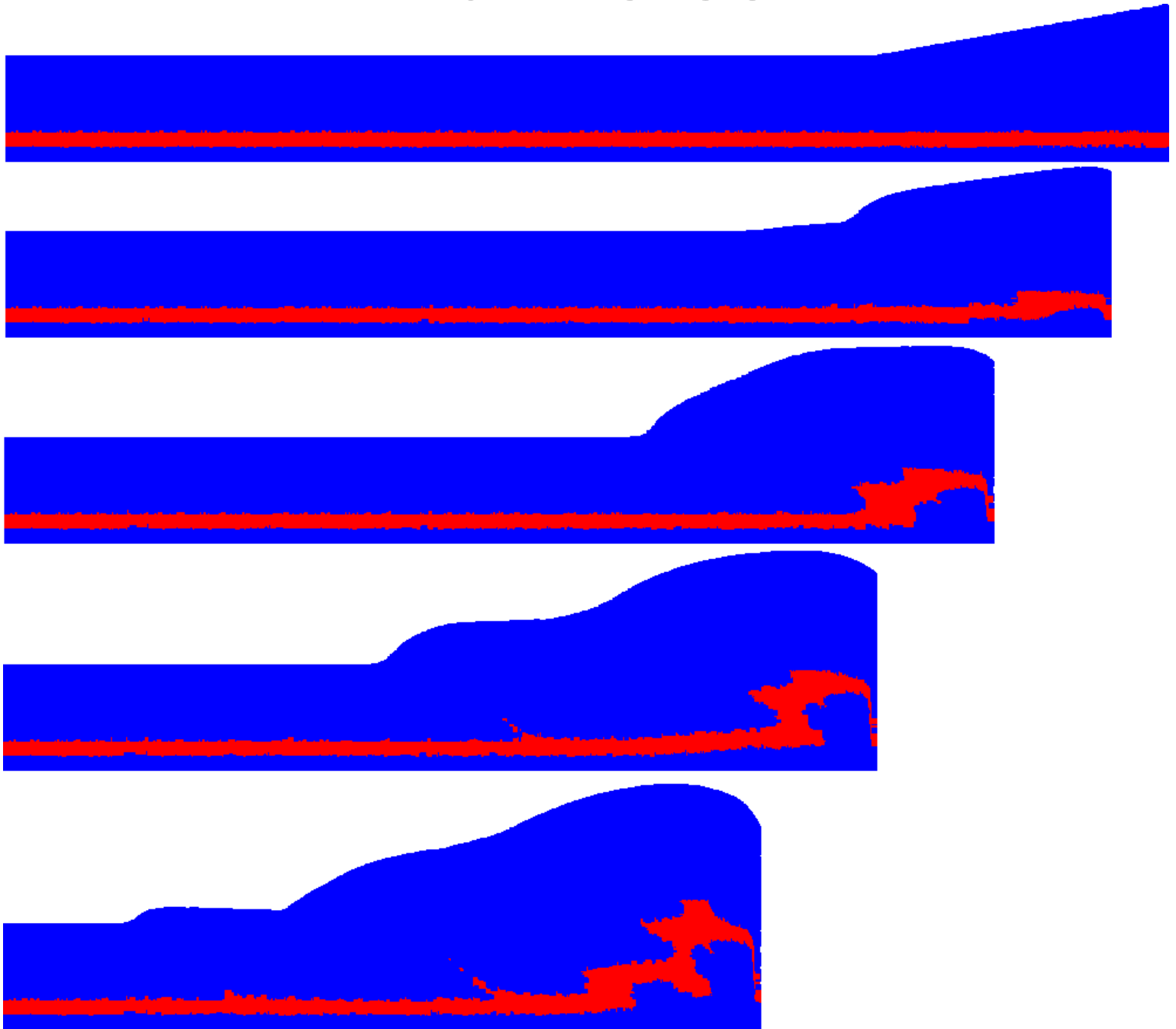
Gale

Walter Landry
Computational Infrastructure for Geodynamics
walter@geodynamics.org

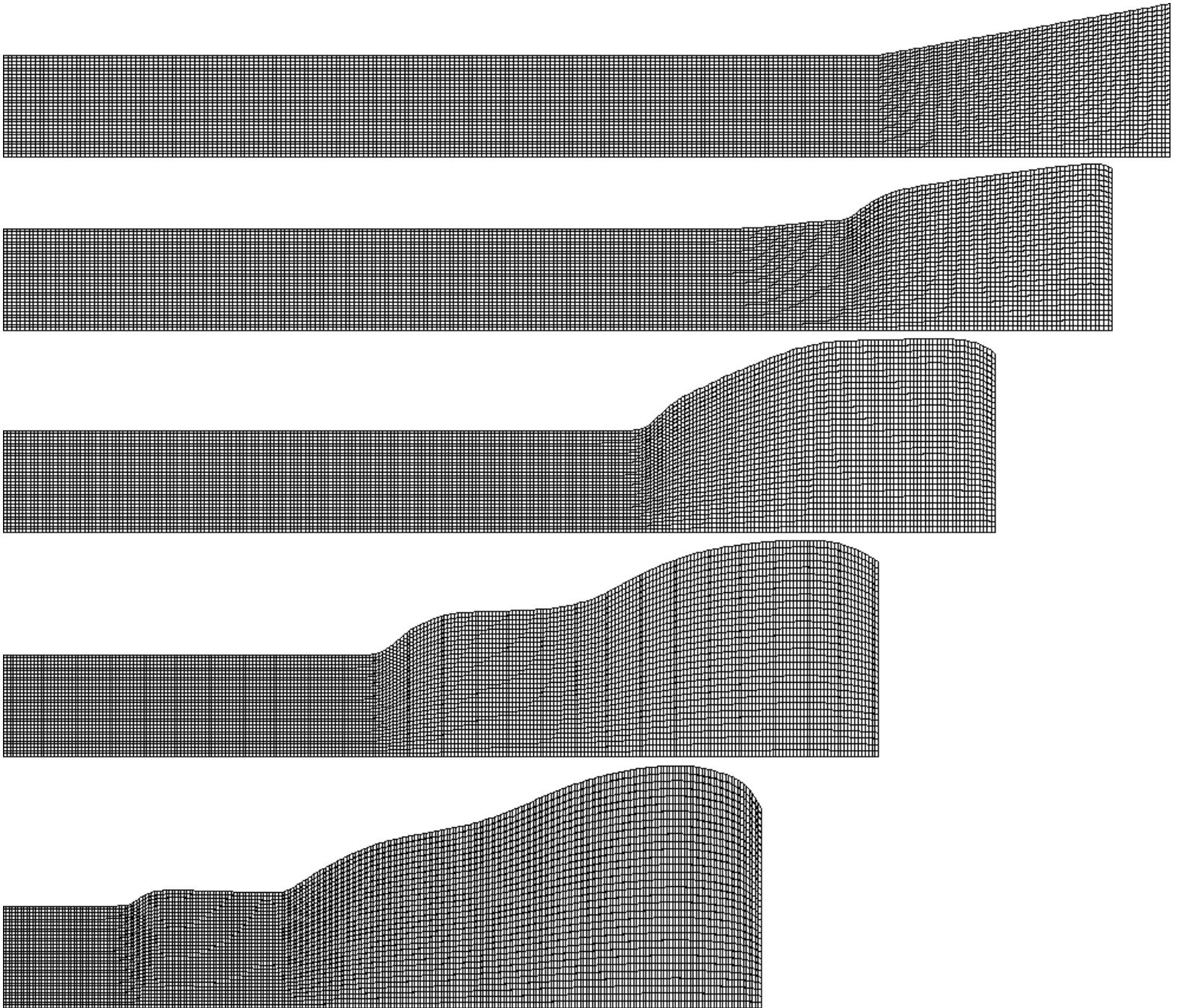
What is Gale

- A 2D/3D parallel code for Stokes and the energy equation
  - Accurately tracks material properties with particles
  - True free surface

Particles



True Free Surface



What is Gale

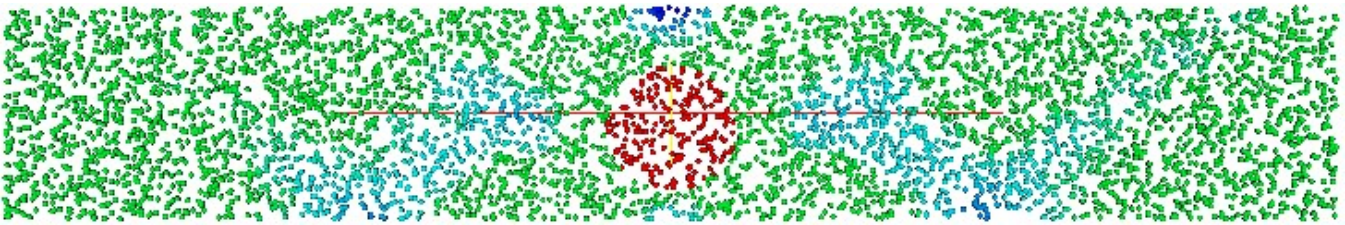
- Developed by CIG in response to a request from the long term tectonics community
- Built upon an existing code, Underworld, which was developed by the Victorian Partnership for Advanced Computing and Louis Moresi's group at Monash University

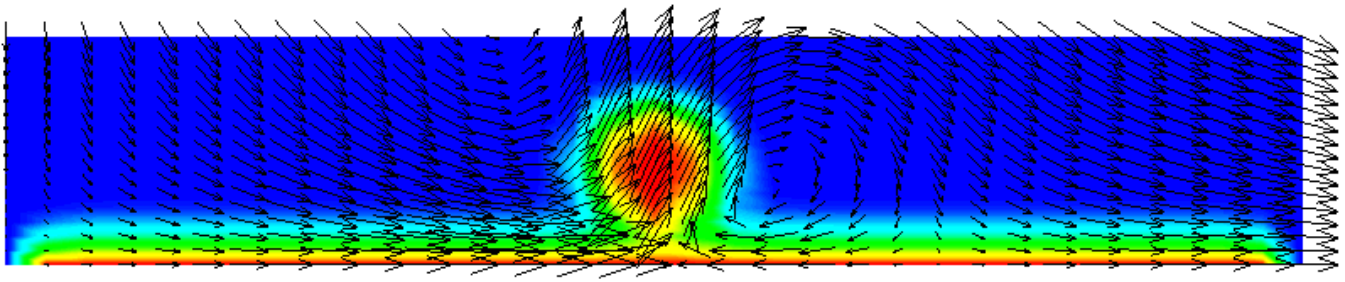
What is Gale

- Free Software
  - No paperwork
  - No license fees
  - No weird restrictions
    - Give it to your friends and enemies
    - Modify and redistribute. You can even try to compete with CIG (Good luck!)
    - Commercial and non-commercial use is fine
  - Just download and go

