LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-665515

GEOS: User Tutorials

P. Fu, R. R. Settgast, S. M. Johnson, W. D. C. Walsh, J. P. Morris, F. J. Ryerson

December 16, 2014

Disclaimer

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

GEOS: User Tutorials

Pengcheng Fu (Editor, fu4@llnl.gov)
Randolph R. Settgast, Scott M. Johnson, Stuart D.C. Walsh, Joseph P. Morris, and Frederick J. Ryerson

Lawrence Livermore National Laboratory
Computational Geosciences Group

December 17, 2014
LLNL-TR-665515

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

Acknowledgments

GEOS was developed at the Lawrence Livermore National Laboratory (LLNL) as part of a Laboratory-Directed Research and Development (LDRD) Strategic Initiative (SI) project: Creating Optimal Fracture Networks (COFN), lead by Frederick J. Ryerson and Tarabay Antoun. The original core development team included Drs. Randolph R. Settgast, Scott M. Johnson, Stuart D.C. Walsh, and Pengcheng Fu. Drs. Joshua White, Yue Hao, Bjorn Sjogreen, Chandrasekhar Annavarapu, Xiao Chen, and others have made important contributions to individual modules of GEOS.

As the development of GEOS is an active ongoing effort, the descriptions and examples provided in this document might become incompatible with the current and newer versions of GEOS. THIS DOCUMENT ALONG WITH THE EXAMPLES IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. We expect competent users to be able to resolve various problems by properly interpreting the source codes.

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

Contents

Chapter 1

Introduction

GEOS is a massively parallel, multi-physics simulation application utilizing high performance computing (HPC) to address subsurface reservoir stimulation activities with the goal of optimizing current operations and evaluating innovative stimulation methods. GEOS enables coupling of different solvers associated with the various physical processes occurring during reservoir stimulation in unique and sophisticated ways, adapted to various geologic settings, materials and stimulation methods. Developed at the Lawrence Livermore National Laboratory (LLNL) as a part of a Laboratory-Directed Research and Development (LDRD) Strategic Initiative (SI) project, GEOS represents the culmination of a multi-year ongoing code development and improvement effort that has leveraged existing code capabilities and staff expertise to design new computational geosciences software.

The overall architecture of the framework includes consistent data structures, generalized parallel communication and input/output functions, and interfaces for incorporating additional physics solvers and materials models as demanded by future applications. Along with predicting the initiation, propagation and reactivation of fractures, GEOS also generates synthetic microseismic source terms that can be used to generate motions at surface and downhole array positions. GEOS can also be linked with existing, non-intrusive uncertainty quantification schemes to constrain uncertainty in its predictions and sensitivity to the various parameters describing the reservoir and stimulation operations.

GEOS's basic infrastructure automatically or semi-automatically takes care of many computer science aspects of HPC, including parallel I/O, MPI, and parallel linear system solving. It is suitable for being used as a generic HPC numerical platform

for applications beyond the original scope of the LDRD effort.

This document is a collection of user training materials developed during ongoing collaborative research between LLNL and external collaborators. Constantly driven by various applications, GEOS's capability development often outpaces the updating of these documentations. This document means to be a useful reference rather than a complete user manual.

Chapter 2

Installation

GEOS is typically distributed in the form of source files. The execution of GEOS relies on a number of external libraries for tasks like xml file parsing, linear system solving, and output file writing. These external libraries are mostly open-source projects available in the form of source codes too. In principle, GEOS can be run on any system on which the external libraries and GEOS itself can be compiled.

This chapter uses Ubuntu 14.04 as an example platform to describe the procedure of compiling and installing the external libraries and GEOS, particularly the former. This chapter was originally prepared for users who are not familiar with Linux systems. Users familiar with Unix-like systems, especially the use of Makefile, will find the compilation process to be fairly straightforward. You are welcome to make changes to the procedure as along as you know what you are doing. If your Linux/Unix version is different from what we use as examples or you use compilers of different versions, you might have to make some minor modifications to the Makefiles.

2.1 Installing packages available through the package manager

Some libraries can be directly installed through Ubuntu's (or other Linux distributions') package manager whereas the user needs to compile and install some others manually.

The procedure is as follows:

1. Using Ubuntu's Software Center to install another package manager called `Synaptic`, which we will use to install some of the libraries.

2. Install the following packages using `Synaptic`: `gfortran (4.8.X)`; `mpich2`; `cmake`; `libblas3`; `liblapack3`; `libxml2`; `libxslt`; and `xmlstarlet`.

3. The names of the files installed by `blast` and `lapack` might be different from what some other packages expect. For instance, `blas` might have put the library in `/usr/lib/libblas/libblas.so.3` while `trilinos` (will be installed later) is expecting it to be at `/usr/lib/libblas/libblas.so`. If this is the case, you will need to do the following:

   ```
   cd /usr/lib/libblas
   sudo ln -s libblas.so.3 libblas.so
   cd /usr/lib/
   sudo ln -s liblapack.so.3 liblapack.so
   ```

2.2 Installing the other external libraries

We have to manually install the rest of the external libraries.

1. We will create the directory into which we will compile and install other external libraries. Here we use `/home/GEOS` as the example.

   ```
   cd /home
   sudo mkdir GEOS
   sudo chown username:username GEOS
   ```

   The last command above changes the owner and group of the new folder from 'root:root' to 'username:username'. 'username' is the user's user name on this Ubuntu system.

2. Create a sub-folder `LIB` in the `GEOS` folder. Copy the source tarballs of the libraries, such as `hdf5-1.8.5-patch1.tar.gz` along with the `Makefile` into the new folder and make it the working directory.

   You will need to inform the Makefile in `geoscore` the directory of your external libraries by having the line
   `EXTERNAL_LIBS=../LIB`
   in a proper `Make.local.defs` file.

3. Figure out your default `gcc` version by typing:

   ```
   gcc --version
   ```

   We have had some trouble in using gcc 4.9.x to compile the libraries while gcc 4.8.x seems to work fine. If the default version is 4.9.X, check if `gcc-4.8` exists. Just type `gcc-4.8`. If it complains about no input file, then it is there. If it does not exist, install it through `Synaptic`.

4. Note that the header part of `Makefile` in the `LIB` folder includes a file at:
   `-include ../geoscore/Make.local.defs`.
   where `geoscore` is main folder of the GEOS source. You might need make some changes to this `Make.local.defs` file depending on the default version of `gcc` in your system. If the default `gcc` version is 4.8.X, you can leave the following arguments as they are:

   ```
   SERIAL_CPP=g++
   SERIAL_CC=gcc
   SERIAL_F90=gfortran
   SERIAL_F77=gfortran
   ```

   If the default version is 4.9.X and version 4.8.X exists, change these arguments to:

   ```
   SERIAL_CPP=g++-4.8
   SERIAL_CC=gcc-4.8
   SERIAL_F90=gfortran-4.8
   SERIAL_F77=gfortran-4.8
   ```

   which explicitly instructs the `Make` utility to use the correct version. You can also adjust `PARALLEL_SIZE` in the `Makefile` according to the number of processors available to Ubuntu. We also tell `trilinos` the locations of `blas` and `lapack` in the following two lines in the `Makefile`,

   ```
   -D TPL_BLAS_LIBRARIES:STRING=/usr/lib/libblas/libblas.so \
   -D TPL_LAPACK_LIBRARIES:STRING=/usr/lib/liblapack.so \
   ```

   Please make sure the files referred to by these two lines are indeed valid. They should have been created in the previous section if they had not been installed automatically by `blas` and `lapack`.

5. Next we install the required packages.

   ```
   make hdf5
   make silo
   sudo make trilinos
   make fp
   make dc3d
   ```

   The compilation and installation of trilinos might take more than an hour depending on the speed and number of cores of your computer. The other four should be pretty fast. If the compiling of trilinos fails, before you try again, run `make trilinosclean` to remove all the temporary files generated by the failed "make".

If any of the `make` command goes wrong, you can append "`&> logfile`" to the command line such like `sudo make trilinos &> trilinos.log`. This will write the on-screen log into a text file which you can share with other people for diagnosis.

2.3 Compiling GEOS

The tarball(s) you have downloaded should be extracted into two folders: `geoscore` and `gmodb`. The `src` directory in `geoscore` contains source files related the core functionality of GEOS, which should be relatively mature. The `src_external` directory in `gmodb` contains materials models, additional physical solvers, and other utilities that expand GEOS's functionality. Both are often under continued development by ongoing research efforts.

1. We need to create a link of the `gmodb` source files in `geoscore` by:

   ```
   cd geoscore
   ln -s ../gmodb/src_external/ src_external
   ```

2. Make a copy of file `Make.local.defs.usertemplate` and rename it to `Make.local.defs.username`. Both this file and `Make.local.defs` are included in the `geoscore/Makefile`. Try to understand the content of these three files. In most cases, you will only need to modify `Make.local.defs.username`.

3. There are a few bash scripts in `geoscore` such as `make_icc.bash` and `make_gcc.bash`, and you can run one of them to compile GEOS depending on what compiler is available to you.

2.4 Running GEOS

To run GEOS in serial, do the following:

```
cd folder/that/contains/the/xml
Path/to/GEOSexe/GEOS.ubuntu -i input.xml
```

Also note that to run GEOS in parallel, you will need to use

```
mpirun -n num_procs path/GEOS.ubuntu -i input.xml
```

On Livermore Computing machines, you should use the `srun` command.

Chapter 3

Running GEOS on Livermore Computing machines

Prepared in September 2013
Updated in January 2014 for running job on Vulcan and shared locations of executables and documentations

3.1 Userful links

- Livermore Computing (LC)'s high performance computing portal
  `https://computing.llnl.gov/`
- An introduction to LC file system
  `https://computing.llnl.gov?set=resources&page=index#file_management`
- Useful training tutorials provided by LC
  `https://computing.llnl.gov?set=training&page=index#training_materials`
- VisIt home (downloads; manuals)
  `https://wci.llnl.gov/codes/visit/`

3.2 Connecting to LC machines

Use secure shell to connect to LC machines:

```
cab670@user: ssh -Y username@MachineName.llnl.gov
```

The -Y switch is to tell the system where to forward the x-window. When connecting from an LLNL IP, the `.llnl.gov` part can be omitted. `MachineName` is cab, vulcan, sierra, etc. We have observed that sometimes Vulcan appears to be unresponsive when we directly connect it. A workaround is to connect to another machine, say Cab, and then connect to Vulcan through Cab:

```
cab670@user:ssh -Y vulcan
```

3.3 Open a remote file browser

Type command:

```
cab670@user: nautilus --browser &
```

The ampersand (&) at the end of the line tells the system that this command runs in background, so that the command line is still available for the user to type more commands.

3.4 Directly go to your home directory

On Unix-like operating systems (including AIX, BSD, GNU/Linux and Mac OS X), tilde (/) often indicates the current user's home directory. Type

```
cab670@user: cd ~
```

you can directly go to your home directory. If you have an executable in your home directory, say "GEOS_EXE", you can directly execute it by typing

```
cab670@user: ~/GEOS_EXE
```

regardless where your current directory is.

3.5 Where is GEOS

GEOS executables are located in a shared folder

```
/usr/gapps/GEOS/external_users/.
```

GEOS.cab will run on Cab and Sierra, while GEOS.vulcan only runs on Vulcan. To expedite future use, it is strongly recommended that you create a symbolic link (similar to shortcut in Windows) to these executables in a convenient location (say in our home directory as shown in the example below) using the following commands:

```
cd ~
ln -s /usr/gapps/GEOS/external_users/GEOS.cab .
```

The dot “;” in the end of the command line refers to the current directory, which in this case is your home directory. After creating the link, you can invoke the executable from anywhere by simply typing `~/GEOS.cab` . In the rest of this document, we refer to the GEOS executable as GEOS_EXE. User documentations are placed in

```
/usr/gapps/GEOS/external_users/user_docs
```

Examples are placed in

```
usr/gapps/GEOS/external_users/user_examples
```

/ These folders and files are regularly updated along with code development work. Please always refer to the newest documents and examples. Users cannot write to these folders, so you need to copy the examples to your own folder before running them.

3.6 Running GEOS interactively

Most LLNL machines (except for Vulcan) allow small serial runs on login nodes:

```
GEOS_EXE -i input.xml
```

GEOS supports specifying an external mesh file using the -m switch, but we recommend specifying the mesh file in the xml file. The intended use of login nodes is for file editing, job submission, and other light-duty work. Try to be a good neighbor and don't abuse LC resources. The login nodes of Vulcan do not run jobs. You will have to use the `srun` command to submit jobs, such as

```
srun -n 1 -p pdebug GEOS_EXE -i input.xml
```

even if it is a serial job. Parallel run on debug node:

```
srun -n num_processes -p pdebug GEOS_EXE -i input.xml
```

where "pdebug" is the debug partition name. Most machines have a debug partition called "pdebug" and a batch partition called "pbatch". See the LC resource page for the size of the partitions on each machine
`https://computing.llnl.gov/?set=resources&page=OCF_resources`

The number of processes to use must be consistent with the partitioning scheme specified in the xml file. Detailed description of the `srun` command is available at
`https://computing.llnl.gov/linux/slurm/srun.html`

You can also request a Moab `xterm` from the batch partition to run parallel jobs.

```
mxterm 8 128 120 -q pbatch -A frnet
mxterm 8 128 120 -q psmall -A gasnet
```

will request 8 computing nodes from the batch partition with 128 processors (cores; 16 cores per node on Vulcan and Cab) for 120 minutes. The computing time will charged to bank frnet/gasnet. See
`https://computing.llnl.gov/tutorials/bgq/samples/mxterm.txt`
for detailed description of the mxterm utility. Once an mxterm window pops up, the user can run GEOS parallel job as

```
srun -n num_processes GEOS_EXE -i input.xml
```

`num_processes` cannot be more than the number of cores that has been requested.

3.7 Running GEOS in batch mode

LC uses a workload manager called Moab to manage batch jobs. A detailed tutorial of Moab is available at
`https://computing.llnl.gov/tutorials/moab/`

Typically, you will create a plain txt file containing your job control script and submit this script using command msub. An example script is shown below

```
#!/bin/csh
#MSUB -A gasnet
#MSUB -l nodes=64
#MSUB -l partition=vulcan
#MSUB -l walltime=16:00:00
#MSUB -q pbatch
#MSUB -m be
#MSUB -V
#MSUB -o /some/path/logFileNameToStoreStandardOutput

##### These are shell commands
date
cd /path/to/your/input/file
srun -n 1024 ~/GEOS_EXE -i input.xml
echo 'Done'
```

Note that all the paths given must be valid. Neither Moab nor GEOS will create directories for you even if the specified directories do not exist. To submit this script, type

```
msub myscript
```

The following commands are useful for monitoring and managing submitted jobs.

```
showq [-u username]
squeue [-u username]
checkjob jobID
canceljob jobID
scancel jobID
```

These commands are explained at

`https://computing.llnl.gov/tutorials/moab/#CommandSummary`

`mshare` or `mshare -u username` can be used to check bank usage.

Apart from the `pdebug` and `pbatch` partitions existing on most machines, Vulcan has an additional partition called "psmall" for small batch jobs. Jobs running on less than 1024 nodes (16k cores/processors) should be run on "psmall", and "pbatch" only accept larger jobs. Use the following command to see the actual job limits for each machine:

```
news job.lim.ThisMachineName
```

3.8 LC parallel file systems

Detailed description of the LC file system is available at
`https://computing.llnl.gov/?set=resources&page=index#file_management`

On the LC home page, follow the link `System Status -> CZ File Systems` on the upper left corner, you can see a table summarizing the connection status between computing machines and each file server.

Most machines have access to lscractchc, d, and e, whereas Vulcan has its own scratch system called lscrachv. Edge, the visualization machine also has access to lscrachv. All scratch drives are mounted on /p/lscrachx.

3.9 Running VisIt remotely

This method is recommend if you have very fast connections to the host machine, such as when you are at LLNL campus or the open campus.

`/usr/gapps/visit/bin/visit &`
The `-v version_number` option can be used to invoke a particular version of VisIt. The default version is the newest version, currently 2.6. If VisIt often crashes, try version 2.3.

To expedite the launch of VisIt, you can create a blank text file (say named "visit")
`cd ~ ; gedit visit`

in your home directory and copy the above command line, namely `/usr/gapps/visit/bin/visit -v 2.3` into this file. Add an executable permission to this file by `chmod +x ./visit` Then you can invoke visit from anywhere by simply typing
`~/visit &`

3.10 Running VisIt in client mode

To set it up for the first run, install VisIt on your local computer. Click `Host Profiles and Configuratio` in the "Options" menu. Pick "LLNL open network" as the computing center and "LLNL" in the lower box, and install.
When this is done, a number of LC computers will appear in the "Host" field of the "File open" dialogue box. You can launch the VisIt computation engine on any one of those that you have an account on.

