TPV12 and TPV13

October 1, 2009

Contents

Part 1: What's New

Part 2: Description of the 3D Model

- Fault Geometry
- Material Properties
- Initial Stress Tensor
- Initial Normal and Shear Stress on the Fault
- Friction Parameters and Nucleation
- Running Time, Node Spacing, and Results

Part 3: Description of the 2D Model

Part 4: Linear Slip-Weakening Friction

Part 5: Drucker-Prager Plasticity

- Constitutive Law
- Return Map Algorithm
- Sample Fortran Code
- Test Problem #1 (S-Wave Single-Element Test)
- Test Problem #2 (P-Wave Single-Element Test)

Part 6: Methods for Implementing Gravity and Boundary Tractions

- Method 1 — Using Stress Change
- Method 2 — Using Explicit Gravity and Boundary Tractions

Part 7: On-Fault Stations, and Time-Series File Format

Part 8: Off-Fault Stations, and Time-Series File Format

Part 9: Contour-Plot File Format

Part 10: Using the Web Server to Upload Files

Part 1: What's New

Our goal with TPV12 and TPV13 is to introduce a benchmark with off-fault plastic yielding. To achieve this goal, we had to make several changes from previous benchmarks, which are highlighted below.

- **Benchmark TPV13 uses non-associative Drucker-Prager plasticity with yielding in shear.** Benchmark TPV12, like all prior benchmarks, uses linear elasticity. We include a detailed description of Drucker-Prager plasticity.

- **There are two test problems you can use to check your implementation of plasticity.** We ***strongly*** urge you to run the two test problems before you attempt TPV13. Analytic solutions to the test problems are provided.

- **We are doing both 2D and 3D versions of these benchmarks.** The 2D versions are called TPV12-2D and TPV13-2D. The 3D versions are called simply TPV12 and TPV13.

- **The initial stress tensor is specified everywhere throughout the entire model volume.** Previous benchmarks specified initial stress only on the fault surface. In TPV12 and TPV13, the initial stress on the fault is 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.

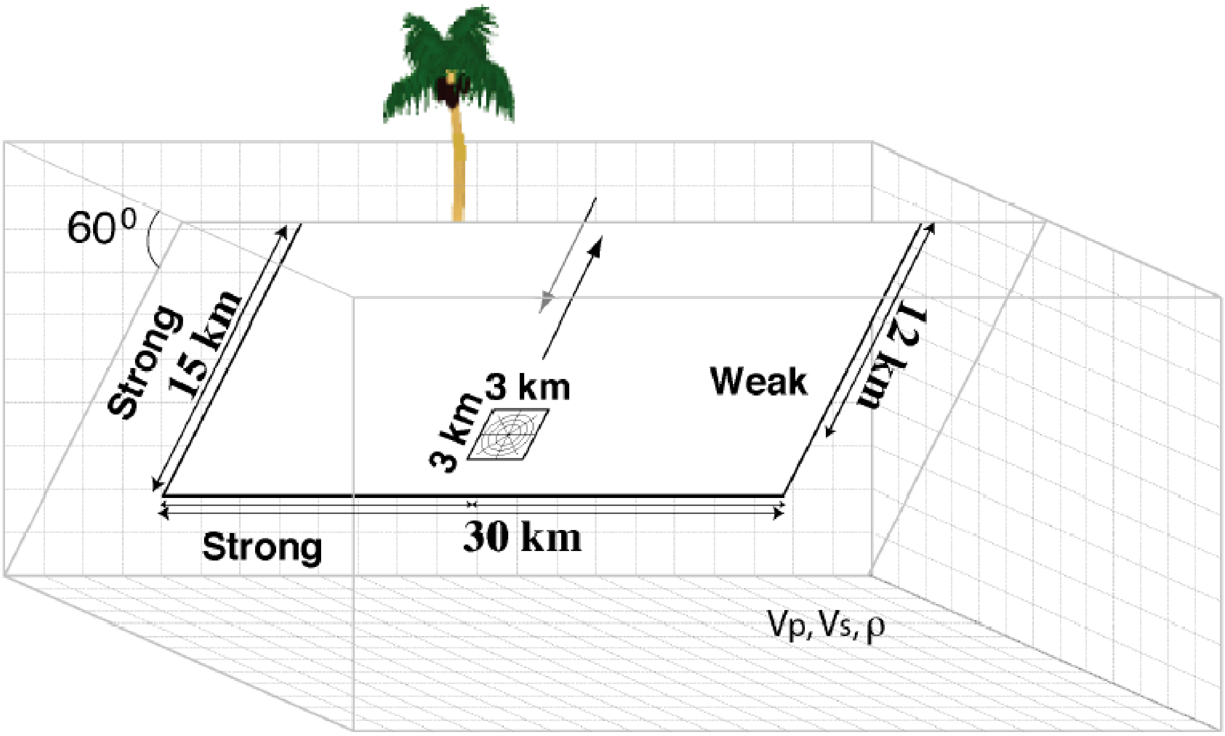
- **Fluid pressure is included in the model.** Fluid pressure is hydrostatic, with water table at the earth's surface. The fluid pressure enters into the plastic constitutive law, and the friction law.