Extensive Documentation

- 100+ page manual
- 9 complete cookbook examples

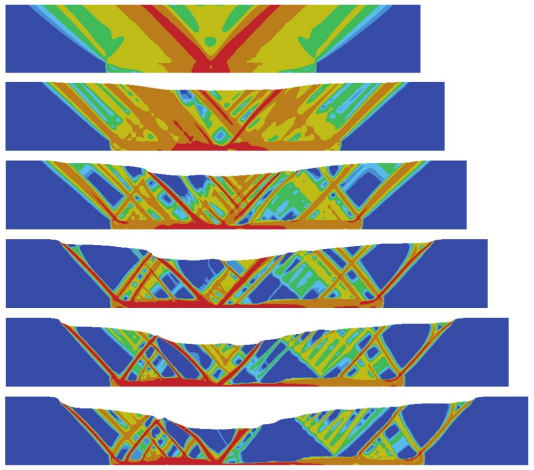




COMPUTATIONAL INFRASTRUCTURE FOR GEODYNAMICS (CIG)
VICTORIA PARTNERSHIP FOR ADVANCED COMPUTING (VPAC)
MONASH UNIVERSITY

Gale

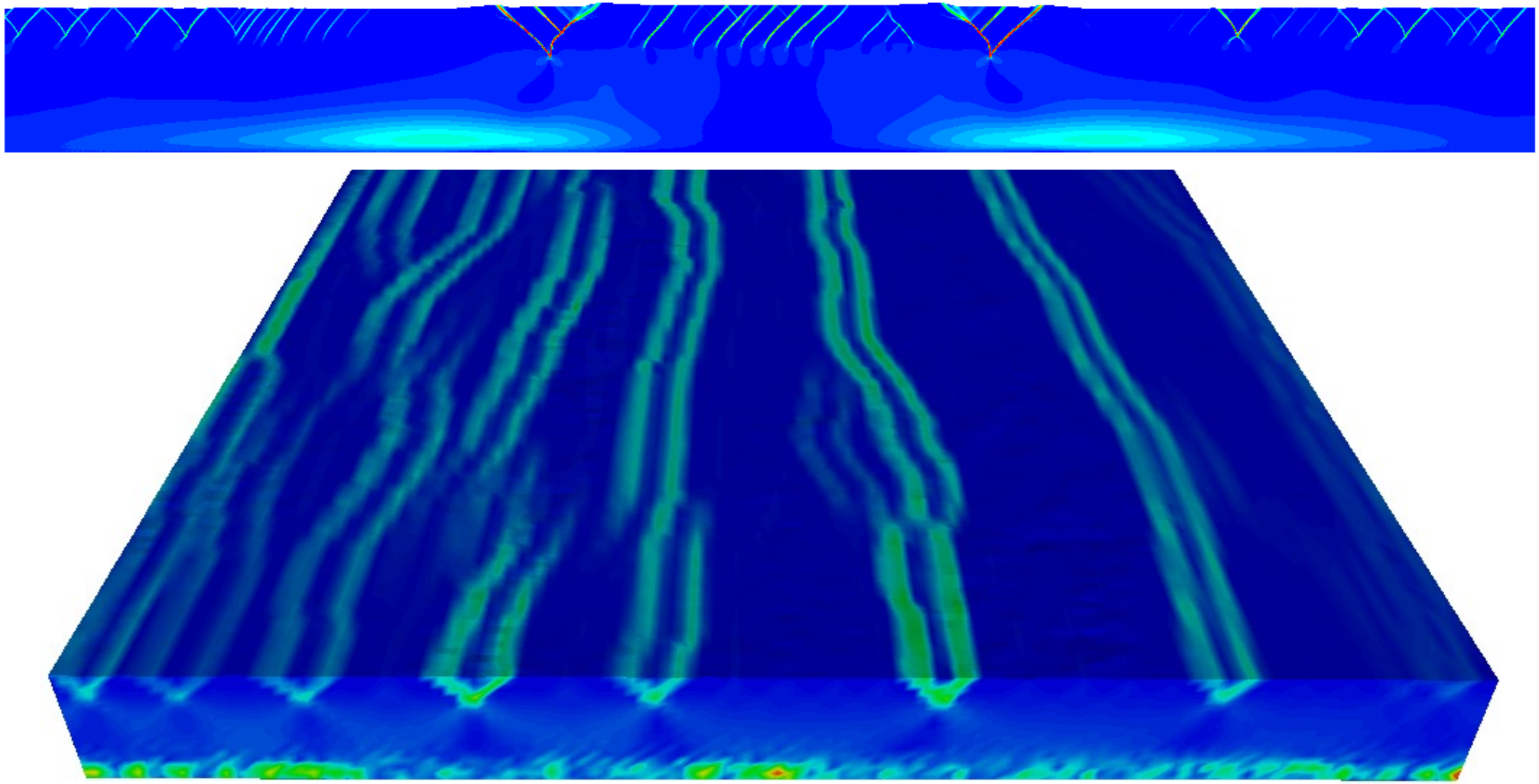
User Manual
Version 1.3



Walter Landry
Luke Hodkinson
Susan Kientz

www.geodynamics.org

Geologic Example



Binaries

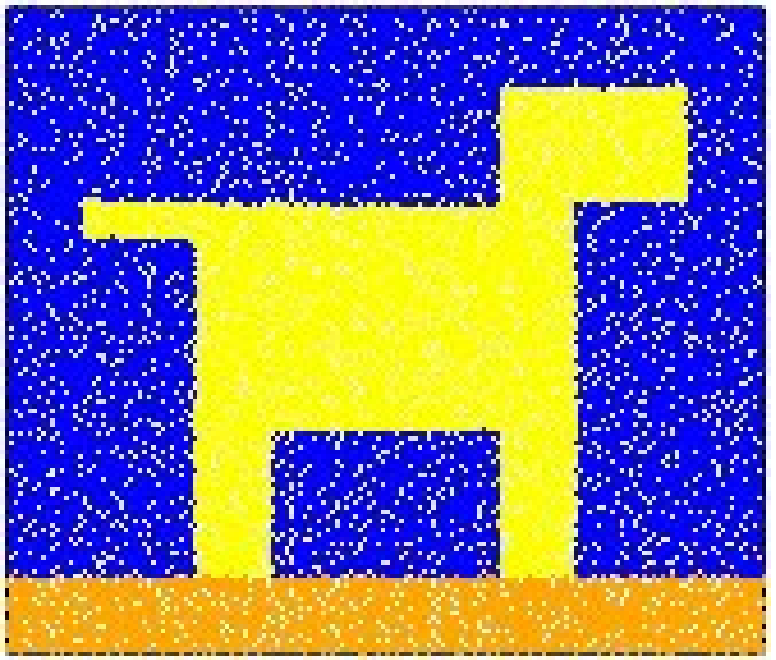
- Precompiled binaries are available for Linux x86, Linux AMD64, Mac PPC, Mac-Intel, Win32
- Gale is also installed on TACC Lonestar

Gale Capabilities

- Limited API for surface processes
- Many rheologies built in
  - Mohr-Coulomb
  - Drucker-Prager
  - Von Mises
  - Frank-Kamenetskii
  - Or make your own

Gale Capabilities

- Complex shapes which you mix and match just by changing the input file
  - Ellipsoid
  - Box
  - Half-plane
  - Polygon
  - Cylinder

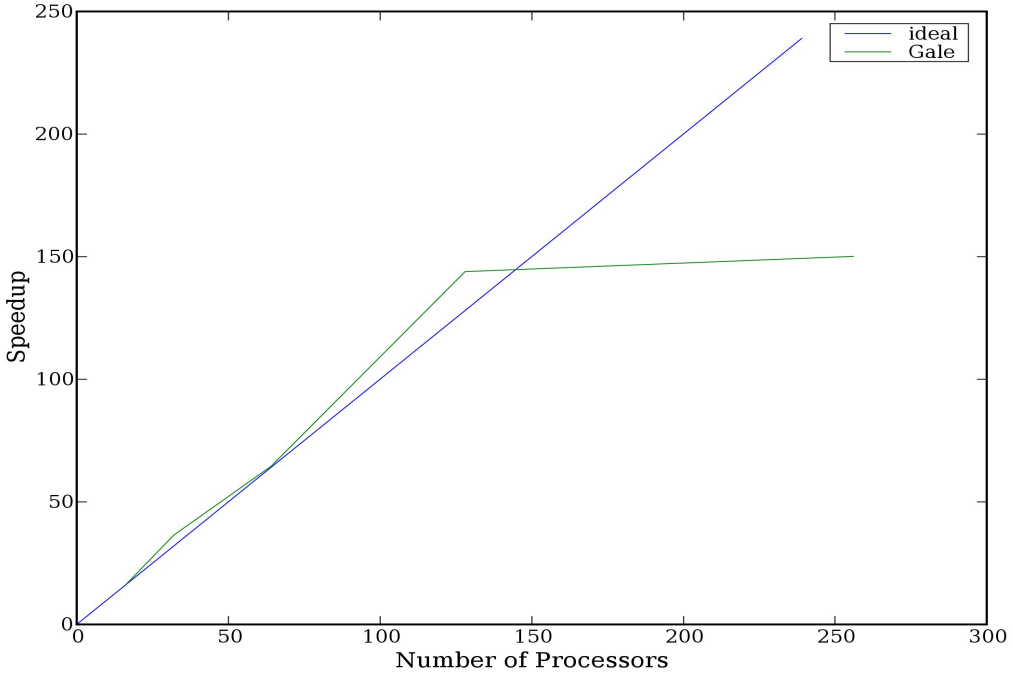


Gale Capabilities

- Many different boundary conditions
  - Fixed
  - Moving with a wide range of velocity profiles
  - Free slip
  - Stress: normal and tangential
  - Friction: static and kinetic
  - Inflow/Outflow

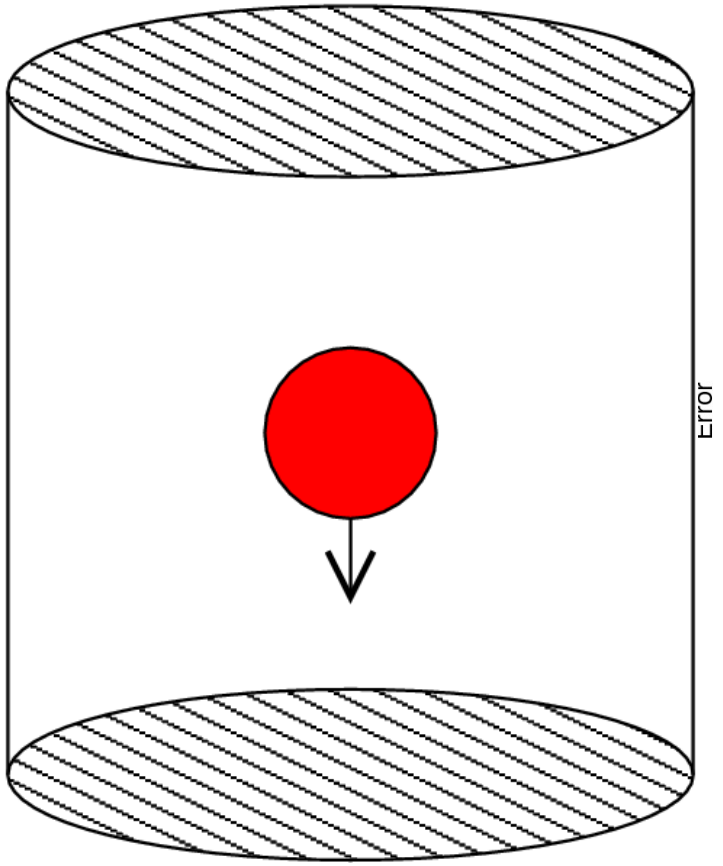
Gale Capabilities

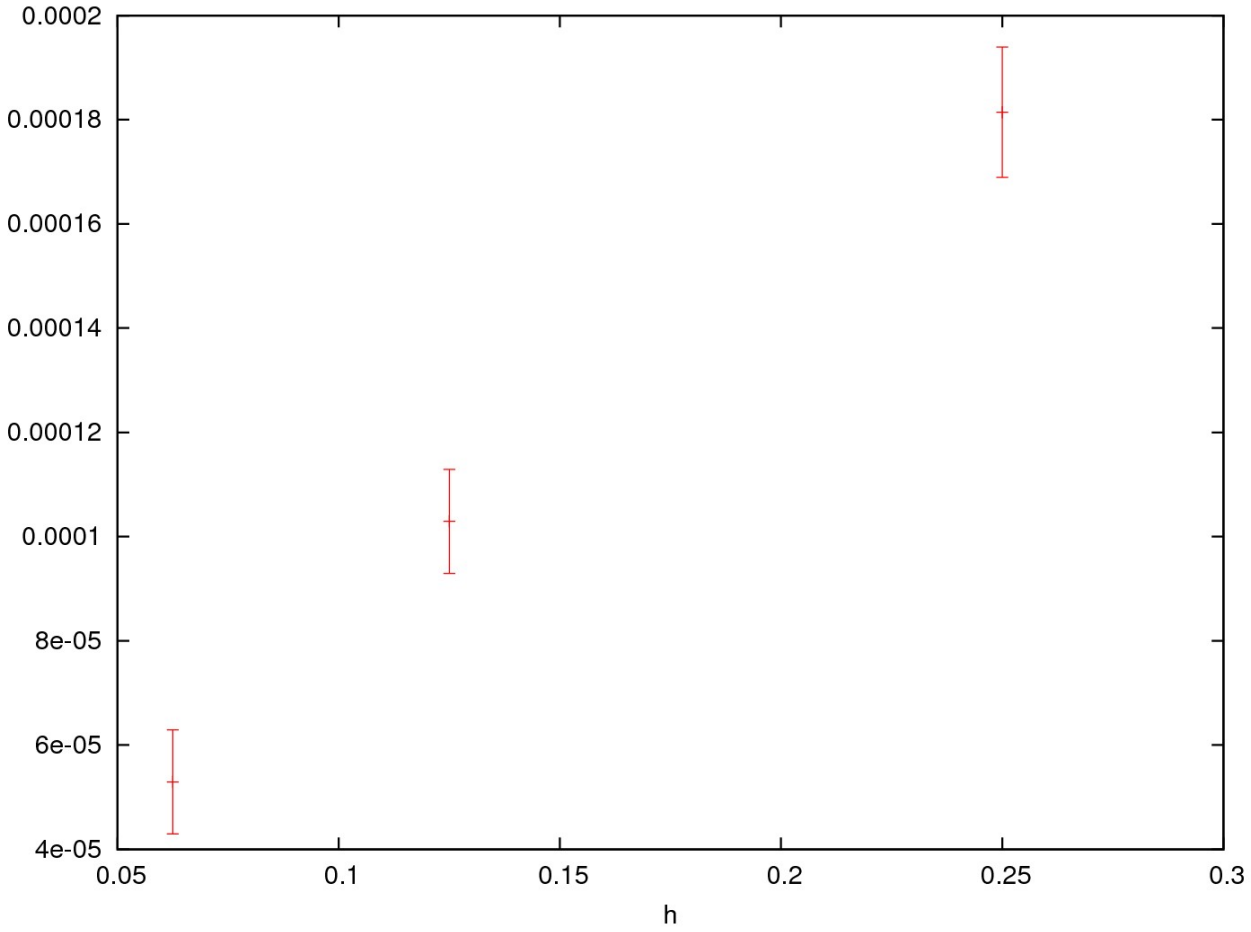
- Works in parallel
  - Linear scaling demonstrated from 16 to 128 processors for a small problem
  - No fundamental limit