3.11 Some commonly used Linux commands

`cd; cp; mv; rm; ls; ln; mkdir; grep; tar; top; kill; less; rsh`
See any Linux tutorial for detailed descriptions.
"baobab" is a useful tool with a friendly GUI for checking your home directory usage. It requires x-forwarding though.

3.12 Using Eclipse

If you use a Windows machine or a Mac connecting to LC through relatively slow network, you might find the "Remote System Explorer" perspective of Eclipse to be useful. It allows you to edit files on remote systems and run command through an ssh terminal in Eclipse. No X-forwarding is available through Eclipse.
Note that the default XML editor for xml files in Eclipse has two modes: Design and Source. You can switch between these two modes by clicking the buttons at the lower left corner of the editor window.

Chapter 4

Simulating hydraulic fracturing using GEOS's explicit coupled solver

Initially prepared by Pengcheng Fu (fu4@llnl.gov) in June 2013.
With addition from Scott John and other GEOS developers.
Subjected to continued updates.

This internal memo of the LLNL Computational Geomechanics group describes the input file for explicit hydraulic fracturing simulations, as well as result visualization in VisIt. Note that the development of GEOS is an ongoing effort and substantial changes have been made to the code since this memo was prepared. Although some content is somewhat out of date, the tutorial can still be a useful reference for the users to begin with.

4.1 xml file format and xml file validation

eXtensible Markup Language (XML) is a standardized, tagged document format that can be validated against schema definitions of the expected structure and syntax. An xml input file is typically composed of multiple "blocks". We organize the majority of this document by blocks.

If an attribute name is spelled wrong, GEOS will not report an error or warning unless the misspelling causes the absence of mandatory attribute. GEOS will simply ignore the misspelled attributes and use the default values. Most attributes have default values hard-coded. You can, however, use any XML validation tool (e.g. the xmllint executable in the GEOS `external_libs/bin` directory on Livermore Computing [LC] machines) along with the XML schema file included with the GEOS source (`geoscore/src/schema/gpac.xsd`) to validate your input file before running. This can identify and help you correct many common syntax problems and is the method used when GEOS is regression tested against stored XML input files. The XML schema file reflects the most up-to-date syntax for the GEOS input files, so deprecated syntax in this document can also be identified by referring to the schema.

To run the validator on LC machines, run the following commands. The first command store the path to the scheme file in an environment variable that is used in the shell script in the second command.

```
export XML_SCHEMA_PATH=~/SomePath/geos.xsd
bash ~/SomePath/ValidateXML.sh YouXMLFile.xml
```

Please note that because this external xml validator and the xml parser in GEOS might enforce rules differently, passing the validation is neither a sufficient nor a necessary condition for the xml file to run by GEOS.

4.2 The Solvers block

The "Solvers" block defines behavior and parameters for solvers to be used in the simulation. An example is given below.

```
<Solvers>
  <LagrangeDynamicsParallelPlateFlowExplicit
                name="solver1"
                courant="0.8"
                ppcourant="3"
                BulkModulus="20.0e6"
                dampingM="800"
                MaximumAperture="3.0e-3"
                BartonJointParameters="0.5e-3 10e6 0.1e-3"
                normalJointStiffness="5e9"
                shearJointStiffness="5e9"
```

```
                leakoffCoefficient="3e-4"
                pressurecap="6e6" />
</Solvers>
```

Currently only the `LagrangeDynamicsParallelPlateFlowExplicit` solver is available for hydraulic fracturing simulation. This solver invokes the `LarangeLargeStrain` solver to handle solid deformation and the `ParallelPlateFlowSolver` to handle fluid flow in fractures.

Some commonly used solvers:

- LagrangeExplicitDynamicsSolver The legacy explicit solid mechanics solver; not directly accessible anymore.
- LagrangeLargeStrain The new explicit solid mechanics solver.
- LagrangeSmallStrainLinearElastic Implicit solid mechanics solvers. Note that as of January 2014, this solver handles both 2D and 3D.

4.2.1 name

Name of the solver chosen by user. It will be referred to by the "SolverApplications"block.

4.2.2 courant

The Courant factor for time stepping. The time step size will be the automatically calculated critical time step size times this factor. However, if "dt" is specified in the solver application block, it will override the effects of the courant factor and the automatic time stepping.

4.2.3 ppcourant

The Courant factor for the parallel plate flow solver. The value of "courant"will be used for the flow solver if "ppcourant"is not given. Our experience shows that the solid solver is generally stable if "courant"is smaller than 1.0. The flow solver could be stable if "ppcourant"is smaller than 0.5, although sometimes a value as high as 1.0 could work.

4.2.4 BulkModulus

The bulk modulus of the fluid (K). The larger the K value, the smaller the compressibility of the fluid. The governing equations for fluid flow in fractures (equations 4–6 in NAG paper) are for incompressible fluids, so we mostly use this parameter only to make the explicit flow solver to work. The critical time step for the flow solver is inversely proportional to K, so smaller values will allow longer time steps and thus faster simulation. The rule of thumb is, if K is significantly greater (say by one order of magnitude) than the highest fluid pressure expected in the system, then its value should not significantly affect the simulation results. However, in scenarios where the compressibility of the fluid is important to the simulation results, proper bulk modulus must be given. The default value of this parameter is 2.0e9.

4.2.5 dampingM

The coefficient of the mass-proportional component of Rayleigh damping. Default value is zero. For quasi-static problems, the value of damping should be chosen in a way that a moderate variation of the value would not affect the simulation results. To result in the same or at least comparable system damping behavior, the damping coefficient is dependent on material properties and mesh resolution. A useful reference value is $(E/\rho)^{0.5}/L$, where E is the Young's modulus of the solid, ρ is the density of the solid medium, and L is a characteristic length scale that is relevant to the mode that the user intends to damp. Note that for a given damping coefficient, the frequency-dependent damping ratio is smaller for higher frequencies. If the damping ratio is too low, the response might be noisy. Since we are primarily interested in pseudo-static responses of the system, low damping might result in some false dynamic modes caused by the numerical method. If the damping ratio is too high, on the other hand, the response of the system might be retarded.

If the user is unfamiliar with the concept of "damping" or "Rayleigh damping", please consult any Structural Dynamic or Elastic Wave textbook or the following web pages:

- `http://en.wikipedia.org/wiki/Damping`
- `http://help.solidworks.com/2013/English/SolidWorks/cworks/c_Rayleigh_Damping.htm`

4.2.6 MaximumAperture

The cut-off aperture size for calculating fracture permeability. This feature is undocumented in the NAG paper. As long as the value is significantly larger than the smaller aperture sizes that control the flow rate, it should not affect the simulation results. The rationale is that the majority of the fluid pressure drop takes place in small apertures. If a segment of the fracture has significantly wider aperture than the narrow parts of the network, how wide the aperture is does not really matter. The critical time step for the flow solver is inversely proposal to the square of the widest aperture in the system, so a larger value of this parameter will significantly slow down the simulation. Default value 0.004.

4.2.7 BartonJointParameters

Three parameters for the Barton-Bandis joint model. The three values are:
w_{max}, aperture width at zero-effective stress; σ'_{ref}, a reference effective stress; and w_{ref}, aperture width at the reference effective stress.

These parameters determine the relationship between aperture width and effective stress for a closed joint. The parameters for the original Barton-Bandis model are calculated as

$b = \frac{w_{max} - w_{ref}}{\sigma'_{ref} w_{ref}}$, $a = w_{max} b$

For a given effective stress σ;, the aperture is

$w = w_{max} - \frac{a\sigma'}{1 + b\sigma'}$

In the NAG paper, the aperture width of a closed joint is a constant regardless of the stress state of the joint. The implementation of the Barton-Bandis model is a significant improvement.

4.2.8 normalJointStiffness, shearJointStiffness

The normal and shear stiffness of a closed joint. Default values are 1.0e10. If the physical mechanisms being simulated are not very relevant to joint stiffness, then some arbitrary values that is high enough to avoid over-penetration and low enough to not require small time steps can be used. If the values are significantly greater

than E/L, then the contact algorithm requires a small critical time step smaller than that required by the solid solver.

4.2.9 leakoffCoefficient

Coefficient for Carter's leakoff model. According to the model, the leakoff velocity
$u_L = \frac{C_L}{\sqrt{t-t_0}}$
where C_L is the leakoff coefficient (in length/$\sqrt{time}$),$tisthecurrenttime, andt_0$ is the time when a flow cell became saturated. Note that the leakoff formulation is on a "per fracture area (length in 2D)" basis. Rock matrix in each side of the fracture receives a half of the total leakoff volume. C_L could be experimentally or analytically determined, and it should be the coefficient for the expected pressure. The user can also isolate pressure out of the coefficient in an alternative formulation
$u_L = \frac{C'_L(P_f-P_i)}{\sqrt{t-t_0}}$
where P_f is the current fluid pressure in the flow cell, P_i is the far-field pore pressure. The coefficient C'_L has different meaning than C_L, but is read in from the same attribute `leakoffCoefficient`. To activate this alternative formulation use attribute
`pressureDependentLeakoff="1"`
and the far-field pore pressure is given by
`farFieldPorePressre="1.0e5"`
with a default value of zero.

Once the leakoff model is activated by setting a positive leakoff coefficient, there will be a face field named `totalLeakedVolume` in the output silo files attached to face elements representing fracture walls. Its value is the total volume, on a per unit area basis, that has leaked off from this side of the fracture wall.

4.2.10 pressurecap

The highest fluid pressure we allow in the system. This is to prevent sudden pressure spikes due to noises in the solid phase motion. This mechanism also helps mitigate the initial instability caused by constant flow rate boundary condition.

This is how this is enforced:

```
P = K_o*(rho - rho_o)/rho;
if ( P > 0.5 * pressureCap)  P = 0.5 * pressureCap + ( P - 0.5 * pressureCap) /
```

```
(K_o - 0.5 * pressureCap) * 0.5 * pressureCap;
```

The value of `pressurecap` should not be greater than a half of the fluid bulk modulus.

4.2.11 COFJoint

The coefficient of friction of a closed joint. Default value 0.5.

4.2.12 mu

Dynamic viscosity of the fluid. Default value 0.001.

4.2.13 rho_o

Reference density of the fluid. Default value 1000.

4.3 The ElementResgions Block

```
<ElementRegions>
 <ElementRegion name="RegionA" elementtype="linear">
     <LinearElasticMaterial ShearModulus="8.3333e9" Density="26500"
                BulkModulus="11.111e9"/>
     ...
 <ElementRegion name="RegionF" elementtype="linear">
     <LinearElasticMaterial ShearModulus="8.3333e9" Density="26500"
                BulkModulus="11.111e9"/>
  </ElementRegion>
</ElementRegions>
```

`elementtype` defines the numerical integration rules. For triangle and tetrahedron elements, it must be "linear"; for quads and hexes, it should be "poly". Quads and hexes also accept "uniformstrain" as the `elementtype`, which means a single integration point (reduced integration).

Conversion between elastic constants can be easily done at
`http://www.efunda.com/formulae/solid_mechanics/mat_mechanics/calc_elastic_constants.cfm`

4.4 The SolverApplications Block

The following example is self-explanatory.

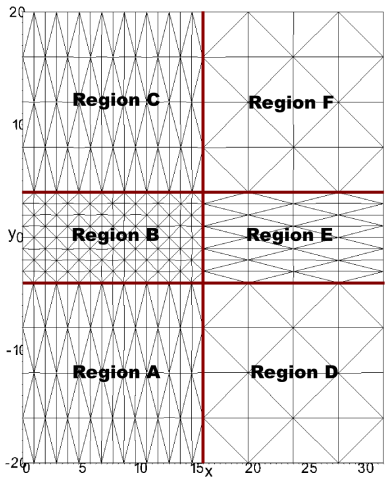
```
<SolverApplications>
  <Application name="app1" begintime="0.0" endtime="1000.0">
    <Apply solver="solver1" toregions="RegionA RegionB RegionC
  RegionD RegionE RegionF"/>
  </Application>
</SolverApplications>
```

4.5 The Mesh Block

The `Mesh` block provides the solid mesh.

```
<Mesh
   externalMeshFile="0"
   xcoords="0 16 32"
   ycoords="-20 -4 4 20"
   nx="16 4"
   ny="4 8 4"
   elementTypes="STRI"
   regionNames="RegionA RegionB RegionC RegionD RegionE RegionF"
   perturbationFactor = "0.01"
   perturbationSeed = "0"
   />
```

The block above should generate a mesh shown below



In the old version of GEOS mesher (before Dec. 2013), this block looked like to result in a similar mesh:

```
<Mesh
      externalMeshFile="0"
xmin="0   -4  0"
xmax="16   4  0"
nx="16"
ny="8"
nz="1"
elementType="STRI"
wExtensionMin ="0   16  0"
wExtensionMax ="16  16  0"
nExtensionLayersMin = "0  4  0"
nExtensionLayersMax = "4  4  0"
   perturbationFactor = "0.01"
   perturbationSeed = "0"
/>
```

4.5.1 coord's and n's

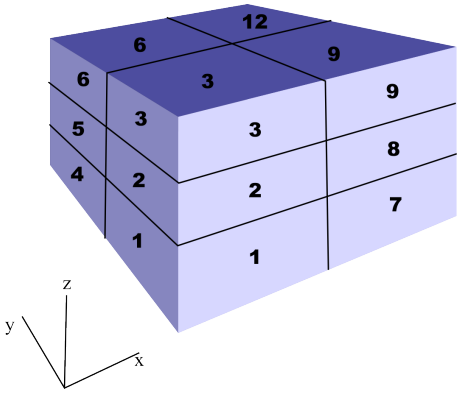
The mesh created by the internal mesh align with the Cartesian coordinate system. The boundaries of the blocks are specified by the `xcoords`, `ycoords`, and `zcoords`

attributes. `nx`, `ny`, and `nz` specify the number of elements in each block in each direction. If this is a 2D problem, all z-related attributes will be ignored.

4.5.2 regionNames

Region names for each block. The number of names is generally be the same as $nx \times ny \times nz$. Six regions are used in this example for illustrative purpose. In real simulations, it is beneficial to use a small number of regions, because using multiple regions make VisIt visualization cumbersome. If the same region name is given to multiple blocks, then these blocks belong to the same region. If only one region name is given, then all the blocks belong to the same region. The sequence of the names is as follows:
$(\mathrm{i}_1, j_1, k_1), (i_1, j_1, k_2) \ldots (i_1, j_1, k_{nz}), (i_1, j_2, k_1), (i_1, j_2, k_2) \ldots (i_{nx}, j_{ny}, k_{nz})$
$where i, j, and k are the indices running in the x-, y-, and z- directions, respectively.$

Suppose we have a $2 \times 2 \times 3$ blocks, the sequence of these blocks in the regionNames array is as blow:



4.5.3 elementType

STRI for triangle (2D); CPE4 for quads (2D); C3D4 for tetrahedrons; and C3D8 for hexahedrons. The number of element types given in this attribute in general should be consistent with the number of blocks. If all the blocks have the same element type, the user can just provide one element type, just like how `regionNames` is handled.

4.5.4 perturbationFactor

The ratio of maximum nodal position perturbation distance to element size. The amount of perturbation follows a uniform distribution.

4.5.5 perturbationSeed

The seed value for the random number generator used for nodal position perturbation. The same seed will yield the same perturbation pattern.

Note that the perturbation does not affect the geometrical boundaries of space partitions, so a node can be geometrically outside the partition that owns this node. This might generate some unintended consequences.

4.5.6 Reading external mesh

GEOS can read in mesh from an external text file in the so called "Abaqus" format, and the mesh block will be simply
`<Mesh file="mesh_file.name"/>`

GEOS understands four element geometries: tet, hex, quad, and triangle, and GEOS assumes the element type keys are C3D4, C3D8, CPE4, and STRI, respectively. The information that GEOS actually takes from an Abaqus mesh file is the nodal locations and node connectivity for each element. GEOS also reads in nodeset information. There is some other information in an Abaqus file such as material properties, but GEOS will ignore it. The Abaqus code uses the element type keys in a different way than how they are used in GEOS. For instance, CPE4 actually means plane strain quad for Abaqus, and CPS4 means plane stress quad. So if you have a mesh file with element type "CPS4", GEOS will not be able to read it. However, you can just change the key to "CPE4" and GEOS will be able to read it. Whether the model is for a plane-stress or plane-strain problem is specified in the solver block in the GEOS xml file. Since the Abaqus mesh file only provides the geometry information, CPE4 and CPS4 do not make any difference. Abaqus also has different element type keys for different integration rules. For instance, C3D8R in Abaqus means hex elements with reduced integration. Again this information is not useful for GEOS because the integration rule is defined by the `elementtype` attribute (linear/poly/uniformstrain) in the ElementRegion block. So if you have a mesh file with element C3D8R, you can just change it to C3D8 and define `elementtype="uniformstrain"`.

