TPV29 and TPV30
Rough Fault with Viscoplasticity Benchmarks

January 17, 2015

These 3D benchmarks use a rough vertical fault in a half-space. We are doing two benchmarks, as shown in the following table.

Benchmarks			
Benchmark	*Dimension*	*Rupture Type*	*Material Properties*
TPV29	3D	Right-lateral, vertical strike-slip	Linear elastic
TPV30	3D	Right-lateral, vertical strike-slip	Drucker-Prager viscoplastic

We request that you run each of these two benchmarks using 50 meter resolution.

Benchmark Summary

- The geometry is a vertical strike-slip fault in a half-space. The fault is 40 km long and 20 km deep. The fault has randomly-generated roughness, with Hurst exponent equal to 1.

- Material properties are uniform throughout the model volume. The benchmarks include gravity and fluid pressure. The initial stress tensor varies with depth, and is in static equilibrium.

- The material properties are the only difference between the two benchmarks.

- The benchmarks use linear slip-weakening friction. Friction parameters are uniform over the fault surface, except that there is increased frictional cohesion in the uppermost 4 km to suppress free-surface effects.

- The fault boundary condition is that slip goes to zero at the border of the fault. So, a node which lies precisely on the border of a fault should *not* be permitted to slip. The free surface is not considered to be a border of the fault.

- Nucleation is done using a smoothed forced rupture. The forced rupture is applied over an interval of 0.5 seconds, and the forced rupture velocity decreases with distance from the hypocenter.

Contents

Part 1: Considerations for Plastic Benchmarks

Benchmarks with plasticity are more complicated to set up than benchmarks with linear elastic material properties. The following list highlights some special considerations to keep in mind when implementing a benchmark with off-fault plastic yielding.

- **Benchmark TPV30 uses non-associative Drucker-Prager viscoplasticity with yielding in shear.** Benchmark TPV29 uses linear elasticity. We include a detailed description of Drucker-Prager viscoplasticity.

- **The initial stress tensor is specified everywhere throughout the entire model volume.** Benchmarks with linear elasticity only need to specify initial shear and normal stresses on the fault surfaces. In TPV29 and TPV30, the initial shear and normal stresses on the faults are implied by resolving the initial stress tensor onto the fault surface.

- **Gravity must be accounted for.** The initial stress increases with depth, so gravitational acceleration is required to balance the gradient in the initial stress tensor.

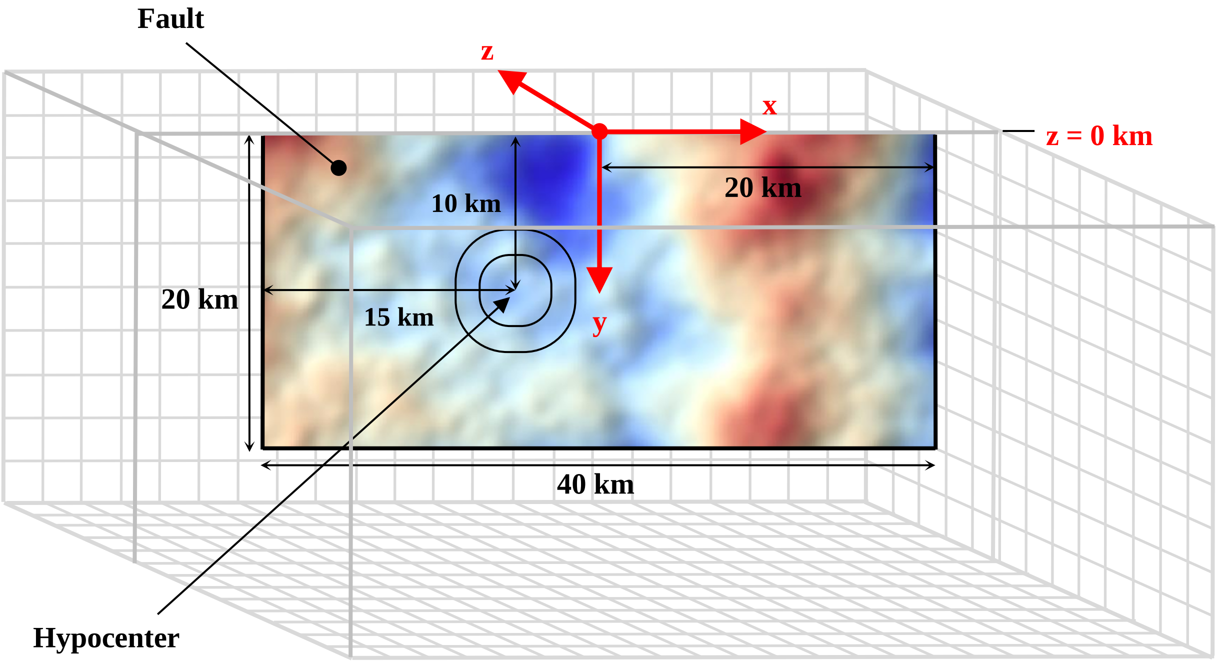
- **You may need to apply traction forces to the boundary of the model volume.** The initial stresses exert traction forces on the boundary of the model volume. Depending on your implementation, you may need to balance them with externally-applied tractions.

- **Fluid pressure is included in the model.** Fluid pressure is hydrostatic, with water table at the earth's surface. The fluid pressure enters into the plastic constitutive law, and the friction law.

- **Nucleation is done using forced rupture.** Benchmarks with linear elasticity can simply impose higher shear stress near the hypocenter, but that is not possible with plasticity because it would be inconsistent with the initial stress tensor. In TPV29 and TPV30, we nucleate by using a zone of forced rupture surrounding the hypocenter. A detailed description is provided.

TPV29 and TPV30 use a linear slip-weakening friction law.

Part 2: Fault Geometry for TPV29 and TPV30



The model volume is a half-space.

The fault is a rectangle measuring 40000 m along-strike and 20000 m deep. The fault is a strike-slip, right-lateral fault. The fault reaches the earth's surface. The fault surface has randomly-generated roughness.

The hypocenter is located 15 km from the left edge of the fault, at a depth of 10 km.

Introduce an (x, y, z) coordinate system, where x increases from left to right, y increases from top to bottom, and z increases from front to back. The origin is on the earth's surface, 20000 m from either edge of the fault. In this coordinate system, the fault is:

$$\begin{aligned} -20000\text{ m} \le x &\le 20000\text{ m} \\ 0\text{ m} \le y &\le 20000\text{ m} \\ z &= f(x, y) \end{aligned}$$

The function $f(x, y)$ defines the three-dimensional shape of the fault. We have supplied the function $f(x, y)$ as a data file. The data file format is described in part 3. There are two versions of the data file for you to choose from:

- A “short” data file named **`tpv29_tpv30_geometry_25m_data.txt`**. The short file describes the rough fault surface, which measures 40 km by 20 km.

- An “extended” data file named **`tpv29_tpv30_geometry_ext_25m_data.txt`**. The extended file describes a larger rough surface, which measures 50 km by 30 km. The fault surface is the central part of this larger surface. The extended file is useful if you need to know the rough surface geometry outside the 20 km by 40 km fault surface.

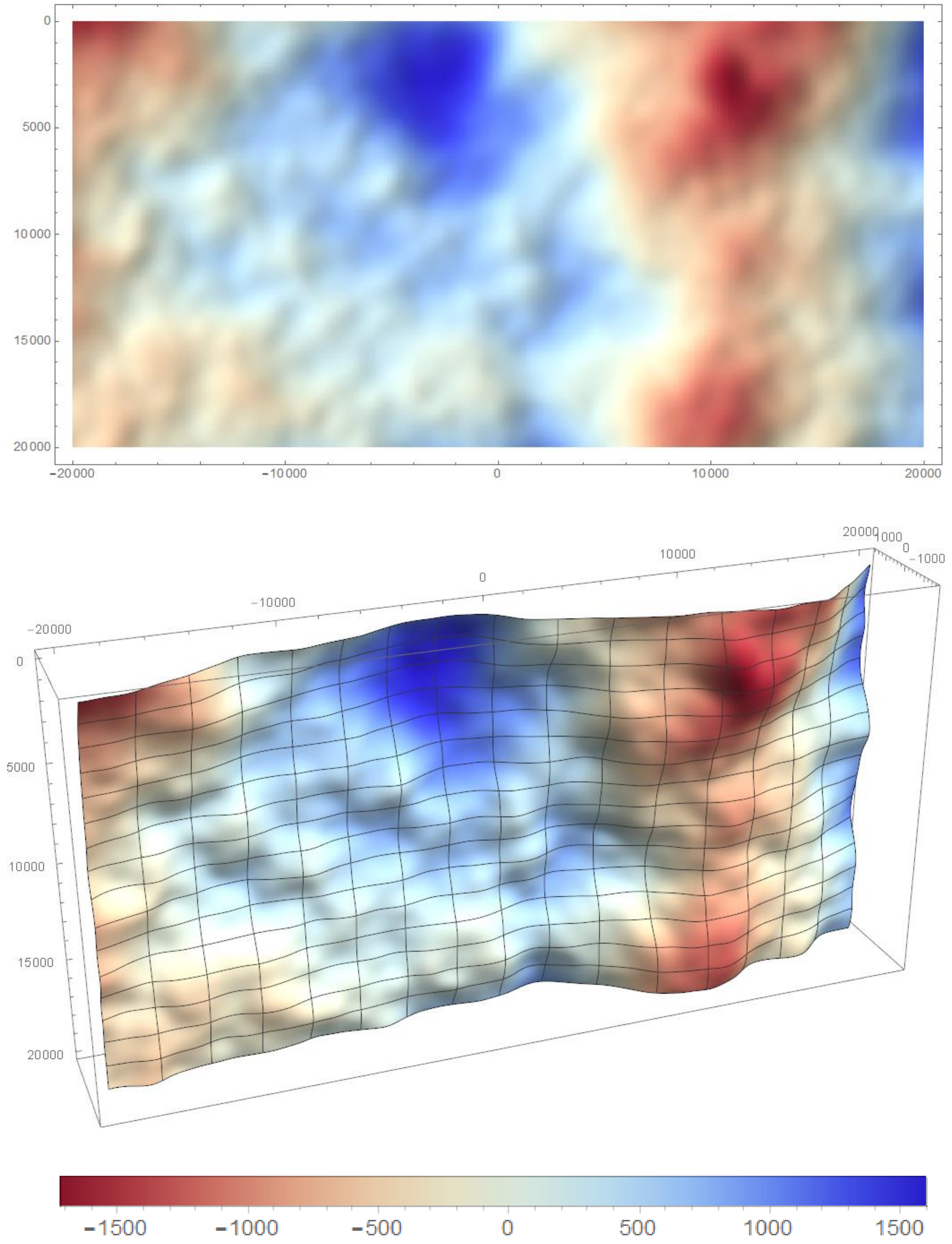
The two data files describe the same rough fault surface. You can use either one.