Benchmarks

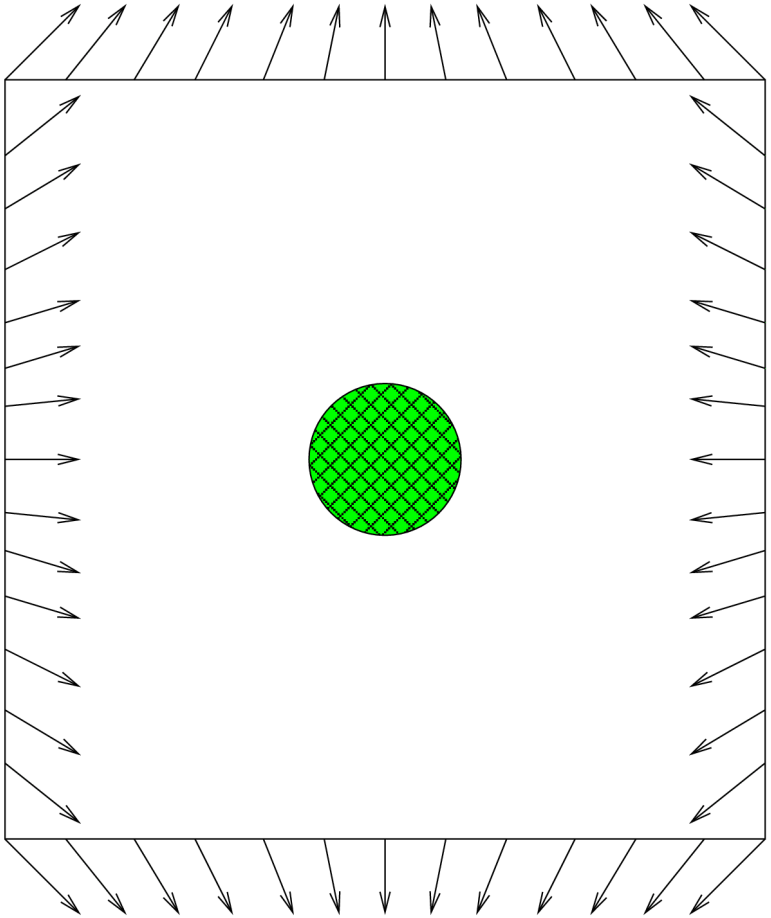
- Falling sphere in a cylinder

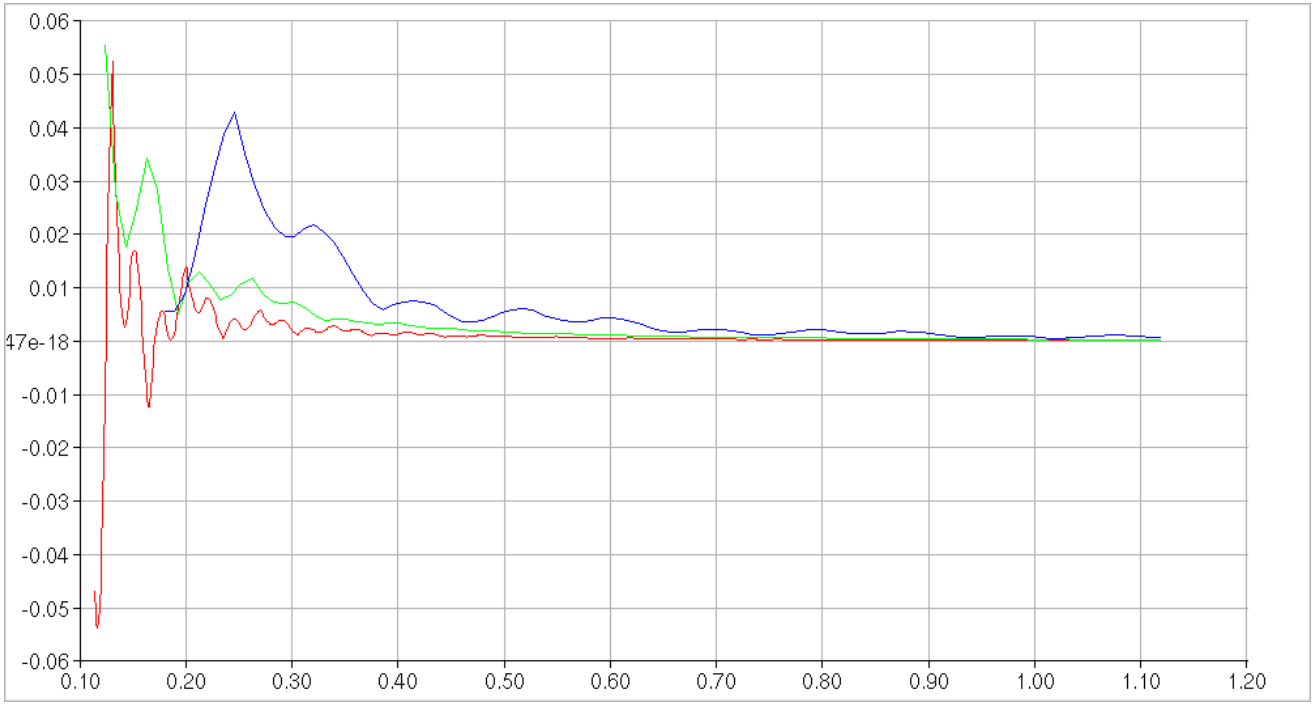




Benchmarks

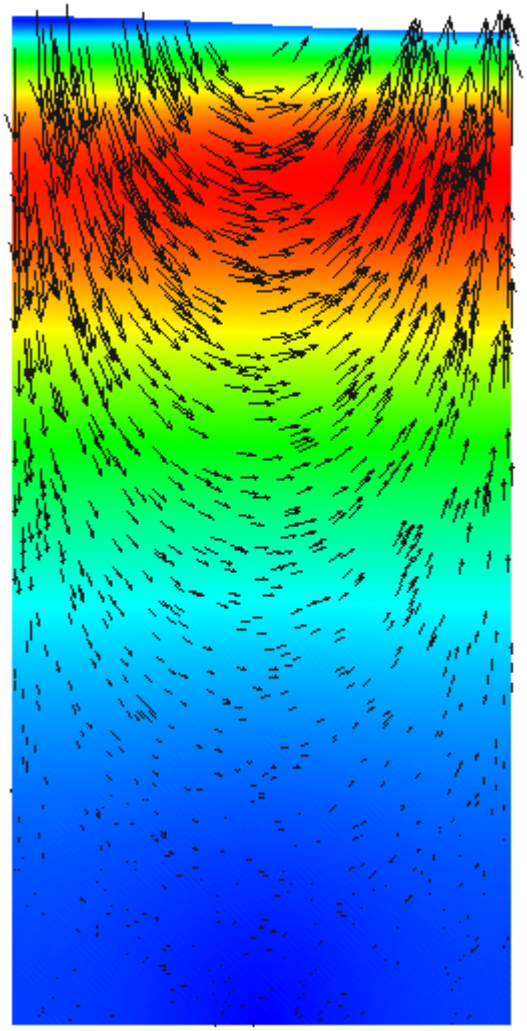
- Circular inclusion

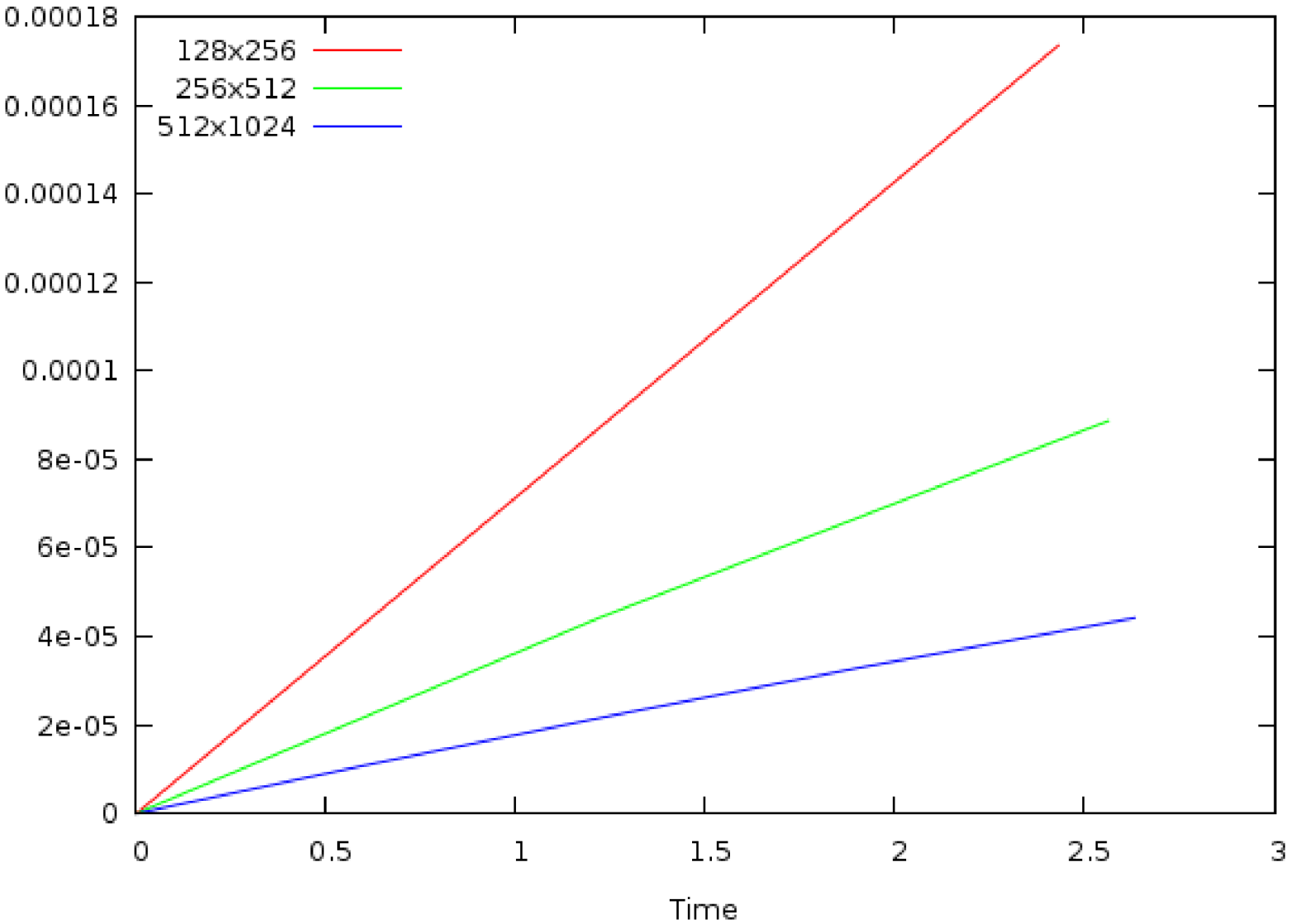




Benchmarks

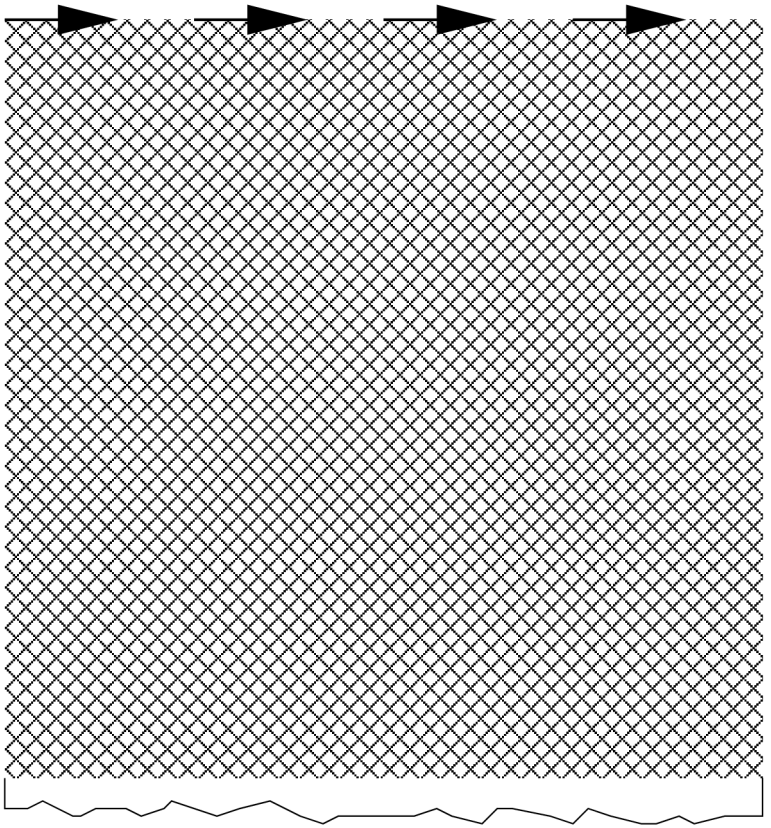