4.6 The Nodesets Block

This block defines "nodesets" to be used in other blocks.

```
<Nodesets>
  <Nodeset name="source"
        type="0"
        xmin="-5 -1.0 0"
        xmax=" 5  1.0 0" />
  <Nodeset name="source"
        type="0"
        xmin="-5 -81.0 0"
        xmax=" 5 -79.0 0" />
  <Nodeset name="core"
        type="0"
        xmin="-400.1 -120.1 0"
        xmax=" 400.1  120.1 0" />
</Nodesets>
```

If the same set name is used multiple times, all the applicable nodes will be appended to the same set. This feature is useful for defining complex node sets.

`type = "0"` means we will include all nodes within the box defined by `xmin` and `xmax` in the nodeset. `type = "1"` or `type = "3"` uses a cylinder defined by `point1`, `point2`, and `radius` as

```
<Nodeset name="disk_shape_prefrac"
type="1"
point1="7   0.0 3.5"
point2="7.1 0.0 3.6"
  radius="3.0" />
```

For `type = "1"`, `point1` is the center of the cylinder and `point2` is the center of one of end faces. For `type = "3"`, `point1` and `point2` are the centers of the two end faces.

Seven node sets are automatically generated by the internal mesh generator: "xneg", "xpos", "yneg", "ypos", "zneg", "zpos", and "all". Note that when a node set is defined, the corresponding element set, face set, and edge set are automatically generated.

4.7 The BoundaryConditions Block

```
<BoundaryConditions>
   <BoundaryCondition object="Face"
                      fieldname="combinedFlowRate"
                      setnames="source"
                      scale="0.006"
                      timetable="ttable" />
   <BoundaryCondition object="Face"
                      fieldname="Pressure"
                      setnames="source2"
                      scale="1.0e6"
                      timetable="ttable"/>
   <BoundaryCondition object="Node" fieldname="Velocity" setnames="
  xneg" component="0" scale="0.0" fieldtype="Vector"/>
   <BoundaryCondition object="Node" fieldname="Velocity" setnames="
  xpos" component="0" scale="0.0" fieldtype="Vector"/>
   <BoundaryCondition object="Node" fieldname="Velocity" setnames="
  yneg" component="1" scale="0.0" fieldtype="Vector"/>
   <BoundaryCondition object="Node" fieldname="Velocity" setnames="
  ypos" component="1" scale="0.0" fieldtype="Vector"/>
</BoundaryConditions>
```

The first element in the example above uses "combinedFlowRate" to apply the injection flow rate boundary condition to a face set. If the face set consists of multiple faces, the combined injection rate is the value specified.

The second element applies pressure boundary condition. The other three elements applied zero-velocity boundary condition to the boundary of the mesh. Note that for explicit solid solver (say `LagrangeLargeStrain`), the "Displacement" boundary condition does not apply. (I believe `fieldtype` is not used anymore. Need to confirm.)

4.8 The Table Block

```
<Tables>
  <Table1D name="ttable"
           coord  = "0.0, 0.01, 1.0e9"
           value = "1.0, 1.0, 1.0" />
</Tables>
```

1D tables are usually used as time-tables for specifying a time-varying quantity, such as the value for a boundary condition. 2D, 3D, and 4D tables can also be defined, but these functionalities are not covered in this document.

4.9 The Partition Block

The total number of partitions must match the corresponding augment of `srun`. See the "Cheat Sheet" for the use of `srun` and Moab scripts.

```
<Partition>
  <SpatialPartition xpar="8" ypar="4" zpar="1" />
</Partition>
```

4.10 The Fracture Block

```
<Fracture fractureFlag="100"
          preFractureSetName=""
          separableSet="core"
          failCriterion="1"
          insitu_Stress="-3e6 -1e6 0"
          insituStress3D="-1e6, -2e6, 0, 0, 0, 0"
          failstress="2e5"
          rockToughness="1e6"
          failgap="0.0004"
          verbose="0"
          maxKinkAngle = "134.0"
          kinkStrength = "-5.0e5"
          allowVacuumFrac="0"
          x1_PreFrac="  -0.12000E+02,   -0.12000E+02,   -0.12000E+02"
          y1_PreFrac="   0.00000E+00,   -0.80000E+02,   -0.40000E+02"
          z1_PreFrac="   0.00000E+00,    0.00000E+00,    0.00000E+00"
          x2_PreFrac="   0.12000E+02,    0.12000E+02,    0.12000E+02"
          y2_PreFrac="   0.00000E+00,   -0.80000E+02,   -0.40000E+02"
          z2_PreFrac="   0.00000E+00,    0.00000E+00,    0.00000E+00"
  />
```

4.10.1 fractureFlag

This is the flag for fracturing handling. If ="0", no fracture check will be performed. The non-zero value given will be the number of time steps between two adjacent fracturing criterion evaluations.

4.10.2 How preexisting fractures are defined[1]

For both 2D and 3D problems, if "preFractureSetName" is defined, all faces defined by this set will be fractured in the initialization of the problem. The user can give multiple set names separated by white spaces to this attribute. ther `separableSet` attribute does not affect the creation of pre-existing fractures. As an additional option for 2D, preexisting fractures can be defined by line segments connecting points defined by `x1_Prefrac` ... `y2_PreFrac`. The values in these arrays must be divided by commas, whereas white spaces are often used as separators for most other attributes.

4.10.3 failCriterion

If `failCriterion ="1"`, use stress intensity factor-based criterion; if ="0", use stress-based criterion; "2" means combining these two criteria. Default is "0" for 3D and "1" for 2D. "0" is recommended for hydraulic fracturing simulations. Criterion 1 only allows fracture to propagate from existing fracture tips. By using the "kink" options, i.e. `kinkStrength` and `maxKinkAngle`, it is possible to initiate new fractures by splitting nodes along fracture faces. However, criterion 1 does not allow initiate fracture inside intact continuum. Criterion 0 or 2 is needed to nucleate fractures inside intact solid bodies.
See sections "failStress" and "rockToughness" for more information.

4.10.4 separableSet

A face cannot be split if it is not within this set. If left empty, everything is separable.

[1] We have unified how preexising fractures are defined for 2D and 3D, and the "preFractureSetName" attribute works for both 2D and 3D. The "xyz_PreFrac" attribute still works for 2D.

4.10.5 insitu_Stress

This option might still work but the recommended method to initialize in situ stress is to use "Initial Conditions".

The in situ stress in the rock body. Three values are needed: stress_xx, stress_yy, and stress_xy for 2D, and stress_xx, _yy, _zz, _yz, _xz, and _xy for 3D (6 independent components). Note that the 3D stress components must be separated by commas. Signing of stress follows solid mechanics conventions: tension is positive, so the normal components should be negative for geological settings. Note if the in situ stress provided is not compatible with the boundary condition, the latter will prevail and the former will relax very quickly. It is therefore recommended to use zero-velocity boundary conditions.

4.10.6 failgap

Might be obsolete.

4.10.7 failStress

When stress-based fracturing criteria are used, i.e. `failCriterion ="0"` or `"2"`, a face with a normal stress greater than the `failStress` will be marked "fracture-ready", which enables fracture growth through this face. However, this does not necessarily create new fracture faces because certain other conditions, such as face fluid saturation and node topology compatibility, must also be satisfied.

4.10.8 rockToughness

The critical stress intensity factor. When stress intensity factor-based fracturing criteria are used, i.e. `failCriterion ="1"` or `"2"`, if a fracture tip edge's stress intensity factor (the combination of two modes) exceeds this critical value, we will mark certain faces in front of this tip edge "fracture-ready", which generally will lead to fracture propagation from this edge if other conditions, such as face fluid saturation and node topology compatibility, are also satisfied..

4.10.9 maxKinkAngle, kinkStrength

If the kink angle of a kink node is smaller than `maxKinkAngle` and there is an edge connected to this node with normal stress greater than `kinkStrength`, this kink will break. Currently only apply for 2D simulations. The default value for `maxKinkAngle` is zero and for `kinkStrength` is a large value.

4.10.10 maxTurnAngle

The greatest angle a fracture can turn from its tip. In degrees; default value is 91.

4.10.11 allowVacuumFrac

If ="0"(default), a fracture will not propagate if the flow cell next to the tip is not saturated, even when the solid mechanics criterion is met.

4.11 The Output Block

```
<Output   writePlot="1"
          plot_interval="5.0"
          restart_interval="1000.0"
          plotfile_root="FileRoot"
          slave_directory="sub"
          parallel_silo="32"
          writeFEMFaces="1"
          writeFlowText="0"
          fieldsToPlot="CommonHydroFrac ghostRank" />
```

4.11.1 parallel$_s$*ilo*

The number of the silo files to be written. If the number of partitions is big, a value of 32 or 64 is recommended for this attribute, because there are usually several dozens of I/O channels between each computational machine and the scratch file server.

4.11.2 slave$_d$*irectory*

If a slave-directory name is given, only the master silo file will be written to the working directory and all the slave files will be written to the specified sub-directory. This arrangement significantly improves the efficiency of VisIt in loading database. VERY IMPORTANT: GEOS will not create the sub-directory. The user will have to create it manually, such as through the msub script. If the specified sub-directory does not exist, GEOS will hang there and waste all the allocated CPU-hours without giving an error message.

4.11.3 writeFEMFaces

Default value is "0". Since flow fields are attached on faces, we need to write face information to silo files for analysis.

4.11.4 writeFlowText

Default value is "0". If activated, GEOS will write information about the flow cells into a single txt file at each output step. This functionality only works for 2D hydraulic fracturing models.

4.11.5 plot_interval

Specifies uniform interval for writing silo files. If non-uniform intervals are needed, a `Table1D` can be defined and use `plotIntervalTable` to refer to that table. For instance, in the Tables block, we can define:

```
<Table1D name="ttable3" coord="0.0, 800000, 804000, 820000" value="
    20000, 1000, 2000, 20000" />
```

and in the Output block, we use `plotIntervalTable="ttable3"`. Between time=0.0 and 800,000 seconds, the plot interval will be 20,000 seconds; in the next 4,000 seconds, the interval will be 1,000 second, and so on.

4.11.6 fieldsToPlot

User specified which fields (variables) to be included in the silo output files. If this attribute is left blank, most of the fields used in the code will be written to the files, resulting in large file sizes. Fields that are available for plotting are listed in the appendix. The field names should be separated using spaces. If the same field name is used by multiple objects, the plotting switch applies to all the fields. For instance, in the example above, the "ghostRanks" of nodes, faces, and elements will all be plotted.

User can also choose to plot pre-defined field sets, which are a currently hard-coded. "CommonHydroFrac" is such a set for typical hydraulic fracturing simulations and it include the following fields:

```
sigma_x
sigma_xy
sigma_y    // 2D stress components, element
Aperture     // aperture width, face
Density    // density of fluid (face) and solid (element)
flowFaceType  // =0 if this is a flow face; =-1 otherwise
flowRate
Mass
Pressure     // fluid pressure for faces and mean stress for elements
ruptureState
stressNOnFace
stressTOnFace
Volume
contactForce
hydroForce
kinkAngle
netDisplacement
SIF_I
SIF_II
Velocity
```

4.12 Tips For Visualization

4.12.1 File handling

For a big parallel job, the number of silo files written by the code could be enormous. Trying to `ls` the directory or browse the folder using a file manager could take a very long time. Using the `slave_directory` option is recommended.

4.12.2 To magnify fracture width

Apply Operators-¿Transforms-¿Displace. Give a large displacement multiplier, say 1000. Using NodalFields/netDisplacement as the displacement variable.

4.12.3 To get smooth stress plot

Use the nodal stress fields instead of the zonal (region field) stress and hide the stress in ghost zones.

4.12.4 ghostRank

The `ghostRank` field can be used to differentiate ghost elements from real elements.

If the current domain owns one element, the ghostRank of that element is negative. If the ghostRank is -1, it means that this element also exists as a ghost in a neighbor domain. If the ghostRank is zero or positive, the element is a ghost in the current partition. The value is the rank of the domain that actually owns this element.

If we use ghostRank as the "threshold" operator variable and set "-0.5" as the max value, all ghost zones will be hidden.

4.12.5 To separate flow faces from solid faces

Apply Operators-¿Selection-¿Threshold. Delete the default threshold variable. Use FaceFields/flowFaceType as the new threshold variable, and use "-0.5" as the lower bound. Note that flowFaceType=-1 for solid faces and =0 for flow faces.

You can also use the `isExternal` fields because flow faces are external. Because the exterior faces of the model are also external, you will have to use a "selection/box" operator to exclude these exterior faces.

4.12.6 Node and element ID's

The ID's are only unique in each partition (called "domain" in VisIt). For parallel simulations, the domain number must be provided if you want to pick an element or node by its ID. The domain numbers start from ONE in VisIt, although partition numbers used by GEOS start from Zero, so rank 0 partition in GEOS will be domain 1 in VisIt.

4.12.7 Displacement vs. netDisplacement

Both are NodalFields. Displacement uses the zero-stress state as the reference state; netDisplacement uses the in situ stress state (typically the initial state) as the reference state.

4.12.8 If you want to use VisIt for your own data

GEOS outputs are in the "silo" format, for which the detailed description is available at:
`https://wci.llnl.gov/codes/silo/`

VisIt can also process a great variety data formats, and a list of compatible file format is available in the "File Open" dialogue box of VisIt. More detailed description of some of these file formats is available at
`https://wci.llnl.gov/codes/visit/FAQ.html#12`

A particularly useful format for finite element postprocessing is the plain text `vtk` file, which is described in
`http://www.vtk.org/VTK/img/file-formats.pdf`

4.13 Userful links

- Livermore Computing (LC)'s high performance computing portal
  `https://computing.llnl.gov/`
- An introduction to LC file system
  `https://computing.llnl.gov?set=resources&page=index#file_management`
- Useful training tutorials provided by LC
  `https://computing.llnl.gov?set=training&page=index#training_materials`
- VisIt home (downloads; manuals)
  `https://wci.llnl.gov/codes/visit/`

4.14 Connecting to LC machines

Use secure shell to connect to LC machines:

```
cab670@user: ssh -Y username@MachineName.llnl.gov
```

The -Y switch is to tell the system where to forward the x-window. When connecting from an LLNL IP, the ".llnl.gov"part can be omitted. "MachineName"is cab, vulcan, sierra, etc. We have observed that sometimes Vulcan appears to be unresponsive when we directly connect it. A workaround is to connect to another machine, say Cab, and then connect to Vulcan through Cab:

```
cab670@user:ssh -Y vulcan
```

4.15 Open a remote file browser

Type command:

```
cab670@user: nautilus --browser &
```

The ampersand (&) at the end of the line tells the system that this command runs in background, so that the command line is still available for the user to type more commands.

4.16 Directly go to your home directory

On Unix-like operating systems (including AIX, BSD, GNU/Linux and Mac OS X), tilde (/) often indicates the current user's home directory. Type

```
cab670@user: cd ~
```

you can directly go to your home directory. If you have an executable in your home directory, say "GEOS_EXE", you can directly execute it by typing

```
cab670@user: ~/GEOS_EXE
```

regardless where your current directory is.

4.17 Where is GEOS on LC machines

GEOS executables are located in a shared folder

```
/usr/gapps/GEOS/external_users/.
```

GEOS.cab will run on Cab and Sierra, while GEOS.vulcan only runs on Vulcan. To expedite future use, it is strongly recommended that you create a symbolic link (similar to shortcut in Windows) to these executables in a convenient location (say in our home directory as shown in the example below) using the following commands:

```
cd ~
ln -s /usr/gapps/GEOS/external_users/GEOS.cab .
```

The dot "." in the end of the command line refers to the current directory, which in this case is your home directory. After creating the link, you can invoke the executable from anywhere by simply typing `~/GEOS.cab` . In the rest of this document, we refer to the GEOS executable as GEOS_EXE. User documentations are placed in

```
/usr/gapps/GEOS/external_users/user_docs
```

Examples are placed in

`usr/gapps/GEOS/external_users/user_examples`

/ These folders and files are regularly updated along with code development work. Please always refer to the newest documents and examples. Users cannot write to these folders, so you need to copy the examples to your own folder before running them.