The figures on the next page show the fault geometry we are using for these benchmarks.

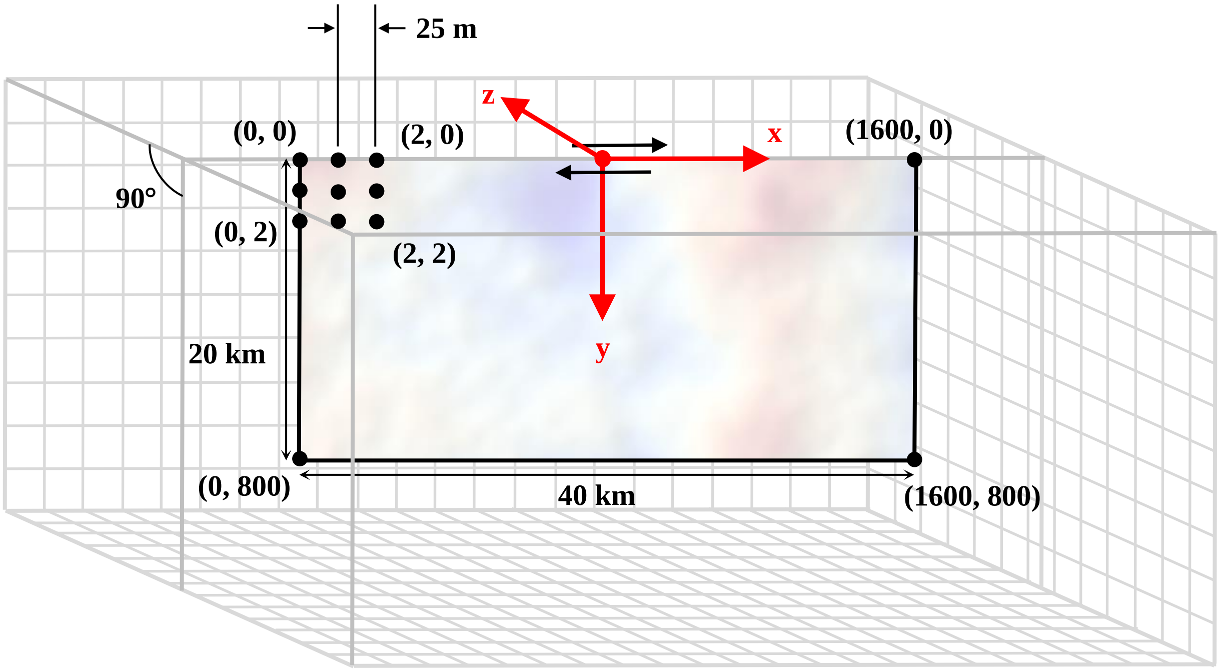
The top figure shows the xy-plane, color-coded to indicate the value of $z = f(x, y)$. Areas where the fault lies in front of the xy-plane, with negative z, are colored red. Areas where the fault lies in back of the xy-plane, with positive z, are colored blue. As shown by the color scale, the values of z range from approximately −1700 m to +1600 m.

The bottom figure shows a 3D oblique view of the fault surface. **The roughness has been exaggerated to make the fault geometry easy to see.**

Remark: This rough surface was generated using a random process. The surface is a linear superposition of sine waves, with randomly-selected phases. The wavelengths range from 1000 m to 40000 m. Each sine wave has amplitude proportional to the square of the wavelength, which produces a surface with Hurst exponent equal to 1. After forming the linear superposition of sine waves, the surface is smoothed by applying a moving-average filter with a radius of 1000 m.


0
5000
10000
15000
20000
−20000
−10000
0
10000
20000
−20000
−10000
0
10000
20000
1000
0
−1000
0
5000
10000
15000
20000
−1500
−1000
−500
0
500
1000
1500

Part 3: Input File Format for TPV29 and TPV30



The input file gives the values of the function $z = f(x, y)$ at a set of grid points on the fault surface. It also contains the two partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$.

This section describes **`tpv29_tpv30_geometry_25m_data.txt`**, which is the "short" input file. The "extended" file is described later.

The grid points are spaced 25 meters along-strike, and 25 meters along-dip. The fault surface has 1601 grid points along-strike, and 801 grid points along-dip, making for a total of 1282401 grid points.

Each grid point is labeled by a pair of integer coordinates (nx, ny). The coordinate nx is an integer ranging from 0 at the left edge of the fault to 1600 at the right edge. The coordinate ny is an integer ranging from 0 at the top of the fault to 800 at the bottom. The grid point at the upper left corner of the fault has coordinates $(0, 0)$.

The following excerpt shows the first six lines of the data file.

```
978171814
1600  800  40000.  20000.
0  0  -20000.  0.  -1526.3657  0.0087117  0.2553249
1  0  -19975.  0.  -1526.1162  0.0113073  0.2556886
2  0  -19950.  0.  -1525.7974  0.0142544  0.2559769
3  0  -19925.  0.  -1525.4007  0.0175341  0.2561871
```

The first two lines are a header, and the remaining lines contain data. The header contains information about the roughness realization, but is not actually needed to run the simulation. In your code, you may wish to skip over the two-line header, and just read the data.

Line 1: Identifies the random number seed used to generate the roughness. The single field is:

- Random number seed, an integer.

Line 2: Contains the total size of the fault surface. The four fields are:

- Number of grid cells along-strike (this is one less than the number of grid points along-strike). It is an integer. For TPV29-30, it is 1600.

- Number of grid cells down-dip (this is one less than the number of grid points down-dip). It is an integer. For TPV29-30, it is 800.

- Size of fault surface along-strike, in meters, as a floating-point number. For TPV29-30, it is 40000.

- Size of fault surface down-dip, in meters, as a floating-point number. For TPV29-30, it is 20000.

Remaining lines give the function $z = f(x, y)$ and the two partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ for each grid point on the fault surface. For TPV29-30, there are 1282401 of these lines. The seven fields on each line are:

- Grid point location along-strike, measured in grid cells (counting from 0 at the left edge of the fault surface). This is coordinate nx. It is an integer.

- Grid point location down-dip, measured in grid cells (counting from 0 at the earth's surface). This is coordinate ny. It is an integer.

- Grid point location along-strike, in meters, measured from the center of the fault surface. This is the x-coordinate, and it ranges from -20000 to 20000. It is a floating-point number.

- Grid point location down-dip, in meters, measured from the earth's surface. This is the y-coordinate, and it ranges from 0 to 20000. It is a floating-point number.

- The function value $z = f(x, y)$, in meters. Positive values denote displacement of the fault surface toward the far side of the fault. It is a floating-point number.

- The partial derivative $\partial f/\partial x$, which is dimensionless. It is a floating-point number.

- The partial derivative $\partial f/\partial y$, which is dimensionless. It is a floating-point number.

Grid points are listed in a specific order. The grid point in the upper left corner of the fault surface is listed first. Next come the remaining grid point at the earth's surface, listed left to right. Next come the grid point one cell (25 m) below the earth's surface, listed left to right. And so on. At the end of the file come the grid point at the bottom of the fault surface, listed left to right, with the grid point in the lower right corner of the fault surface appearing last.

The Extended Input File

Some codes may benefit from knowing the rough surface geometry beyond the 40 km by 20 km fault surface. So, we have provided an "extended" input file, which contains the function values $z = f(x, y)$ and the derivatives $\partial f/\partial x$ and $\partial f/\partial y$ over a surface that measures 50 km by 30 km. The extended input file is named **`tpv29_tpv30_geometry_ext_25m_data.txt`**.

The extended file has the same format as the short file, except that the grid is enlarged so that it extends 5000 meters to the left, to the right, below, and above the fault surface. In particular:

- The grid coordinate nx ranges from −200 to 1800.
- The grid coordinate ny ranges from −200 to 1000.
- The location x along-strike ranges from −25000 to 25000 meters.
- The location y down-dip ranges from −5000 to 25000 meters.

The enlarged grid has 2001 grid points along-strike, and 1201 grid points along-dip, making for a total of 2403201 grid points.

The following excerpt shows the first six lines of the extended data file:

```
978171814
2000  1200  50000.  30000.
-200  -200  -25000.  -5000.  -1111.8221  -0.1385928  -0.3117154
-199  -200  -24975.  -5000.  -1115.2390  -0.1347650  -0.3104137
-198  -200  -24950.  -5000.  -1118.5608  -0.1309947  -0.3088505
-197  -200  -24925.  -5000.  -1121.7894  -0.1273091  -0.3070421
```

Notice that the third line in the file, which is the first data line, has (nx, ny) = (−200, −200) and (x, y) = (−25000, −5000). This is the upper left corner of the enlarged grid, which lies 5000 meters to the left of the fault, and 5000 meters above the earth's surface. Also, notice that the second line of the file gives the size of the entire enlarged grid.

