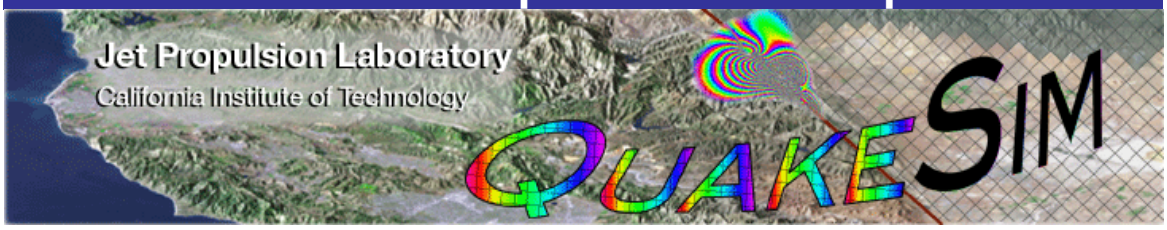
Second Code Improvement Completed



Numerical Simulations For Active Tectonic Processes: Increasing Interoperability And Performance

JPL Task Order: 10650

GeoFEST Documentation Only

Milestone G – Code Improvement due date: 6/30/2004

2nd code improvement - further optimization for some codes, pick up others that were neglected in 1st improvement - documented source code made publicly available via the Web.

- PARK on 1024 CPU machine with 400,000 elements, 50,000 time steps in 5 times the baseline code
- GeoFEST (assuming availability of 880 processor machine) 16M elements, 1000 time steps in the same time as the baseline code using the PYRAMID AMR libraries
- Virtual California with N=700 segments for 10,000 time steps in 1 hour or less, MPI parallel implementation, running on M-processor machine, with 2 GB of memory per CPU, speedup of approximately M/2 on up to 256 processors. Investigation of fast multipole method for this code.

Team

Andrea Donnellan:
Principal Investigator
Jet Propulsion Laboratory
Mail Stop 183-335
4800 Oak Grove Drive
Pasadena, CA 91109-8099
donnellan@jpl.nasa.gov
818-354-4737

Michele Judd:
Technical Task Manager
Jet Propulsion Laboratory
Mail Stop 183-335
4800 Oak Grove Drive
Pasadena, CA 91109-8099
michele.judd@jpl.nasa.gov
818-354-4994

Jay Parker:
Overall Software Engineer
Jet Propulsion Laboratory
Mail Stop 238-600
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Jay.W.Parker@jpl.nasa.gov
818-354-6790

Terry Tullis:
Fast Multipole Methods
Brown University
Box 1846, Brown University
Providence, RI 02912-1846
Terry_Tullis@Brown.edu
401-863-3829

Geoffrey Fox:
Information Architect
Community Grid Computing Laboratory
Indiana University
501 N. Morton, Suite 224
Bloomington, IN 47404-3730
gcf@indiana.edu
812-856-7977

Dennis McLeod:
Database Interoperability
Professor
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
mcleod@usc.edu
213-740-4504

John Rundle:
Pattern Recognizers
Center for Computational Science and Engineering
U. C. Davis
Davis, CA 95616
rundle@geology.ucdavis.edu
530-752-6416

Gleb Morein:
Pattern Recognizers
Center for Computational Science and Engineering
U. C. Davis
Davis, CA 95616
gleb@cse.ucdavis.edu

Greg Lyzenga:
Finite Element Models
Jet Propulsion Laboratory
Mail Stop 126-347
4800 Oak Grove Drive
Pasadena, CA 91109-8099
greg.lyzenga@jpl.nasa.gov
818-354-6920

Marlon Pierce:
Code Interoperability Software Engineer
Community Grid Computing Lab
Indiana University
501 N. Morton, Suite 224
Bloomington, IN 47404-3730
marpierc@indiana.edu
812-856-1212

Lisa Grant:
Fault Database Architect
University of California, Irvine
Environmental Analysis and Design
Irvine, CA 92697-7070
lgrant@uci.edu
949-824-5491

Robert Granat:
Pattern Recognizers
Jet Propulsion Laboratory
Mail Stop 126-347
4800 Oak Grove Drive
Pasadena, CA 91109-8099
robert.granat@jpl.nasa.gov
818-393-5353

Maggi Glasscoe:
GeoFEST Code Verification
Jet Propulsion Laboratory
Mail Stop 300-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Margaret.T.Glasscoe@jpl.nasa.gov
818-393-4834

AD HOC Team Member
Charles Norton:
PYRAMID/GeoFEST
Jet Propulsion Laboratory
Mail Stop 169-315
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Charles.Norton@jpl.nasa.gov
818-393-3920

Overview

Code	Machine Wallclock Time	Processors	Date	Elements	Time Steps
GeoFEST Milestone **E**	Solaris workstation (JPL) 13.7 Hours	1	July 30, 2002	55,369	1,000
GeoFEST Milestone **F**	*Thunderhead* (GSFC) 2.8 Hours	64	September 1, 2003	1,400,198	1,000
GeoFEST Milestone **G**	*Cosmos* (JPL) 12.0 Hours (under 13.7 allowable)	490 (well under 880 allowable)	March 26, 2005	16,021,034	1,000

Table 1: Computer runs demonstrating baseline, Milestone F performance enhancements and Milestone G performance enhancements for GeoFEST.

Milestone G Supporting Documents

The top-level web site for the QuakeSim task is at http://quakesim.jpl.nasa.gov. Source code for the GeoFEST code may be found at http://quakesim.jpl.nasa.gov/download.html. Files required for the GeoFEST baseline case (Milestone E) may be found at http://quakesim.jpl.nasa.gov/milestones.html.

GeoFEST

GeoFEST simulates stress evolution, fault slip and plastic/elastic processes in realistic materials. The products of such simulations are synthetic observable time-dependent surface deformation on scales from days to decades. Scientific applications of the code include the modeling of static and transient co- and post-seismic Earth deformation, Earth response to glacial, atmospheric and hydrological loading, and other scenarios involving the bulk deformation of geologic media.

Diverse types of synthetic observations will enable a wide range of data assimilation and inversion techniques for ferreting out subsurface structure and stress history. In the short term, such a tool allows rigorous comparisons of competing models for interseismic stress evolution, and the sequential GeoFEST system is being used for this at JPL and UC Davis. Parallel implementation is required to go from local, single-event models to regional models that cover many earthquake events and cycles.

GeoFEST uses stress-displacement finite elements to model stress and flow in a realistic model of the Earth's crust and upper mantle in a complex region such as the Los Angeles Basin. The model includes stress and strain due to the elastic response to an earthquake event in the region of the slipping fault, the time-dependent viscoelastic relaxation, and the net effects from a series of earthquakes. The physical domain may be two or three dimensional and may contain heterogeneous materials and an arbitrary network of faults. The physics models supported by the code include isotropic linear elasticity and both Newtonian and power-law viscoelasticity via implicit/explicit quasi-static time stepping. In addition to triangular, quadrilateral, tetrahedral and hexahedral continuum elements, the program supports split-node faulting, body forces and surface tractions.