4.18 Running GEOS interactively

Most LLNL machines (except for Vulcan) allow small serial runs on login nodes:

```
GEOS_EXE -i input.xml
```

GEOS supports specifying an external mesh file using the -m switch, but we recommend specifying the mesh file in the xml file. The intended use of login nodes is for file editing, job submission, and other light-duty work. Try to be a good neighbor and don't abuse LC resources. The login nodes of Vulcan do not run jobs. You will have to use the `srun` command to submit jobs, such as

```
srun -n 1 -p pdebug GEOS_EXE -i input.xml
```

even if it is a serial job. Parallel run on debug node:

```
srun -n num_processes -p pdebug GEOS_EXE -i input.xml
```

where "pdebug" is the debug partition name. Most machines have a debug partition called "pdebug" and a batch partition called "pbatch". See the LC resource page for the size of the partitions on each machine
`https://computing.llnl.gov/?set=resources&page=OCF_resources`

The number of processes to use must be consistent with the partitioning scheme specified in the xml file. Detailed description of the `srun` command is available at
`https://computing.llnl.gov/linux/slurm/srun.html`

You can also request a Moab `xterm` from the batch partition to run parallel jobs.

```
mxterm 8 128 120 -q pbatch -A frnet
mxterm 8 128 120 -q psmall -A gasnet
```

will request 8 computing nodes from the batch partition with 128 processors (cores; 16 cores per node on Vulcan and Cab) for 120 minutes. The computing time will charged to bank frnet/gasnet. See
`https://computing.llnl.gov/tutorials/bgq/samples/mxterm.txt`
for detailed description of the mxterm utility. Once an mxterm window pops up, the user can run GEOS parallel job as

```
srun -n num_processes GEOS_EXE -i input.xml
```

`num_processes` cannot be more than the number of cores that has been requested.

4.19 Running GEOS in batch mode

LC uses a workload manager called Moab to manage batch jobs. A detailed tutorial of Moab is available at
`https://computing.llnl.gov/tutorials/moab/`

Typically, you will create a plain txt file containing your job control script and submit this script using command msub. An example script is shown below

```
#!/bin/csh
#MSUB -A gasnet
#MSUB -l nodes=64
#MSUB -l partition=vulcan
#MSUB -l walltime=16:00:00
#MSUB -q pbatch
#MSUB -m be
#MSUB -V
#MSUB -o /some/path/logFileNameToStoreStandardOutput

##### These are shell commands
date
cd /path/to/your/input/file
srun -n 1024 ~/GEOS_EXE -i input.xml
echo 'Done'
```

Note that all the paths given must be valid. Neither Moab nor GEOS will create directories for you even if the specified directories do not exist. To submit this script, type

```
msub myscript
```

The following commands are useful for monitoring and managing submitted jobs.

```
showq [-u username]
squeue [-u username]
checkjob jobID
canceljob jobID
scancel jobID
```

These commands are explained at

`https://computing.llnl.gov/tutorials/moab/#CommandSummary`

`mshare` or `mshare -u username` can be used to check bank usage.

Apart from the "pdebug"and "pbatch"partitions existing on most machines, Vulcan has an additional partition called "psmall" for small batch jobs. Jobs running on less than 1024 nodes (16k cores/processors) should be run on "psmall", and "pbatch" only accept larger jobs. Use the following command to see the actual job limits for each machine:

```
news job.lim.ThisMachineName
```

4.20 LC parallel file systems

Detailed description of the LC file system is available at
`https://computing.llnl.gov/?set=resources&page=index#file_management`

On the LC home page, follow the link `System Status -> CZ File Systems` on the upper left corner, you can see a table summarizing the connection status between computing machines and each file server.

Most machines have access to lscractchc, d, and e, whereas Vulcan has its own scratch system called lscrachv. Edge, the visualization machine also has access to lscrachv. All scratch drives are mounted on /p/lscrachx.

4.21 Running VisIt remotely

This method is recommend if you have very fast connections to the host machine, such as when you are at LLNL campus or the open campus.

`/usr/gapps/visit/bin/visit &`
The `-v version_number` option can be used to invoke a particular version of VisIt. The default version is the newest version, currently 2.6. If VisIt often crashes, try version 2.3.

To expedite the launch of VisIt, you can create a blank text file (say named "visit")
`cd ~ ; gedit visit`
in your home directory and copy the above command line, namely `/usr/gapps/visit/bin/visit -v 2.3` into this file. Add an executable permission to this file by `chmod +x ./visit` Then you can invoke visit from anywhere by simply typing
`~/visit &`

4.22 Running VisIt in client mode

To set it up for the first run, install VisIt on your local computer. Click `Host Profiles and Configuratio` in the "Options" menu. Pick "LLNL open network" as the computing center and "LLNL" in the lower box, and install.
When this is done, a number of LC computers will appear in the "Host" field of the "File open" dialogue box. You can launch the VisIt computation engine on any one of those that you have an account on.

4.23 Some commonly used Linux commands

`cd; cp; mv; rm; ls; ln; mkdir; grep; tar; top; kill; less; rsh`
See any Linux tutorial for detailed descriptions.
"baobab" is a useful tool with a friendly GUI for checking your home directory usage. It requires x-forwarding though.

4.24 Using Eclipse

If you use a Windows machine or a Mac connecting to LC through relatively slow network, you might find the "Remote System Explorer" perspective of Eclipse to be useful. It allows you to edit files on remote systems and run command through an ssh terminal in Eclipse. No X-forwarding is available through Eclipse.
Note that the default XML editor for xml files in Eclipse has two modes: Design and Source. You can switch between these two modes by clicking the buttons at the lower left corner of the editor window.

Chapter 5

Training exercises

This chapter provides four sets of comprehensive exercises for training purposes.

5.1 Problem 1 Center-cracked infinite strip with a finite width

Practice the following skills:

1. Use different element types;
2. Understand the difference between implicit and explicit solvers;
3. Understand the role of mesh resolution on SIF and tip stress;
4. Understand damping in dynamic responses;
5. Understand parallel partitioning;
6. Basic VisIt manipulation, primarily 2D.

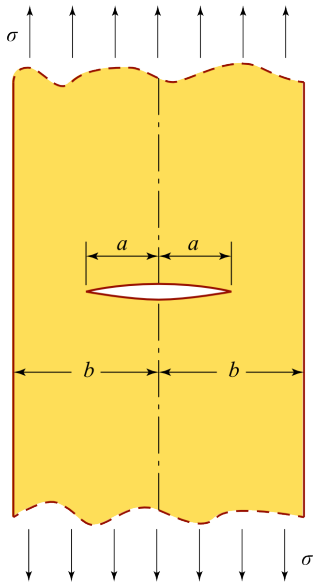
Calculate and analyze the mechanical responses (stress and stress intensity factor) for the fracture configuration shown below. You can also pick a comparable configuration with closed form solutions.

If the strip is sufficiently long, the analytical solution for the SIF at the tip for the given configuration is

$$K_I = \sigma\sqrt{\pi a}F_I(a/b) \tag{5.1}$$

where

$$F_I(a/b) = [1 - 0.025(a/b)^2 + 0.06(a/b)^4](cos\frac{\pi a}{2b})^{-0.5} \tag{5.2}$$



Exercises: Use an implicit solver for exercises 1) through 5). Remember the 2D and 3D implicit mechanical solvers have slightly different names.

1. Determine a baseline configuration (say a/b=3/4) and simulate it with different element types (2D: CPE4 and STRI; 3D: C3D8 and C3D4). Observe the difference in how pre-existing fractures are created between 2D and 3D. Compare the SIF and stress values (say, `sigma_yy` in the element next to the tip) obtained with different element types at the same mesh resolution.

2. Compare the results from the full model as shown in the figure (exercise 1) and a half model that exploits the symmetry in the problem. Tricky question: can we use a quarter model?

3. For the same baseline case, observe how the value of SIF and stress near tip change as the mesh is refined. Pay special attention to the case with extremely

low mesh resolution, for instance, when there is only one element between the tip and lateral boundary, and see how bad the SIF estimation gets under such extremely unfavorable conditions (hopefully not too bad).

4. At a fixed mesh resolution, vary the a/b ratio and see if the numerical results of SIF can capture the trend of the analytical solution.

5. Vary the partition scheme and see if you get exactly the same results. Pay particular attention to the case where partition boundaries cut through the fracture or the tip. If you simulate a large problem (high resolution, at least thousands of nodes), you should see the significant speed improvement due to parallelization.

6. In the exercises above, you should have used the implicit solver to get the static solution. You can try the explicit solver to see how a dynamic system evolves to the static solution. Tips:

   - First, calculate a characteristic time scale, i.e. the time it takes elastic wave to propagate from one end of the model to the other end. The simulation time should be much longer than this time scale while the output interval should be much shorter than this time scale. Note that the simulation time step is automatically calculated. Elastic wave speed is approximately $\sqrt{E/\rho}$, where E is the Young's modulus and ρ is the density of the solid.
   - Try different damping values and observe how damping affects the system response. The tutorial gives a reference value for damping. A useful reference value is $\sqrt{E/\rho}/L$, where L is a characteristic length scale this is relevant to the mode that the user intends to damp.
   - If you use an example from the training material as the template, you will have to make sure the fracturing criterion

7. Instead of pulling from the two ends of the model, try to apply pressure from inside of the crack, using pressure of the same magnitude as the pulling stress. Compare the SIF and stress results between the two loading scenarios. Tips:

   - See section 6.6 to see how internal pressure along a fracture can be applied.
   - The damping value you used previously that results in fast convergence to the static solution should be useful.

5.2 Problem 2 Penny-shape fracture

A penny-shape fracture in an infinite (or sufficiently large) medium. Apply tensile stress at far field normal to the disk plane. The stress intensity factor is $K_I = 2\sigma\sqrt{r/\pi}$.

You will need to use type 1 nodeset to create a penny-shape fracture. Exercises:

1. 1) Use the internal mesher and a quarter model (+x, +y, full z; Cartesian mesh) to calculate the SIF. SIF is calculated along the perimeter of the fracture plane and is an edge field. You can use the threshold operator in visit to isolate those edges along the perimeter for visualization. Notice the poor representation of a circular shape by hexes and tets.

2. According to Sneddon (1946), the aperture profile of such a penny shape fracture is

$$w(r) = \frac{8\sigma R(1-\nu^2)}{\pi E}\sqrt{1-(\frac{r}{R})^2} \tag{5.3}$$

   where r is the distance of an arbitrary point to the center; R is the radius of the fracture. See how the numerical results, at least at the center, compare with the analytical solution. See if the poor resolution of fracture perimeter shape affects the aperture width results.

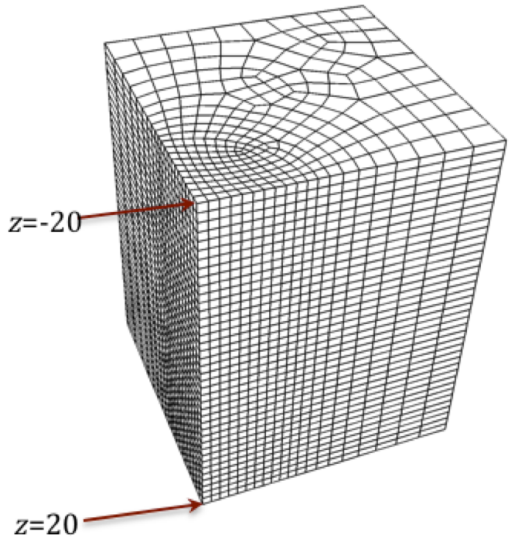
3. If you have a handy mesher, try to generate a mesh that perfectly accommodates the penny-shape and see how it improves the accuracy of SIF estimation. You can also use the mesh `QuarterDisk_Hex.inp` provided as shown in the figure below.

4. Practice various VisIt skills through the following exercises:

   - Isolate the fracture planes from the 3D model; Use a box operator to isolate the plane, and use threshold with isExternal or Pressure to differentiate fractured faces from intact faces.
   - Exaggerate the aperture width;
   - Draw the perimeter of the fracture; Tip: Plot edge mesh with thick line and isolate the tip edges using a threshold operator.
   - Cut the model with an arbitrary plane and plot stress on the obtained cross-section;

- Plot the `stress_zz` curve along the center axis. Tip: use line-out query tool.

Mesh of `QuarterDisk_Hex.inp`:



5.3 Problem 3 Penny-shape hydraulic fracture

The analytical solution for a viscosity-dominated penny shape fracture with no leak-off was provided in Perkins and Kern (1961) and Geertsma and de Klerk (1969). The aperture width at the well bore (center of the fracture) is

$$w = 2.17(\frac{\mu^2 q^3}{E'^2})^{1/9} t^{1/9} \tag{5.4}$$

and the radius of the fracture is

$$R = 0.52(\frac{E' q^3}{\mu})^{1/9} t^{4/9} \tag{5.5}$$

where E' is called the "plane strain modulus" and defined as $E/(1-\nu^2)$ with ν being the Poisson's ratio.

The baseline xml input file and the moab script for a penny shape hydraulic fracture will be provided. Please review it carefully and make sure you understand all the lines. You can change the parameters in the file, but it might be useful to discuss with the instructor the implications for computational cost. This simulation takes 288 processors about 4.5 hours to run on cab. Remember to create a sub-folder consistent with the `slave_directory` attribute in the working folder. This exercise will focus on result analysis.

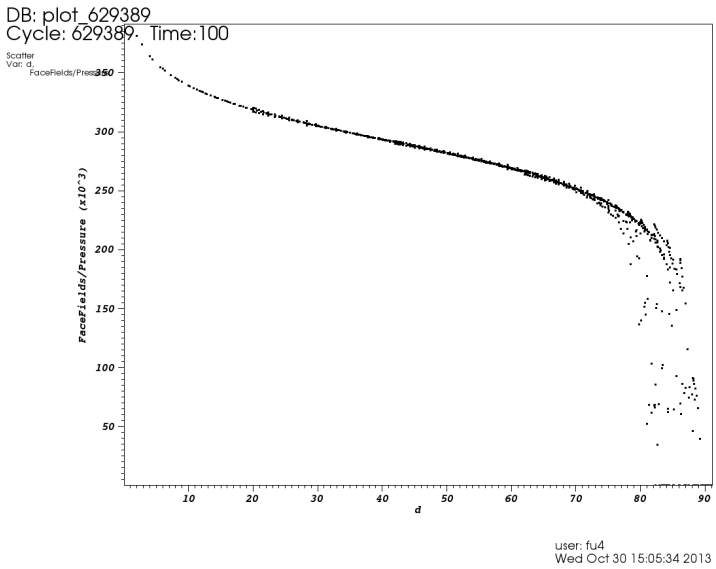
1. Plot the evolution of the fracture radius with respect to time and compare it with the analytical solution.

   - You can perform a time query of fracture surface area and convert it to equivalent radius. There is a variable in the query panel called "3D surface area" that can be used for this purpose.
   - Use a box to isolate the fracture plane and use "isExternal" threshold to differentiate fracture faces from intact ones. Remember to exclude ghost faces (those with ghostRank ¿=0). Because both sides of the fracture will be isolated and included in the query, the query result is actually twice the fracture area.
   - You can select "time" as the x-axis variable in Controls/Query over time options. The default is cycle .

2. Plot the evolution of the center aperture width with respect to time and compare it with the analytical solution.

3. At the end of the simulation (t=100) plot the pressure as a function of distance to fracture center.

   - In the `expression` panel, create a variable "d" as `sqrt(coord(face_mesh)[0]^2+coord(face_mesh)[1]^2)`.
   - Create a scatter plot with "d" being the x coordinate and "Pressure" as the y coordinate. Because this is a 2D scatter plat and 3D and 2D plots cannot coexist in the same window, you will have to hide all the 3D plots. The result should look like



- The data in this scatter plot can be saved into a text file if you save the window using the ply as the file type.

sectionProblem 4 Penny-shape hydraulic fracture that encounters a sink

Based on the above penny-shape hydraulic fracture model, place a sink at some distance from the injection point. The following describes some considerations for building this model and example parameters used for your reference.

1. Once the sink is placed, the axisymmetry of Problem 3 is broken, but the problem is still symmetric again the plane connecting the injection point and the sink. A half model will work.

2. In situ stress determines the orientation of penny shape fracture but the magnitude of the in situ stress does not affects behavior of the fracture, so we dont need to explicitly model the in situ stress. We can constrain the propagation path of the fracture in a plane by using the separableSet attribute.

3. We will use a coarse mesh to speed up the simulation. We can still check the behavior of the hydraulic fracture before it hits the sink again the analytical solution given in Problem 3, which will give us a good idea of the effects of mesh resolution.