Interpolating the Input Data

The input file supplies the value of $f(x, y)$ at grid points spaced 25 meters apart. If you need the value of $f(x, y)$ at other points, then you have to interpolate the supplied values. For these benchmarks, you can interpolate using whatever method you prefer.

If you select an interpolation method that requires values from outside the fault surface, then you may obtain those values from the extended input file.

Simple linear interpolation yields values of $f(x, y)$ that are accurate to within 0.1 meters.

If you want greater accuracy, we suggest using biquadratic interpolation, which produces values of $f(x,y)$ that are accurate to within 0.003 meters. It does not require any points from outside the fault surface. Here we give the formulas for biquadratic interpolation.

Biquadratic interpolation is a nine-point interpolation method, which means that $f(x,y)$ is calculated from nine of the supplied input values. Imagine that the fault surface is partitioned into square cells that are 50 meters by 50 meters. Then each cell contains nine grid points: four at the corners, four at the edge midpoints, and one at the center. To interpolate, we first locate the square cell that contains (x,y), and then use the function values at the cell's nine grid points to compute an estimate of $f(x,y)$.

Assume that we are given x and y with $-20000\text{ m} \le x \le 20000\text{ m}$ and $0\text{ m} \le y \le 20000\text{ m}$. Let $h = 25$ m be the grid point spacing. To begin, calculate:

$$x' = 2h \min(\lfloor x/2h \rfloor, 399)$$

$$y' = 2h \min(\lfloor y/2h \rfloor, 399)$$

The notation $\lfloor x/2h \rfloor$ means the largest integer that is less than or equal to $x/2h$. For example, $\lfloor 5.7 \rfloor = 5$ and $\lfloor -7.3 \rfloor = -8$. Then (x', y') is the grid point located at the upper left corner of the square cell that contains (x,y). Taking the minimum with 399 ensures (x', y') does not lie on the right or bottom edge of the fault, so that the square cell is inside the fault surface. Next, calculate:

$$u = (x - x')/h - 1$$

$$v = (y - y')/h - 1$$

Then u and v give the position of (x,y) within the square cell, scaled so $-1 \le u \le 1$ and $-1 \le v \le 1$. Next, calculate the interpolation coefficients:

$$a_0 = u(u-1)/2$$

$$a_1 = (1-u)(1+u)$$

$$a_2 = u(u+1)/2$$

$$b_0 = v(v-1)/2$$

$$b_1 = (1-v)(1+v)$$

$$b_2 = v(v+1)/2$$

Then the interpolated function value is:

$$f(x,y) \approx \sum_{m=0}^{2} \sum_{n=0}^{2} a_m \, b_n \, f(x' + mh, y' + nh)$$

Part 4: Description of the 3D Benchmarks

Fault Geometry

Fault geometry for TPV29 and TPV30 is shown in part 2.

The hypocenter is 15000 m from the left edge of the fault, and 10000 m deep, at location $x = -5000$ m and $y = 10000$ m.

Slip goes to zero at the border of a fault. So, a node which lies precisely on the border of a fault should not be permitted to slip. The free surface is not considered to be a border of the fault.

Material Properties

In TPV29, the entire model volume is a linear elastic material, with the following parameters.

Density $\rho = 2670\ \mathrm{kg/m^3}$

Shear-wave velocity $V_s = 3464\ \mathrm{m}/s$

Pressure-wave velocity $V_p = 6000\ \mathrm{m/s}$

In TPV30, the entire model volume is a non-associative Drucker-Prager viscoplastic material that yields in shear, with the following parameters. Drucker-Prager viscoplasticity is described later, in part 7.

Density $\rho = 2670\ \mathrm{kg/m^3}$

Shear-wave velocity $V_s = 3464\ \mathrm{m}/s$

Pressure-wave velocity $V_p = 6000\ \mathrm{m/s}$

Cohesion $c = 1.18\ \mathrm{MPa}$

Bulk friction $\nu = 0.1680$

Viscoplastic relaxation time $T_v = 0.05\ \mathrm{s}$

The material properties are the only difference between the elastic benchmark (TPV29), and the viscoplastic benchmark (TPV30).

Initial Stress Tensor

In an inelastic calculation, the initial stress tensor needs to be specified throughout the medium. The total (time-varying) stress, which equals the initial stress plus the stress change, is used in the plastic yield calculation. The (time-varying) normal and shear stresses acting on the fault surface are implied by the stress tensor in the adjacent rock.

For TPV29 and TPV30, the initial stress tensor varies only with depth. The components of the stress tensor are defined in the following table.

Stress Tensor Components	
Component	*Definition*
σ_{22}	Compressive stress in the vertical direction. Negative values denote compression.
σ_{11}	Compressive stress in the direction parallel to the fault. Negative values denote compression.
σ_{33}	Compressive stress in the direction perpendicular to the fault. Negative values denote compression. This equals the negative of the total normal stress on the fault.
σ_{13}	Shear stress in the horizontal plane. Positive values denote right-lateral shear stress on the fault. This equals the horizontal shear stress on the fault.
σ_{23}	Shear stress in a vertical plane perpendicular to the fault.
σ_{12}	Shear stress in a vertical plane parallel to the fault.

The stress tensor components can vary with time during the simulation.

Fluid pressure P_f is hydrostatic with water table at the surface. It varies with depth, but is constant in time.

For these 3D benchmarks, the fluid pressure and the initial values of the stress tensor are:

$$P_f = (1000\ \mathrm{kg/m^3})(9.8\ \mathrm{m/s^2})(\text{depth in meters})$$

$$\sigma_{22} = -(2670\ \mathrm{kg/m^3})(9.8\ \mathrm{m/s^2})(\text{depth in meters})$$

$$\sigma_{11} = \Omega(\text{depth})\big(b_{11}\,(\sigma_{22} + P_f) - P_f\big) + (1 - \Omega(\text{depth}))\sigma_{22}$$

$$\sigma_{33} = \Omega(\text{depth})\big(b_{33}\,(\sigma_{22} + P_f) - P_f\big) + (1 - \Omega(\text{depth}))\sigma_{22}$$

$$\sigma_{13} = \Omega(\text{depth})\left(b_{13}\left(\sigma_{22} + P_f\right)\right)$$

$$\sigma_{23} = 0$$

$$\sigma_{12} = 0$$

The tapering coefficient $\Omega(\text{depth})$ causes the deviatoric component of stress to taper down to zero at depths between 17000 m and 22000 m.

$$\Omega(\text{depth}) = \begin{cases} 1\,, & \text{if depth} \leq 17000\text{ m} \\ (22000\text{ m} - \text{depth})/(5000\text{ m})\,, & \text{if } 17000\text{ m} \leq \text{depth} \leq 22000\text{ m} \\ 0\,, & \text{if depth} \geq 22000\text{ m} \end{cases}$$

The coefficients b_{11}, b_{33}, and b_{13} are given in the following table.

Coefficient	*Value for TPV29 and TPV30*
b_{11}	1.025837
b_{33}	0.974162
b_{13}	-0.158649

Note that the gravitational acceleration g is *exactly* 9.8 m/s^2. Do not substitute a “more accurate” value.

Note that the initial axis of intermediate compressive stress is vertical, and its initial value is lithostatic. The initial axes of maximum and minimum compressive stress are horizontal, and their average value is lithostatic. They are oriented so that the xy-plane is an optimal plane for fracture formation, that is, has the maximum ratio of shear stress to effective normal stress. For TPV30, the initial deviatoric stress at hypocenter depth is 93% of the plastic yield stress.

Note that we defined the stress tensor components σ_{ij} to be the **total** stress tensor, which includes both elastic stress and fluid pressure. To obtain the **effective** stress tensor, which includes only elastic stress, you need to add the fluid pressure P_f to the diagonal components. This can be shown as:

$$\text{total stress tensor} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{pmatrix}$$

$$\text{effective stress tensor} = \begin{pmatrix} \sigma_{11} + P_f & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} + P_f & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} + P_f \end{pmatrix}$$

Friction Parameters and Nucleation

We use a linear slip-weakening friction law, which has the following six parameters.

Friction Parameters		
Symbol	*Parameter*	*Unit*
μ_s	Static coefficient of friction.	Dimensionless
μ_d	Dynamic coefficient of friction.	Dimensionless
d_0	Slip-weakening critical distance.	Meter
C_0	Frictional cohesion.	Pascal
T	Time of forced rupture.	Second
t_0	Forced rupture decay time	Second

The operation of the slip-weakening friction law is described in detail later, in part 5.

The friction parameter values are as follows, where r denotes distance to the hypocenter:

$$\mu_s = 0.18$$

$$\mu_d = 0.12$$

$$d_0 = 0.30 \text{ m}$$

$$C_0 = \begin{cases} 0.40 \text{ MPa} + (0.00020 \text{ MPa/m})(4000 \text{ m} - \text{depth}), & \text{if depth} \leq 4000 \text{ m} \\ 0.40 \text{ MPa}, & \text{if depth} \geq 4000 \text{ m} \end{cases}$$

$$r_{\text{crit}} = 4000 \text{ m}$$

$$T = \begin{cases} \dfrac{r}{0.7\,V_S} + \dfrac{0.081\, r_{\text{crit}}}{0.7\,V_S}\left(\dfrac{1}{1-(r/r_{\text{crit}})^2} - 1\right), & \text{if } r < r_{\text{crit}} \\ 1.0\text{E}{+}9, & \text{if } r \geq r_{\text{crit}} \end{cases}$$

$$t_0 = 0.50 \text{ s}$$

The variation in the value of T near the hypocenter causes the rupture to nucleate, in a manner that is described later, in part 6.

The frictional cohesion C_0 (not to be confused with the plastic cohesion c) is 1.20 MPa at the earth's surface. It is 0.40 MPa at depths greater than 4000 m, and is linearly tapered in the uppermost 4000 m.

Running Time, Node Spacing, and Results

Run the model for times from **0.0 to 20.0 seconds after nucleation**.

Please submit results using **50 m node spacing** on the fault plane.

