
GeoFEST was run on the 1.86M finite element domain on 64 processors of the Cosmos system at JPL. The initial elastic solution supplies a map of strain energy, which is used to produce a refined mesh with about 5 million elements, producing a better solution. The changes in stress on the adjacent faults has not yet been extracted and analyzed.

Contact Information *Jay.W.Parker@jpl.nasa.gov*
NASA QuakeSim Project: http://quakesim.org/
SCEC Community Fault Model: http://structure.harvard.edu/cfm/
LaGriT Mesh Generation: http://meshing.lanl.gov/
GeoFEST download:
http://www.openchannelsoftware.org/projects/GeoFEST

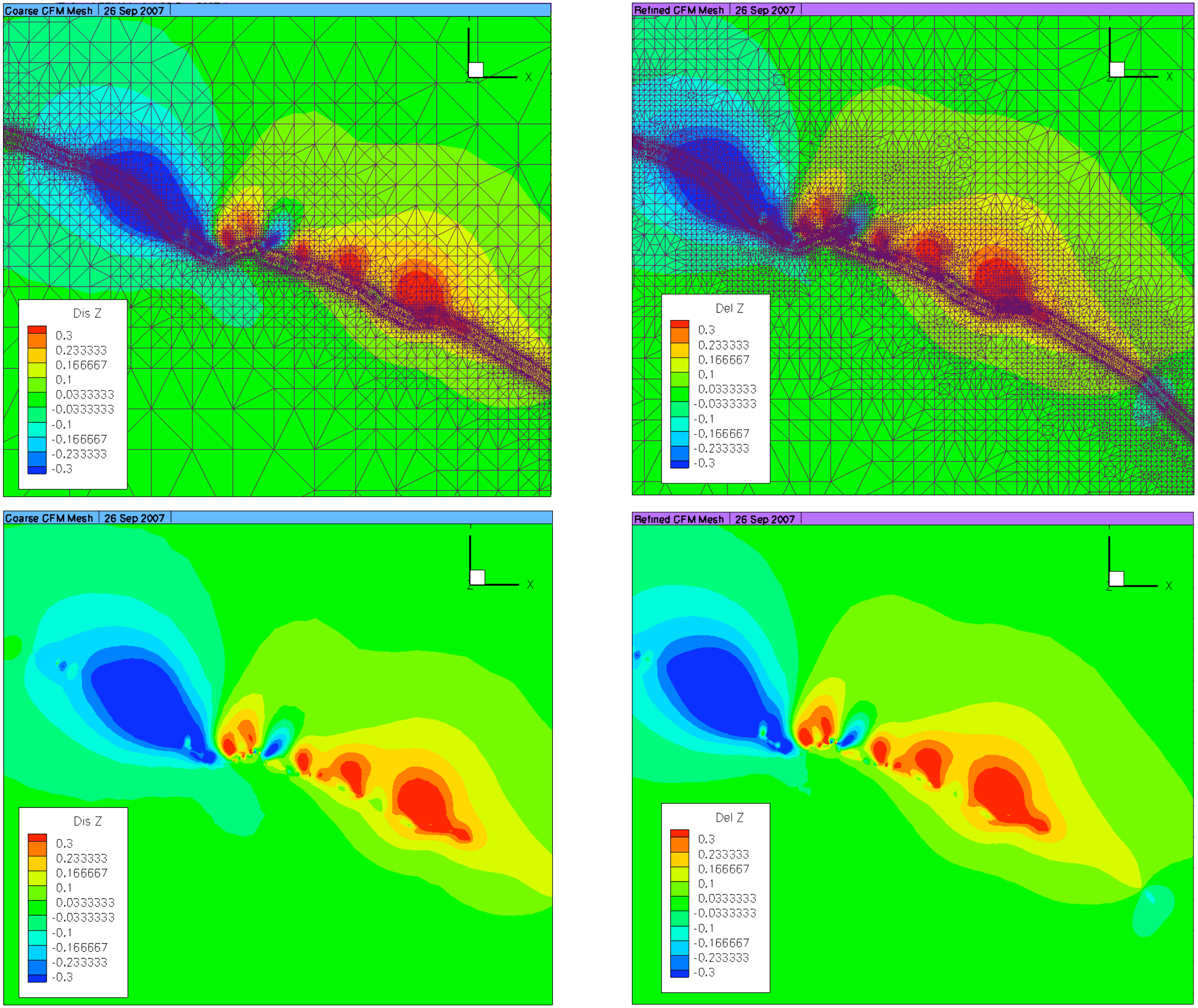
Elastic Solutions with AMR on CFM using Quakesim