- Relaxation of topography





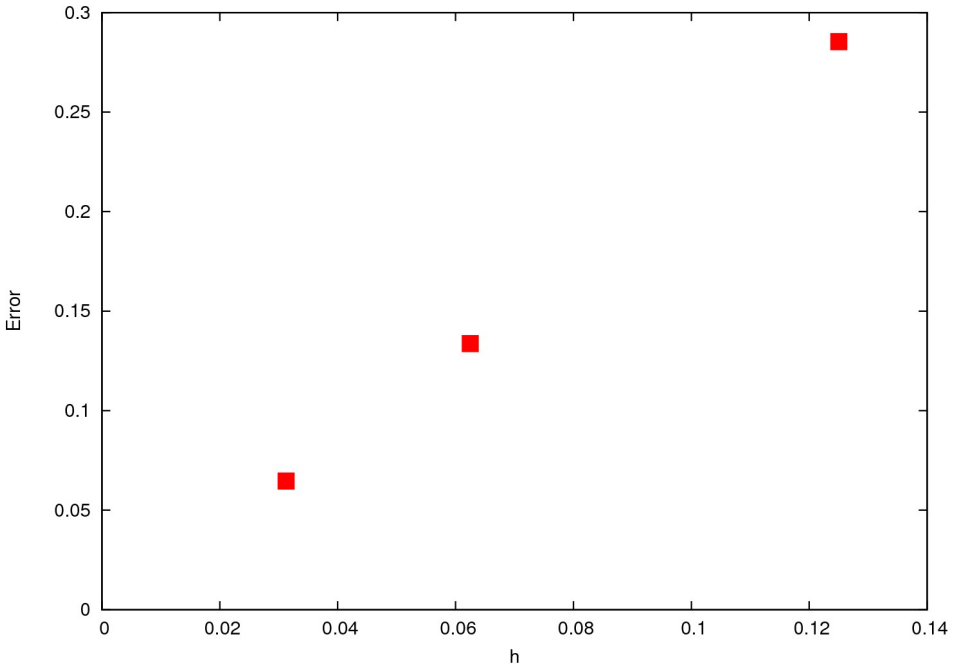
Benchmarks

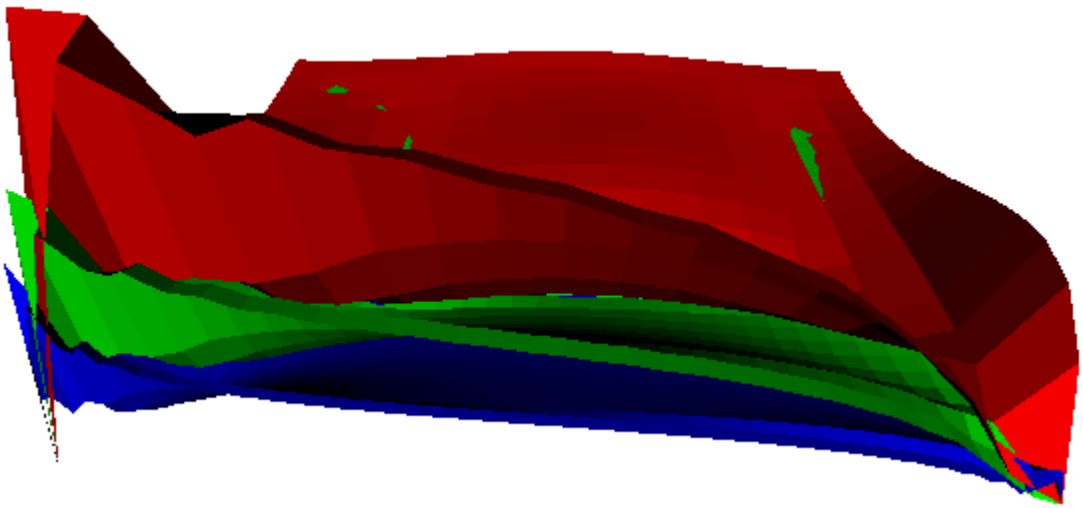
- Shear



Shear Benchmark

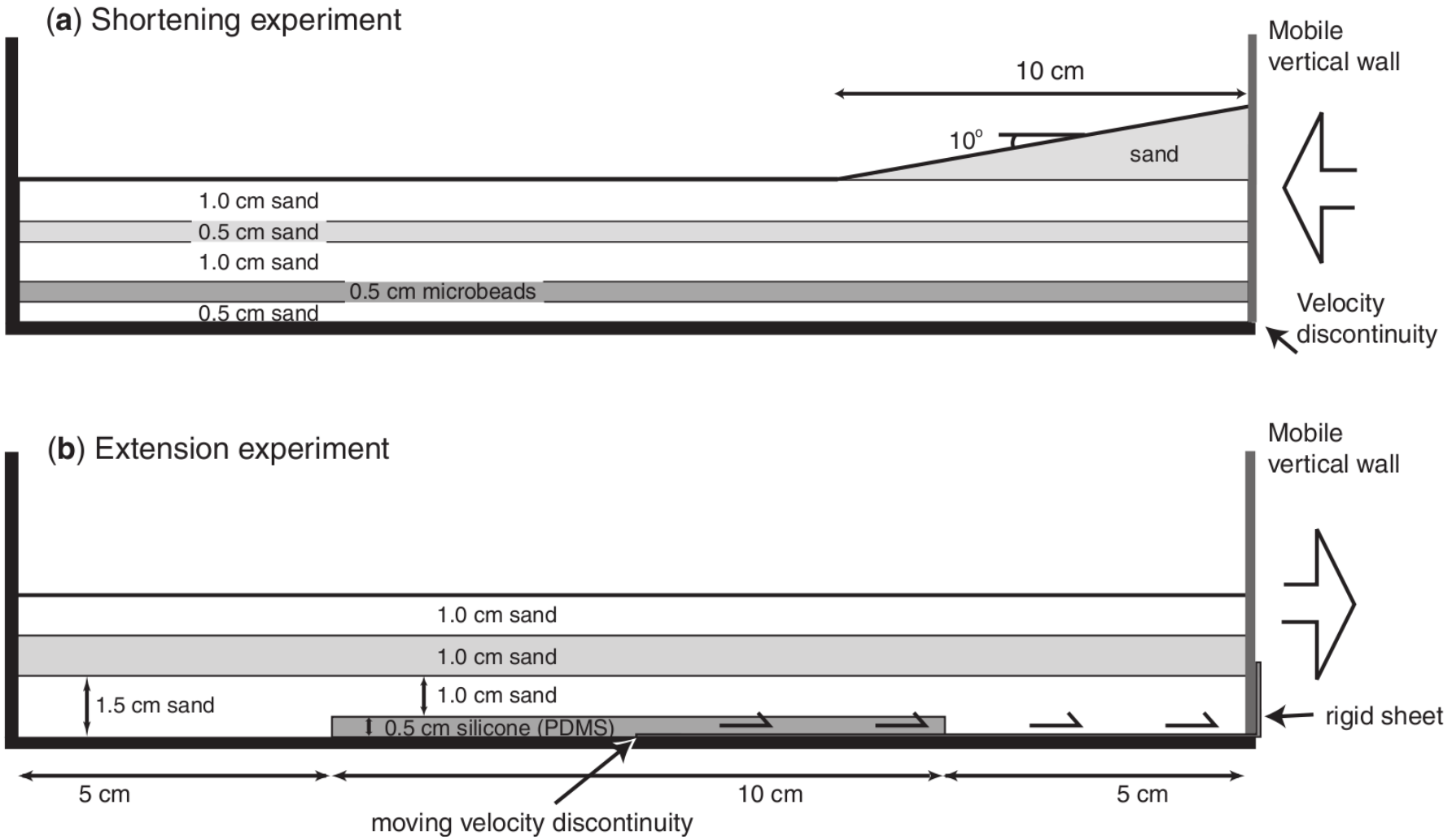
- Static Friction
- Kinetic Friction



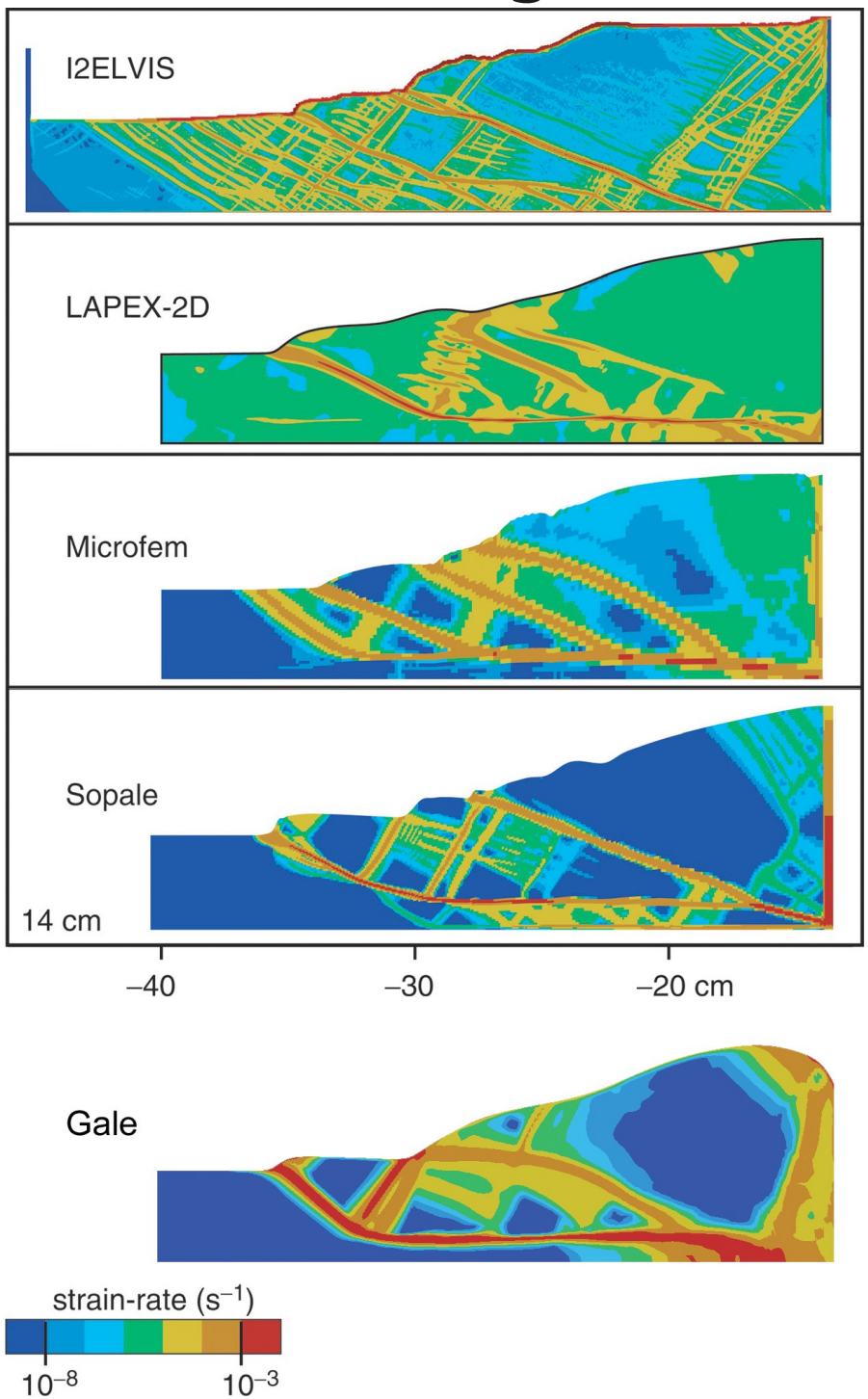


Benchmarks