If you are unable to run the simulation with 50 m node spacing, then it is OK to use 100 m node spacing. However, we strongly suggest that you run the simulation with 50 m resolution if at all possible.

The requested output files are:

- **On-fault time-series files**, which give slips, slip rates, and stresses for each on-fault station at each time step. These files are described in part 9.

- **Off-fault time-series files**, which give displacements and velocities for each off-fault station at each time step. These files are described in part 10.

- **A contour-plot file** which, for each node on the fault, gives the time at which the slip rate first changes from 0 to greater than 0.001 m/s. This file is described in part 11.

Part 5: Linear Slip-Weakening Friction

Benchmarks TPV29 and TPV30 use linear slip-weakening friction, with frictional cohesion and forced rupture. This friction law has the following parameters and variables:

Friction Parameters		
Symbol	*Parameter*	*Unit*
μ_s	Static coefficient of friction.	Dimensionless
μ_d	Dynamic coefficient of friction.	Dimensionless
d_0	Slip-weakening critical distance.	Meter
C_0	Frictional cohesion.	Pascal
T	Time of forced rupture.	Second
t_0	Forced rupture decay time	Second

Friction Variables		
Symbol	*Parameter*	*Unit*
σ_n	Total normal stress acting on the fault, taken to be positive in compression.	Pascal
τ	Shear stress acting on the fault.	Pascal
P_f	Fluid pressure (which is assumed to be constant in time).	Pascal
$\sigma_n - P_f$	Effective normal stress acting on the fault, which is positive in compression.	Pascal

When the fault is sliding, the shear stress τ at a given point on the fault is given by:

$$\tau = C_0 + \mu \max(0,\ \sigma_n - P_f)$$

The time-varying coefficient of friction μ is given by the following formulas, where D is the total distance the node has slipped, and t is the time since the start of the simulation:

$$\mu = \mu_s + (\mu_d - \mu_s)\max(f_1,\ f_2)$$

$$f_1 = \begin{cases} D/d_0\,, & \text{if } D < d_0 \\ 1\,, & \text{if } D \geq d_0 \end{cases}$$

$$f_2 = \begin{cases} 0\,, & \text{if } t < T \\ (t - T)/t_0\,, & \text{if } T \leq t < T + t_0 \\ 1\,, & \text{if } t \geq T + t_0 \end{cases}$$

The distance D that the node has slipped is path-integrated. For example, if the node slips 0.4 m in one direction and then 0.1 m in the opposite direction, the value of D is 0.5 m (and not 0.3 m).

The time of forced rupture T is used to nucleate the rupture, as described in part 6. At time $t = T$, the coefficient of friction begins to decrease, reaching its final dynamic value at time $t = T + t_0$. So, the node is forced to begin sliding somewhere in the interval $T < t < T + t_0$. Notice that because of the expression $\max(f_1, f_2)$, forcing has no effect if friction has already weakened due to the accumulated slip. If a given node does not undergo forced rupture, then the value of T is 1.0E9.

Tension on the fault: If you encounter tension on the fault, you should **treat tension on the fault the same as if the effective normal stress equals zero**. This is shown in the above formulas by the expression $\max(0, \sigma_n - P_f)$.

You should **constrain the motion of the node so that the fault cannot open (that is, only permit sliding parallel to the fault), even when the fault is in tension**. During the time the fault is in tension, continue to accumulate the slip distance D as usual.

In your time-series files, where you are asked to report "normal stress," you should report the effective normal stress which is defined to be $\sigma_n - P_f$.

Part 6: Nucleation

Nucleation is performed by forcing the fault to rupture, within a circular zone surrounding the hypocenter. Forced rupture is achieved by artificially reducing the friction coefficient, beginning at a specified time T. The parameter t_0 specifies how long it takes for the friction coefficient to be artificially reduced from its static value to its dynamic value. So, the friction coefficient reaches its dynamic value at time $T + t_0$. We reduce the friction coefficient gradually, over an interval of time, in order to smooth the nucleation process and reduce unwanted oscillations.

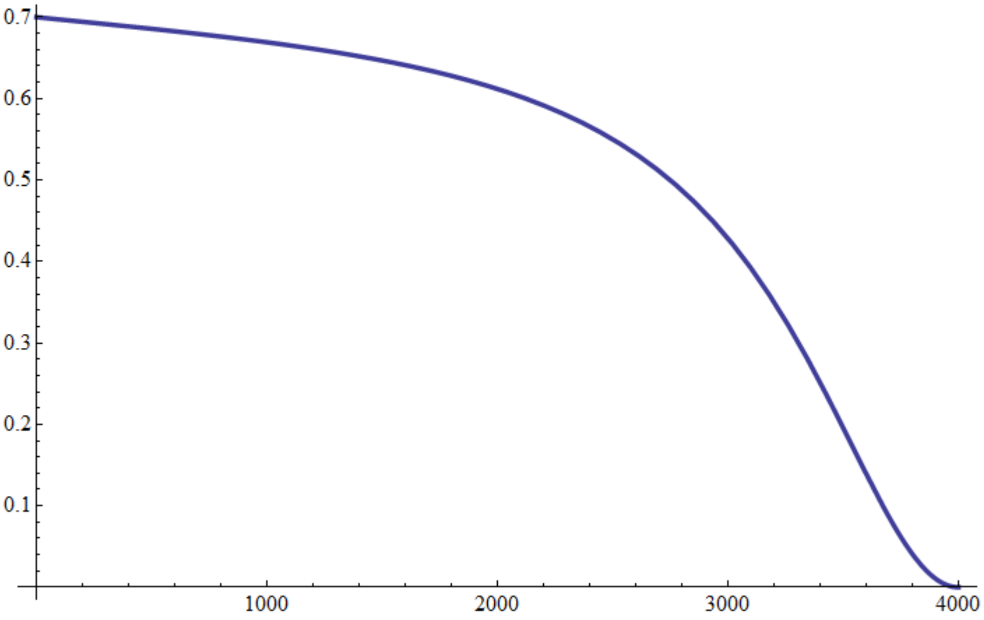
Recall that we selected t_0 = 0.5 s, and that the time of forced rupture is defined to be

$$T = \begin{cases} \dfrac{r}{0.7\,V_S} + \dfrac{0.081\,r_{\text{crit}}}{0.7\,V_S}\left(\dfrac{1}{1-(r/r_{\text{crit}})^2} - 1\right), & \text{if } r < r_{\text{crit}} \\ 1.0\text{E+}9\,, & \text{if } r \geq r_{\text{crit}} \end{cases}$$

where r is distance from the hypocenter, and where r_{crit} = 4000 m is the radius of the nucleation zone. At the hypocenter, $T = 0$. The value of T then increases with distance from the hypocenter, which creates an expanding circular region of forced rupture.

The time T is computed so that the forced rupture expands at a variable speed. Near the hypocenter, the forced rupture expands at a speed of $0.7\,V_S$. The speed decreases with increasing distance from the hypocenter, finally reaching a speed of zero at a distance of 4000 m from the hypocenter. The variable speed allows for a smooth transition between forced rupture and spontaneous rupture, because the spontaneous rupture should gradually overtake the ever-slowing forced rupture. The graph below shows the forced rupture speed $V_R \equiv dr/dT$ as a function of distance from the hypocenter.

Speed of forced rupture V_R/V_S.



Distance from hypocenter r, meters.

Part 7: Drucker-Prager Viscoplasticity

Constitutive Law

For TPV30 we are using non-associative Drucker-Prager viscoplasticity with yielding in shear. The material is characterized by seven constitutive parameters:

Density $\rho = 2670 \text{ kg/m}^3$

Shear-wave velocity $V_s = 3464 \text{ m/s}$

Pressure-wave velocity $V_p = 6000 \text{ m/}s$

Cohesion $c = 1.18 \text{ MPa}$

Bulk friction $\nu = 0.1680$

Viscoplastic relaxation time $T_v = 0.05 \text{ s}$

Fluid pressure $P_f = (1000 \text{ kg/m}^3)(9.8 \text{ m/s}^2)(\text{depth in meters})$

Except for the fluid pressure, all the constitutive parameters are uniform. The “cohesion” listed here is not to be confused with the “frictional cohesion” that appears in the fault friction law. From these six constitutive parameters, we can derive several additional ones:

Shear modulus $\mu = \rho V_s^2$

Lame volumetric parameter $\lambda = \rho(V_p^2 - 2V_s^2)$

Bulk modulus $K = \lambda + \frac{2}{3}\mu$

Angle of friction $\phi = \tan^{-1} \nu$

Let σ_{ij} denote the stress tensor. Define:

Mean stress $\sigma_m = \frac{1}{3}(\sigma_{11} + \sigma_{22} + \sigma_{33})$

Stress deviator $s_{ij} = \sigma_{ij} - \sigma_m \delta_{ij}$

Second invariant of the stress deviator $J_2(\sigma) = \frac{1}{2}\sum_{i,j} s_{ij} s_{ji}$

Drucker-Prager yield stress $Y(\sigma) = \max(0,\ c\cos\phi - \left(\sigma_m + P_f\right)\sin\phi)$

Drucker-Prager yield function $F(\sigma) = \sqrt{J_2(\sigma)} - Y(\sigma)$

Drucker-Prager plastic projection operator $P(\sigma) = \sigma^P$ where

$$\sigma_{ij}^{P} = \begin{cases} \sigma_{ij}\,, & \text{if } F(\sigma) \le 0 \\ \sigma_m \delta_{ij} + \left(Y(\sigma) \,/\, \sqrt{J_2(\sigma)} \right) s_{ij}\,, & \text{if } F(\sigma) \ge 0 \end{cases}$$