The GeoFEST problem geometry for Milestone G is shown in Figure 1 below. Elastic and viscoelastic deformation is simulated for 16 million finite elements, for a duration of 1000 time steps. The significant increase is in the number of elements, which was about 55,000 for the baseline case and 1.4 million elements for the Milestone F code improvement (both of which also run 1000 time steps). Note that the baseline case was modeled on the 1994 Northridge earthquake, using a single fault of 300 square km in a domain of 240 x 240 x 100 km. The Milestone F demonstration case is based on the 1992 Landers event, using three closely arranged faults within an 865 square km area in a domain volume area of 1000 x 1000 x 60 km.] The Milestone G demonstration is still based on the 1992 Landers event – but now has higher mesh density near the faults, and 200 km depth, to reduce boundary effects for the viscoelastic relaxation.

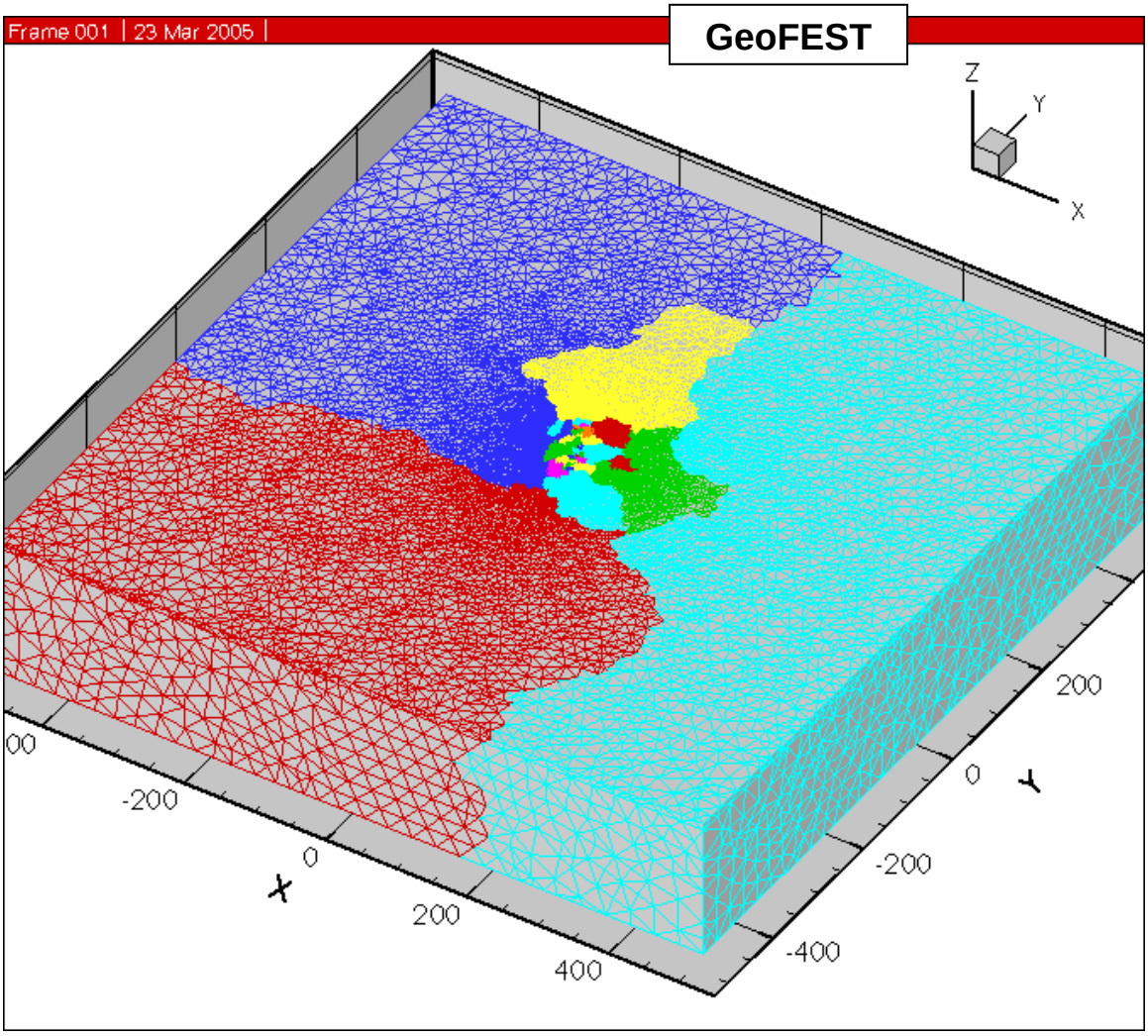


Figure 1: Finite element mesh LandersDeep for GeoFEST milestone G code improvement problem. Colors indicate partitioning among processors (limited to 16 processors in this image for clarity, although 490 processors were actually used). Partitions cluster near domain center due to the high mesh density that is used near the faults.