4. The sink is modeled as a face element with a pressure-boundary condition. We can just apply this boundary to a single face element (actually the 4-node node set for this face element). We dont need to split this face as a pre-existing fracture. The pressure boundary condition there will only be activated when the fracture hits the face and create a flow element on that face. I recommend using a small positive pressure value (say 1.0 Pa) for the boundary condition, even if you want to apply a zero pressure boundary. This is because the inversion of zero pressure into fluid density through the equation of state (EOS) is ambiguous. Tricky question: if you make a node set that only includes a single face element and make it a preFractureSet, will the face be split?

5. Here are the parameters that I have tried:

   - he core domain is 360 x 180 (360 full model) x 120 and the mesh resolution is 6 x 6 x 10. Extension to each direction of x and y is 120 and the extension to each direction of z is 300 m.
   - The hydraulic fracture source is centered at the center of the domain. The distance between the sink and the source is 96.
   - The injection rate is 0.5 (1.0 for full model) and the sink pressure is 1.0e4.
   - According to the analytical solution, the hydraulic fracture hits the sink after about 50 (seconds).
   - Considering we have about 100,000 nodes, I used a 4x4x4 partition. It took cab about 10 hours to simulate 600 seconds of the process.

Expected analysis tasks:

1. Before the hydraulic fracture hits the sink, compare the numerical solution with the analytical solution.

2. Isolate the fracture faces and see how the sink affects the fracture profile in the radial direction and in the out-of-plane direction.

3. Track the source pressure evolution and surface area growth and see how the sink affects those.

4. The two following snapshots are provided for your information.

Chapter 6

Notes on example input files

This chapter provides some brief notes on a series of example input files offered as a training material. The xml input files are provided in the Appendix. Another very useful source of examples is the test cases located in directory
`geoscore/test/FullTests`

6.1 Example 1: Wave propagation in a 1D bar

Notes:

1. Elastic wave propagating along a 1D bar.

2. The solid deformation in GEOS's hydraulic fracturing simulations is handled by the same solver.

3. The meaning of Rayleigh damping. See any structural dynamics textbook. Also briefly covered in subsection 4.2.5.

4. The kinematic boundary condition for explicit solid mechanics solver should be Velocity; Displacement boundary condition is invalid.

5. For kinematic boundary conditions, the direction attribute is understood by GEOS as DOF component, not a real direction.

6. Compare fixed and free surface boundary condition on the reflection end of the bar to see how it affects the direction of the reflected velocity pulse and the sign of the reflected stress wave.

To learn:

1. Basic concept of explicit dynamic solid mechanics solver;
2. Abaqus mesh file;
3. The concept of nodeset and how they are defined in the mesh file;
4. Kinematic boundary condition;
5. 1D time table;
6. Basis VisIt operation.
7.

To explore:

1. The role of damping in dynamic response;
2. Boundary condition for dynamic problems.

6.2 Example 2: Cantilever beam bending

Notes:

1. The kinematic boundary condition for implicit solid mechanics solver should be Displacement; Velocity boundary condition is invalid.
2. No inertial effects although we still specify time in the problem and can have time varying boundary conditions.

To learn:

1. Implicit solid mechanics solver;
2. Traction boundary condition;

6.3 Example 3: Internal mesher

section 4.5 provides a relatively comprehensive overview of the internal mesher. To learn:

1. Using the internal mesher to generate mesh;
2. The concept of region.

6.4 Example 4: Notched beam, 2D

Example 4 has been expanded to four related examples. Note:

1. The direction attribute in traction boundary condition means actual direction.
2. We give rockToughness a very high value, so that the fracture wont propagate.
3. Example 4B and 4C simulate fracture propagation in the same beam using implicit and explicit solvers, respectively.
4. In example 4D, we apply tension and shear to the same specimen with a fracture in it to calculate mode-I SIF and mode-II SIF.
5. In 2D, SIF is a node field while in 2D is an edge field.

To learn:

1. Define nodeset for boundary condition application.
2. Creating a pre-existing fracture.
3. Fracture propagation.

6.5 Example 5: Notched beam, 3D

To learn:

1. Create a pre-existing fracture in 3D.
2. Try a parallel run.

6.6 Example 6: KGD problem

chapter 4 covers this example in great detail. Note that the updated (2014.12) example contains certain new features that have not been included in chapter 4 yet, but most of these new features should be self-explanatory.

6.7 Example 7: Static penny-shape fracture

This example is the solution to problem 2 in the Problem Set. Three solutions are given:

1. uses the internal mesher, which results in jagged fracture perimeter;
2. uses an externally generated mesh;
3. is based on 2) but the loading is internal pressure in the fracture instead of pulling from the far-field boundary. The internal pressure is activated by the "applyNormalTraction" switch in the TractionBoundaryCondition block.

A VisIt plot session file `StaticPlot.session` is given as an example of isolating and plotting data in 3D. Another VisIt plot session file `StressOnPlane.session` demonstrates the calculation of normal and shear stress on planes with a given orientation. See the Expression window for the formulations of the added variables.

6.8 Example 8: Penny-shape hydraulic fracture

This example is corresponding to problem 3 in the "Problem Set". The folder also contains an msub script and two VisIt plot sessions.

6.9 Example 9: Penny-shape hydraulic fracture with a sink

This example is the solution to problem 4 in the "Problem Set". The folder also contains an msub script and a VisIt plot sessions.

6.10 Example 10: 3D hydraulic fracturing - PKN

With relatively low mesh resolution to enable fast running.

6.11 Example 11: Spatially varying material property

A 3D table is given to specify the location-dependent elastic properties (shear modulus and Lames parameter) in Region C. The properties (e.g. ShearModulus) can be viewed in the restart files (which are also silo files that can be opened by VisIt), while the mechanical responses can be seen in the regular plot files.

6.12 Example 12: Bore hole mixed mesh

A 2D model of a bore hole. The mesh consists of a mix of triangle and quad elements. Apply pressure inside the borehole and fractures will be initiated. Note that this kind of mixing of element types works for solid solvers but will not work for the hydrofracture solver, because the element-to-face mapping is messed up.

6.13 Example 13: Placeholder for new method of applying in situ stress

Obsolete. The method of applying in situ stress has been covered by the updated Example 6.

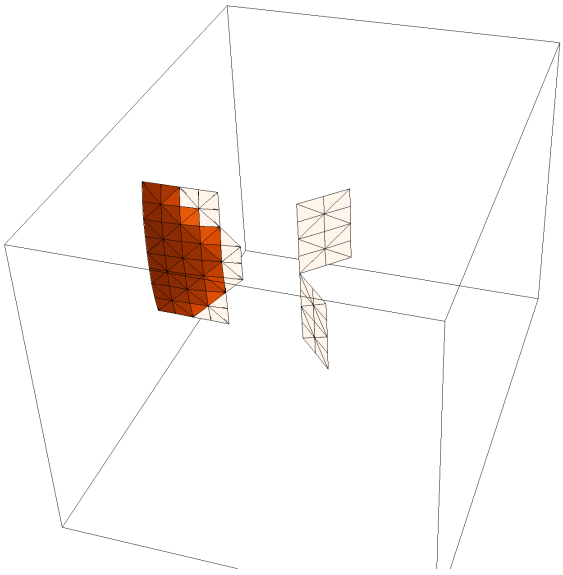
6.14 Example 14: Joint shearing

A joint (pre-existing fracture) is oriented 45 degrees from the principal stress directions. The two principal stress components are sigma$_x$ $= -1MPaandsigma_y = -3MPa.Onthisfracture, theshearstressis1MPaandthetotalnormalstressis2.0MPa.Wepumpfluidintot$

Note: Magnify the netDisplacement by 500 to 1,000 times by using Transform/Displace operator to see the shearing. Plot Face Field/contact stress as a vector field to see the evolution of the contact stress as the pressure changes.

6.15 Example 15: Bifurcating wings

A propagating hydraulic fracture propagates and encounters two small natural fractures oriented at different angles. This will cause the hydraulic fracture to bifurcate. Since no in situ stress presents, the fracture trajectory is not regulated. Small anomalies of fracture paths might be observed.



6.16 Example 16: 3D hydraulic fracturing in arbitrary stress field

Only the xml template is provided. The user needs to provide the space tables of the in situ stresses.

6.17 Example 17: Generic elastic orthotropy material model

Only the xml template is provided. Material properties are not included.

Appendix A

Example xml input files

This chapter provides some brief notes on a series of example input files offered as a training material.

A.1 Example 1: Wave propagation in a 1D bar

```
<?xml version="1.0" ?>
<!-- This is a problem that tests 1D wave propagation in a bar composed of
  2D quad elements. -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">
  <Mesh file="bar.geom" />
  <Solvers>
    <LagrangeLargeStrain name="solver1" courant="0.5"
      timeIntegrationOption="2" dampingM="10000" />
  </Solvers>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="0.04">
      <Apply solver="solver1" toregions="EB1" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="EB1" elementtype="uniformstrain">
      <LinearElasticMaterial ShearModulus="15.0e9"
        Density="2650" BulkModulus="15.0e9" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="NS1" direction="1.0 0.0 0.0" scale="0.01" timetable="ttable" />
```

```
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 0.001, 0.002, 0.003, 1.0e9"
      value="0.0,    1.0,    1.0,    0.0,  0.0" />
  </Tables>

  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>

  <Output writePlot="1" plot_interval="4e-4" restart_interval="0.02"
    plotfile_root="output" parallel_silo="1" />

</Problem>
```

A.2 Example 2: Cantilever beam bending

This example uses two files: one is referred to by the other.

A.2.1 The main file:

```
<?xml version="1.0" ?>
<!-- This is a standard benchmark problem, a straight cantilever beam. Meshes
  with different refinement levels are given. PFU. Reference R.H. Macneal and
  R.L.Harder,   A Proposed Standard Set of Problems To Test Finite Element Accuracy
    ,
  Finite Elements in Analysis and Design, 1, pp 3-20, 1985. -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Include file="LagrangeSolvers.xml" />

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="2.0" dt="1.0 s">
      <Apply solver="implag2" toregions="EB1" />
    </Application>
  </SolverApplications>

  <Contact active="0">
    <PenaltyCoulomb stressYield="100 GPa"
      normalApproachYield="4.99e-3" aperture="0.00" />
  </Contact>

  <ElementRegions>
    <ElementRegion name="EB1" elementtype="poly">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
  </ElementRegions>
```

```
  <BoundaryConditions>
    <TractionBoundaryCondition object="Node"
      setnames="NS2" direction="0.0 0.0 -1" scale="1000 Pa" timetable="ttable" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="NS1" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="NS1" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="NS1" component="2" scale="0.0" fieldtype="Vector" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 10.0" value="0.0, 1.0" />
  </Tables>

  <Mesh file="CantileverBeamHex.abq" />

  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>

  <Output writePlot="1" restart_interval="100.0" plot_interval="1.0"
    plotfile_root="plot" parallel_silo="1" />

</Problem>
```

A.2.2 The included file:

```
<Parameters>
  <Parameter name="SolidDensity" value="2650 kg/m^3" />
  <Parameter name="SolidBulkModulus" value="8.33333 MPa" />
  <Parameter name="SolidShearModulus" value="3.84615 MPa" />
  <Parameter name="deltaTime" value="0.005 s" />
  <Parameter name="endTime" value="0.01 s" />
</Parameters>

<Solvers>
  <ImplicitMechanicsSolver3D name="implag" tol="1e-10" />

  <LagrangeSmallStrainLinearElastic
    name="implag2" tol="1e-6" useMLPrecoditioner="0" />

  <LagrangeSmallStrainLinearElastic
    name="implag2d" tol="1e-16" useMLPrecoditioner="0" />

</Solvers>
```

A.3 Example 3: Internal mesher

A.3.1 A mesh with one element type

```
<?xml version="1.0" ?>
<!-- -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="gpac.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 MPa" />
    <Parameter name="SolidShearModulus" value="3.84615 MPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
  </Parameters>

  <Solvers>
    <LagrangeSmallStrainLinearElastic
      name="implag2d" tol="1e-10" useMLPreconditioner="0" />
  </Solvers>

  <Mesh xcoords="-30 -10 10 30" ycoords="-5 0 5" nx="8 16 8" ny="4 4 "
    elementType="CPE4" regionNames="RegionA RegionB RegionB RegionA RegionA RegionB"
     />

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="2.0" dt="1.0 s">
      <Apply solver="implag2d" toregions="RegionA RegionB" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="RegionA" elementtype="poly">
      <LinearElasticMaterial ShearModulus="1e7"
        Density="2650" BulkModulus="1e7" />
    </ElementRegion>
    <ElementRegion name="RegionB" elementtype="poly">
      <LinearElasticMaterial ShearModulus="2e7"
        Density="2650" BulkModulus="2e7" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <TractionBoundaryCondition object="Node"
      setnames="xpos" direction="1.0 0 0.0" scale="3.0 MPa" timetable="ttable" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="xneg" component="1" scale="0.0" fieldtype="Vector" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 2.0" value="0.0, 1.0" />
  </Tables>

  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>
```

```
  <Output writePlot="1" restart_interval="1.0" plot_interval="1.0"
    plotfile_root="plot" parallel_silo="1" />

</Problem>
```

A.3.2 A mesh with mixed element types

```
<?xml version="1.0" ?>
<!-- A strip load on a half space. Use two different element types -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="gpac.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 MPa" />
    <Parameter name="SolidShearModulus" value="3.84615 MPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
  </Parameters>

  <Solvers>
    <LagrangeSmallStrainLinearElastic
      name="implag2d" tol="1e-16" useMLPreconditioner="0" />
  </Solvers>

  <Mesh xcoords="-100 -10 10 100" ycoords="-100 -16 0" nx="10 20 10"
    ny="10 16" elementTypes="CPE4 CPE4 CPE4 STRI CPE4 CPE4 "
    regionNames="RegionA RegionA RegionA RegionB RegionA RegionA" />

  <Nodesets>
    <Nodeset name="strip" type="0" xmin="-3.1 -0.1 0" xmax="3.1 0.1 0" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="2.0" dt="1.0 s">
      <Apply solver="implag2d" toregions="RegionA RegionB" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="RegionA" elementtype="poly">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
    <ElementRegion name="RegionB" elementtype="linear">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <TractionBoundaryCondition object="Node"
      setnames="strip" direction="0.0 -1 0.0" scale="1.0 MPa" timetable="ttable" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
```

```
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="xpos" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 10.0" value="0.0, 1.0" />
  </Tables>

  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>

  <Output writePlot="0" restart_interval="1.0" plotfile_root="plot"
    parallel_silo="1" />

</Problem>
```

A.4 Example 4: Notched beam, 2D

A.4.1 Static simulation for SIF calculation

```
<?xml version="1.0" ?>
<!-- -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 GPa" />
    <Parameter name="SolidShearModulus" value="3.84615 GPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
  </Parameters>

  <Solvers>
    <LagrangeSmallStrainLinearElastic
      name="implag2d" twoD_option="1" tol="1e-16" useMLPreconditioner="0" />
  </Solvers>

  <Mesh xcoords="-20 20" ycoords="0 8" nx="80" ny="16" elementType="CPE4"
    regionNames="Region" />

  <Nodesets>
    <Nodeset name="leftsupport" type="0" xmin="-20.1 -0.1 0" xmax="-19.9 0.1 0" />
    <Nodeset name="rightsupport" type="0" xmin="19.9 -0.1 0" xmax="20.1 0.1 0" />
    <Nodeset name="loadstrip" type="0" xmin="-1.1 7.9 0" xmax=" 1.1 8.1 0" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="2.0" dt="1.0 s">
```

```
    <Apply solver="implag2d" toregions="Region" />
  </Application>
</SolverApplications>

<ElementRegions>
  <ElementRegion name="Region" elementtype="poly">
    <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
      Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
  </ElementRegion>
</ElementRegions>

<BoundaryConditions>
  <TractionBoundaryCondition object="Node"
    setnames="loadstrip" direction="0.0 -1 0.0" scale="1.0e6" timetable="ttable" /
  >
  <BoundaryCondition object="Node" fieldname="Displacement"
    setnames="leftsupport" component="0" scale="0.0" fieldtype="Vector" />
  <BoundaryCondition object="Node" fieldname="Displacement"
    setnames="leftsupport" component="1" scale="0.0" fieldtype="Vector" />
  <BoundaryCondition object="Node" fieldname="Displacement"
    setnames="rightsupport" component="1" scale="0.0" fieldtype="Vector" />
</BoundaryConditions>

<Tables>
  <Table1D name="ttable" coord="0.0, 2.0" value="0.0, 1.0" />
</Tables>

<Fracture fractureFlag="1" preFractureSetName="" separableSet="all"
  rockToughness="0.5e100" failCriterion="1" failgap="0.0004" x1_PreFrac="0"
  y1_PreFrac="0" z1_PreFrac="0" x2_PreFrac="0" y2_PreFrac="4"
  z2_PreFrac="0" />


<Partition>
  <SpatialPartition xpar="1" ypar="1" zpar="1" />
</Partition>

<Output writePlot="1" restart_interval="1.0" plot_interval="1.0"
  plotfile_root="plot" parallel_silo="1" writeFEMFaces="1"
  writeFEMEdges="1" />

</Problem>
```