Let ϵ_{ij} denote the strain tensor, let $\epsilon_{ij}^{\mathrm{vp}}$ denote the viscoplastic strain tensor, and let $\epsilon_{ij}^{\mathrm{e}}$ denote the elastic strain tensor. Then the material constitutive equations are:

$$\epsilon_{ij}^{\mathrm{e}} = \epsilon_{ij} - \epsilon_{ij}^{\mathrm{vp}}$$

$$\sigma_{ij} = \lambda(\epsilon_{11}^{\mathrm{e}} + \epsilon_{22}^{\mathrm{e}} + \epsilon_{33}^{\mathrm{e}})\delta_{ij} + 2\mu\epsilon_{ij}^{\mathrm{e}}$$

$$\frac{d}{dt}\left(\lambda\left(\epsilon_{11}^{\mathrm{vp}} + \epsilon_{22}^{\mathrm{vp}} + \epsilon_{33}^{\mathrm{vp}}\right)\delta_{ij} + 2\mu\epsilon_{ij}^{\mathrm{vp}}\right) = \frac{1}{T_v}\left(\sigma_{ij} - \sigma_{ij}^{P}\right)$$

(Remarks: In writing these equations, we implicitly assume that the strain tensor ϵ_{ij} is a known function of time, and we want to solve for the stress tensor σ_{ij} as a function of time. The first equation is the definition of the viscoplastic strain tensor $\epsilon_{ij}^{\mathrm{vp}}$, and the second equation is the definition of the elastic strain tensor $\epsilon_{ij}^{\mathrm{e}}$. Then the third equation is a set of coupled differential equations which are to be solved for σ_{ij}. The third equation can be simplified, because for our choice of $P(\sigma)$ it can be shown that $\epsilon_{11}^{\mathrm{vp}} + \epsilon_{22}^{\mathrm{vp}} + \epsilon_{33}^{\mathrm{vp}} = 0$. We retain the term because we prefer to write the constitutive equations in a form that is valid for any choice of $P(\sigma)$.)

When $F(\sigma) < 0$, the material behaves like a linear elastic material, with Lame parameters λ and μ. The viscoplastic strain tensor $\epsilon_{ij}^{\mathrm{vp}}$ remains constant while the material is behaving elastically.

When $F(\sigma) \ge 0$, the material *yields viscoplastically*. Viscoplastic yielding can be understood as follows:

- If the material is subjected to an instantaneous (or very rapid) increment of strain, then it responds like a linear elastic material, with Lame parameters λ and μ. The viscoplastic strain tensor $\epsilon_{ij}^{\mathrm{vp}}$ remains unchanged.

- If time passes with no change in the strain tensor, then the stress tensor σ_{ij} decays exponentially to the value σ_{ij}^{P}, with characteristic time T_v. The change in the stress tensor is accompanied by a corresponding change in the viscoplastic strain tensor $\epsilon_{ij}^{\mathrm{vp}}$. We say that the material *yields in shear* because the stress tensor σ_{ij} changes by an amount proportional to the stress deviator s_{ij}, and the viscoplastic strain tensor $\epsilon_{ij}^{\mathrm{vp}}$ is purely deviatoric (*i.e.*, has zero trace).

Technical remarks: We call P a *projection operator* because $P\big(P(\sigma)\big) = P(\sigma)$. When viscoplasticity is described by a projection operator, it is called a *Duvaut-Lions model*. We chose P so that $\sigma - P(\sigma)$ is purely deviatoric, and is proportional to the stress deviator s. The *yield cone* is the set $\{\sigma \mid F(\sigma) \le 0\}$. We say the material is *non-associative* because $P(\sigma)$ is not the point on yield cone that is closest to σ.

Return Map Algorithm

The return map algorithm is a popular algorithm for implementing viscoplasticity, so we describe it here.

Given:

σ_{ij}^0 = Stress tensor at the start of a time step.

ϵ_{ij} = Strain tensor increment applied during the time step.

Δt = Duration of the time step.

We want to find:

σ_{ij} = Stress tensor at the end of the time step.

Step 1. Calculate a *trial stress tensor* by assuming that the material behaves elastically:

$$\sigma_{ij}^{\mathrm{trial}} = \sigma_{ij}^0 + \lambda(\epsilon_{11} + \epsilon_{22} + \epsilon_{33})\delta_{ij} + 2\mu\epsilon_{ij}$$

Step 2. Calculate the following:

Trial mean stress $\sigma_m^{\mathrm{trial}} = \frac{1}{3}\left(\sigma_{11}^{\mathrm{trial}} + \sigma_{22}^{\mathrm{trial}} + \sigma_{33}^{\mathrm{trial}}\right)$

Trial stress deviator $s_{ij}^{\mathrm{trial}} = \sigma_{ij}^{\mathrm{trial}} - \sigma_m^{\mathrm{trial}}\delta_{ij}$

Second invariant of the trial stress deviator $J_2\left(\sigma^{\mathrm{trial}}\right) = \frac{1}{2}\sum_{i,j} s_{ij}^{\mathrm{trial}} s_{ji}^{\mathrm{trial}}$

Trial Drucker-Prager yield stress $Y\left(\sigma^{\mathrm{trial}}\right) = \max\left(0,\ c\cos\phi - \left(\sigma_m^{\mathrm{trial}} + P_f\right)\sin\phi\right)$

Trial Drucker-Prager yield function $F\left(\sigma^{\mathrm{trial}}\right) = \sqrt{J_2(\sigma^{\mathrm{trial}})} - Y\left(\sigma^{\mathrm{trial}}\right)$

Step 3. If $F\left(\sigma^{\mathrm{trial}}\right) \leq 0$, then this is an *elastic step*. In this case, we take:

$$\sigma_{ij} = \sigma_{ij}^{\mathrm{trial}}$$

Step 4. If $F\left(\sigma^{\mathrm{trial}}\right) > 0$, then this is a *viscoplastic step*. In this case, we take:

$$r = \exp(-\Delta t/T_v) \ + \ (1 - \exp(-\Delta t/T_v))\, Y\left(\sigma^{\mathrm{trial}}\right) / \sqrt{J_2(\sigma^{\mathrm{trial}})}$$

$$\sigma_{ij} = \sigma_m^{\mathrm{trial}}\delta_{ij} + r s_{ij}^{\mathrm{trial}}$$

Sample Fortran Code

Courtesy of Joe Andrews and Michael Barall, here is sample Fortran code for the return map algorithm.

```
c  ---[ Constitutive parameters ]---
c     Density
      density = 2670.0
c     Vs
      vs = 3464.0
c     Vp
      vp = 6000.0
c     Shear modulus
      mu = density*(vs**2)
c     Lame volumetric parameter
      lambda = density*(vp**2 - 2.0*(vs**2))
c     Cohesion is 1.36 MPa
      cohes = 1.18e6
c     Bulk friction
      blkfric = 0.1680
c     Angle of friction
      angfric = atan(blkfric)
c     Viscoplastic relaxation time
      tvisc = 0.05
c     Fluid pressure, depth-dependent
      fluidpresh = 1000.0*9.8*depth

c  ---[ Calculate trial stress tensor ]---
c     Stress tensor components: sxx,syy,szz,sxy,syz,sxz
c     Strain tensor increment components: exx,eyy,ezz,exy,eyz,exz
      etrace = exx + eyy + ezz
      sxx = sxx + lambda*etrace + 2.0*mu*exx
      syy = syy + lambda*etrace + 2.0*mu*eyy
      szz = szz + lambda*etrace + 2.0*mu*ezz
      sxy = sxy + 2.0*mu*exy
      syz = syz + 2.0*mu*eyz
      sxz = sxz + 2.0*mu*exz

c  ---[ Calculate trial yield stress ]---
c     Mean stress
      sm = (sxx + syy + szz)/3.0
c     Find stress deviator components
      sdxx = sxx - sm
      sdyy = syy - sm
      sdzz = szz - sm
      sdxy = sxy
      sdyz = syz
      sdxz = sxz
c     Second invariant of stress deviator
      secinv = 0.5*(sdxx**2+sdyy**2+sdzz**2)+sdxy**2+sdyz**2+sdxz**2
c     Scalar measure of shear stress
      tau = sqrt(secinv)
c     Yield stress
      taulim = cohes*cos(angfric) - (sm + fluidpresh)*sin(angfric)
      taulim = max(0.0, taulim)

c  ---[ Yielding in shear ]---
c     Duration of time step: deltat
c     Stress deviators are adjusted, no change to mean stress
      if (tau .gt. taulim) then
        decay = exp(-deltat/tvisc)
        yldfac = decay + (1.0-decay)*taulim/tau
        sxx = sdxx*yldfac + sm
        syy = sdyy*yldfac + sm
        szz = sdzz*yldfac + sm
        sxy = sdxy*yldfac
        syz = sdyz*yldfac
        sxz = sdxz*yldfac
      end if
```

Part 8: Methods for Implementing Gravity and Boundary Tractions

In an inelastic calculation, the initial stress needs to be specified throughout the medium. The total stress, which equals initial stress plus stress change, is used in the yield calculation. This creates two challenges.

- Because the initial stress varies with depth, a gravitational force is required to balance the gradient of the initial stress tensor.

- The initial stress tensor exerts traction forces on the boundary of the mesh. If the code permits the mesh boundaries to move, then external tractions must be applied to balance these forces.

There are two basic methods for dealing with gravity and boundary tractions, described below. You may choose either method.

Method 1 — Using Stress Change

The stress tensor that is calculated and stored in the code is the change of stress from some initial state. The initial state is assumed to be in static equilibrium. With this method, the gravity and boundary tractions do not appear explicitly in the calculation, because as far as the code is concerned, all the initial stresses and forces are equal to zero.

In an elastic calculation, the initial state does not need to be known. In an inelastic calculation, the initial stress tensor must be specified and stored within each element (it varies only with depth in this case). Performing the plastic yield calculation then requires three steps:

1. Add the stress change to the initial stress, obtaining the total stress tensor.
2. Perform the plastic yield calculation (for example, with the return map algorithm), obtaining a new total stress tensor.
3. Subtract the initial stress from the new total stress tensor, obtaining a new stress change.