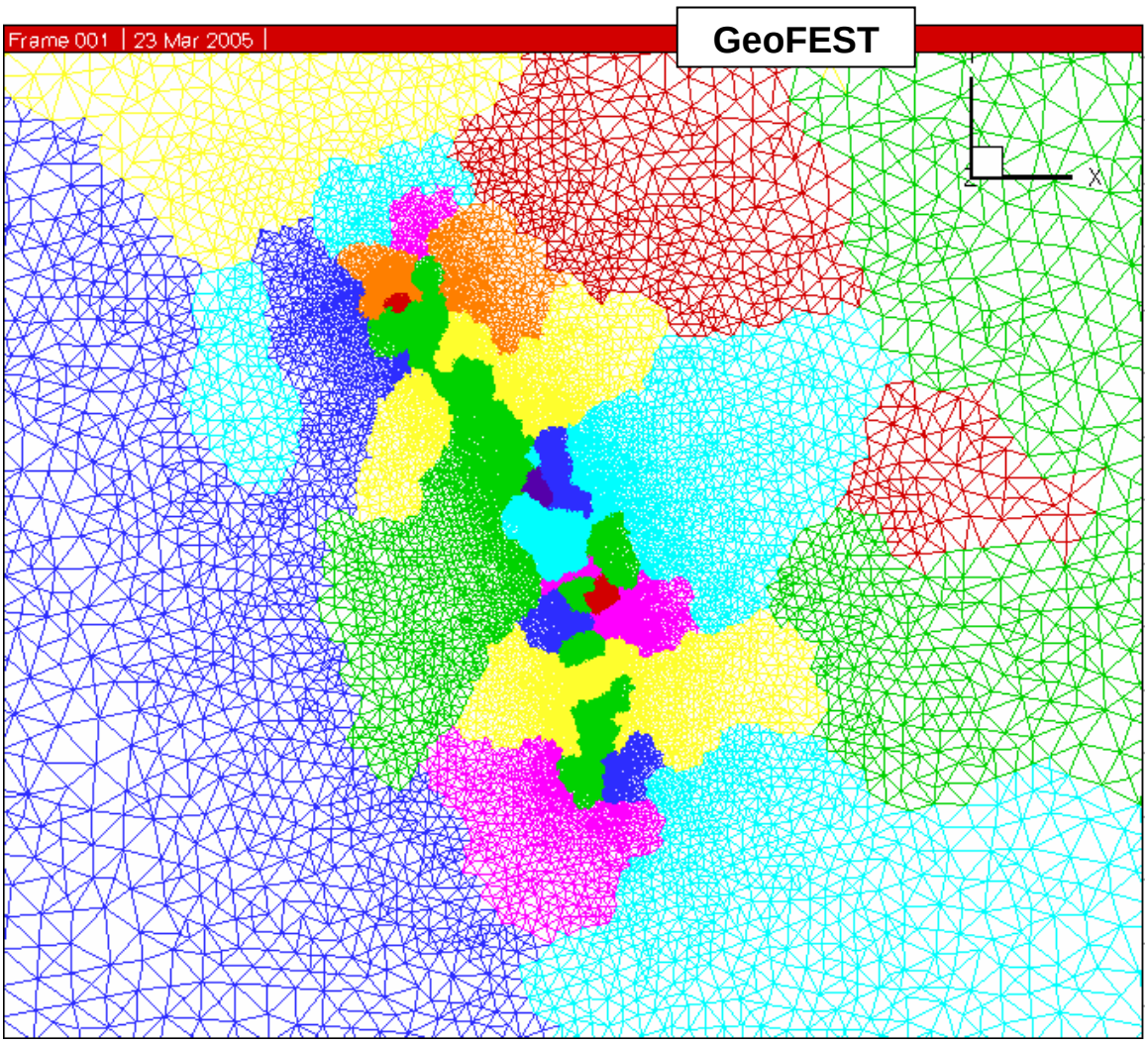


Figure 2: Top view (X-Y surface plane) of LandersDeep mesh in Figure 1 (above). This is the central portion of the mesh that has been zoomed in to provide detail on the region of active, slipping faults. Since the number of elements per partition is roughly equal, smaller regions represent areas of highest element density where strain energy is largest near the faults. Every partition shown in the image, independent of color, is uniquely owned by a specific processor. For example, each of the four red partitions belongs to distinct processors.

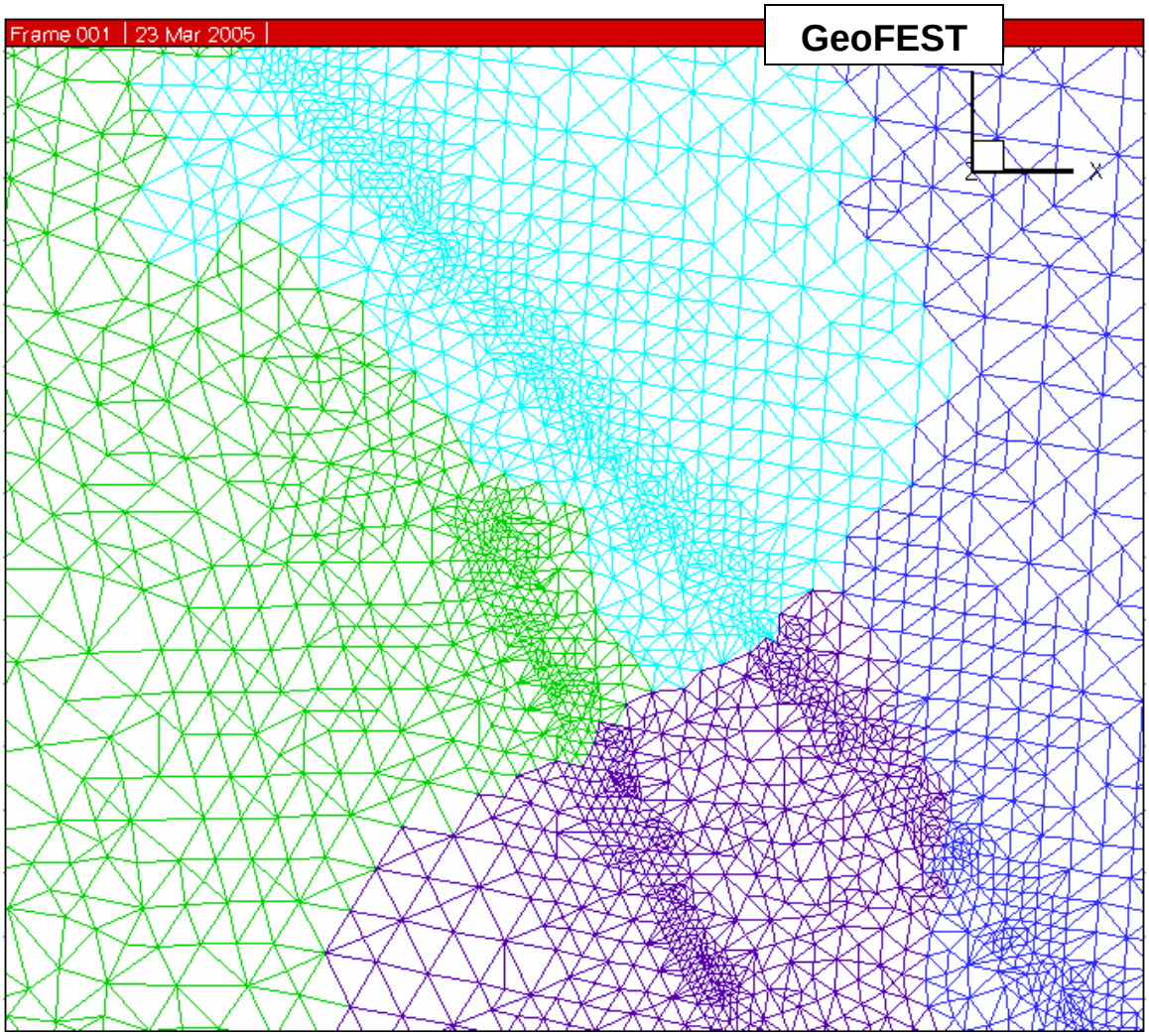


Figure 3: Detail of top view (X-Y surface plane) of the 16 million element LandersDeep mesh created by strain energy solution-adaptive refinement using PYRAMID. The close approach of two neighboring faults is shown. The previous code improvement Milestone F used the GuiVISCO sequential mesh generator to produce a ~1.4 million element mesh. The input mesh for Milestone G, also created with that tool, was limited to nearly 10 million elements. The mesh shown above required use of the parallel adaptive approach.