- **There is a new nucleation method.** In previous benchmarks, nucleation was achieved by imposing a higher initial shear stress within a nucleation zone. In TPV12 and TPV13, nucleation is achieved by selecting a lower static coefficient of friction within a nucleation zone, so that the initial shear stress (which is implied by the initial stress tensor) is greater than the yield stress.

We continue to use a linear slip-weakening friction law. As with TPV10 and TPV11, **we use a 60-degree dipping fault in a uniform half-space**.

The parameters for TPV12 and TPV13 are chosen to produce a powerful super-shear rupture. As a result, **you only need to run the model for 8 seconds**.

Part 2: Description of the 3D Model



Our goal is for each modeler to implement *exactly* the model described here.

Fault Geometry

The model volume is a half-space.

The fault is a 60-degree dipping, planar, normal fault. The fault reaches the earth's surface.

Rupture is allowed within a rectangular area measuring 30000 m along-strike and 15000 m down-dip. Note that 15000 m down-dip corresponds to a depth of 12990.38 m. A node which lies exactly on the border of the 30000 m × 15000 m rectangle is considered to be inside the rectangle, and so should be permitted to rupture.

The portions of the fault below, to the left of, and to the right of the 30000 m × 15000 m rectangle are a strength barrier, within which the fault is not allowed to rupture.

The nucleation zone is a square measuring 3000 m × 3000 m. The center of the square is located 12000 m down-dip (at a depth of 10392.30 m), and is centered along-strike.

Material Properties

In TPV12, the entire model volume is a linear elastic material, with the following parameters.

Density $\rho = 2700\ \mathrm{kg/m^3}$

Shear-wave velocity $V_s = 3300\ \mathrm{m/s}$

Pressure-wave velocity $V_p = 5716\ \mathrm{m/s}$

In TPV13, the entire model volume is a non-associative Drucker-Prager plastic material that yields in shear, with the following parameters. A description of Drucker-Prager plasticity is given later, in *Part 5*.

Density $\rho = 2700\ \mathrm{kg/m^3}$

Shear-wave velocity $V_s = 3300\ \mathrm{m/s}$

Pressure-wave velocity $V_p = 5716\ \mathrm{m/s}$

Cohesion $c = 5\ \mathrm{MPa}$

Bulk friction $\nu = 0.85$

The choice of material properties is the only difference between TPV12 and TPV13.

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 TPV12 and TPV13, the initial stress varies only with depth. In this discussion, the sign convention for principal stress components is positive in compression but, of course, the signs need to be reversed to have tensor stress for use in the code.

The initial maximum compressive principal stress, σ_1, is vertical and is equal to the overburden.

$$\begin{aligned}\sigma_1 &= (2700\ \mathrm{kg/m^3})(9.8\ \mathrm{m/s^2})(\text{depth in meters}) \\ &= (26460\ \mathrm{Pa/m})(\text{depth in meters})\end{aligned}$$

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

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

Note that we have chosen the gravitational acceleration g to be *exactly* 9.8 m/s^2. Do not substitute a "more accurate" value of g.