Because the code starts off with all stresses and forces equal to zero, you must “manually” specify the initial shear stress and initial effective normal stress at each node on the fault surface. To calculate the frictional sliding on the fault, you must (a) calculate the tractions on the fault surface implied by the change in stress, and then (b) add the initial shear stress and initial effective normal stress, to obtain the total shear stress and total effective normal stress which appear in the friction law.

Method 2 — Using Explicit Gravity and Boundary Tractions

The stress tensor that is calculated and stored in the code is the total stress tensor. The stress tensor throughout the medium must be initialized to a state that is in static equilibrium. The equilibrium stress state might come from a static calculation. In this case, because it depends only on depth, the initial stress state is simple enough to specify directly.

The acceleration of gravity must be added explicitly to the equations of motion. The gradient of the initial stress tensor is balanced by gravitational forces. For TPV29 and TPV30, you should set the gravitational acceleration g to be exactly 9.8 m/s^2.

If your code allows the mesh boundaries to move, then you must also apply external tractions to the mesh boundary, to balance the tractions caused by the initial stress. You need to calculate and store the force exerted on each boundary node by the initial stress conditions (including the gravitational force). Then, during the dynamic calculation, apply the negative of the stored force to each node lying in the boundary, thereby eliminating unwanted motion of the boundary. There are two ways to calculate the forces:

1. Perform a preliminary static calculation.

2. Analytically calculate the forces, in a manner that depends on the details of your code.

Because of approximation errors in the equations of motion, there may be small velocities throughout the mesh before real signals arrive. They are caused by small residual forces that exist because the nodal forces calculated from the initial stress tensor may not precisely cancel out the calculated gravitational forces. This effect is generally small enough to be ignored.

(Remark: If you do a preliminary static calculation to obtain boundary tractions, you can at the same time calculate the residual forces. Then, during the dynamic calculation, you can cancel out both the boundary tractions and the residual forces, thereby eliminating the small velocities. Just be careful that you don't also cancel out the traction forces acting on the fault surface!)

The plastic yield calculation has the total stress tensor available, so that the initial stress tensor does not need to be stored.

The code automatically calculates the total shear stress and total normal stress acting on the fault surface. So, it is not necessary to "manually" specify initial stresses acting on the fault surface.

However, the frictional sliding calculation also requires the fluid pressure, so that effective normal stress can be computed. There are two ways you could obtain the fluid pressure at each node on the fault surface. You may use either technique.

1. Calculate the fluid pressure by applying the formula

   $$P_f = (1000\ \mathrm{kg/m^3})(9.8\ \mathrm{m/s^2})(\text{depth in meters})$$

   This should work well for fault nodes not located on the earth's surface. However, special considerations apply to a fault node located at the earth's surface. For most codes it is incorrect to assign a fluid pressure of zero. We recommend that the fluid pressure be set as follows: (a) If your code has an initial stress tensor that is uniform within each element, then a fault node at the earth's surface should be assigned a fluid pressure corresponding to a depth of 1/2 element. (b) If your code has an initial stress tensor that varies linearly with depth inside each element, then a fault node at the earth's surface should be assigned a fluid pressure corresponding to a depth of 1/3 element.

2. Perform a preliminary static calculation to obtain the traction exerted on the fault surface by fluid pressure. Set up the static calculation to use density and initial stress as

   $$\rho = 1000\ \mathrm{kg/m^3}$$
   $$\sigma_{11} = \sigma_{22} = \sigma_{33} = -(1000\ \mathrm{kg/m^3})(9.8\ \mathrm{m/s^2})(\mathrm{depth\ in\ meters})$$

   The static calculation should also include the gravitational force. You can numerically calculate the normal stress exerted at each node on the fault surface by fluid pressure, and store the calculated normal stress. Then, during the dynamic calculation, you use the stored value as the fluid pressure.

   (Because of numerical approximation errors, the static calculation may also yield a small shear stress acting on the fault surface. If you wish, you may store the calculated shear stress, and deduct it from the total shear stress (using vector subtraction) during the dynamic calculation.)

(Remark: Some codes may prefer to calculate and store the effective stress tensor at each point of the medium, rather than the total stress tensor. This does not fundamentally change the method, although some details are changed. In particular, because resolving the effective stress tensor onto the fault surface yields effective normal stress, it is not necessary to separately calculate the fluid pressure on the fault surface.)

Part 9: On-Fault Stations, and Time-Series File Format

The benchmarks use 24 stations on the fault, which are listed below. A diagram of station locations is given following the table. You need to supply one time-series file for each station.

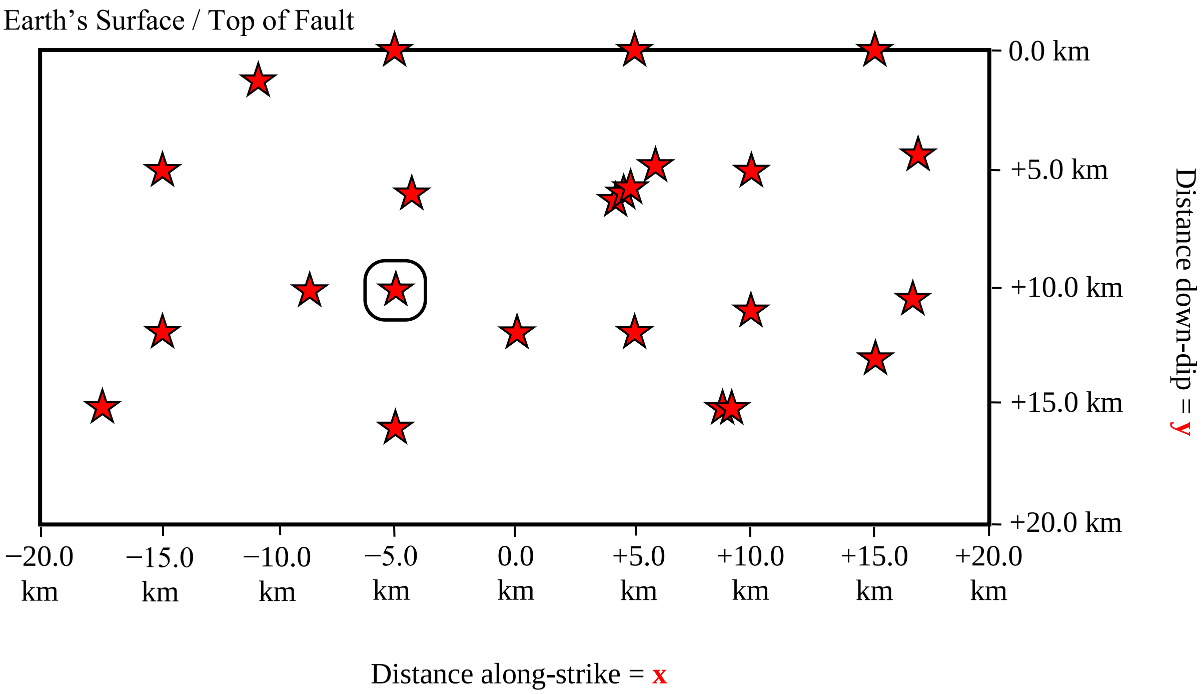
On-Fault Stations, for TPV29 and TPV30	
Station Name	**Location**
faultst-180dp156	On fault, −18.0 km along strike, 15.6 km down-dip.
faultst-150dp050	On fault, −15.0 km along strike, 5.0 km down-dip.
faultst-150dp120	On fault, −15.0 km along strike, 12.0 km down-dip.
faultst-110dp014	On fault, −11.0 km along strike, 1.4 km down-dip.
faultst-089dp101	On fault, −8.9 km along strike, 10.1 km down-dip.
faultst-050dp000	On fault, −5.0 km along strike, 0 km down-dip.
faultst-050dp100	On fault, −5.0 km along strike, 10.0 km down-dip (hypocenter).
faultst-050dp160	On fault, −5.0 km along strike, 16.0 km down-dip.
faultst-042dp061	On fault, −4.2 km along strike, 6.1 km down-dip.
faultst000dp120	On fault, 0 km along strike, 12.0 km down-dip.
faultst043dp062	On fault, 4.3 km along strike, 6.2 km down-dip.
faultst046dp060	On fault, 4.6 km along strike, 6.0 km down-dip.
faultst050dp000	On fault, 5.0 km along strike, 0 km down-dip.
faultst050dp120	On fault, 5.0 km along strike, 12.0 km down-dip.
faultst051dp057	On fault, 5.1 km along strike, 5.7 km down-dip.
faultst059dp047	On fault, 5.9 km along strike, 4.7 km down-dip.
faultst090dp153	On fault, 9.0 km along strike, 15.3 km down-dip.
faultst093dp153	On fault, 9.3 km along strike, 15.3 km down-dip.
faultst100dp050	On fault, 10.0 km along strike, 5.0 km down-dip.

`faultst100dp110`	On fault, 10.0 km along strike, 11.0 km down-dip.
`faultst150dp000`	On fault, 15.0 km along strike, 0 km down-dip.
`faultst150dp130`	On fault, 15.0 km along strike, 13.0 km down-dip.
`faultst167dp105`	On fault, 16.7 km along strike, 10.5 km down-dip.
`faultst170dp045`	On fault, 17.0 km along strike, 4.5 km down-dip.

If you do not have a node at the location of a station, there are two options: (1) you can move the station to the nearest node, or (2) you can interpolate the data values from nodes near the station location.

Note: Location along-strike is measured relative to the origin of the (x, y, z) coordinate system Positive locations are to the right of the origin.

Note: The filenames and descriptions give the x and y coordinates of the station. For example, station `faultst-180dp156` is located at $(x, y, z) = (-18000\text{ m},\ 15600\text{ m},\ z)$ where the value of z is obtained from the fault geometry input file.



Each time series file is an ASCII file that contains 8 data fields, as follows.