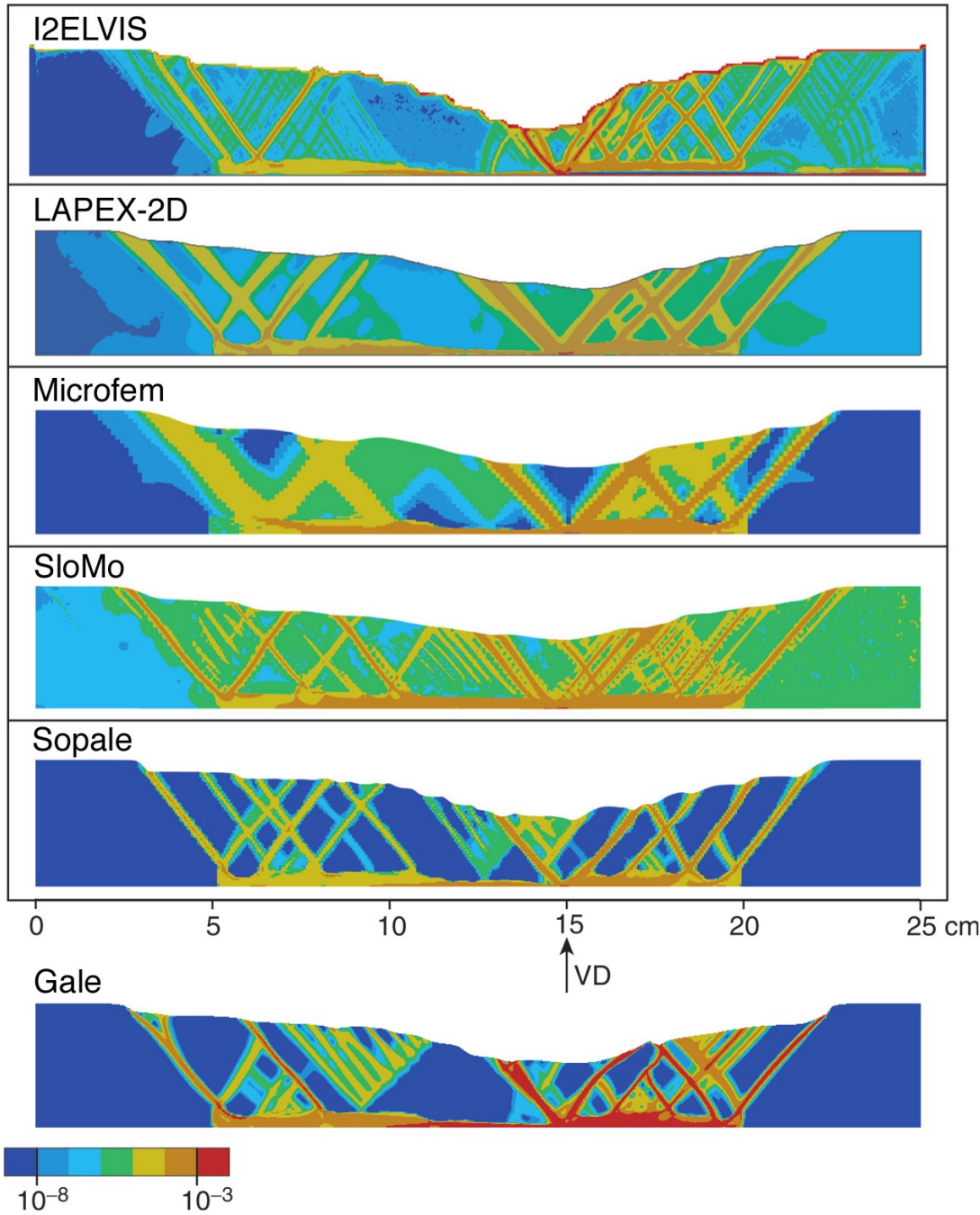
- Geomod 2004



- Shortening

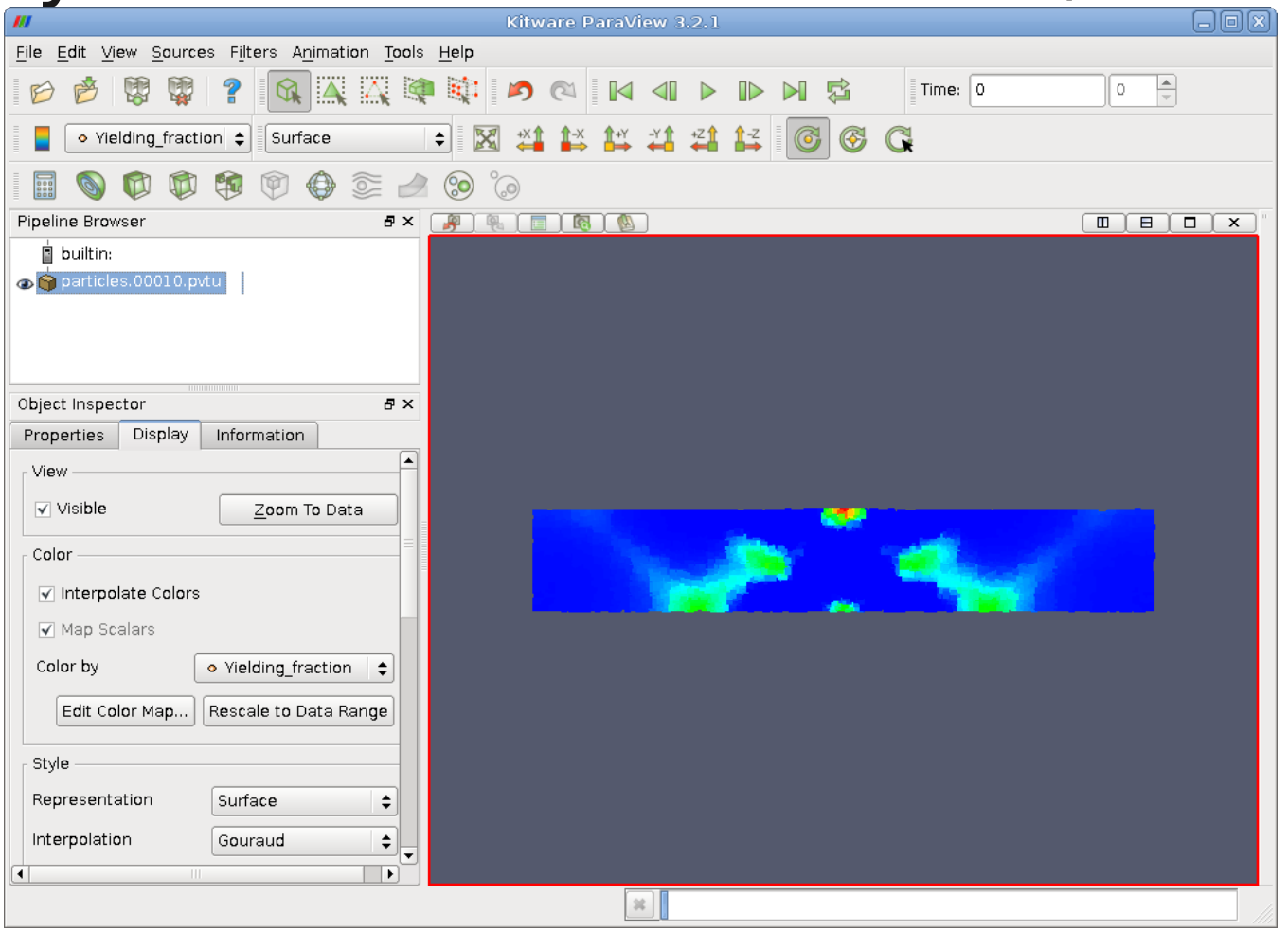


- Extension



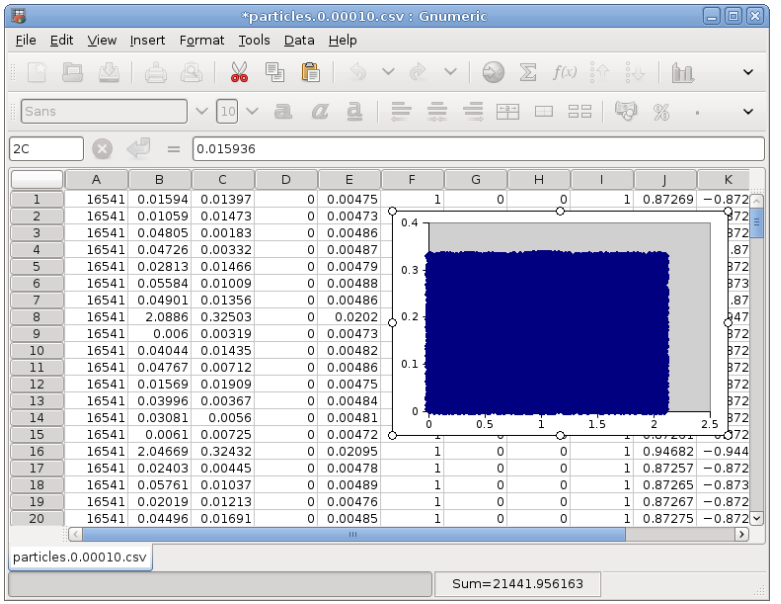
Visualization

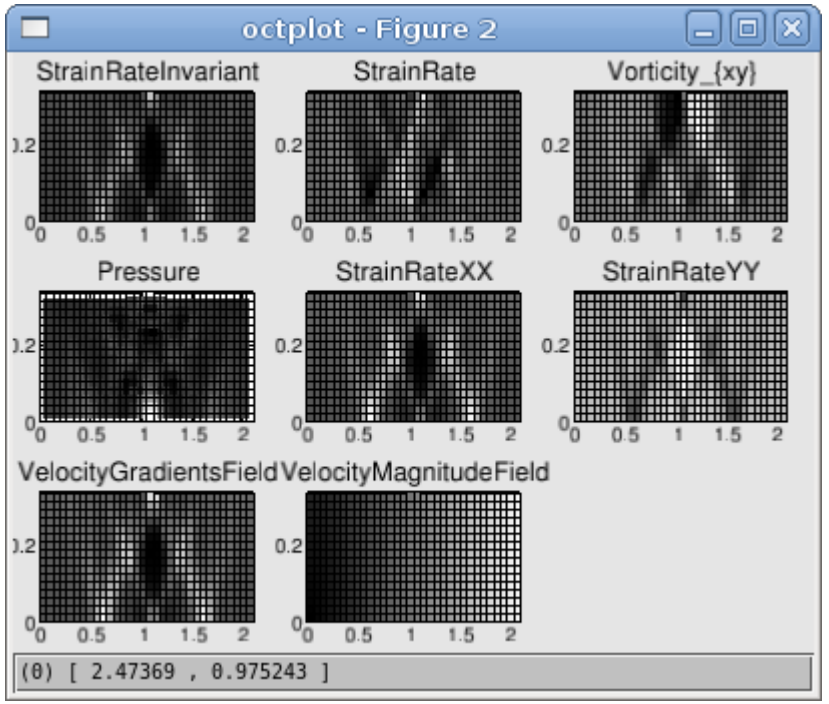
- Gale outputs in parallel VTK format, making it easy to visualize in Paraview, Mayavi, or Visit.