- **AMR views** of elastic solution (surface of 3D mesh)
- Coarse Mesh: 1,863,336 Elements and 331,136 Nodes
- Refined Mesh: 5,164,419 Elements and 890,810 Nodes
- SCEC CFM fault model provided by Carl Gable (LANL)

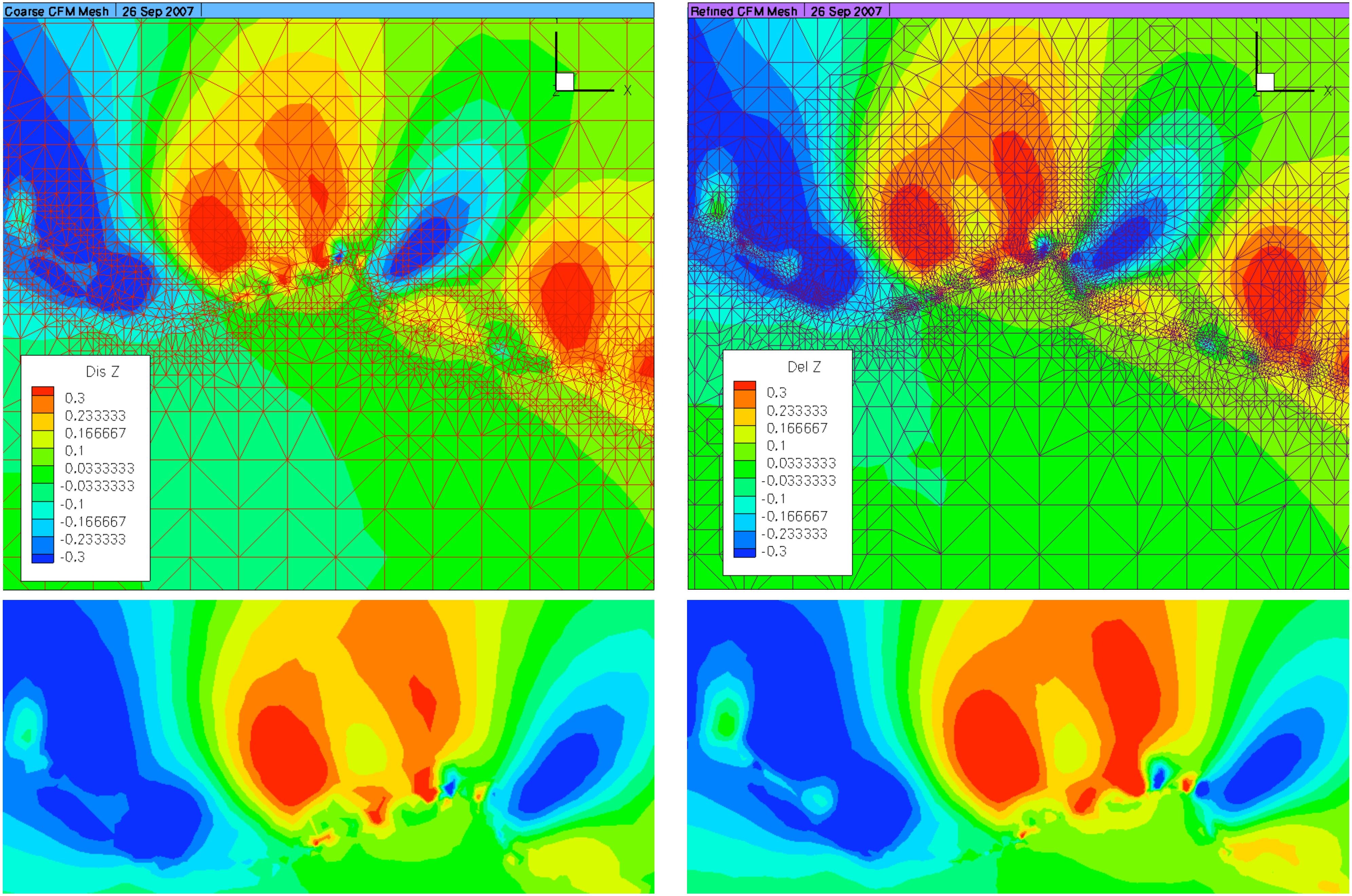




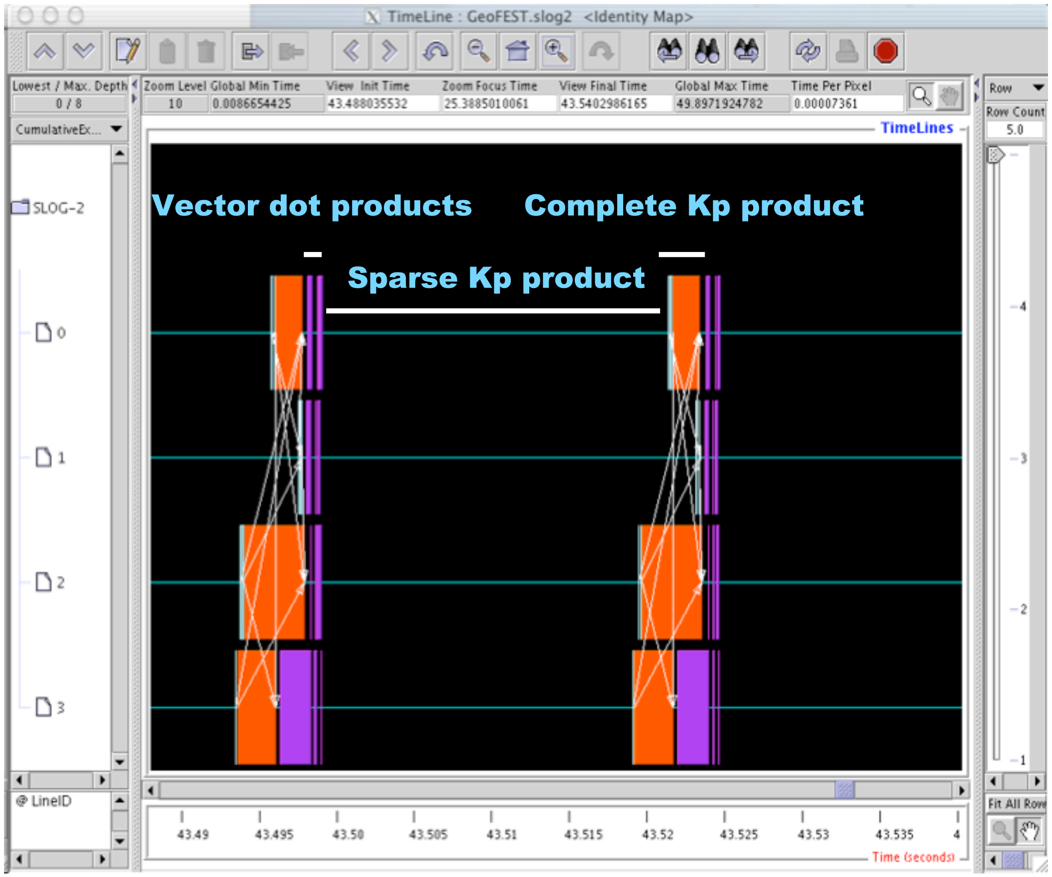
Elastic Solutions with AMR on CFM using Quakesim