On-Fault Time Series Data Fields for TPV29 and TPV30	
Field Name	**Description, Units, and Sign Convention**
`t`	Time (s).
`h-slip`	Horizontal slip along-strike (m). Sign convention: Positive means **right lateral** slip.
`h-slip-rate`	Horizontal slip rate along-strike (m/s). Sign convention: Positive means **right lateral** motion.
`h-shear-stress`	Horizontal shear stress along-strike (MPa). Sign convention: Positive means shear stress that tends to cause **right-lateral** slip.
`v-slip`	Vertical along-dip slip (m). Sign convention: Positive means **downward slip** (that is, the far side of the fault moving downward relative to the near side of the fault).
`v-slip-rate`	Vertical along-dip slip rate (m/s). Sign convention: Positive means **downward motion** (that is, the far side of the fault moving downward relative to the near side of the fault).
`v-shear-stress`	Vertical along-dip shear stress (MPa). Sign convention: Positive means shear stress that tends to cause **downward slip** (that is, the far side of the fault moving downward relative to the near side of the fault).
`n-stress`	Normal stress (MPa). Note: You should report **effective normal stress**, which is defined to be total normal stress minus fluid pressure. Sign convention: Positive means **extension**.

The **near side** of a fault is in the front of the diagram (the $-z$ side of the fault).
The **far side** of a fault is in the back of the diagram (the $+z$ side of the fault).

The on-fault time series file consists of three sections, as follows.

On-Fault Time Series File Format for TPV29 and TPV30	
File Section	**Description**
File Header	A series of lines, each beginning with a # symbol, that gives the following information: • Benchmark problem (TPV29 or TPV30) • Author • Date • Code • Code version • Node spacing or element size • Time step • Number of time steps in file • Station location • Descriptions of data columns (7 lines) • Anything else you think is relevant
Field List	A **single line**, which lists the names of the 8 data fields, in column order, separated by spaces. It should be: `t h-slip h-slip-rate h-shear-stress v-slip v-slip-rate v-shear-stress n-stress` (all on one line). The server examines this line to check that your file contains the correct data fields.
Time History	A series of lines. Each line contains 8 numbers, which give the data values for a single time step. The lines must appear in order of increasing time. **C/C++ users:** For all data fields except the time, we recommend using 14.6E or 14.6e floating-point format. For the time field, we recommend using 20.12E or 20.12e format (but see the note on the next page). **Fortran users:** For all data fields except the time, we recommend using E15.7 or 1PE15.6 floating-point format. For the time field, we recommend using E21.13 or 1PE21.12 format (but see the note on the next page). The server accepts most common numeric formats. If the server cannot understand your file, you will see an error message when you attempt to upload the file.

Note: We recommend higher precision for the time field so the server can tell that your time steps are all equal. (If the server thinks your time steps are not all equal, it will refuse to apply digital filters to your data.) If you use a “simple” time step value like 0.01 seconds or 0.005 seconds, then there is no need for higher precision, and you can write the time using the same precision as all the other data fields. When you upload a file, the server will warn you if it thinks your time steps are not all equal.

Here is an example of an on-fault time-series file. This is an invented file, not real modeling data.

```
# Example on-fault time-series file.
#
# This is the file header:
# problem=TPV29
# author=A.Modeler
# date=2015/01/23
# code=MyCode
# code_version=3.7
# element_size=100 m
# time_step=0.008
# num_time_steps=1625
# location= on fault, 5.0 km along strike, 10.0km down-dip
# Column #1 = Time (s)
# Column #2 = horizontal slip (m)
# Column #3 = horizontal slip rate (m/s)
# Column #4 = horizontal shear stress (MPa)
# Column #5 = vertical slip (m)
# Column #6 = vertical slip rate (m/s)
# Column #7 = vertical shear stress (MPa)
# Column #8 = normal stress (MPa)
#
# The line below lists the names of the data fields:
t h-slip h-slip-rate h-shear-stress v-slip v-slip-rate v-shear-stress n-stress
#
# Here is the time-series data.
# There should be 8 numbers on each line, but this page is not wide enough
#  to show 8 numbers on a line, so we only show the first five.
0.000000E+00   0.000000E+00   0.000000E+00   7.000000E+01   0.000000E+00  ...
5.000000E-03   0.000000E+00   0.000000E+00   7.104040E+01   0.000000E+00  ...
1.000000E-02   0.000000E+00   0.000000E+00   7.239080E+01   0.000000E+00  ...
1.500000E-02   0.000000E+00   0.000000E+00   7.349000E+01   0.000000E+00  ...
2.000000E-02   0.000000E+00   0.000000E+00   7.440870E+01   0.000000E+00  ...
2.500000E-02   0.000000E+00   0.000000E+00   7.598240E+01   0.000000E+00  ...
# ... and so on.
```

Part 10: Off-Fault Stations, and Time-Series File Format

The benchmarks use the 12 off-fault stations listed below. All stations are at the earth’s surface.

Refer to the next page for a diagram of station locations. You need to supply one time-series file for each station.

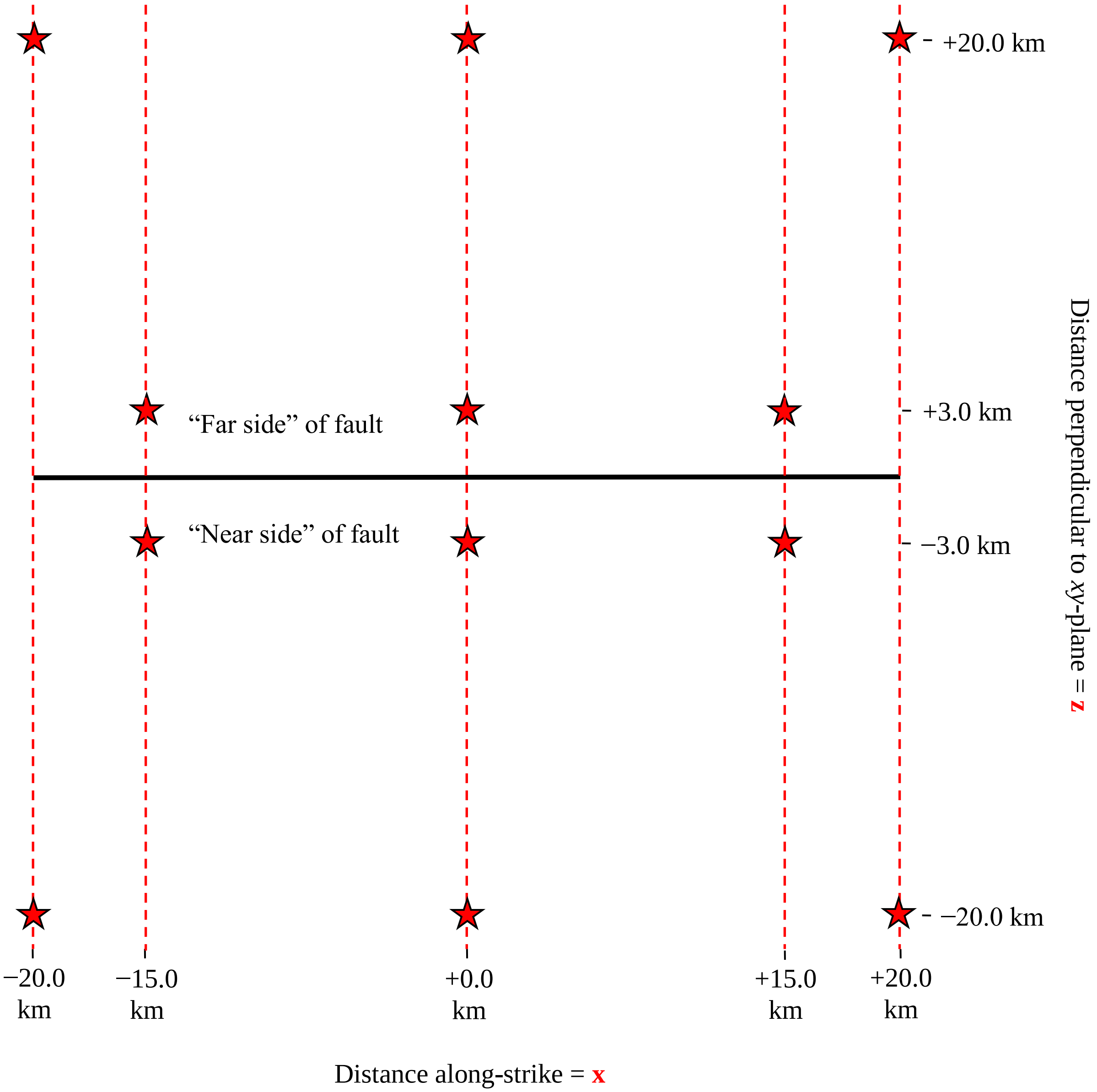
Off-Fault Stations for TPV29 and TPV30	
Station Name	**Location**
`body-200st-200dp000`	−20.0 km off xy-plane (near side), −20.0 km along strike, 0 km depth.
`body-200st000dp000`	−20.0 km off xy-plane (near side), 0.0 km along strike, 0 km depth.
`body-200st200dp000`	−20.0 km off xy-plane (near side), 20.0 km along strike, 0 km depth.
`body-030st-150dp000`	−3.0 km off xy-plane (near side), −15.0 km along strike, 0 km depth.
`body-030st000dp000`	−3.0 km off xy-plane (near side), 0.0 km along strike, 0 km depth.
`body-030st150dp000`	−3.0 km off xy-plane (near side), 15.0 km along strike, 0 km depth.
`body030st-150dp000`	3.0 km off xy-plane (far side), −15.0 km along strike, 0 km depth.
`body030st000dp000`	3.0 km off xy-plane (far side), 0.0 km along strike, 0 km depth.
`body030st150dp000`	3.0 km off xy-plane (far side), 15.0 km along strike, 0 km depth.
`body200st-200dp000`	20.0 km off xy-plane (far side), −20.0 km along strike, 0 km depth.
`body200st000dp000`	20.0 km off xy-plane (far side), 0.0 km along strike, 0 km depth.
`body200st200dp000`	20.0 km off xy-plane (far side), 20.0 km along strike, 0 km depth.