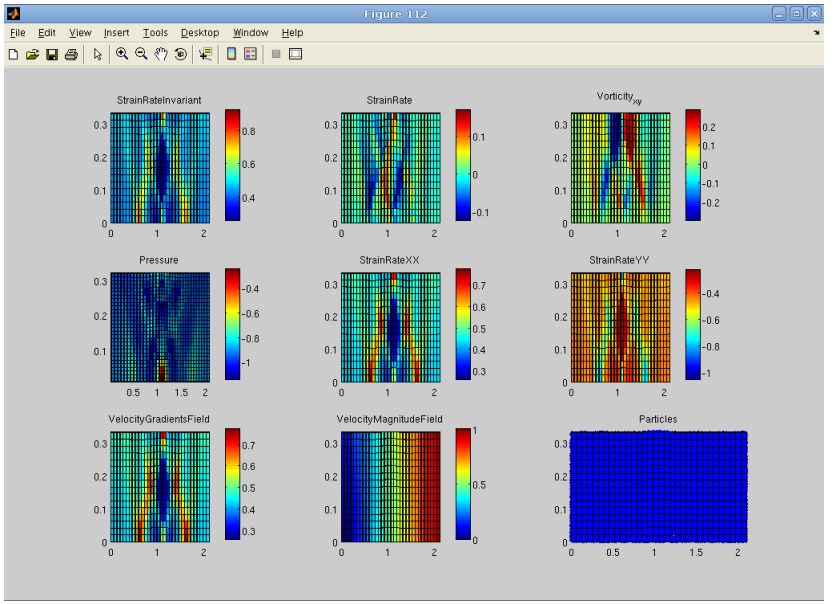


Visualization

- There are also scripts to visualize with Octave, Matlab, and Excel







Future Work

- Deformed lower boundaries
- Parallel surface processes
- Better normal stress boundary conditions
- Geomod 2008 benchmarks

Installation

- Easy: Use the binaries (even on TACC)
- Failing that, you will have to build it yourself.
- Gale requires
  - libxml2 (including development headers)
  - Petsc 2.3.2 (not 2.3.3)
  - mpi
- Generally, Petsc is the hardest part

Petsc Installation

- Get the 2.3.2 source tarball from
  - http://www-unix.mcs.anl.gov/petsc/ petsc-as/download/index.html
- Unpack it
  - tar -zxf petsc-2.3.2-p10.tar.gz
- cd petsc-2.3.2-p10

Petsc Configuration

- Try configuring it
  - config/configure.py
- If this works, you will have to set two environment variables
  - PETSC_ARCH
  - PETSC_DIR
- Then type “make all”

Petsc Troubleshooting

- Unfortunately, this often does not work.
- If configuration seems to hang forever, the problem could be that you have to submit all mpi jobs to the queue.
- Add the option –-with-batch=1
  - config/configure.py --with-batch=1

Petsc Troubleshooting

- Then petsc will ask you to submit a script “conftest” to the batch system.
- In practice, I have always been able to just run the script manually
- Then run “python reconfigure”

Petsc Troubleshooting

- Your mpi compiler may be in a non standard location
  - --with-mpi-dir=/opt/mpich/myrinet/intel
- Fortran binding may cause problems
  - --fc=0

Petsc Troubleshooting

- Petsc can not find a blas or lapack library.
  - If you are using the Intel compiler, then find out where mkl is and add the option --with-blas-lapack-dir=/path/to/mkl
  - Otherwise, use the option --download-c-blas-lapack=yes

Petsc Troubleshooting

- You can get a complete list of options to configure
  - config/configure.py --help

Petsc Optimized and Debug

- Petsc, by default, creates libraries with debugging information.
- To turn on optimization, add the option --with-debugging=0

Gale Installation

- Gale uses the same configuration mechanism as Petsc.
- So you will usually copy the compiler information from Petsc to Gale
  - PETSC: config/configure.py –with-mpi-dir=/opt/intel
  - GALE: ./configure --with-mpi-dir=/opt/intel

Gale Installation

- You do not have to copy any information about blas and lapack
- If you set the environment variable PETSC_ARCH and PETSC_DIR, then Gale will find and use that automatically.
- Otherwise, use the –with-petsc-arch and –with-petsc-dir options.

Gale Installation

- Once you are done configuring, type “make” and then “make install”.
- The executable will be in bin/Gale, and will have the same optimization/debug options as Petsc.

Running Serial Jobs

- Gale always needs an input file
  - ./Gale input/cookbook/yielding.xml
- If you do not give an input file, you will get an error
  - Error in _AbstractContext_New: The dictionary is empty, meaning no input parameters have been feed into your program. Perhaps you've forgot to pass any input files (or command-line arguments) in. Gale-debug: build/StGermain/Base/IO/src/Journal.c:603: Journal_Firewall: Assertion `expression' failed. p0_12616: p4_error: interrupt SIGx: 6

Running Serial Jobs

- If you give an input file that does not exist, you will get a different error
  - Error: File foo.xml doesn't exist, not readable, or not valid. Gale-debug: build/StGermain/Base/IO/src/Journal.c:603: Journal_Firewall: Assertion `expression' failed. p0_13533: p4_error: interrupt SIGx: 6

Running Serial Jobs

- Some MPI implementations change the current directory, so that Gale will not be able to find your input file.
- The workaround is to give the complete name of the input file
  - ./Gale /home/walter/gale/input/cookbook/yielding.xml

Running Serial Jobs

- Gale also accepts Petsc's command line options.
- For example, to use a direct LU solve instead of GMRES (Often much faster)
  - ./Gale input/extension.xml -pc_type lu -ksp_type preonly

Running Serial Jobs

- You can also override any parameter in the input file from the command line
  - ./Gale extension.xml –maxTimeSteps=10

Running Parallel Jobs

- If you have your own multi-cpu machine, then to use more than one processor, prepend mpirun or mpiexec to the whole command
  - mpirun -np 2 bin/Gale /home/walter/gale/input/extension.xml

Running Parallel Jobs

- On a shared facility such as the teragrid, you will have to submit a job to the queue.
- This varies from site to site. The site administrators should provide a sample script.

Using a Parallel Direct Solver

- If you have everything working, you can get a parallel direct solver working by configuring Petsc with
  --download-mumps=yes
  --download-blacs=yes
  --download-scalapack=yes
  --with-blacs=yes
  --with-scalapack=yes
- Then add to the command line
  -mat_type aijmumps -ksp_type preonly
  -pc_type lu

Writing Input Files

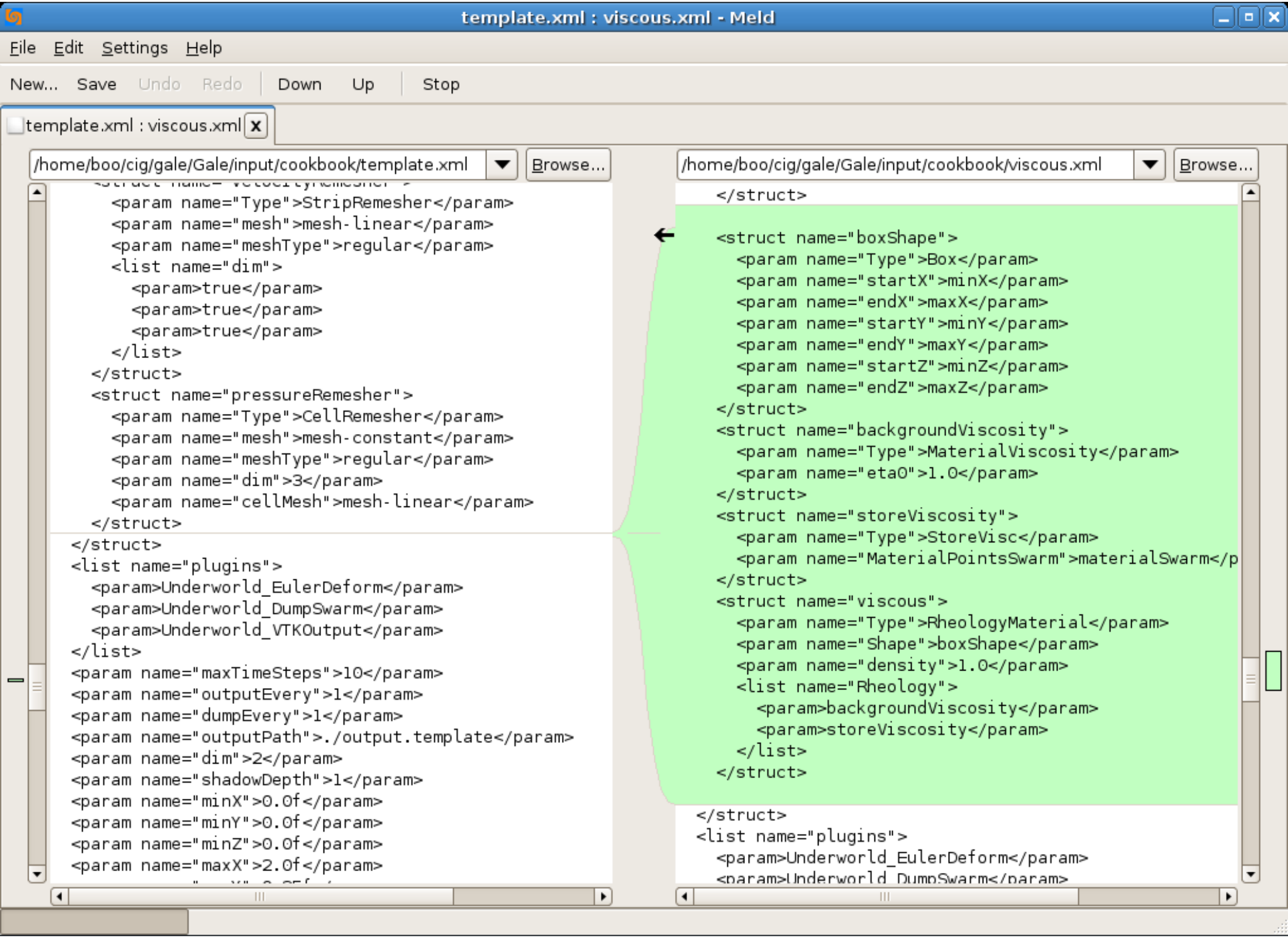
- Gale is based upon a generic library for scientific computing: StGermain
- So you must explicitly specify a number of things which are normally implicit (e.g. we are solving a Stokes flow problem)
- The input format (XML) is similarly generic and (unfortunately) wordy.