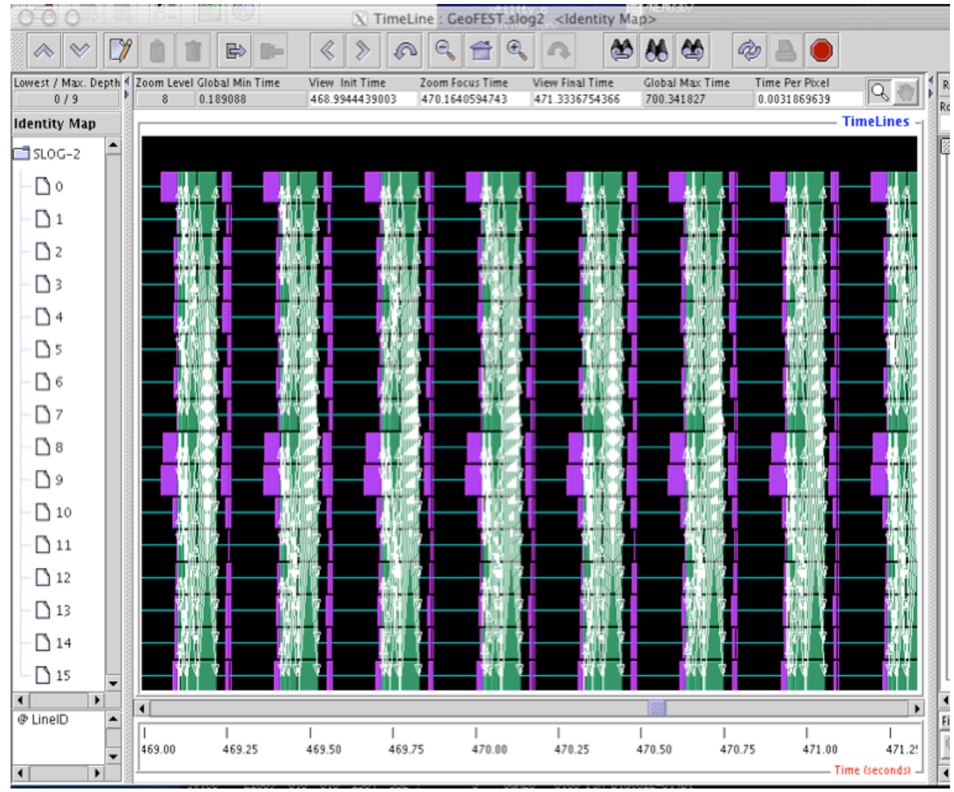


Elastic Solutions with AMR on CFM using Quakesim



Communication Performance and Optimization





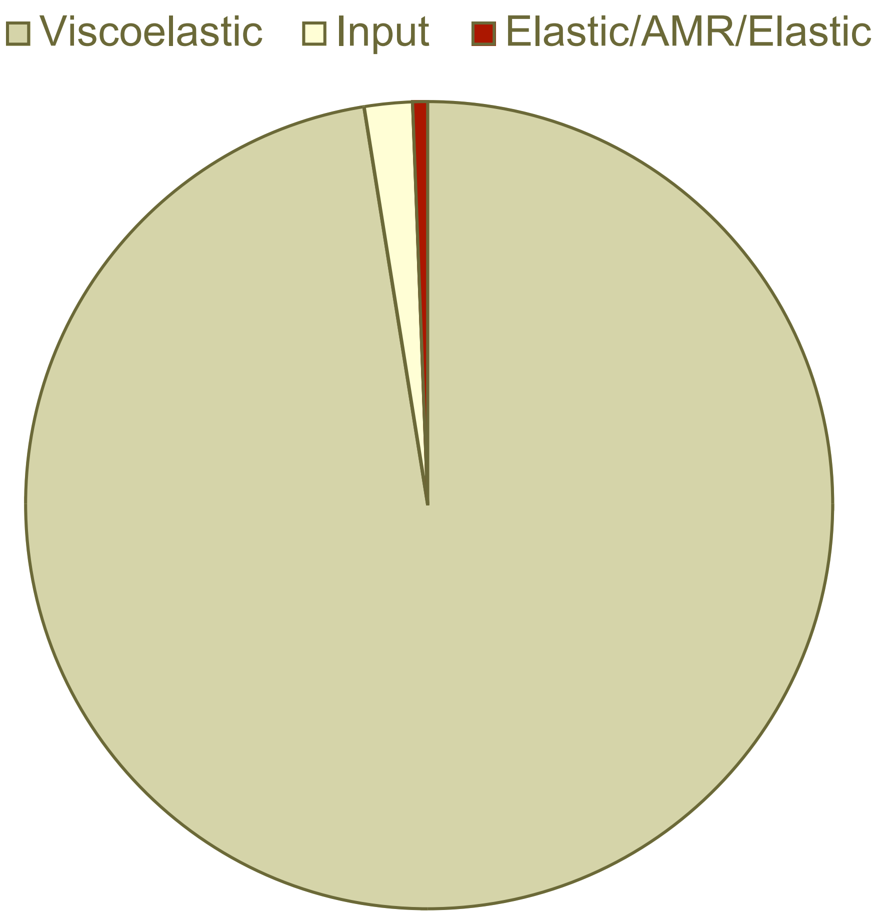
Non-Scalable Solution

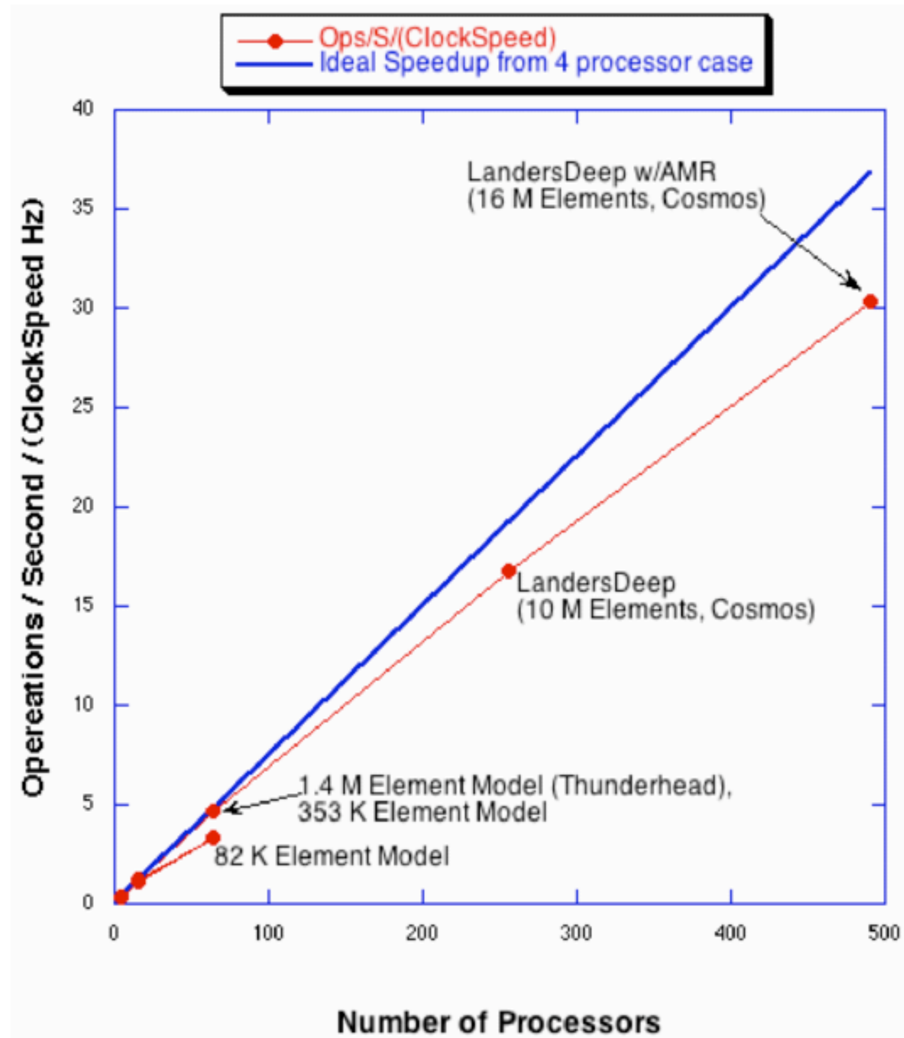
Two Iterations of PCG solve of Thousands Performed show communication efficiency for a 4 processor example. Black shows computation, Red shows WAITALL (completion of matrix-vector products) and violet ALL_REDUCE (global combine of parts of vector dot product) communication. Computation dominates giving scalability.