A.4.2 Simulating fracture propagation using an implicit solver

```
<?xml version="1.0" ?>
<!-- -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 GPa" />
    <Parameter name="SolidShearModulus" value="3.84615 GPa" />
    <Parameter name="deltaTime" value="0.005 s" />
```

```
  <Parameter name="endTime" value="0.01 s" />
</Parameters>

<Solvers>
  <LagrangeSmallStrainLinearElastic
    name="implag2d" twoD_option="1" tol="1e-16" useMLPreconditioner="0" />
</Solvers>

<Mesh xcoords="-20 20" ycoords="0 8" nx="160" ny="32" elementType="CPE4"
  regionNames="Region" />

<Nodesets>
  <Nodeset name="leftsupport" type="0" xmin="-20.1 -0.1 0" xmax="-19.9 0.1 0" />
  <Nodeset name="rightsupport" type="0" xmin="19.9 -0.1 0" xmax="20.1 0.1 0" />
  <Nodeset name="loadstrip" type="0" xmin="-0.1 7.9 0" xmax=" 0.1 8.1 0" />
</Nodesets>

<SolverApplications>
  <Application name="1" begintime="0.0" endtime="3.0" dt="0.1 s">
    <Apply solver="implag2d" toregions="Region" />
  </Application>
</SolverApplications>

<ElementRegions>
  <ElementRegion name="Region" elementtype="poly">
    <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
      Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
  </ElementRegion>
</ElementRegions>

<BoundaryConditions>
  <BoundaryCondition object="Node" fieldname="Displacement"
    setnames="loadstrip" component="1" scale="-0.03" fieldtype="Vector"
    timetable="ttable" />
  <BoundaryCondition object="Node" fieldname="Displacement"
    setnames="leftsupport" component="0" scale="0.0" fieldtype="Vector" />
  <BoundaryCondition object="Node" fieldname="Displacement"
    setnames="leftsupport" component="1" scale="0.0" fieldtype="Vector" />
  <BoundaryCondition object="Node" fieldname="Displacement"
    setnames="rightsupport" component="1" scale="0.0" fieldtype="Vector" />
</BoundaryConditions>

<Tables>
  <Table1D name="ttable" coord="0.0, 3.0" value="0.0, 1.0" />
</Tables>

<Fracture fractureFlag="1" preFractureSetName="" separableSet="all"
  rockToughness="4e6" failCriterion="1" failgap="0.0004" x1_PreFrac="0"
  y1_PreFrac="0" z1_PreFrac="0" x2_PreFrac="0" y2_PreFrac="4"
  z2_PreFrac="0" />

<Partition>
  <SpatialPartition xpar="1" ypar="1" zpar="1" />
</Partition>

<Output writePlot="1" restart_interval="1.0" plot_interval="0.1"
  plotfile_root="fine" parallel_silo="1" writeFEMFaces="1"
```

```
    writeFEMEdges="1" />

</Problem>
```

A.4.3 Simulating fracture propagation using an explicit solver

```
<?xml version="1.0" ?>
<!-- -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 GPa" />
    <Parameter name="SolidShearModulus" value="3.84615 GPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
  </Parameters>

  <Solvers>
    <LagrangeLargeStrain name="solver1" courant="0.5"
      timeIntegrationOption="2" dampingM="100" />
  </Solvers>

  <Mesh xcoords="-20 20" ycoords="0 8" nx="80" ny="16" elementType="CPE4"
    regionNames="Region" />

  <Nodesets>
    <Nodeset name="leftsupport" type="0" xmin="-20.1 -0.1 0" xmax="-19.9 0.1 0" />
    <Nodeset name="rightsupport" type="0" xmin="19.9 -0.1 0" xmax="20.1 0.1 0" />
    <Nodeset name="loadstrip" type="0" xmin="-0.1 7.9 0" xmax=" 0.1 8.1 0" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="3.0">
      <Apply solver="solver1" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="poly">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="loadstrip" component="1" scale="-0.01" fieldtype="Vector"
      timetable="ttable" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="leftsupport" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="leftsupport" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
```

```
      setnames="rightsupport" component="1" scale="0.0" fieldtype="Vector" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 3.0" value="1.0, 1.0" />
  </Tables>

  <Fracture fractureFlag="1" preFractureSetName="" separableSet="all"
    rockToughness="4e6" failCriterion="1" failgap="0.0004" x1_PreFrac="0"
    y1_PreFrac="0" z1_PreFrac="0" x2_PreFrac="0" y2_PreFrac="4"
    z2_PreFrac="0" />

  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>

  <Output writePlot="1" writeRestart="0" restart_interval="100.0"
    plot_interval="0.1" plotfile_root="ex" parallel_silo="1"
    writeFEMFaces="1" writeFEMEdges="1" />

</Problem>
```

A.4.4 Mixed mode fracture propagation

```
<?xml version="1.0" ?>
<!-- -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 GPa" />
    <Parameter name="SolidShearModulus" value="3.84615 GPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
  </Parameters>

  <Solvers>
    <LagrangeSmallStrainLinearElastic
      name="implag2d" twoD_option="1" tol="1e-16" useMLPreconditioner="0" />
  </Solvers>

  <Mesh xcoords="-10 10" ycoords="-10 10" nx="40" ny="40"
    elementType="CPE4" regionNames="Region" />

  <Nodesets>
    <Nodeset name="leftsupport" type="0" xmin="-10.1 -10.1 0"
      xmax=" -9.9 -9.9 0" />
    <Nodeset name="rightsupport" type="0" xmin=" 9.9 -10.1 0"
      xmax="10.1 -9.9 0" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="1.0" dt="1.0 s">
      <Apply solver="implag2d" toregions="Region" />
```

```
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="poly">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <TractionBoundaryCondition object="Node"
      setnames="ypos" direction="0 1 0" scale="1.0e6" timetable="ttable" />
    <TractionBoundaryCondition object="Node"
      setnames="yneg" direction="0 -1 0" scale="1.0e6" timetable="ttable" />
    <TractionBoundaryCondition object="Node"
      setnames="xpos" direction="0 1 0" scale="1.0e6" timetable="ttable" />
    <TractionBoundaryCondition object="Node"
      setnames="ypos" direction="1 0 0" scale="1.0e6" timetable="ttable" />
    <TractionBoundaryCondition object="Node"
      setnames="yneg" direction="-1 0 0" scale="1.0e6" timetable="ttable" />
    <TractionBoundaryCondition object="Node"
      setnames="xneg" direction="0 -1 0" scale="1.0e6" timetable="ttable" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="leftsupport" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="leftsupport" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="rightsupport" component="1" scale="0.0" fieldtype="Vector" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 2.0" value="1.0, 1.0" />
  </Tables>

  <Fracture fractureFlag="1" preFractureSetName="" separableSet="all"
    rockToughness="0.5e100" failCriterion="1" failgap="0.0004" x1_PreFrac="-4"
    y1_PreFrac="0" z1_PreFrac="0" x2_PreFrac="4" y2_PreFrac="0"
    z2_PreFrac="0" />

  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>

  <Output writePlot="1" restart_interval="1.0" plot_interval="1.0"
    plotfile_root="plot" parallel_silo="1" writeFEMFaces="1"
    writeFEMEdges="1" />

</Problem>
```

A.5 Example 5: Notched beam, 3D

```
<?xml version="1.0" ?>
<!-- -->
<!--# # -->
```

```
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 GPa" />
    <Parameter name="SolidShearModulus" value="3.84615 GPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
  </Parameters>

  <Solvers>
    <LagrangeSmallStrainLinearElastic
      name="implag2d" tol="1e-14" useMLPreconditioner="0" />
  </Solvers>

  <Mesh xcoords="-20 20" ycoords="0 8" zcoords="-0.75 0.75" nx="80"
    ny="16" nz="3" elementType="C3D4" regionNames="Region" />

  <Nodesets>
    <Nodeset name="leftsupport" type="0" xmin="-20.1 -0.1 -100"
      xmax="-19.9 0.1 100" />
    <Nodeset name="rightsupport" type="0" xmin="19.9 -0.1 -100"
      xmax="20.1 0.1 100" />
    <Nodeset name="loadstrip" type="0" xmin="-1.1 7.9 -100" xmax=" 1.1 8.1 100" />
    <Nodeset name="notch" type="0" xmin="-0.1 -0.1 -100" xmax=" 0.1 4.1 100" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="2.0" dt="1.0 s">
      <Apply solver="implag2d" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="linear">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <TractionBoundaryCondition object="Node"
      setnames="loadstrip" direction="0.0 -1 0.0" scale="1.0e6" timetable="ttable" /
    >
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="leftsupport" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="leftsupport" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="rightsupport" component="1" scale="0.0" fieldtype="Vector" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 2.0" value="0.0, 1.0" />
  </Tables>
```

```
  <Fracture fractureFlag="1" preFractureSetName="notch"
    separableSet="all" rockToughness="0.5e100" failCriterion="1" failgap="0.0004" />

  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>

  <Output writePlot="1" restart_interval="1.0" plot_interval="1.0"
    plotfile_root="plot" parallel_silo="1" writeFEMFaces="1"
    writeFEMEdges="1" />

</Problem>
```

A.6 Example 6: Verification against the KGD problem

```
<?xml version="1.0" ?>
<Problem>

  <Solvers>
    <Hydrofracture name="solver1" timeIntegration="Explicit"
      ppSolverName="ppsolve" lgSolverName="lagsolve" />

    <LagrangeLargeStrain name="lagsolve" courant="0.6"
      timeIntegrationOption="2" dampingM="1000" />

    <ParallelPlateFlowSolver name="ppsolve"
      BulkModulus="1e7" ppcourant="0.4" mu="0.001" BartonJointParameters="0.5e-3 5e6
     0.1e-3"
      MaximumAperture="3e-2" pressurecap="3e6" apertureMovingAverageCoeff="0.99" />
  </Solvers>

  <Mesh xcoords="0 100 300" ycoords="-260 -60 60 260" nx="50 10" ny="10 60 10"
    nz="1" elementType="CPE4" regionNames="Region" />

  <Nodesets>
    <Nodeset name="source" type="0" xmin="-0.1 -0.1 0" xmax=" 2.1  0.1 0" />
    <Nodeset name="perf" type="0" xmin="-0.1 -0.1 0" xmax=" 4.1  0.1 0" />
    <Nodeset name="core" type="0" xmin="-0.1 -0.1 0" xmax=" 100.1  0.1 0" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="100.0">
      <Apply solver="solver1" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="poly">
      <LinearElasticMaterial ShearModulus="8.3333e9"
        Density="26500" BulkModulus="11.111e9" />
    </ElementRegion>
  </ElementRegions>
```

```
  <BoundaryConditions>
    <BoundaryCondition object="Face" fieldname="combinedFlowRate"
      setnames="source" scale="0.002" timetable="ttable" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xpos" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="ypos" component="1" scale="0.0" fieldtype="Vector" />
  </BoundaryConditions>

  <InitialConditions>
    <ConstantInitialCondition toregions="Region"
      fieldname="sigma_x" fieldtype="Scalar" object="Element" value="-1e6" />
    <ConstantInitialCondition toregions="Region"
      fieldname="sigma_y" fieldtype="Scalar" object="Element" value="-0.5e6" />
  </InitialConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 0.01, 1.0e9" value="1.0, 1.0, 1.0" />
  </Tables>

  <Partition>
    <SpatialPartition xpar="4" ypar="3" zpar="1" />
  </Partition>

  <Fracture fractureFlag="10" preFractureSetName="perf"
    separableSet="core" failCriterion="1" rockToughness="0.5e4" failgap="0.0004"
    verbose="0" maxKinkAngle="134.0" kinkStrength="0.0e6" />

  <Output writePlot="1" plot_interval="1" restart_interval="1000.0"
    plotfile_root="kgd" slave_directory="sub" parallel_silo="4"
    writeFEMFaces="1" />
</Problem>
```

A.7 Example 7: Static penny-shape fracture

A.7.1 Static penny-shape fracture in a structured mesh

```
<?xml version="1.0" ?>
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 GPa" />
    <Parameter name="SolidShearModulus" value="3.84615 GPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
```

```
  </Parameters>

  <Solvers>
    <LagrangeSmallStrainLinearElastic
      name="implag2d" tol="1e-10" useMLPreconditioner="0" />
  </Solvers>

  <Mesh xcoords="0 16 48" ycoords="0 16 48" zcoords="-56 -16 16 56"
    nx="16 8" ny="16 8" nz="8 12 8" elementType="C3D4" regionNames="Region" />

  <Nodesets>
    <Nodeset name="crack" type="1" point1="0 0.0 0.1" point2="0 0.0 -0.1"
      radius="8.1" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="1.0" dt="1.0 s">
      <Apply solver="implag2d" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="linear">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <TractionBoundaryCondition object="Node"
      setnames="zneg" direction="0.0 0.0 -1.0" scale="1.0e6" timetable="ttable" />
    <TractionBoundaryCondition object="Node"
      setnames="zpos" direction="0.0 0.0 1.0" scale="1.0e6" timetable="ttable" />

  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 1.0" value="0.0, 1.0" />
  </Tables>

  <Fracture fractureFlag="1" preFractureSetName="crack"
    separableSet="all" rockToughness="0.5e100" failCriterion="1" failgap="0.0004" />

  <Partition>
    <SpatialPartition xpar="2" ypar="2" zpar="4" />
  </Partition>

  <Output writePlot="1" restart_interval="1.0" plot_interval="1.0"
    plotfile_root="plot" parallel_silo="1" writeFEMFaces="1"
    writeFEMEdges="1" />

</Problem>
```

A.7.2 Static penny-shape fracture in a geometrically conforming mesh

```
<?xml version="1.0" ?>

<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 GPa" />
    <Parameter name="SolidShearModulus" value="3.84615 GPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
  </Parameters>

  <Solvers>
    <LagrangeSmallStrainLinearElastic
      name="implag2d" tol="1e-10" useMLPreconditioner="0" />
  </Solvers>

  <Mesh file="QuarterDisk_Hex.inp" />

  <Nodesets>
    <Nodeset name="crack" type="1" point1="0 0.0 -0.1" point2="0 0.0 0.1"
      radius="10.1" />
    <Nodeset name="top" type="1" point1="0 0.0 19.9" point2="0 0.0 20.1"
      radius="1000.1" />
    <Nodeset name="bot" type="1" point1="0 0.0 -20.1" point2="0 0.0 -19.9"
      radius="1000.1" />
    <Nodeset name="xneg" type="0" xmin="-0.1 -100.1 -100" xmax="0.1 100.1 100" />
    <Nodeset name="yneg" type="0" xmin="-100.1 -0.1 -100" xmax="100.1 0.1 100" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="1.0" dt="1.0 s">
      <Apply solver="implag2d" toregions="EB1" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="EB1" elementtype="poly">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <TractionBoundaryCondition object="Node"
      setnames="bot" direction="0.0 0.0 -1.0" scale="1.0e6" timetable="ttable" />
    <TractionBoundaryCondition object="Node"
      setnames="top" direction="0.0 0.0 1.0" scale="1.0e6" timetable="ttable" />
```

```
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 1.0" value="0.0, 1.0" />
  </Tables>

  <Fracture fractureFlag="1" preFractureSetName="crack"
    separableSet="all" rockToughness="0.5e100" failCriterion="1" failgap="0.0004" />

  <Partition>
    <SpatialPartition xpar="2" ypar="2" zpar="4" />
  </Partition>

  <Output writePlot="1" restart_interval="1.0" plot_interval="1.0"
    plotfile_root="abqs" parallel_silo="1" writeFEMFaces="1"
    writeFEMEdges="1" />

</Problem>
```