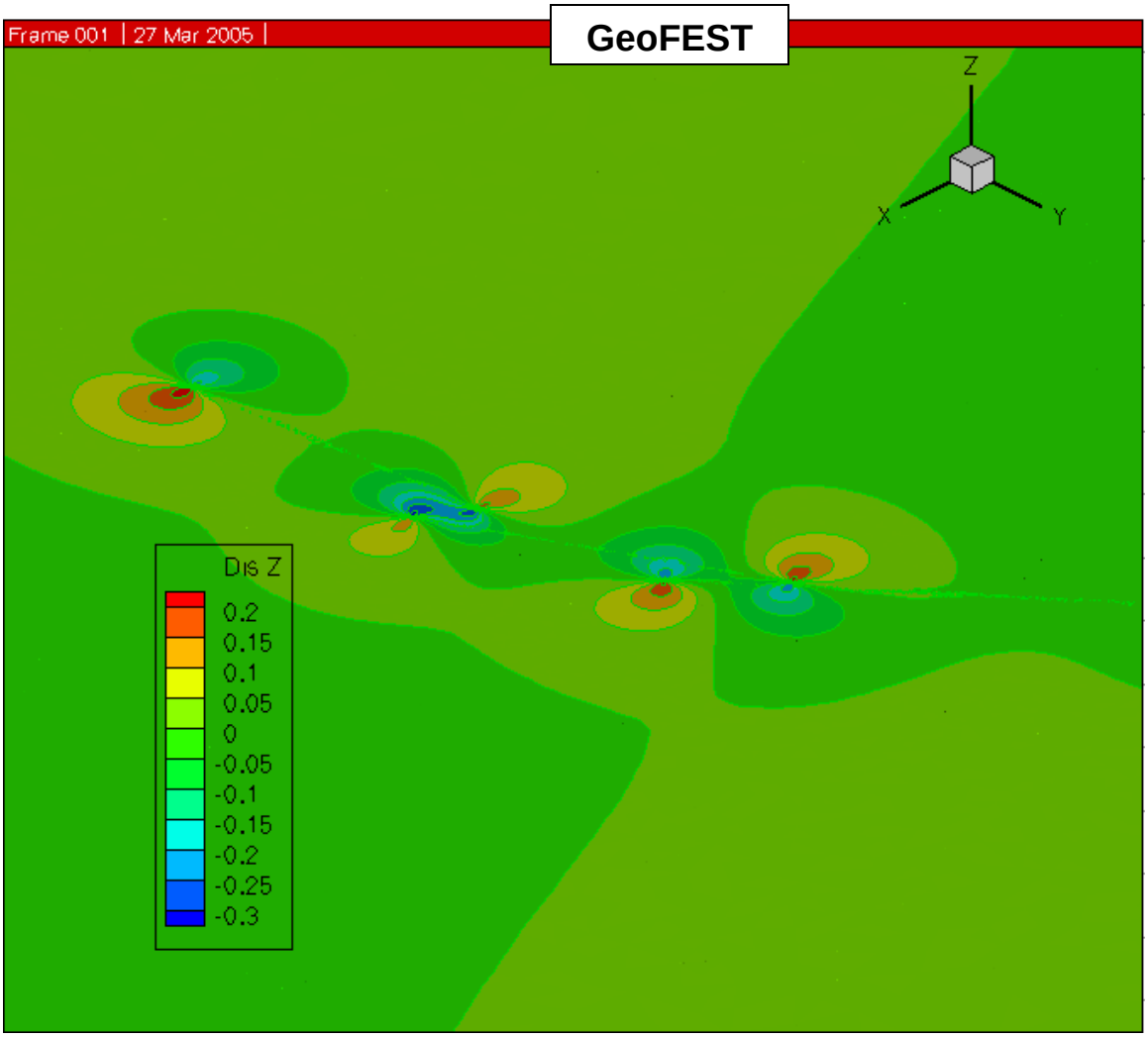


Figure 4: Top view of vertical displacement of elastic response of region to Landers earthquake based on Milestone G simulation using LandersDeep mesh.

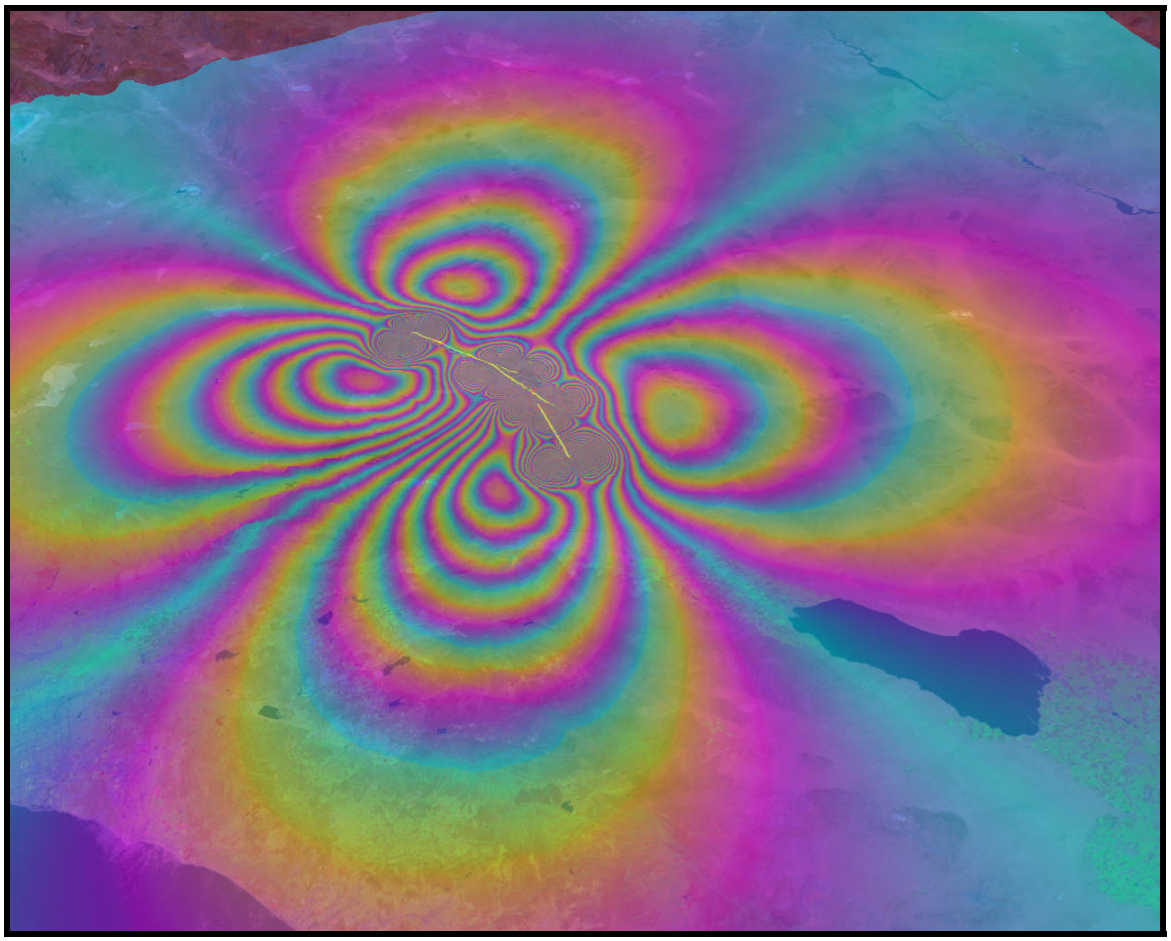


Figure 5: GeoFEST simulated surface displacement from coseismic Landers model, displayed as InSAR fringes.