Writing Input Files

- So the input files tend to be rather large.
- Most of the parameters will not change. So you only need to copy in a template
  - input/cookbook/template.xml

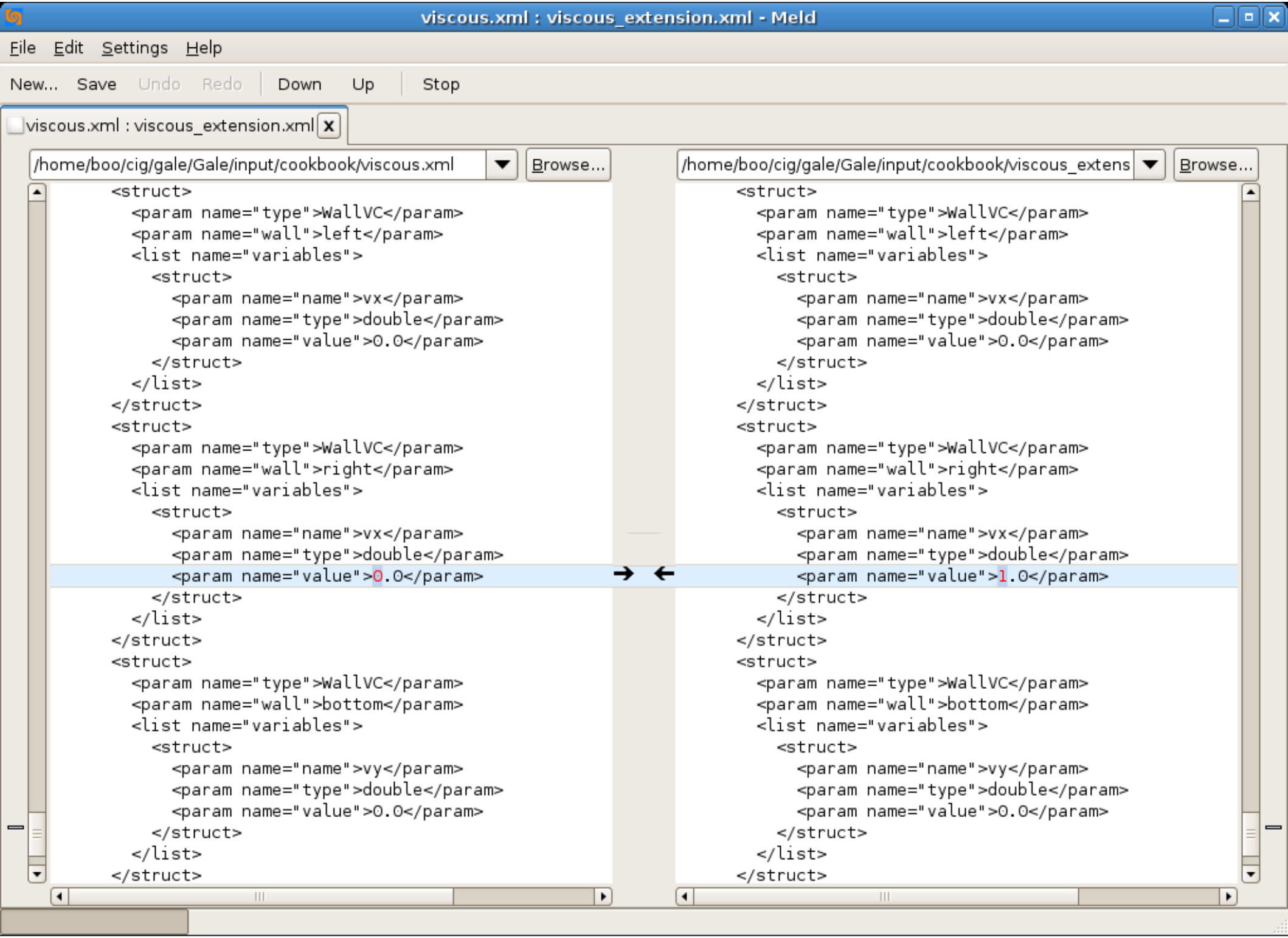
Writing Input Files

- The manual leads you through the modification of this template to create different models.
- We will start with the first example, where we add a viscous material to the template.


template.xml : viscous.xml - Meld
File Edit Settings Help
New... Save Undo Redo Down Up Stop
template.xml : viscous.xml
/home/boo/cig/gale/Gale/input/cookbook/template.xml
Browse...
/home/boo/cig/gale/Gale/input/cookbook/viscous.xml
Browse...
<param name="Type">StripRemesher</param>
<param name="mesh">mesh-linear</param>
<param name="meshType">regular</param>
<list name="dim">
<param>true</param>
<param>true</param>
<param>true</param>
</list>
</struct>
<struct name="pressureRemesher">
<param name="Type">CellRemesher</param>
<param name="mesh">mesh-constant</param>
<param name="meshType">regular</param>
<param name="dim">3</param>
<param name="cellMesh">mesh-linear</param>
</struct>
</struct>
<list name="plugins">
<param>Underworld_EulerDeform</param>
<param>Underworld_DumpSwarm</param>
<param>Underworld_VTKOutput</param>
</list>
<param name="maxTimeSteps">10</param>
<param name="outputEvery">1</param>
<param name="dumpEvery">1</param>
<param name="outputPath">./output.template</param>
<param name="dim">2</param>
<param name="shadowDepth">1</param>
<param name="minX">0.0f</param>
<param name="minY">0.0f</param>
<param name="minZ">0.0f</param>
<param name="maxX">2.0f</param>
</struct>
<struct name="boxShape">
<param name="Type">Box</param>
<param name="startX">minX</param>
<param name="endX">maxX</param>
<param name="startY">minY</param>
<param name="endY">maxY</param>
<param name="startZ">minZ</param>
<param name="endZ">maxZ</param>
</struct>
<struct name="backgroundViscosity">
<param name="Type">MaterialViscosity</param>
<param name="eta0">1.0</param>
</struct>
<struct name="storeViscosity">
<param name="Type">StoreVisc</param>
<param name="MaterialPointsSwarm">materialSwarm</p
</struct>
<struct name="viscous">
<param name="Type">RheologyMaterial</param>
<param name="Shape">boxShape</param>
<param name="density">1.0</param>
<list name="Rheology">
<param>backgroundViscosity</param>
<param>storeViscosity</param>
</list>
</struct>
</struct>
<list name="plugins">
<param>Underworld_EulerDeform</param>
<param>Underworld_DumpSwarm</param>

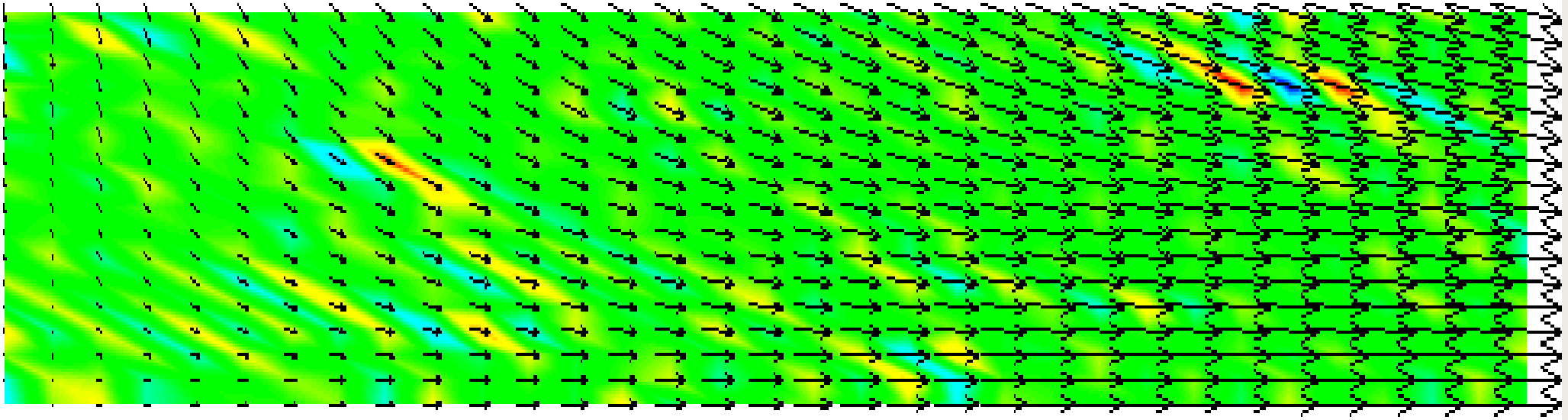
Writing Input Files

- Next, we add a moving boundary


viscous.xml : viscous_extension.xml - Meld
File Edit Settings Help
New... Save Undo Redo Down Up Stop
viscous.xml : viscous_extension.xml
/home/boo/cig/gale/Gale/input/cookbook/viscous.xml
Browse...
/home/boo/cig/gale/Gale/input/cookbook/viscous_extens
Browse...
<struct>
<param name="type">WallVC</param>
<param name="wall">left</param>
<list name="variables">
<struct>
<param name="name">vx</param>
<param name="type">double</param>
<param name="value">0.0</param>
</struct>
</list>
</struct>
<struct>
<param name="type">WallVC</param>
<param name="wall">right</param>
<list name="variables">
<struct>
<param name="name">vx</param>
<param name="type">double</param>
<param name="value">0.0</param>
</struct>
</list>
</struct>
<struct>
<param name="type">WallVC</param>
<param name="wall">bottom</param>
<list name="variables">
<struct>
<param name="name">vy</param>
<param name="type">double</param>
<param name="value">0.0</param>
</struct>
</list>
</struct>
<param name="value">1.0</param>

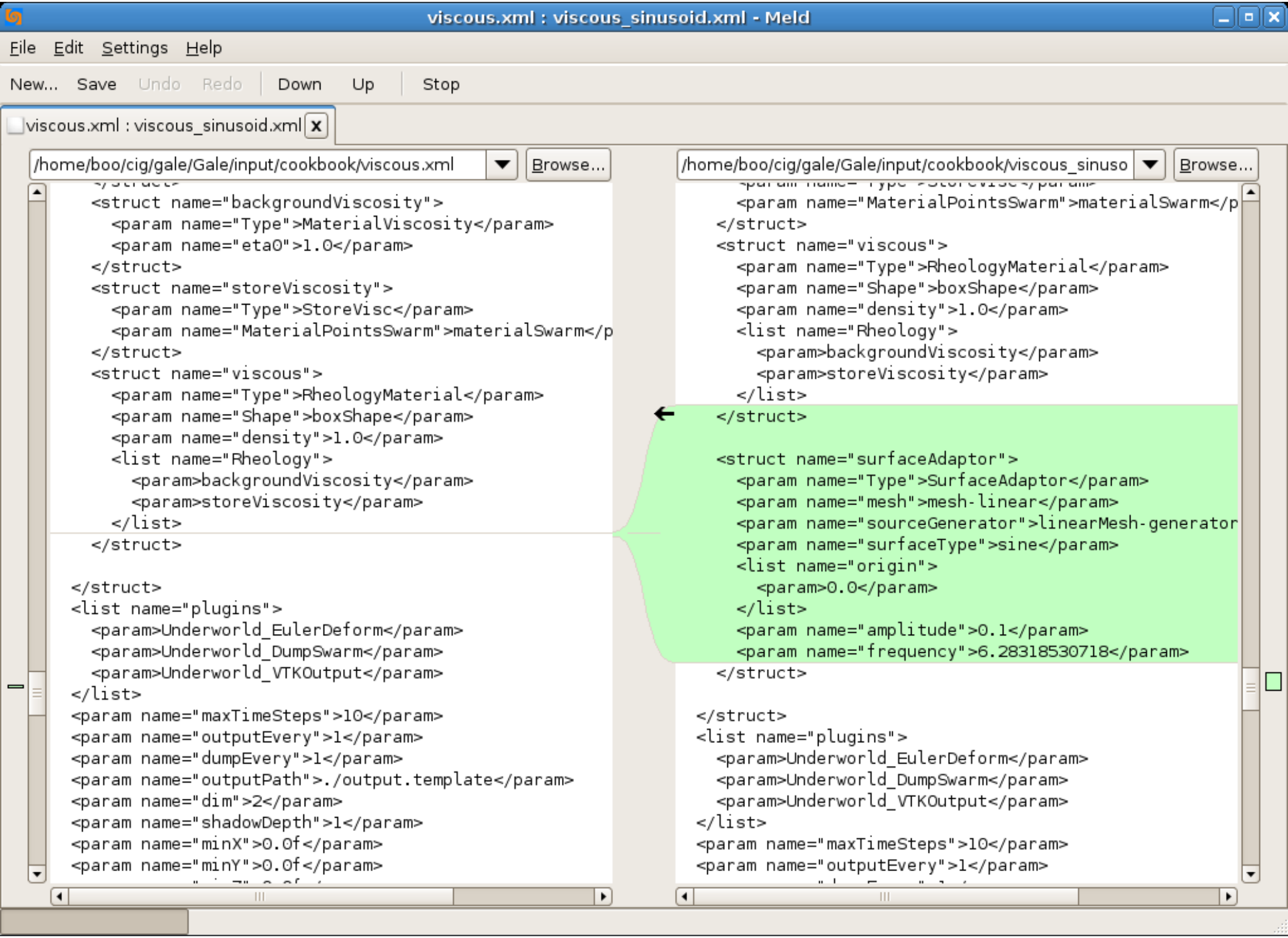
Visualization

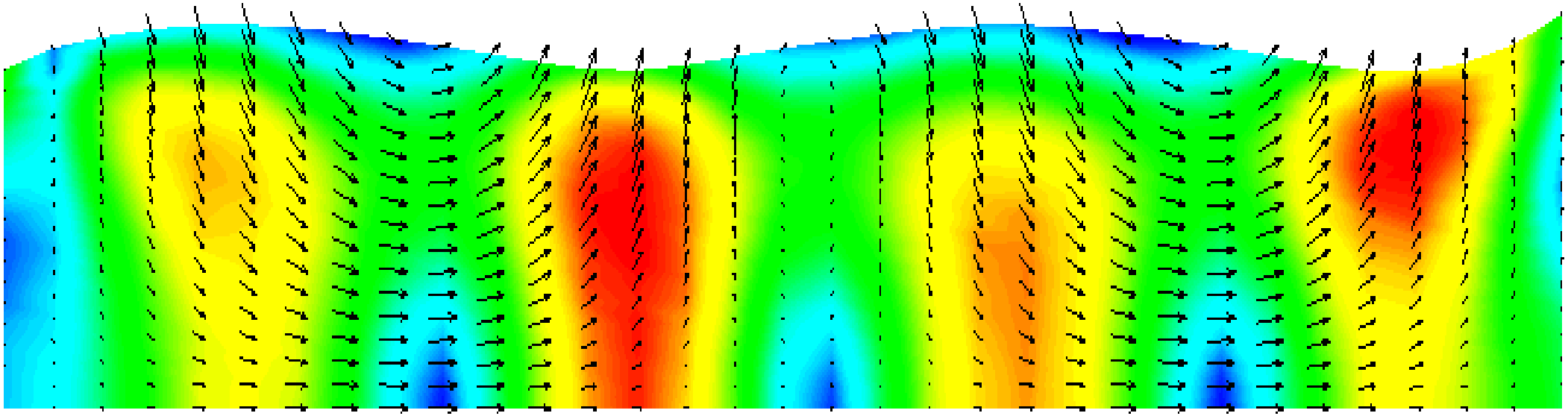
- We can then visualize the results in Paraview