A.7.3 Static penny-shape fracture pressurized internally

```
<?xml version="1.0" ?>

<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 GPa" />
    <Parameter name="SolidShearModulus" value="3.84615 GPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
  </Parameters>

  <Solvers>
    <LagrangeSmallStrainLinearElastic
      name="implag2d" tol="1e-10" useMLPreconditioner="0" />
  </Solvers>

  <Mesh file="QuarterDisk_Hex.inp" />

  <Nodesets>
    <Nodeset name="crack" type="1" point1="0 0.0 -0.1" point2="0 0.0 0.1"
      radius="10.1" />
    <Nodeset name="top" type="1" point1="0 0.0 19.9" point2="0 0.0 20.1"
      radius="1000.1" />
    <Nodeset name="bot" type="1" point1="0 0.0 -20.1" point2="0 0.0 -19.9"
      radius="1000.1" />
    <Nodeset name="xneg" type="0" xmin="-0.1 -100.1 -100" xmax="0.1 100.1 100" />
    <Nodeset name="yneg" type="0" xmin="-100.1 -0.1 -100" xmax="100.1 0.1 100" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="1.0" dt="1.0 s">
      <Apply solver="implag2d" toregions="EB1" />
```

```
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="EB1" elementtype="poly">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <TractionBoundaryCondition object="Node"
      setnames="crack" applyNormalTraction="1" direction="0.0 0.0 1.0"
      scale="1.0e6" timetable="ttable" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 1.0" value="1.0, 1.0" />
  </Tables>

  <Fracture fractureFlag="1" preFractureSetName="crack"
    separableSet="all" rockToughness="0.5e100" failCriterion="1" failgap="0.0004" />

  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>

  <Output writePlot="1" restart_interval="1.0" plot_interval="1.0"
    plotfile_root="abqs" parallel_silo="1" writeFEMFaces="1"
    writeFEMEdges="1" />

</Problem>
```

A.8 Example 8: Penny-shape hydraulic fracture

```
<?xml version="1.0" ?>
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Solvers>
    <LagrangeDynamicsParallelPlateFlowExplicit
      name="hf1" courant="0.8" ppcourant="0.5" BulkModulus="10.0e6"
      dampingM="1000" MaximumAperture="1.0e-2" pressurecap="3e6"
      BartonJointParameters="0.2e-4 5e6 0.1e-3" />
  </Solvers>

  <Mesh xcoords="0 100 220" ycoords="0 100 220" zcoords="-260 -60 60 260"
    nx="50 12" ny="50 12" nz="20 40 20" elementType="C3D8" regionNames="Region" />

  <Nodesets>
    <Nodeset name="crack" type="0" xmin="-0.1 -0.1 -0.1" xmax=" 4.1  4.1  0.1" />
```

```
    <Nodeset name="source" type="0" xmin="-0.1 -0.1 -0.1" xmax=" 2.1  2.1  0.1" />
    <Nodeset name="fracturePlane" type="0" xmin="-0.1    -0.1 -0.1"
      xmax=" 120.1 120.1  0.1" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="100.0">
      <Apply solver="hf1" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="poly">
      <LinearElasticMaterial ShearModulus="8.3333e9"
        Density="26500" BulkModulus="11.111e9" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xpos" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="ypos" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zneg" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zpos" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Face" fieldname="combinedFlowRate"
      setnames="source" scale="0.1" timetable="ttable" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 2.0" value="1.0, 1.0" />
  </Tables>

  <Fracture fractureFlag="1" preFractureSetName="crack"
    separableSet="fracturePlane" rockToughness="1e5" failCriterion="1"
    failgap="0.0004" />

  <Partition>
    <SpatialPartition xpar="6" ypar="6" zpar="8" />
  </Partition>

  <Output writePlot="1" restart_interval="1000.0" plot_interval="1.0"
    plotfile_root="plot" parallel_silo="32" slave_directory="sub"
    writeFEMFaces="1" writeFEMEdges="1" />

</Problem>
```

A.9 Example 9: Penny-shape hydraulic fracture with a sink

```
<?xml version="1.0" ?>
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Solvers>
    <LagrangeDynamicsParallelPlateFlowExplicit
      name="hf1" courant="0.8" ppcourant="0.5" BulkModulus="10.0e6"
      dampingM="1000" MaximumAperture="1.0e-2" pressurecap="3e6"
      BartonJointParameters="0.5e-4 5e6 0.1e-3" />
  </Solvers>

  <Mesh xcoords="-300 -180 180 300" ycoords="0 180 300" zcoords="-360 -60 60 360"
    nx="6 60 6" ny="30 6" nz="10 12 10" elementType="C3D8" regionNames="Region" />

  <Nodesets>
    <Nodeset name="perf" type="0" xmin="-12.1 -0.1 -0.1" xmax=" 6.1 12.1  0.1" />
    <Nodeset name="source" type="0" xmin="-6.1 -0.1 -0.1" xmax=" 0.1  6.1  0.1" />
    <Nodeset name="fracturePlane" type="0" xmin="-180.1 -0.1 -0.1"
      xmax=" 180.1 180.1  0.1" />
    <Nodeset name="sink" type="0" xmin=" 89.9 -0.1 -0.1" xmax=" 96.1 6.1  0.1" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="300.0">
      <Apply solver="hf1" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="poly">
      <LinearElasticMaterial ShearModulus="8.3333e9"
        Density="26500" BulkModulus="11.111e9" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xpos" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="ypos" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zneg" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zpos" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Face" fieldname="combinedFlowRate"
      setnames="source" scale="0.1" timetable="ttable" />
    <BoundaryCondition object="Face" fieldname="Pressure"
```

```
      setnames="sink" scale="1e4" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 2.0" value="1.0, 1.0" />
  </Tables>

  <Fracture fractureFlag="1" preFractureSetName="perf"
    separableSet="fracturePlane" rockToughness="1e5" failCriterion="1"
    failgap="0.0004" />

  <Partition>
    <SpatialPartition xpar="4" ypar="4" zpar="4" />
  </Partition>

  <Output writePlot="1" restart_interval="1000.0" plot_interval="3.0"
    plotfile_root="sink" parallel_silo="32" slave_directory="sub"
    writeFEMFaces="1" writeFEMEdges="1" />
</Problem>
```

A.10 Example 10: 3D hydraulic fracturing - verification against the PKN model

```
<?xml version="1.0" ?>
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Solvers>
    <LagrangeDynamicsParallelPlateFlowExplicit
      name="hf1" courant="0.8" ppcourant="0.5" BulkModulus="10.0e6"
      dampingM="1000" MaximumAperture="1.0e-2" pressurecap="3e6"
      BartonJointParameters="0.5e-4 5e6 0.1e-3" />
  </Solvers>

  <Mesh xcoords="0 200 300 400" ycoords="-400 -150 -50 50 150 400"
    zcoords="-400 -150 -50 50 150 400" nx="20 5 3" ny="5 5 10 5 5" nz="5 5 10 5 5"
    elementType="C3D8" regionNames="Region" />

  <Nodesets>
    <Nodeset name="perf" type="0" xmin="-0.1 -0.1 -10.1" xmax="20.1 0.1 10.1" />
    <Nodeset name="source" type="0" xmin="-0.1 -0.1 -10.1" xmax="10.1 0.1 10.1" />
    <Nodeset name="fracturePlane" type="0" xmin="-0.1 -0.1 -50.1"
      xmax=" 300.1 0.1 50.1" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="3000.0">
      <Apply solver="hf1" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
```

```
    <ElementRegion name="Region" elementtype="poly">
      <LinearElasticMaterial ShearModulus="8.3333e9"
        Density="26500" BulkModulus="11.111e9" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xpos" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="ypos" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zneg" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zpos" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Face" fieldname="combinedFlowRate"
      setnames="source" scale="0.1" timetable="ttable" />

  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 2.0" value="1.0, 1.0" />
  </Tables>

  <Fracture fractureFlag="10" preFractureSetName="perf"
    separableSet="fracturePlane" rockToughness="1e5" failCriterion="1"
    failgap="0.0004" />

  <Partition>
    <SpatialPartition xpar="4" ypar="2" zpar="3" />
  </Partition>

  <Output writePlot="1" restart_interval="10000.0" plot_interval="20.0"
    plotfile_root="pkn" parallel_silo="16" slave_directory="sub"
    writeFEMFaces="1" writeFEMEdges="1" />

</Problem>
```

A.11 Example 11: Spatially varying material property

```
<?xml version="1.0" ?>
<!-- -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="gpac.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
```

```
  <Parameter name="SolidBulkModulus" value="8.33333 MPa" />
  <Parameter name="SolidShearModulus" value="3.84615 MPa" />
  <Parameter name="deltaTime" value="0.005 s" />
  <Parameter name="endTime" value="0.01 s" />
</Parameters>

<Solvers>
  <LagrangeSmallStrainLinearElastic
    name="implag2d" tol="1e-10" useMLPreconditioner="0" />
</Solvers>

<Mesh externalMeshFile="0" xcoords="-30 -10 10 30" ycoords="-5 0 5"
  nx="8 16 8" ny="4 4 " elementType="CPE4"
  regionNames="RegionA RegionB RegionB RegionC RegionA RegionB" />

<SolverApplications>
  <Application name="1" begintime="0.0" endtime="1.0" dt="1.0 s">
    <Apply solver="implag2d" toregions="RegionA RegionB RegionC" />
  </Application>
</SolverApplications>

<ElementRegions>
  <ElementRegion name="RegionA" elementtype="poly">
    <LinearElasticMaterial ShearModulus="1e7"
      Density="2650" BulkModulus="1e7" />
  </ElementRegion>
  <ElementRegion name="RegionB" elementtype="poly">
    <LinearElasticMaterial ShearModulus="2e7"
      Density="2650" BulkModulus="2e7" />
  </ElementRegion>
  <ElementRegion name="RegionC" elementtype="poly">
    <LinearElasticMaterial ShearModulus="2e7"
      Density="2650" BulkModulus="2e7" />
  </ElementRegion>
</ElementRegions>

<InitialConstitutiveValues>
  <InitialConstitutiveValue object="FiniteElement_ElementRegion"
    propertytype="ShearModulus" tablename="G" toregion="RegionC" />
  <InitialConstitutiveValue object="FiniteElement_ElementRegion"
    propertytype="Lame" tablename="G" toregion="RegionC" />
</InitialConstitutiveValues>

<BoundaryConditions>
  <TractionBoundaryCondition object="Node"
    setnames="xpos" direction="1.0 0 0.0" scale="3.0 MPa" timetable="ttable" />
  <BoundaryCondition object="Node" fieldname="Displacement"
    setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
  <BoundaryCondition object="Node" fieldname="Displacement"
    setnames="xneg" component="1" scale="0.0" fieldtype="Vector" />
</BoundaryConditions>

<Tables>
  <Table1D name="ttable" coord="0.0, 2.0" value="1.0, 1.0" />
  <Table3D name="G" x_file="x" y_file="y" z_file="z" voxel_file="G" />
</Tables>
```

```
  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>

  <Output writePlot="1" writeRestart="1" restart_interval="1.0"
    plot_interval="1.0" plotfile_root="plot" parallel_silo="1" />

</Problem>
```

A.12 Example 12: Bore hole mixed mesh

```
<?xml version="1.0" ?>

<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 GPa" />
    <Parameter name="SolidShearModulus" value="3.84615 GPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
  </Parameters>

  <Solvers>
    <LagrangeSmallStrainLinearElastic
      name="implag2d" tol="1e-10" useMLPreconditioner="0" />
  </Solvers>

  <Mesh file="BoreHoleMixedMesh.inp" />

  <Nodesets>
    <Nodeset name="interior" type="1" point1="0 0.0 -0.1" point2="0 0.0 0.1"
      radius="1.01" />
    <Nodeset name="xneg" type="0" xmin="-5.01 -5.01 -0.1" xmax="-4.99 5.01 0.1"
    />
    <Nodeset name="xpos" type="0" xmin="4.99 -5.01 -0.1" xmax="5.01 5.01 0.1" />
    <Nodeset name="yneg" type="0" xmin="-5.01 -5.01 -0.1" xmax="5.01 -4.99 0.1" />
    <Nodeset name="ypos" type="0" xmin="-5.01 4.99 -0.1" xmax="5.01 5.01 0.1" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="10.0" dt="0.5 s">
      <Apply solver="implag2d" toregions="R1 R2" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="R1" elementtype="poly">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
    </ElementRegion>
    <ElementRegion name="R2" elementtype="linear">
      <LinearElasticMaterial ShearModulus="$:SolidShearModulus"
        Density="$:SolidDensity" BulkModulus="$:SolidBulkModulus" />
```

```
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <TractionBoundaryCondition object="Node"
      setnames="interior" applyNormalTraction="1" direction="0.0 1.0 0"
      scale="1e6" timetable="ttable" />
    <TractionBoundaryCondition object="Node"
      setnames="xneg" direction="1.0 0.0 0" scale="1.0e6" />
    <TractionBoundaryCondition object="Node"
      setnames="xpos" direction="-1.0 0.0 0" scale="1.0e6" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="yneg ypos" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Displacement"
      setnames="yneg ypos" component="1" scale="0.0" fieldtype="Vector" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 10.0" value="0.0, 10.0" />
  </Tables>

  <Fracture fractureFlag="1" preFractureSetName="" separableSet=""
    rockToughness="1e6" failCriterion="1" failgap="0.0004" verbose="1"
    x1_PreFrac="1, -1" y1_PreFrac="0, 0" z1_PreFrac="0, 0" x2_PreFrac="1.2, -1.2"
    y2_PreFrac="0, 0" z2_PreFrac="0, 0" />

  <Partition>
    <SpatialPartition xpar="1" ypar="1" zpar="1" />
  </Partition>

  <Output writePlot="1" restart_interval="1000.0" plot_interval="0.5"
    plotfile_root="bore" parallel_silo="1" writeFEMFaces="1"
    writeFEMEdges="1" />

</Problem>
```

A.13 Example 13: Placeholder for new method of applying in situ stress

Obsolete. The method of applying in situ stress has been covered by the updated Example 6.

```
<?xml version="1.0" ?>
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="gpac.xsd">

  <Parameters>
    <Parameter name="SolidDensity" value="2650 kg/m^3" />
    <Parameter name="SolidBulkModulus" value="8.33333 MPa" />
    <Parameter name="SolidShearModulus" value="3.84615 MPa" />
    <Parameter name="deltaTime" value="0.005 s" />
    <Parameter name="endTime" value="0.01 s" />
```

```
</Parameters>

<Solvers>
  <LagrangeLargeStrain name="solver1" courant="0.5"
    timeIntegrationOption="2" dampingM="100" />
</Solvers>

<Mesh xcoords="-30 -10 10 30" ycoords="-30 -10 10 30" nx="5 40 5"
  ny="5 20 5 " elementType="CPE4" regionNames="Region" />

<SolverApplications>
  <Application name="1" begintime="0.0" endtime="2.0">
    <Apply solver="solver1" toregions="Region" />
  </Application>
</SolverApplications>

<ElementRegions>
  <ElementRegion name="Region" elementtype="poly">
    <LinearElasticMaterial ShearModulus="1e7"
      Density="2650" BulkModulus="1e7" />
  </ElementRegion>
</ElementRegions>

<Nodesets>
  <Nodeset name="crack" type="0" xmin="-6.1 -0.1 -100" xmax=" 6.1  0.1  100" />
</Nodesets>

<InitialConditions>
  <InitialConditionTable toregions="Region"
    fieldname="sigma_y" fieldtype="Scalar" object="Element" table="sY" />
</InitialConditions>

<BoundaryConditions>
  <BoundaryCondition object="Node" fieldname="Velocity"
    setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
  <BoundaryCondition object="Node" fieldname="Velocity"
    setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
  <BoundaryCondition object="Node" fieldname="Velocity"
    setnames="xpos" component="0" scale="0.0" fieldtype="Vector" />
  <BoundaryCondition object="Node" fieldname="Velocity"
    setnames="ypos" component="1" scale="0.0" fieldtype="Vector" />
  <TractionBoundaryCondition object="Node"
    setnames="crack" applyNormalTraction="1" direction="0.0 1.0 0.0"
    scale="1.0e6" timetable="ttable" />
</BoundaryConditions>

<Tables>
  <Table1D name="ttable" coord="0.0, 2.0" value="0.0, 1.0" />
  <Table3D name="sY" x_file="x" y_file="y" z_file="z"
    voxel_file="stressY" />
</Tables>

<Fracture fractureFlag="1" preFractureSetName="crack"
  separableSet="all" rockToughness="0.5e100" failCriterion="1" failgap="0.0004" />

<Partition>
  <SpatialPartition xpar="1" ypar="1" zpar="1" />
```

```
  </Partition>

  <Output writePlot="1" writeRestart="0" restart_interval="1.0"
    plot_interval="0.1" plotfile_root="plot" parallel_silo="1" />

</Problem>
```