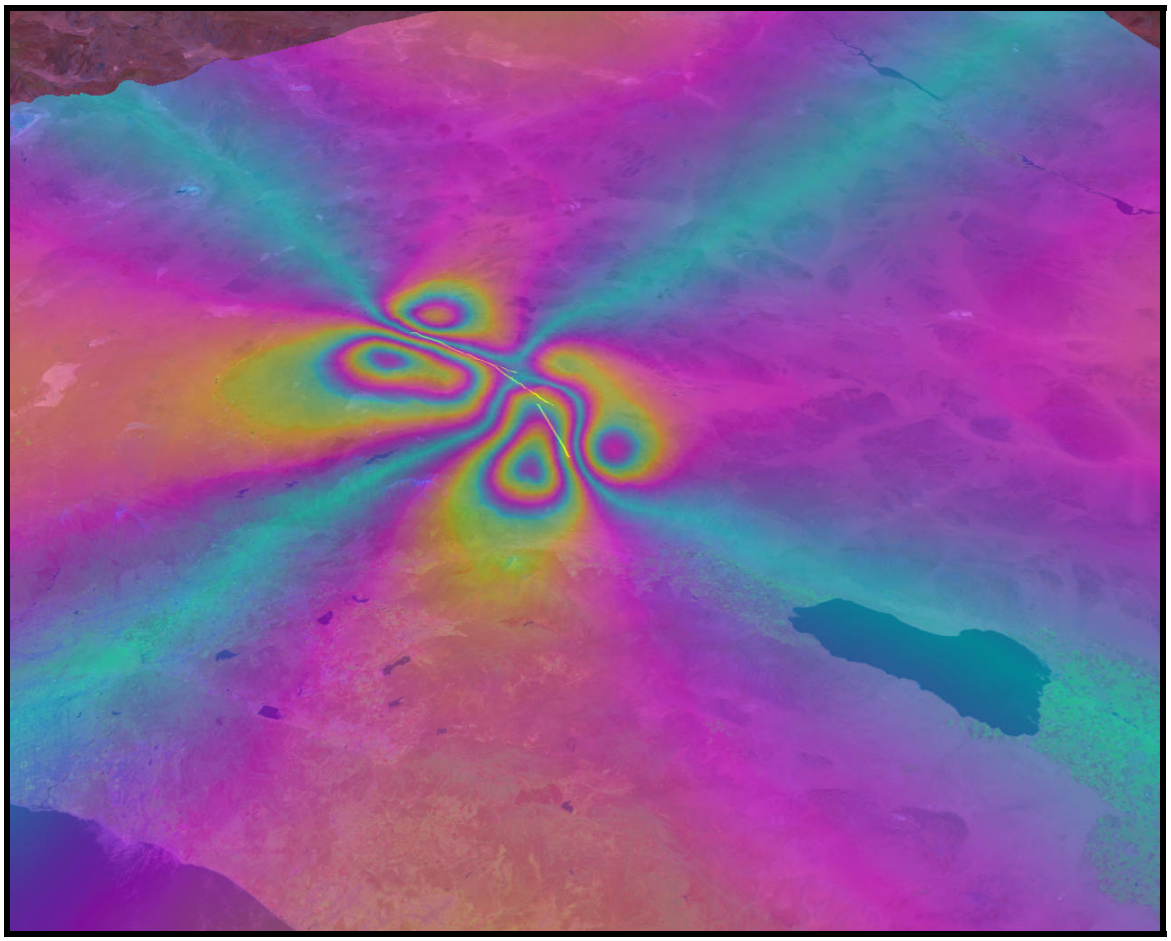


Figure 6: GeoFEST simulated postseismic surface displacement from Landers model after 500 years of viscoelastic relaxation (at the end of the GeoFEST Milestone G case of Table 1). Color scale of InSAR fringes is that of Figure 5.

GeoFEST Algorithm

GeoFEST reads in a tetrahedral mesh and information on boundary conditions, faulting events and variations in time. The equilibrium conditions are computed based on solution of the elastostatic equations through the finite element method. For the Milestone G computation, a metric of strain energy per element is used to indicate where the mesh needs to be refined. PYRAMID produces a refined mesh (partitioned among the processors), which becomes the basis for a new equilibrium elastic solution and the subsequent simulation. Then viscoelastic evolution of the stress field is computed based on an implicit technique applied on a series of time steps. User-indicated parts of the displacement and stress fields are collected to a single ordered file at requested time steps.

GeoFEST Numerical Methods

Details are found in the GeoFEST User's Guide, which is posted at:

http://quakesim.jpl.nasa.gov/GeoFEST_User_Guide.pdf.

In brief, the elastostatic solution is computed using standard elastostatic finite elements, as found in (e.g.) Hughes (2000). The sparse positive definite system of equations is solved by the Diagonally Preconditioned Conjugate Gradient (DPCG) method.

For the viscoelastic time steps we form a modified stiffness matrix and right-hand side terms according to the method of Hughes and Taylor (1978), which again results in a positive definite system. Each time-step solution is found by the same DPCG function.

Slip on faults is accommodated by a split node technique (Melosh and Raefsky, 1981) that modifies the right-hand side of the matrix system at nodes local to the fault that slips.

GeoFEST Parallel Implementation

We have linked GeoFEST to the PYRAMID library

(http://www.openchannelsoftware.org/projects/PYRAMID) and rely on PYRAMID functions for parallel mesh refinement, as well as parallel domain partitioning and communication between nodes. The initial mesh (constructed according to the User's Guide) and a derived connectivity file (produced from the initial mesh by the application gfmeshparse, available with the GeoFEST software) are read by GeoFEST and mesh structures are passed to PYRAMID functions. PYRAMID is used for partition information, resulting in each processor having a compact segment of the finite element domain for its formation of the local part of the

stiffness matrix and solution. PYRAMID is also used for combining these local solution estimates and conjugate gradient vectors at each iteration within of each time step using the "globalize" function that combines the local information and ensures each processor has valid data. When used with mesh refinement enabled, GeoFEST uses a strain energy per element threshold to mark the elements requiring refinement. All node-and element-based mesh properties (including split nodes) are written to PYRAMID attributes for preservation through the refinement process. PYRAMID refines the mesh and passes the mapped attributes back to GeoFEST, which updates its structures to represent the problem on the new mesh. The new mesh is used to solve again the problem at the current time step, and is the basis for the subsequent simulation steps. When the output displacement and stress fields are required, a final merging within PYRAMID allows GeoFEST to write a single result file with every node and element represented correctly and uniquely.

Performance is dominated by the elastic and viscoelastic iterative matrix solution. The scaling of this parallel iterative solution is explained in terms of component operations in the Milestone F report. PYRAMID refinement does not significantly affect scaling when done once, early in the simulation, as it is in the Milestone G computation. Overall scaling is shown in Figure 7 (the GeoFEST scaling plot).

GeoFEST Documentation

This section will describe how to replicate the Milestone G case using mesh refinement, GeoFEST version 4.5G. Version 4.5 is also available and useful for validation and development purposes, but will not be discussed here.