The initial minimum compressive principal stress, σ_3, is horizontal and normal to the fault trace. At down-dip distances less than 13800 meters (that is, at depths less than 11951.15 meters), the initial minimum compressive stress is chosen to satisfy

$$\left(\sigma_3 - P_f\right) = 0.3496\ (\sigma_1 - P_f)$$

which works out to

$$\sigma_3 = (15624.34\ \mathrm{Pa/m})(\mathrm{depth\ in\ meters})$$

The initial intermediate principal stress, σ_2, is horizontal and parallel to the fault trace. It is

$$\sigma_2 = (\sigma_1 + \sigma_3)/2$$

At down-dip distances greater than 13800 meters (that is, at depths greater than 11951.15 meters), shear stress vanishes and all initial principal stresses equal the overburden:

$$\sigma_1 = \sigma_2 = \sigma_3 = (2700\ \mathrm{kg/m^3})(9.8\ \mathrm{m/s^2})(\mathrm{depth\ in\ meters})$$

If an individual element extends both above and below a down-dip distance of 13800 m (depth of 11951.15 m), then we recommend assigning that element a weighted average for σ_3 and σ_2.

Because the initial stress varies with depth, it is necessary to include gravitational forces in the model to balance out the gradient in stress. Also, the initial stress exerts tractions on the mesh boundary, and it may be necessary to balance them out by including externally-applied traction forces. Two basic methods for implementing gravitational forces and boundary tractions are described later, in *Part 6*.

Initial Normal and Shear Stress on the Fault

Ideally, your code would calculate the normal and shear stresses acting on the fault, by resolving the stress tensor onto the fault surface. Then there is no need to separately specify the initial normal and shear stresses on the fault.

But, we recognize that with some codes it may be necessary, or simply convenient, to "manually" assign initial normal and shear stresses to nodes on the fault. So we provide the required formulas.

Rotating the stress state to the 60-degree dipping fault orientation gives initial effective normal stress

$$\left(\sigma_n - P_f\right) = 0.25\left(\sigma_1 - P_f\right) + 0.75\ (\sigma_3 - P_f)$$

and initial down-dip shear stress

$$\tau = (3^{1/2}/4)(\sigma_1 - \sigma_3)$$

At down-dip distances less than 13800 meters (depths less than 11951.15 meters), this works out to

$$\begin{aligned}(\sigma_n - P_f) &= (8533.25\ \mathrm{Pa/m})(\text{depth in meters}) \\ &= (7390.01\ \mathrm{Pa/m})(\text{down-dip distance in meters})\end{aligned}$$

$$\tau = 0.549847(\sigma_n - P_f)$$

At down-dip distances greater than 13800 meters (depths greater than 11951.15 meters), this works out to

$$\begin{aligned}(\sigma_n - P_f) &= (16660\ \mathrm{Pa/m})(\text{depth in meters}) \\ &= (14427.98\ \mathrm{Pa/m})(\text{down-dip distance in meters})\end{aligned}$$

$$\tau = 0$$

A fault node represents a sub-fault with finite extent. If a sub-fault extends both above and below 13800 m down-dip, the initial normal and shear stresses should be a weighted average.

Special considerations apply to a fault node located at the earth's surface. For most codes it is incorrect to assign initial stresses of zero. We recommend that initial stresses 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 initial normal and shear stresses 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 initial normal and shear stresses corresponding to a depth of 1/3 element.

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 $\sigma_n - P_f$.

Friction Parameters and Nucleation

For fault nodes located inside the nucleation zone, we use the following friction parameters.

Static coefficient of friction $\mu_s = 0.54$

Dynamic coefficient of friction $\mu_d = 0.10$

Slip-weakening critical distance $d_0 = 0.50$ m

Frictional cohesion $c_0 = 0.2$ MPa

For fault nodes located outside the nucleation zone but inside the 30000 m × 15000 m rectangular rupture area (including nodes located exactly on the border of the 30000 m × 15000 m rectangle), we use the following friction parameters.