Note: The filenames and descriptions give the (x, y, z) coordinates of the station. For example, station `body-030st-150dp000` is located at $(x, y, z) = (-15000\text{ m},\ 0\text{ m},\ -3000\text{ m})$.

In the station names, the first number is the horizontal perpendicular distance from the station to the xy-plane. A positive number means that the station is located on the **far side**.

If you do not have a node at the location of a station, there are two options: (1) you can move the station to the nearest node, or (2) you can interpolate the data values from nodes near the station location.

Off-Fault Station Locations



The diagram shows the earth's surface, looking downwards.

There are 12 stations at the earth's surface:

- At −20.0 km, +0.0 km, and +20.0 km along-strike, and +20.0 km perpendicular distance from the x-axis.

- At −15.0 km, +0.0 km, and +15.0 km along-strike, and +3.0 km perpendicular distance from the x-axis.

- At −15.0 km, +0.0 km, and +15.0 km along-strike, and −3.0 km perpendicular distance from the x-axis.

- At −20.0 km, +0.0 km, and +20.0 km along-strike, and −20.0 km perpendicular distance from the x-axis.

The **near side** of the fault is in the front of the diagram (the $-z$ side of the fault).
The **far side** of the fault is in the back of the diagram (the $+z$ side of the fault).
Positive perpendicular distance from the xy-plane means that the station is on the **far side**.

Each time series file is an ASCII file that contains 7 data fields, as follows.

Off-Fault Time Series Data Fields for TPV29 and TPV30	
Field Name	**Description, Units, and Sign Convention**
`t`	Time (s).
`h-disp`	Horizontal displacement, parallel to the fault strike (m). Sign convention: Positive means displacement **to the right** relative to the station's initial position (that is, in the $+x$ direction).
`h-vel`	Horizontal velocity, parallel to the fault strike (m/s). Sign convention: Positive means motion **to the right** (that is, in the $+x$ direction).
`v-disp`	Vertical displacement (m). Sign convention: Positive means displacement **downward** relative to the station's initial position (that is, in the $+y$ direction).
`v-vel`	Vertical velocity (m/s). Sign convention: Positive means motion **downward** (that is, in the $+y$ direction).
`n-disp`	Horizontal displacement, perpendicular to the fault strike (m). Sign convention: Positive means displacement **away from the viewer, into the paper** (away from near side of the fault and toward the far side of the fault) relative to the station's initial position. In other words, displacement in the $+z$ direction.
`n-vel`	Horizontal velocity, perpendicular to the fault strike (m/s). Sign convention: Positive means motion **away from the viewer, into the paper** (that is, away from near side of the fault and toward the far side of the fault). In other words, motion in the $+z$ direction.

The **near side** of the fault is in the front of the diagram (the $-z$ side of the fault).
The **far side** of the fault is in the back of the diagram (the $+z$ side of the fault).

The off-fault time series file consists of three sections, as follows.

<table>
<tr><th colspan="2">Off-Fault Time Series File Format for TPV29 and TPV30</th></tr>
<tr><th>File Section</th><th>Description</th></tr>
<tr><td>File Header</td><td>A series of lines, each beginning with a # symbol, that gives the following information:
• Benchmark problem (TPV29 or TPV30)
• Author
• Date
• Code
• Code version
• Node spacing or element size
• Time step
• Number of time steps in file
• Station location
• Descriptions of data columns (7 lines)
• Anything else you think is relevant</td></tr>
<tr><td>Field List</td><td>A <b>single line</b>, which lists the names of the 7 data fields, in column order, separated by spaces. It should be:

<code>t h-disp h-vel v-disp v-vel n-disp n-vel</code>

(all on one line). The server examines this line to check that your file contains the correct data fields.</td></tr>
<tr><td>Time History</td><td>A series of lines. Each line contains 7 numbers, which give the data values for a single time step. The lines must appear in order of increasing time.

<b>C/C++ users:</b> For all data fields except the time, we recommend using 14.6E or 14.6e floating-point format. For the time field, we recommend using 20.12E or 20.12e format (but see the note on the next page).

<b>Fortran users:</b> For all data fields except the time, we recommend using E15.7 or 1PE15.6 floating-point format. For the time field, we recommend using E21.13 or 1PE21.12 format (but see the note on the next page).

The server accepts most common numeric formats. If the server cannot understand your file, you will see an error message when you attempt to upload the file.</td></tr>
</table>

Note: We recommend higher precision for the time field so the server can tell that your time steps are all equal. (If the server thinks your time steps are not all equal, it will refuse to apply digital filters to your data.) If you use a “simple” time step value like 0.01 seconds or 0.005 seconds, then there is no need for higher precision, and you can write the time using the same precision as all the other data fields. When you upload a file, the server will warn you if it thinks your time steps are not all equal.

Here is an example of an off-fault time-series file. This is an invented file, not real modeling data.

```
# Example off-fault time-series file.
#
# This is the file header:
# problem=TPV29
# author=A.Modeler
# date=2015/01/23
# code=MyCode
# code_version=3.7
# element_size=100 m
# time_step=0.008
# num_time_steps=1625
# location= 3.0 km off fault, 15.0 km along strike, 0.0 km depth
# Column #1 = Time (s)
# Column #2 = horizontal displacement (m)
# Column #3 = horizontal velocity (m/s)
# Column #4 = vertical displacement (m)
# Column #5 = vertical velocity (m/s)
# Column #6 = normal displacement (m)
# Column #7 = normal velocity (m/s)
#
# The line below lists the names of the data fields:
t  h-disp  h-vel  v-disp  v-vel  n-disp  n-vel
#
# Here is the time-series data.
# There should be 7 numbers on each line, but this page is not wide enough
#  to show 7 numbers on a line, so we only show the first five.
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00  ...
5.000000E-03  -2.077270E-85  -2.575055E-83  -2.922774E-86  -3.623018E-84  ...
1.000000E-02  -1.622118E-82  -2.005817E-80  -1.387778E-83  -1.713249E-81  ...
1.500000E-02  -9.020043E-80  -1.114231E-77  -4.402893E-81  -5.424313E-79  ...
2.000000E-02  -1.201684E-77  -1.467704E-75  -4.549845E-79  -5.533119E-77  ...
2.500000E-02  -1.528953E-75  -1.866265E-73  -4.126064E-77  -5.004886E-75  ...
# ... and so on.
```

Part 11: Contour-Plot File Format

There is one contour-plot file, as shown here:

Contour-plot file for TPV29 and TPV30	
File Name	**Description**
`cplot`	Rupture times for the fault.

The contour plot file lists the locations of all the nodes on the fault surface, and the time at which each node ruptures.

The contour plot file is an ASCII file that contains three data fields, as follows.

Contour Plot Data Fields for TPV29 and TPV30	
Field Name	**Description, Units, and Sign Convention**
`j`	Distance along strike (m). Sign convention: Positive means a location **to the right** of the origin. For TPV29 and TPV30, the value of `j` can range from −20000 to 20000.
`k`	Distance down-dip (m). Sign convention: Zero is the earth's surface, and positive means **underground**. For TPV29 and TPV30, the value of `k` can range from 0 to 20000.
`t`	Rupture time (s). This is the time at which fault slip-rate first changes from 0 to greater than 0.001 m/s. If this node never ruptures, use the value 1.0E+09.

A pair of numbers `(j,k)` denotes a point on the fault surface. It is equal to the (x, y) coordinates.

The contour plot file consists of three sections, as follows.

Contour Plot File Format for TPV29 and TPV30	
File Section	**Description**
File Header	A series of lines, each beginning with a # symbol, that gives the following information: • Benchmark problem (TPV29 or TPV30) • Author • Date • Code • Code version • Node spacing or element size • Descriptions of data columns (7 lines) • Anything else you think is relevant
Field List	A **single line**, which lists the names of the 3 data fields, in column order, separated by spaces. It should be: `j   k   t` (all on one line). The server examines this line to check that your file contains the correct data fields.
Rupture History	A series of lines. Each line contains three numbers, which give the `(j,k)` coordinates of a node on the fault surface, and the time `t` at which that node ruptures. **C/C++ users:** We recommend using 14.6E or 14.6e floating-point format. **Fortran users:** We recommend using E15.7 or 1PE15.6 floating-point format. If a node never ruptures, the time should be given as 1.0E+09. Nodes may be listed in any order.

Note: The nodes may appear in any order. The nodes do not have to form a rectangular grid, or any other regular pattern.

Note: When you upload a file, the server constructs the Delaunay triangulation of your nodes. Then, it uses the Delaunay triangulation to interpolate the rupture times over the entire fault surface. Finally, it uses the interpolated rupture times to draw a series of contour curves at intervals of 0.5 seconds.

Here is an example of a contour-plot file. This is an invented file, not real modeling data.

```
# Example contour-plot file.
#
# This is the file header:
# problem=TPV29
# author=A.Modeler
# date=2015/01/23
# code=MyCode
# code_version=3.7
# element_size=100 m
# Column #1 = horizontal coordinate, distance along strike (m)
# Column #2 = vertical coordinate, distance down-dip (m)
# Column #3 = rupture time (s)
#
# The line below lists the names of the data fields.
# It indicates that the first column contains the horizontal
#  coordinate (j), the second column contains the vertical
#  coordinate (k), and the third column contains the time (t).
j  k  t
#
# Here is the rupture history
-6.000000E+02   7.000000E+03   3.100000E-02
-6.000000E+02   7.100000E+03   4.900000E-02
-6.000000E+02   7.200000E+03   6.700000E-02
-7.000000E+02   7.000000E+03   1.230000E-01
-7.000000E+02   7.100000E+03   1.350000E-01
-7.000000E+02   7.200000E+03   1.470000E-01
# ... and so on.
```