Three files should be downloaded:

- the "GeoFEST User's Guide" (a pdf file, GeoFEST_User_Guide.pdf)
- the "GeoFEST 4.5(g) Addendum" (a pdf file, GF45Gdescription.pdf)
- the "GeoFest 4.5g" release (a compressed tar file, geofest_4_5g.tgz).

The GeoFEST User Guide and addendum can be downloaded from
http://quakesim.jpl.nasa.gov/GeoFEST_User_Guide.pdf *and*
http://quakesim.jpl.nasa.gov/GF45Gdescription.pdf

The GeoFEST code may be downloaded from:
http://www.openchannelsoftware.org/projects/GeoFEST
(follow the "GET IT!" link).

The User's Guide (with Addendum) covers the following. The Introduction describes the use of the finite element method for stress and deformation simulation for models of earthquake faults that include many of the complexities of crust and mantle materials. The Features section describes the kinds of physics and boundaries currently supported in GeoFEST. The Theory of Operation section covers the mathematical and computational basis of the GeoFEST simulations, including the mathematics of viscoelastic mechanics, the finite element formulation, the implicit time-stepping scheme, the split-node implementation for faults, and the basis of parallel computation. The Input/Output section describes the formats and meanings of the parts of the relevant files. The section titled Running GeoFEST includes compilation details and parallel execution. There is an annotated sample 2D input file (this corresponds to the "GeoFEST example" interactive link from the GeoFEST Open Channel page, mentioned above).

Finally there are two appendices. The first provides flow charts describing the basic organization of the GeoFEST code at the source level, including how GeoFEST links PYRAMID calls for parallel operation. The second describes all GeoFEST functional routines, organized by source file. The addendum contains additional descriptions and instructions relevant to the special 4.5(g) release, which is only supported for parallel computing application (unlike the general purpose 4.5 release intended for the wider community of users.

The compressed file “geofest_4_5g.tgz” may be unpacked using "tar xzf geofest_4_5g.tgz " (on UNIX systems) or the equivalent. This creates a directory **GeoFEST-4.5g** which includes the subdirectories **geofest**, **PYRAMID**, and a text guide "README." Follow the directions in “README” to complete the download and configure the GeoFEST compilation for your machine. One should note that additional libraries are necessary (PYRAMID and ParMetis, both of which are freely distributed at the links listed in “README”), and a soft link must be made to configure GeoFEST with PYRAMID. (The entire process should take just a few minutes). Note that several example files named “Makefile.*” are given in the **geofest** directory to support various systems and compilers. These may be used as examples for systems not yet supported (usually new compilers or parallel systems are a matter of choosing appropriate compiler flags and incorporating these into the make system).

The **PYRAMID** directory is empty, and is provided for creating the necessary soft links for the PYRAMID library (see "README" in the **geofest** directory).

Directions for running the Milestone G test case are below.

GeoFEST Scaling Analysis

GeoFEST processing for the benchmark case naturally divides into three phases:

- Input, including reading the mesh file and creating the mesh partition data for each processor;
- Elastic solution and mesh refinement, where the initial equilibrium state is computed;
- and Viscoelastic evolution, where a sequence of time steps follow the relaxation of shear stresses in viscous material.

Figure 6 shows where time is spent for the Milestone G case (490 processors, 16 million elements, 1,000 time steps)[2]. The preponderance of time is taken up by the viscoelastic time step portion of the code. It is clear that unless many more processors are used (or many fewer time steps of simulation), the scaling of the viscoelastic steps determines the time one may expect for a simulation to complete.

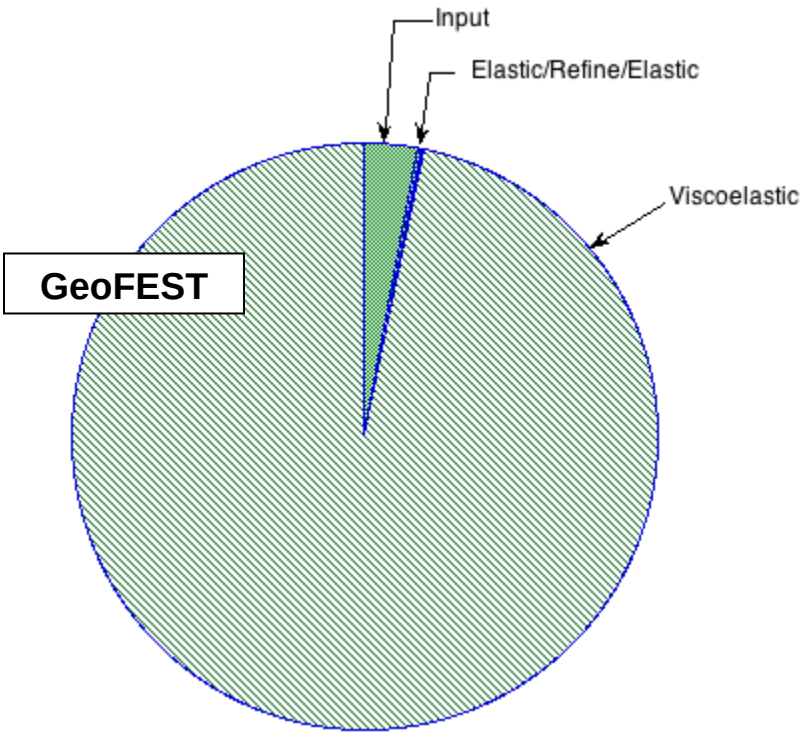


Figure 6: The Milestone G GeoFEST run times for the model input, first elastic step after the earthquake, and the 1000 viscoelastic steps. Input is for the 10 M element LandersDeep mesh. Elastic/Refine/Elastic represents the time for an initial elastic solution, solution-based refinement, and the elastic solution on the 16 M element refined mesh.

[2] For 8 of the 1,000 time steps, a snap-shot of full displacement data was captured to a file (as was done in the baseline case for Milestone E).

Code	Input Time (seconds)	Elastic Time (seconds)	Viscoelastic Time (seconds)
GeoFEST Milestone **F**	186.2	3.5	7,247.6
GeoFEST Milestone **G**	1,273.8	169.7 Elastic/Refine /Elastic	41,838.0

Table 2. The Milestone G case has proportionally larger input time. But this confirms our original design concept: if we didn't have adaptive mesh refinement, we'd eventually be dominated by input time, because that's sequential. We could have started with a mesh like the 1.4 million element milestone F mesh, and reduced our input time to about the same (186 s); that would increase our Elastic/Refine/Elastic time (to reach 16 M elements), but probably not by more than a few hundred seconds. Such tradeoffs will be evaluated in future work.

A single Viscoelastic time step consists of construction of a stiffness matrix (and right-hand side), followed by solution of this matrix by the parallel conjugate gradient technique (many iterative steps). So the Viscoelastic segment of Figure 6 represents 1,000 time steps, each consisting of roughly 200 iterations of the sparse solver. One iterative step consists chiefly of a sparse distributed matrix vector multiply and three distributed vector dot products (plus some scalar-vector operations that may be neglected).