A.14 Example 14: Joint shearing

```
<?xml version="1.0" ?>
<Problem>

  <Solvers>
    <Hydrofracture name="solver1" timeIntegration="Explicit"
      ppSolverName="ppsolve" lgSolverName="lagsolve" COFJoint="0.7" />

    <LagrangeLargeStrain name="lagsolve" courant="0.2"
      timeIntegrationOption="2" dampingM="1000" />

    <ParallelPlateFlowSolver name="ppsolve"
      BulkModulus="1e7" ppcourant="0.2" mu="0.001" BartonJointParameters="0.5e-3 5e6
     0.1e-3"
      MaximumAperture="3e-2" pressurecap="3e6" apertureMovingAverageCoeff="0.99" />
  </Solvers>

  <Mesh externalMeshFile="0" xcoords="-16 16" ycoords="-16 16" nx="32"
    ny="32" elementType="STRI" regionNames="Region" />

  <Nodesets>
    <Nodeset name="source" type="0" xmin="-0.1 -1.1 0" xmax=" 0.1 -0.9 0" />
    <Nodeset name="source" type="0" xmin="0.9 -0.1 0" xmax="1.1 0.1 0" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="100.0">
      <Apply solver="solver1" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="linear">
      <LinearElasticMaterial ShearModulus="8.3333e9"
        Density="2650" BulkModulus="11.111e9" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Face" fieldname="Pressure"
      setnames="source" scale="1.8e6" timetable="ttable" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xpos" component="0" scale="0.0" fieldtype="Vector" />
```

```
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="ypos" component="1" scale="0.0" fieldtype="Vector" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 0.01,  1.0e9" value="1.0, 1.0, 1.0" />
  </Tables>

  <Partition>
    <SpatialPartition xpar="2" ypar="2" zpar="1" />
  </Partition>

  <Fracture fractureFlag="10" preFractureSetName=""
    separableSet="" insitu_Stress="-1e6 -3e6 0" failstress="2e5"
    rockToughness="1e6" failgap="0.0004" verbose="0" maxKinkAngle="134.0"
    kinkStrength="0.0e6" x1_PreFrac="-6" y1_PreFrac="-7" z1_PreFrac="0"
    x2_PreFrac="6" y2_PreFrac="5" z2_PreFrac="0" />

  <Output writePlot="1" plot_interval="0.05" restart_interval="1000.0"
    plotfile_root="JS" slave_directory="sub" parallel_silo="1"
    writeFEMFaces="1" />
</Problem>
```

A.15 Example 15: Bifurcating wings

```
<?xml version="1.0" ?>
<!-- -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Solvers>
    <Hydrofracture name="hf1" timeIntegration="Explicit"
      ppSolverName="ppsolve" lgSolverName="lagsolve" />

    <LagrangeLargeStrain name="lagsolve" courant="0.6"
      timeIntegrationOption="2" dampingM="1000" />

    <ParallelPlateFlowSolver name="ppsolve"
      BulkModulus="1e7" ppcourant="0.4" mu="0.001" BartonJointParameters="0.5e-3 5e6
     0.1e-3"
      MaximumAperture="2e-3" pressurecap="3e6" apertureMovingAverageCoeff="0.99" />
  </Solvers>

  <Mesh externalMeshFile="0" xcoords="-7 11" ycoords="-12 12"
    zcoords="-8 8" nx="18" ny="24" nz="16" elementType="C3D4" regionNames="Region" /
    >

  <Nodesets>
    <Nodeset name="crack" type="0" xmin="-7.1 -0.1 -2.1" xmax="-3.9  0.1 2.1" />
    <Nodeset name="crack" type="0" xmin="0.9 -0.1 -4.1" xmax="1.1  0.1  4.1" />
    <Nodeset name="crack" type="0" xmin="1.9 0.9 -0.1" xmax="2.1 1.1  4.1" />
    <Nodeset name="crack" type="0" xmin="2.9 1.9 -0.1" xmax="3.1 2.1  4.1" />
```

```
    <Nodeset name="crack" type="0" xmin="1.9 -1.1 -4.1" xmax="2.1 -0.9 0.1" />
    <Nodeset name="crack" type="0" xmin="2.9 -2.1 -4.1" xmax="3.1 -1.9 0.1" />
    <Nodeset name="source" type="0" xmin="-7.1 -0.1 -2.1" xmax="-5.9 0.1 2.1" />
    <Nodeset name="core" type="0" xmin="-7.1 -12.1 -4.1" xmax="11.1 12.1 4.1" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="100.0">
      <Apply solver="hf1" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="linear">
      <LinearElasticMaterial ShearModulus="8.3333e9"
        Density="26500" BulkModulus="11.111e9" />
    </ElementRegion>
  </ElementRegions>

  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zneg" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zpos" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Face" fieldname="combinedFlowRate"
      setnames="source" scale="0.02" timetable="ttable" />
  </BoundaryConditions>

  <Tables>
    <Table1D name="ttable" coord="0.0, 2.0" value="1.0, 1.0" />
  </Tables>

  <Fracture fractureFlag="10" preFractureSetName="crack"
    separableSet="core" rockToughness="2e5" failCriterion="1" failgap="0.0004" />

  <Partition>
    <SpatialPartition xpar="4" ypar="4" zpar="4" />
  </Partition>

  <Output writePlot="1" restart_interval="1000.0" plot_interval="0.5"
    plotfile_root="plot" parallel_silo="16" slave_directory="sub"
    writeFEMFaces="1" writeFEMEdges="1" />

</Problem>
```

A.16 Example 16: 3D hydraulic fracturing in arbitrary stress field

```
<?xml version="1.0" ?>
<!-- This is a dummy example showing multi-stage hydraulic fracturing with
  the presence of natural fractures. The xml itself wont run; the user will
```

```
  have to supply the geometries and the 3D table text file for the in situ
  stresses. -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Solvers>
    <Hydrofracture name="hf1" timeIntegration="Explicit"
      ppSolverName="ppsolve" lgSolverName="lagsolve" />

    <LagrangeLargeStrain name="lagsolve" courant="0.6"
      timeIntegrationOption="2" dampingM="500" timeToSnapshotDisp="600" />

    <!-- Because the arbitrary stress fields supplied by the user are not
    necessarily
      in equilibrium, we let the model "settle" for 600 seconds by starting pumping
      at t=600. The user will want to excelude the solid deformation that takes
      place in the first 600 s from the final results. By using this
    timeToSnapshotDisp
      attribute, GEOS will take a snapshot of the displacement field at the
    specified
      time so that the user can use it to construct the real displacement field. -->

    <ParallelPlateFlowSolver name="ppsolve"
      BulkModulus="30.0e6" ppcourant="0.33" mu="0.001"
      BartonJointParameters="3e-4 5e6 1e-4" MaximumAperture="1.5e-3"
      pressurecap="13e6" leakoffCoefficient="5e-6" />
  </Solvers>

  <Mesh xcoords="x0 ..." ycoords="y0 ..." zcoords="z0 ..." nx="nx1 ..."
    ny="ny1 ..." nz="nz1 ..." elementType="C3D4" regionNames="Region" />

  <Nodesets>
    <Nodeset name="perf1" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="perf2" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="source1" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="source2" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="nf1" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="nf2" type="0" xmin="x y z" xmax="x y z" />
    <!-- I removed some additional natural fracture nodesets. -->
    <Nodeset name="core" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="core" type="0" xmin="x y z" xmax="x y z" />
  </Nodesets>

  <SolverApplications>
    <Application name="1" begintime="0.0" endtime="132000.0">
      <Apply solver="hf1" toregions="Region" />
    </Application>
  </SolverApplications>

  <ElementRegions>
    <ElementRegion name="Region" elementtype="linear">
      <LinearElasticMaterial ShearModulus="16.667e9"
        Density="100000" BulkModulus="22.222e9" />
    </ElementRegion>
  </ElementRegions>
```

```
  <BoundaryConditions>
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xpos" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="ypos" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zneg" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zpos" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Face" fieldname="combinedFlowRate"
      setnames="source1" scale="0.12" timetable="ttable1" />
    <BoundaryCondition object="Face" fieldname="combinedFlowRate"
      setnames="source2" scale="0.12" timetable="ttable2" />
  </BoundaryConditions>

  <Tables>
    <Table3D name="ShminNet" x_file="x" y_file="y" z_file="z"
      voxel_file="Shmin_GRv2_Net1000" />
    <Table3D name="SHMaxNet" x_file="x" y_file="y" z_file="z"
      voxel_file="SHmax_GR_syntheticC" />
    <!-- These five files are supposed to be supplied by the user -->
    <Table1D name="ttable1" coord="0.0, 600, 600.01, 3000, 3000.01"
      value="0.0, 0.0, 1.0, 1.0, 0.0" />
    <Table1D name="ttable2" coord="0.0, 3600, 3600.01, 6000, 6000.01"
      value="0.0, 0.0, 1.0, 1.0, 0.0" />
  </Tables>

  <InitialConditions>
    <InitialConditionTable toregions="Region"
      fieldname="sigma_y" fieldtype="Scalar" object="Element" table="ShminNet" />
    <InitialConditionTable toregions="Region"
      fieldname="sigma_x" fieldtype="Scalar" object="Element" table="SHMaxNet" />
    <ConstantInitialCondition toregions="Region"
      fieldname="sigma_z" fieldtype="Scalar" object="Element" value="-15e6" />
  </InitialConditions>

  <Fracture fractureFlag="40" preFractureSetName="perf1 perf2 nf1 nf2"
    separableSet="core" rockToughness="2e6" failCriterion="1" failgap="0.0004"
    maxTurnAngle="30" />

  <Partition>
    <SpatialPartition xpar="12" ypar="12" zpar="16" />
  </Partition>

  <Output writePlot="1" writeRestart="1" restart_interval="600.0"
    plot_interval="5.0" plotfile_root="MA" parallel_silo="64"
    slave_directory="sub" writeFEMFaces="1" writeFEMEdges="1"
    fieldsToPlot="CommonHydroFrac refDisplacement" />
</Problem>
```

A.17 Example 17: Generic elastic orthotropy material model

```
<?xml version="1.0" ?>
<!-- This is a dummy example showing a solid body with five layers with different
  material properties. The user will have put in geometries and generate stress
  tables. -->
<!--# # -->
<Problem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="geos_v0.0.xsd">

  <Solvers>
    <Hydrofracture name="hf1" timeIntegration="Explicit"
      ppSolverName="ppsolve" lgSolverName="lagsolve" />

    <LagrangeLargeStrain name="lagsolve" courant="0.6"
      timeIntegrationOption="2" dampingM="300" />

    <ParallelPlateFlowSolver name="ppsolve"
      BulkModulus="20.0e6" ppcourant="0.4" mu="0.001"
      BartonJointParameters="3e-4 5e6 1e-4" MaximumAperture="2.5.0e-3"
      pressurecap="8e6" />
  </Solvers>

  <Mesh xcoords="x0 x1 x2 x3" ycoords="y0 y1 y2 y3" zcoords="z0 z1 z2 z3 z4 z5"
    nx="8 61 8" ny="6 15 6" nz="13 15 3 6 13" elementType="C3D8"
    regionNames="Layer5 Layer4 Layer3 Layer2 Layer1
                 Layer5 Layer4 Layer3 Layer2 Layer1
                 Layer5 Layer4 Layer3 Layer2 Layer1
                 Layer5 Layer4 Layer3 Layer2 Layer1
                 Layer5 Layer4 Layer3 Layer2 Layer1
                 Layer5 Layer4 Layer3 Layer2 Layer1
                 Layer5 Layer4 Layer3 Layer2 Layer1
                 Layer5 Layer4 Layer3 Layer2 Layer1" />

  <Nodesets>
    <Nodeset name="perf" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="perf" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="perf" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="source" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="source" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="source" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="fractureface" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="fractureface" type="0" xmin="x y z" xmax="x y z" />
    <Nodeset name="fractureface" type="0" xmin="x y z" xmax="x y z" />
  </Nodesets>

  <ElementRegions>
    <ElementRegion name="Layer1" elementtype="poly">
      <LinearElasticMaterial ShearModulus="X"
        Density="X" BulkModulus="X" />
    </ElementRegion>
    <ElementRegion name="Layer2" elementtype="poly">
      <LinearElasticOrthotropic Density="X" E1="X"
```

```
      E2="X" E3="X" nu12="X" nu23="X" nu13="X" G12="X" G23="X" G13="X" />
  </ElementRegion>
  <ElementRegion name="Layer3" elementtype="poly">
    <LinearElasticOrthotropic Density="X" E1="X"
      E2="X" E3="X" nu12="X" nu23="X" nu13="X" G12="X" G23="X" G13="X" />
  </ElementRegion>
  <ElementRegion name="Layer4" elementtype="poly">
    <LinearElasticOrthotropic Density="X" E1="X"
      E2="X" E3="X" nu12="X" nu23="X" nu13="X" G12="X" G23="X" G13="X" />
  </ElementRegion>
  <ElementRegion name="Layer5" elementtype="poly">
    <LinearElasticOrthotropic Density="X" E1="X"
      E2="X" E3="X" nu12="X" nu23="X" nu13="X" G12="X" G23="X" G13="X" />
  </ElementRegion>
</ElementRegions>

<SolverApplications>
  <Application name="1" begintime="0.0" endtime="20000.0">
    <Apply solver="hf1" toregions="Layer1 Layer2 Layer3 Layer4 Layer5" />
  </Application>
</SolverApplications>

<InitialConditions>
  <InitialConditionTable toregions="Layer1"
    fieldname="sigma_x" fieldtype="Scalar" object="Element" table="SX" />
  <InitialConditionTable toregions="Layer1"
    fieldname="sigma_y" fieldtype="Scalar" object="Element" table="SY" />
  <ConstantInitialCondition toregions="Layer1"
    fieldname="sigma_z" fieldtype="Scalar" object="Element" value="-20e6" />
  <InitialConditionTable toregions="Layer2"
    fieldname="sigma_x" fieldtype="Scalar" object="Element" table="SX" />
  <InitialConditionTable toregions="Layer2"
    fieldname="sigma_y" fieldtype="Scalar" object="Element" table="SY" />
  <ConstantInitialCondition toregions="Layer2"
    fieldname="sigma_z" fieldtype="Scalar" object="Element" value="-20e6" />
  <InitialConditionTable toregions="Layer3"
    fieldname="sigma_x" fieldtype="Scalar" object="Element" table="SX" />
  <InitialConditionTable toregions="Layer3"
    fieldname="sigma_y" fieldtype="Scalar" object="Element" table="SY" />
  <ConstantInitialCondition toregions="Layer3"
    fieldname="sigma_z" fieldtype="Scalar" object="Element" value="-20e6" />
  <InitialConditionTable toregions="Layer4"
    fieldname="sigma_x" fieldtype="Scalar" object="Element" table="SX" />
  <InitialConditionTable toregions="Layer4"
    fieldname="sigma_y" fieldtype="Scalar" object="Element" table="SY" />
  <ConstantInitialCondition toregions="Layer4"
    fieldname="sigma_z" fieldtype="Scalar" object="Element" value="-20e6" />
  <InitialConditionTable toregions="Layer5"
    fieldname="sigma_x" fieldtype="Scalar" object="Element" table="SX" />
  <InitialConditionTable toregions="Layer5"
    fieldname="sigma_y" fieldtype="Scalar" object="Element" table="SY" />
  <ConstantInitialCondition toregions="Layer5"
    fieldname="sigma_z" fieldtype="Scalar" object="Element" value="-20e6" />
</InitialConditions>

<BoundaryConditions>
  <BoundaryCondition object="Node" fieldname="Velocity"
```

```
      setnames="xneg" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="xpos" component="0" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="yneg" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="ypos" component="1" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zneg" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Node" fieldname="Velocity"
      setnames="zpos" component="2" scale="0.0" fieldtype="Vector" />
    <BoundaryCondition object="Face" fieldname="combinedFlowRate"
      setnames="source" scale="0.027" timetable="ttable1" />
  </BoundaryConditions>

  <Tables>

    <Table1D name="ttable1" coord="0.0, 10000, 10000.1" value="1.0, 1.0,    0.0" />
    <Table1D name="ttable2" coord="10000, 10000.1" value="0.0,    1.0" />
    <Table3D name="SX" x_file="x" y_file="y" z_file="z"
      voxel_file="SX" />
    <Table3D name="SY" x_file="x" y_file="y" z_file="z"
      voxel_file="SY" />

  </Tables>

  <Partition>
    <SpatialPartition xpar="8" ypar="4" zpar="5" />
  </Partition>

  <Fracture fractureFlag="40" preFractureSetName="perf"
    separableSet="fractureface" rockToughness="1e6" failCriterion="1"
    failgap="0.0004" />
  <Output writePlot="1" writeRestart="0" restart_interval="1000"
    plot_interval="120" plotfile_root="struorth" parallel_silo="12"
    slave_directory="sub" writeFEMFaces="1" writeFEMEdges="1" />

</Problem>
```