Improper Communication Scheme focuses on balanc communication that does not scale since more communicatio operations are applied than is strictly needed.

Performance Analysis (For Landers Case)

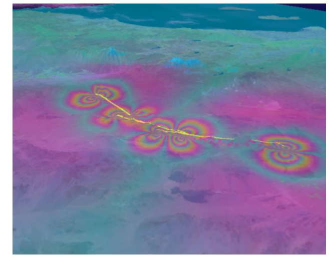
- **Parallel performance is good** (even better than illustrated here)
- 1000 viscoelastic time steps, 16M elements, and 490 processors shown on left



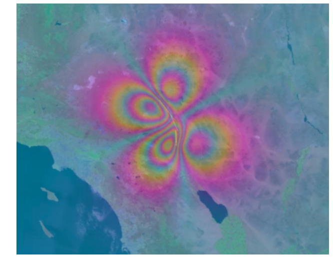


System Performance Comparison (Very Historical)

- Normalized results by processor speed for parallel CG solve iterations (dominate computational stage)

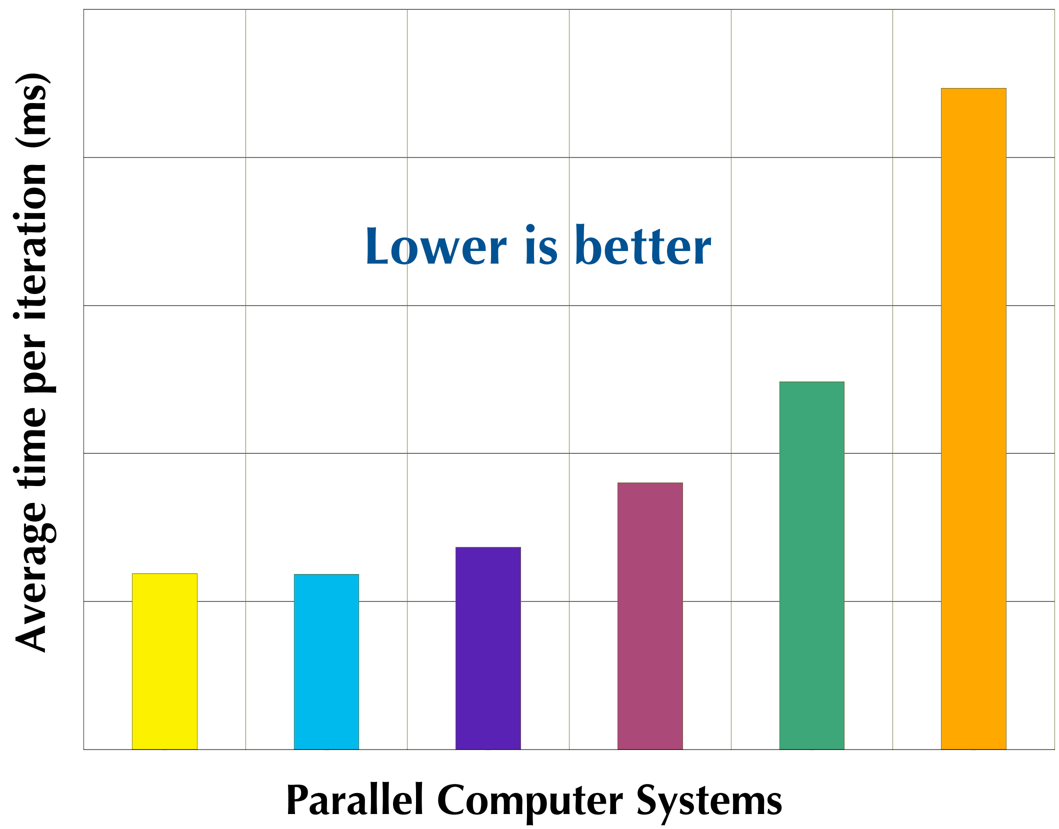


Instantaneous coseismic vertical deformation



Postseismic viscoelastic relaxation at 500 years

Landers Case



- Apple 2.0 GHz PPC G5 (Infiniband)
- Intel 3.0 GHz Pentium IV (GigE)
- Intel 2.4 GHz Pentium Xeon (Myrinet)
- SGI 1.5 GHz Altix 3000 (NUMA)
- HP 1.0 GHz Itanium 2 (QsNET)
- Apple 1.0 GHz PPC G4 (GigE)

Available On-Line via Open Channel Foundation

Not Logged In | Secure Login | New User

OPEN CHANNEL FOUNDATION

PUBLISHING SOFTWARE FROM ACADEMIC & RESEARCH INSTITUTIONS

Quick Application Search:

Search

Pyramid

Foundation :: Parallel Computing :: Pyramid

Get this title!

- Get Pyramid
- Monitor new releases

Basic information

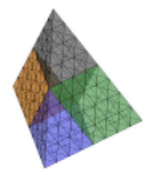
- Pyramid forum
- Contributors
- History

Pyramid [GET IT!]

Parallel Unstructured Adaptive Mesh Refinement

Total downloads from Open Channel to date: 111

✓ SOURCE CODE AVAILABLE



Moderators:
Charles D. Norton
John Lou

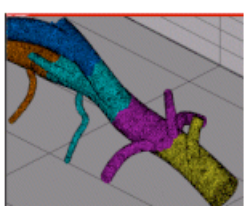
PYRAMID is a software library used for the automatic adaptive refinement of large unstructured meshes on parallel computers and clusters. This software library can be easily used in a variety of unstructured parallel computational applications, including parallel finite element, parallel finite volume, parallel visualization applications using triangular or tetrahedral meshes, and quality controlled parallel mesh generation from initial coarse element models. The library contains a suite of well designed, and efficiently implemented, modules that perform operations in a typical parallel AMR process. Among these are:

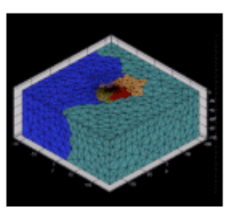
- Mesh quality control during successive parallel adaptive refinement (typically guided by a local-error estimator).
- Parallel adaptive refinement.
- Parallel mesh migration and load balancing.
- Parallel mesh partitioning using ParMetis.
- Visualization (Tecplot format).

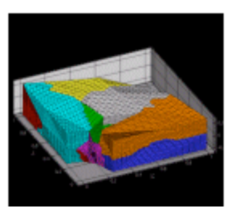
Pyramid 2.0 is Out!!
This release adds various performance improvements and more capabilities. This is the first non-demonstration version of this software. This release has been run on most systems and compilers, but the Intel 7.x/8.x compiler environment is officially supported.

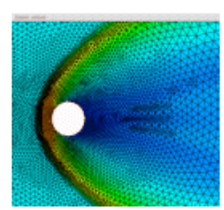
The Pyramid library is implemented in Fortran 90/95 with the Message-Passing Interface (MPI) and parallel partitioning library (ParMetis) supporting code efficiency, modularity, and portability. It is scalable to hundreds of processors and millions of elements. Visualization is supported using Tecplot.

Examples:
Click on each image to see full-scale.









PYRAMID AMR Library

User's Guide and Reference Manual

Parallel Unstructured Adaptive Mesh Refinement Library

Charles D. Norton, John Z. Lou, and E. Robert Tisdale

Official Release Version 2.0

15 April 2005

National Aeronautics and Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

Acknowledgements

- This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration
- NASA Earth Surface and Interiors Program
- NASA Earth Science and Technology Office (ESTO)
- NASA Computational Technologies Program (CT)
- NASA Applied Information Systems Research Program (AISR)
- Thanks to the CIG Workshop Organizers