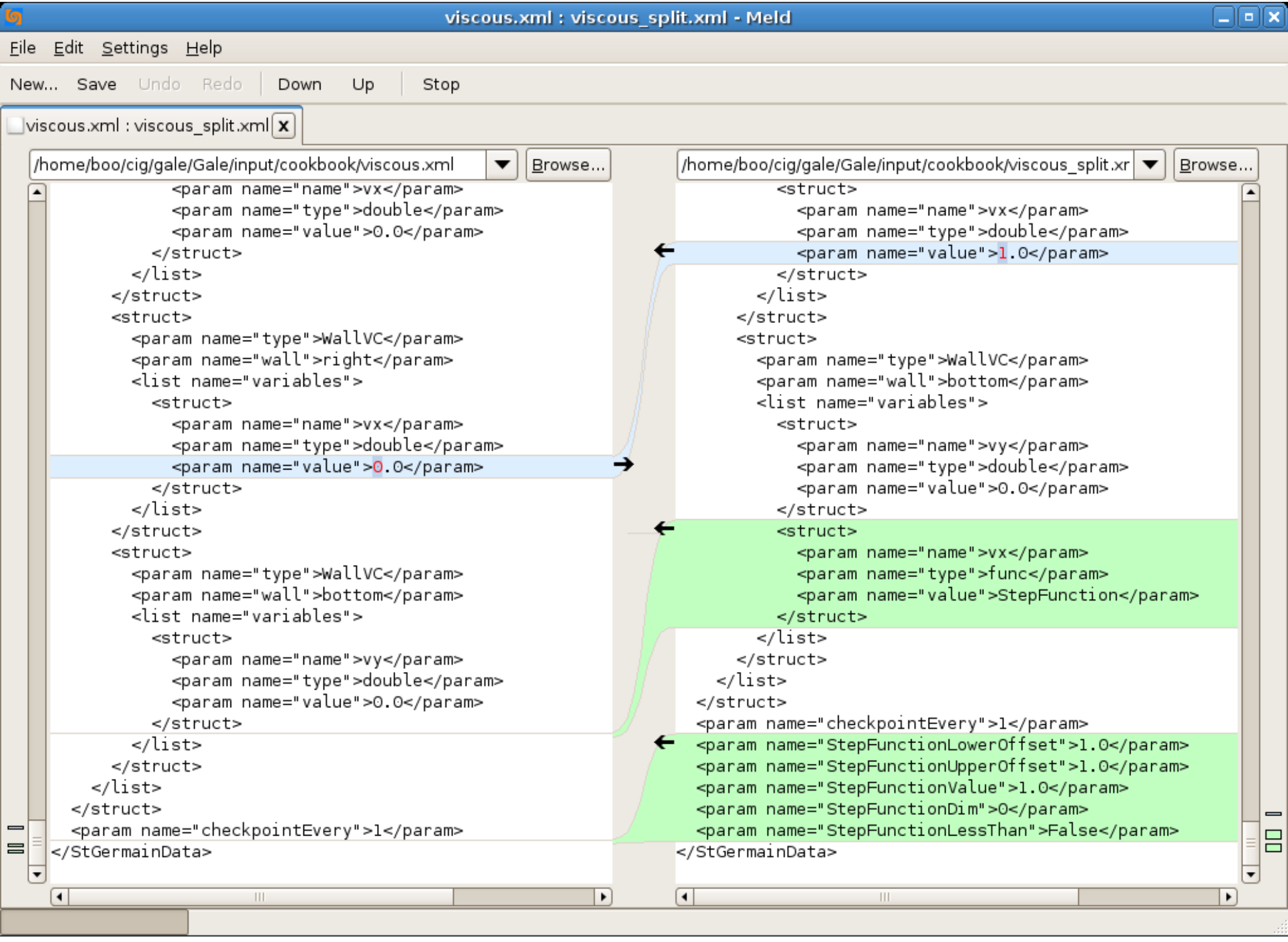


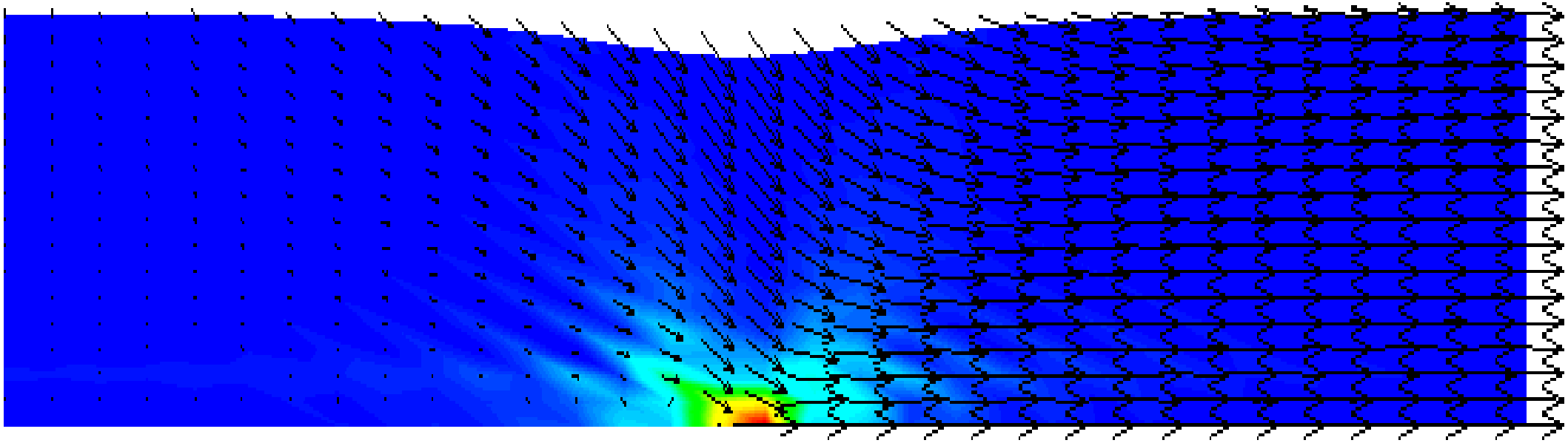
Visualization

- The banding is a numerical artifact. If we increase the number of particles per cell, then the effect decreases.
- For the other simulations, you will not see this banding, even though we have not changed the number of particles per cell.


viscous.xml : viscous_sinusoid.xml - Meld
File Edit Settings Help
New... Save Undo Redo Down Up Stop
viscous.xml : viscous_sinusoid.xml
/home/boo/cig/gale/Gale/input/cookbook/viscous.xml
Browse...
/home/boo/cig/gale/Gale/input/cookbook/viscous_sinuso
Browse...
<struct name="backgroundViscosity">
<param name="Type">MaterialViscosity</param>
<param name="eta0">1.0</param>
</struct>
<struct name="storeViscosity">
<param name="Type">StoreVisc</param>
<param name="MaterialPointsSwarm">materialSwarm</p
</struct>
<struct name="viscous">
<param name="Type">RheologyMaterial</param>
<param name="Shape">boxShape</param>
<param name="density">1.0</param>
<list name="Rheology">
<param>backgroundViscosity</param>
<param>storeViscosity</param>
</list>
</struct>
<struct name="surfaceAdaptor">
<param name="Type">SurfaceAdaptor</param>
<param name="mesh">mesh-linear</param>
<param name="sourceGenerator">linearMesh-generator
<param name="surfaceType">sine</param>
<list name="origin">
<param>0.0</param>
</list>
<param name="amplitude">0.1</param>
<param name="frequency">6.28318530718</param>
</struct>
</struct>
<list name="plugins">
<param>Underworld_EulerDeform</param>
<param>Underworld_DumpSwarm</param>
<param>Underworld_VTKOutput</param>
</list>
<param name="maxTimeSteps">10</param>
<param name="outputEvery">1</param>
<param name="dumpEvery">1</param>
<param name="outputPath">./output.template</param>
<param name="dim">2</param>
<param name="shadowDepth">1</param>
<param name="minX">0.0f</param>
<param name="minY">0.0f</param>
<param name="MaterialPointsSwarm">materialSwarm</p
</struct>
<struct name="viscous">
<param name="Type">RheologyMaterial</param>
<param name="Shape">boxShape</param>
<param name="density">1.0</param>
<list name="Rheology">
<param>backgroundViscosity</param>
<param>storeViscosity</param>
</list>
</struct>
<struct name="surfaceAdaptor">
<param name="Type">SurfaceAdaptor</param>
<param name="mesh">mesh-linear</param>
<param name="sourceGenerator">linearMesh-generator
<param name="surfaceType">sine</param>
<list name="origin">
<param>0.0</param>
</list>
<param name="amplitude">0.1</param>
<param name="frequency">6.28318530718</param>
</struct>
</struct>
<list name="plugins">
<param>Underworld_EulerDeform</param>
<param>Underworld_DumpSwarm</param>
<param>Underworld_VTKOutput</param>
</list>
<param name="maxTimeSteps">10</param>
<param name="outputEvery">1</param>




viscous.xml : viscous_split.xml - Meld
File Edit Settings Help
New... Save Undo Redo Down Up Stop
viscous.xml : viscous_split.xml
/home/boo/cig/gale/Gale/input/cookbook/viscous.xml
Browse...
/home/boo/cig/gale/Gale/input/cookbook/viscous_split.xr
Browse...
<param name="name">vx</param>
<param name="type">double</param>
<param name="value">0.0</param>
</struct>
</list>
</struct>
<struct>
<param name="type">WallVC</param>
<param name="wall">right</param>
<list name="variables">
<struct>
<param name="name">vx</param>
<param name="type">double</param>
<param name="value">0.0</param>
</struct>
</list>
</struct>
<struct>
<param name="type">WallVC</param>
<param name="wall">bottom</param>
<list name="variables">
<struct>
<param name="name">vy</param>
<param name="type">double</param>
<param name="value">0.0</param>
</struct>
</list>
</struct>
</list>
</struct>
<param name="checkpointEvery">1</param>
</StGermainData>
<struct>
<param name="name">vx</param>
<param name="type">double</param>
<param name="value">1.0</param>
</struct>
<struct>
<param name="name">vx</param>
<param name="type">func</param>
<param name="value">StepFunction</param>
</struct>
<param name="StepFunctionLowerOffset">1.0</param>
<param name="StepFunctionUpperOffset">1.0</param>
<param name="StepFunctionValue">1.0</param>
<param name="StepFunctionDim">0</param>
<param name="StepFunctionLessThan">False</param>
</StGermainData>



Evaluating Accuracy

- You must vary the accuracy parameters for every single publishable result.
  - Resolution
  - Number of particles/cell
  - Linear and nonlinear solution tolerance
  - size of the boundary region
  - size of the whole simulation (for some boundary conditions)