We show in Table 3 the time for completing the time step portion for several short (5 time step) runs of varying size, on *Thunderhead*. Problem sizes were constructed to approximate a constant number of elements in each processor for 4, 16, and 64 processors. Figure 7 shows the speed of solution for these cases: "work" is taken to be the operations involved in the iterative steps. This plot proves communication overhead is not growing faster than computational time as we scale the problem size with number of processors. The result is excellent scaling. Problems with > 4,000 elements per processor see negligible parallel

overhead, and even smaller problems show substantial speed up with many processors.

Data for Scaling Runs for GeoFEST				
Number Processors	**Number Elements**	**Viscoelastic Ops**	**Wallclock (s)**	**Operations/ wallclock (s)**
4	82384	2.03E+12	2821.2	7.20E+08
16	82384	2.03E+12	738.9	2.75E+09
64	82384	2.03E+12	252.3	8.05E+09
16	3.53E+05	1.38E+13	4912.7	2.82E+09
64	3.53E+05	1.38E+13	1229.7	1.12E+10
64	1.40E+06	8.11E+13	7247.6	1.12E+10
256	1.05E+07	1.91E+15	46863.0	4.03E+10
490	1.60E+07	4.06E+15	41838.0	9.41E+10

Table 3: Table 3 shows the problem sizes and run times of GeoFEST (for the viscoelastic simulation phase) on *Thunderhead* using different numbers of processors. In order to make this scaling study, the milestone problem was reduced to 5 time steps (rather than 1,000)

Figure 7 demonstrates that the GeoFEST scaling (on the dominating time-stepping loop) is excellent. Sufficiently large problems (such as the 16M element case) have trivial parallel overhead on 490 processors.

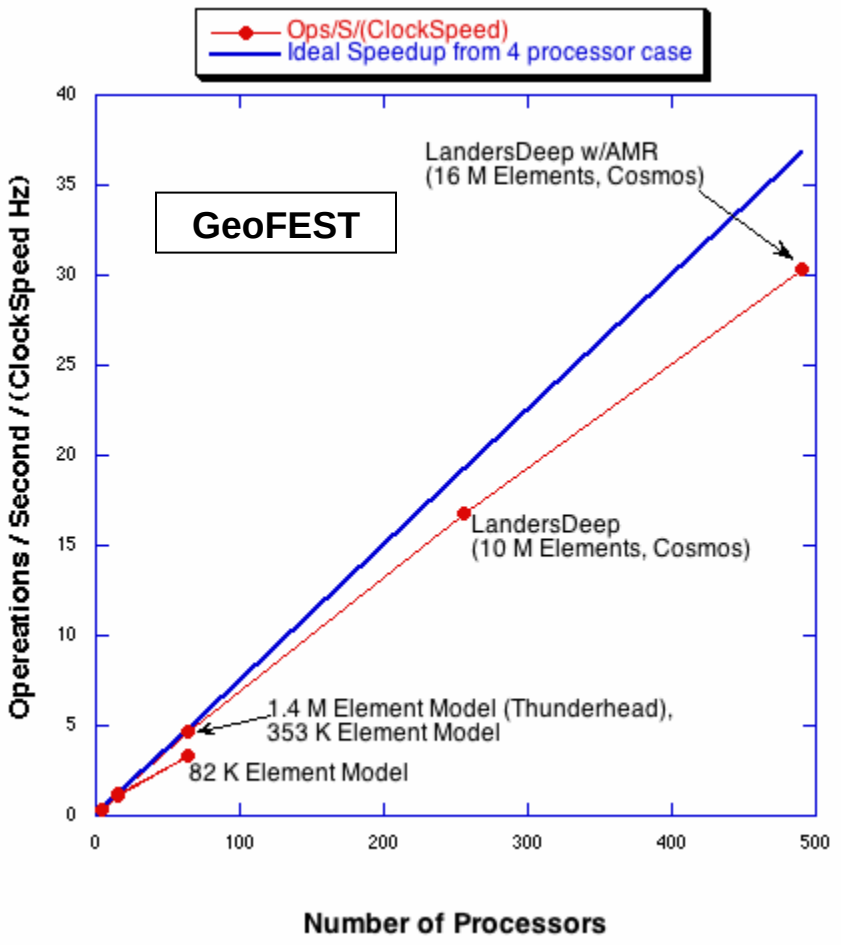


Figure 7: GeoFEST: Scaling of work (on linear scales) in GeoFEST time-step function with number of processors on three sizes of problems (on *Thunderhead* cluster computer, GSFC) plus two very large problems (on *Cosmos* cluster computer, NASA JPL). Blue indicates ideal scaling (from 4 processors). Expressing work in operations/wallclock time scaled by processor speed allows comparison of sizes (and platforms) in a single plot.

***GeoFEST* Scientific and Computational Significance**

GeoFEST has been demonstrated using adaptive mesh refinement based on an element-wise strain energy metric, which concentrates solution resources in the parts of the domain where high resolution is required. Because approximate solutions determine the final mesh structure, this allows higher accuracy solutions for larger domain sizes and greater faulting complexity for any available computer resource. This is a substantial and necessary step in allowing simulations of surface deformation in active tectonic areas, in that it allows accurate solutions on regional scales (such as Southern California), well beyond the scales of single fault segments used in past studies.

GeoFEST Simulation details

This section has two parts. The first describes how to run the milestone case on a unix/linux-based system with the Intel compilers and MPI based on the LSF queuing system. The second part contains details that prove that the milestone case was run successfully on such a system at JPL, the Intel Pentium Xeon-based cluster called *Cosmos*.

Part I

Building the code on Cosmos is accomplished by following these steps:

- Download GeoFEST and PYRAMID from Open Channel Foundation
  - http://openchannelfoundation.org/
- Download ParMetis from University of Minnesota
  - http://www-users.cs.umn.edu/~karypis/metis/parmetis/index.html/
- "tar xvzf GeoFEST.tgz" in your home directory ~/
- Load the Intel 7.x compiler and Myrinet modules
  - module purge
  - module load latest_intel71
  - module load mpich-gm-intel71
- Install ParMetis in ~/GeoFEST/Pyramid/Pyramid/ParMetis/
- cd ~/GeoFEST/Pyramid/ and type "make -f Makefile.Intel"
- cd ~/GeoFEST/geofest/ and type "make -f Makefile.Parallel"

This results in an executable program called GeoFEST in ~/GeoFEST/geofest.

Follow these steps to run the code on *Cosmos*.

- Setup input (input.dat and input.dat.jpl for milestone meshes)
  - ln –s /pathtofile/LandersDeep.dat input.dat
  - ln –s /pathtofile/LandersDeep.dat.jpl input.dat.jpl
- If these files are not present on Cosmos, they may be retrieved from http://www.servogrid.org/slide/GEM/SERVO/GeoFESTBenchmarks/ and expanded with the unzip utility.
- Submit an LSF batch job

```
#!/bin/sh
#BSUB -x -n 490 -R "span[ptile=1]" -W 13:00
```

```
#BSUB -a mpich_gm
#BSUB -q long
#BSUB -J GeoFEST
#BSUB -o /tmp/QSMG.outfile -e /tmp/QSMG.errfile
date
# NOTE: LSF starts in the current working directory by default.
cd ~/GeoFEST/geofest
mpirun.lsf ./GeoFEST input.dat ~/
```

Note: -x –R "span[ptile=1]" means use one CPU per node

-n means number of processors requested

-W hh:mm means wall clock time requested in hours:minutes

Follow these steps to interpret the results.

- When the code runs, some welcome and diagnostic messages will be printed to the screen.
- Visualization and the conjugate gradient history files will be written in ~/
- For scalability timings, capture ("grep -i pyramid QSMG.outfile") the output that looks like this (the bracketed items are explanatory and the data will be on the screen and also in QSMG.outfile if you chose to save it as suggested above):

```
PYRAMID Current Time:  xxxx.xx [A. Loading Data Start]
PYRAMID Current Time:  xxxx.xx [B. Elastic Soln Start]
PYRAMID Current Time:  xxxx.xx [C. Time Stepping Start]
PYRAMID Current Time:  xxxx.xx [D. Program Termination]
```

The total runtime will be in step D.

Cosmos is described at:

http://sc.jpl.nasa.gov/hardware/dell1750. Particulars relevant to this run are:

Cosmos is a 1024-processor Cluster located at the NASA/Jet Propulsion Laboratory. It features 512 dual-CPU 3.2Ghz Pentium 4 Xeons, 2TB memory, 48TB disk space and a 2+2Gpbs (i.e. bi-directional) myrinet fiber interconnection network.

Part II

The following are excerpts from the run script that demonstrate that the milestone was successfully met.

Transcript from LSF log file:

Sender: LSF System <lsfadm@n030>

Job <#!/bin/sh;#BSUB -x -n 490 -R "span[ptile=1]" -W 24:00;#BSUB -a mpich_gm;#BSUB -q test;#BSUB -o /scratch00/nortonc/Milestone/QSM.outfile -e /scratch00/nortonc/Milestone/QSM.errfile # my default stdout, stderr files; date;# NOTE: LS> was submitted from host <cosmos> by user <nortonc>.

Started at Fri Mar 25 18:06:19 2005

Results reported at Sat Mar 26 06:08:31 2005

Your job looked like:

LSBATCH: User input

#!/bin/sh

#BSUB -x -n 490 -R "span[ptile=1]" -W 24:00

#BSUB -a mpich_gm

#BSUB -q test

#BSUB -o /scratch00/nortonc/Milestone/QSM.outfile -e /scratch00/nortonc/Milestone/QSM.errfile # my default stdout, stderr files

date

NOTE: LSF starts in the current working directory by default.

cd /home/nortonc/GeoFEST-4.5/geofest-MG

mpirun.lsf ./GeoFEST input.dat /scratch00/nortonc/Milestone

Successfully completed.

Resource usage summary:

CPU time :21053728.00 sec.

Max Memory : 257131 MB

Max Swap : 361327 MB

Max Processes : 490

Max Threads : 490

The output (if any) follows:

Fri Mar 25 18:06:20 PST 2005

PYRAMID Unstructured AMR Library

DEMO Version 1.15, 3/25/2005 at 18 hrs 6 min 44 sec

Initialized: 490 processors using double precision reals

****

Welcome to the Geophysical Finite Element Simulation Tool (GeoFEST 4.5) (./GeoFEST)

CPU time = 0.520000

Loading Data At: PYRAMID Current Time: 1.58786773681641D-004

Opening output file /scratch00/nortonc/Milestone/LandersDeep.out ...

landers geometry, two layers, three faults, deeper box

500 years, 8 print times

Number of nodes = 1902576

No surface tractions to process . . .

Split nodes all processed . . .

Generation of element group #1 complete.

Doing shape functions...

Element generation complete. Setting up linear algebra...

Equation setup completed.

Time data generation complete...

Starting Elastic Soln At: PYRAMID Current Time: 1273.76971983910

Starting elastic solution...

Initializing globalize comm buffers:

Calling SOLVER

Elastic solution complete.

Starting Save Attributes.

save_attributes complete.

Starting Adaptive Refinement.

Adaptive mesh refinement complete.

Starting Update Phase.

Number of nodes = 2840427

Global b.c. map generated...

Decomposed node and equation totals:

Global number of elements = 16021034

Split nodes all processed . . .

Generation of element group #1 complete.

Starting Cleanup...

Starting Final Free...

Starting elastic solution...

Initializing globalize comm buffers:

Calling SOLVER

Elastic solution complete.

CPU time = 585.400000

Starting Time_Step Soln At: PYRAMID Current Time: 1443.54365301132

Starting time-stepping algorithm...

Performing scheduled output at time = 2.000000 ...

Output step finished.

Performing scheduled output at time = 10.000000 ...

Output step finished.

Performing scheduled output at time = 50.000000 ...

Output step finished.

Performing scheduled output at time = 100.000000 ...

Output step finished.

Performing scheduled output at time = 200.000000 ...

Output step finished.

Performing scheduled output at time = 300.000000 ...

Output step finished.

Performing scheduled output at time = 400.000000 ...

Output step finished.

Performing scheduled output at time = 500.000000 ...

Output step finished.

Normal program termination.

Finishing Time_Step Soln At: PYRAMID Current Time: 43281.3791708946

Transcript from cghist (conjugate gradient history) file:

time=0 converged in 393 iterations norm reduced 9.88859e-08

starting=558122 , ending=0.0551904

time=0 converged in 429 iterations norm reduced 9.92227e-08

starting=710706 , ending=0.0705182

time=0.5 converged in 394 iterations norm reduced 9.87508e-08

starting=710648 , ending=0.070177

time=1 converged in 918 iterations norm reduced 9.8854e-08

starting=0.357578 , ending=3.5348e-08

time=1.5 converged in 799 iterations norm reduced 9.94214e-08

starting=0.149331 , ending=1.48467e-08

…

time=498 converged in 943 iterations norm reduced 9.94877e-08

starting=0.00420323 , ending=4.18169e-10

time=498.5 converged in 948 iterations norm reduced 9.91846e-08

starting=0.000210996 , ending=2.09276e-11

time=499 converged in 941 iterations norm reduced 9.92404e-08

starting=2.78625e-05 , ending=2.76508e-12

time=499.5 converged in 918 iterations norm reduced 9.76287e-08

starting=3.38485e-06 , ending=3.30459e-13

time=500 converged in 834 iterations norm reduced 9.84679e-08

starting=4.34575e-07 , ending=4.27916e-14

GeoFEST References

Thomas J. R. Hughes and Robert Taylor, “Unconditionally Stable Algorithms For Quasi-Static Elasto-Plastic Finite Element Analysis.” Computers & Structures, Vol. 8, pp. 169-173, 1978.

Thomas J. R. Hughes, “The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.” Dover, Publication, INC., Mineola, New York, 2000.

H. J. Melosh and Raefsky, “A Simple and Efficient Method for Introducing Faults into Finite Element Computations.” Bulletin of the Seismological Society of America, Vol. 71, No. 5, October, 1981.