Static coefficient of friction $\mu_s = 0.70$

Dynamic coefficient of friction $\mu_d = 0.10$

Slip-weakening critical distance $d_0 = 0.50\text{ m}$

Frictional cohesion $c_0 = 0.2\text{ MPa}$

A fault node represents a sub-fault with finite extent. If a sub-fault lies partly inside and partly outside the nucleation zone, the friction parameters should be a weighted average.

Example: Suppose that a sub-fault is a square centered on the fault node (this will of course vary with the design of the code). If the fault node is located exactly at a corner of the nucleation zone, its static coefficient of friction should be 0.66 which is 0.75×0.70 + 0.25×0.54. If the fault node is located exactly on an edge of the nucleation zone (but not at a corner), its static coefficient of friction should be 0.62 which is 0.50×0.70 + 0.50×0.54.

Nucleation cannot be done the same way as prior benchmarks, because a patch with elevated initial shear stress would be associated with a 3D variation of stress off the fault. Instead, we specify a reduced static coefficient of friction within the nucleation zone, which causes the initial shear stress to be slightly higher than the yield stress.

Outside the 30000 m × 15000 m rectangular rupture area there is a strength barrier, where nodes are not allowed to slip. Some codes implement the strength barrier by setting the static coefficient of friction and frictional cohesion to very large values. Other codes implement the strength barrier in other ways.

Note: Do not confuse the "frictional cohesion" that appears in the fault friction law with the "cohesion" that appears in the Drucker-Prager plastic constitutive law.

Running Time, Node Spacing, and Results

Run the model for times from **0.0 to 8.0 seconds after nucleation**.

We recommend using **100 m node spacing** on the fault plane.

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 7*.

- **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 8*.

- **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. These files are described in *Part 9*.

Part 3: Description of the 2D Model

One of our objectives for TPV12 and TPV13 is to determine whether a 2D model gives similar results to a 3D model on the fault center line. So, we are doing 2D versions of TPV12 and TPV13, which are called TPV12-2D and TPV13-2D.

The 2D model consists of a planar slice through the 3D model, passing through the center line of the fault. The plane is located at 0 km along-strike. It is a vertical plane, normal to the fault trace.

All the model parameters, initial stresses, material properties, nucleation patch, friction parameters, station locations, and so forth, are exactly the same as in the 3D model.

The fault in the 2D model is a 60-degree dipping line, which reaches the earth's surface. Rupture is permitted along a line extending from the earth's surface to a down-dip distance of 15000 m (depth of 12990.38 m). A node which lies exactly at a down-dip distance of 15000 m should be permitted to rupture.

The nucleation zone is a line 3000 m long. The center of the nucleation zone is located 12000 m down-dip (at a depth of 10392.30 m).

The 2D model has no motion in the along-strike direction. That is, all motion is within the plane.

Run the 2D model for times from **0.0 to 8.0 seconds after nucleation**.

We recommend using **100 m node spacing** on the fault line.

Supply time-series files for on-fault and off-fault stations. The station locations are the same as for the 3D model, except that only stations located at 0 km along-strike are used. File formats are the same as for the 3D model. In the time-series files, data values that refer to motion along-strike should be set to zero.

There is no contour-plot file for the 2D model.

Part 4: Linear Slip-Weakening Friction

Benchmarks TPV12 and TPV13 use linear slip-weakening with frictional cohesion.

This friction law has four parameters:

μ_s = static coefficient of friction.

μ_d = dynamic coefficient of friction.

d_0 = slip-weakening critical distance.

c_0 = frictional cohesion.

The behavior of a node on the fault is influenced by the following variables:

σ_n = total normal stress, taken to be positive in compression.

τ = shear stress.

P_f = fluid pressure.

$\sigma_n - P_f$ = effective normal stress.

Initially, the node is locked. Failure occurs, and the node begins to slip, when

$$\tau > \mu_s \times \max(\sigma_n - P_f, 0) + c_0$$

When the node has slipped a total distance d, the time-varying coefficient of friction μ is

$$\mu = \mu_s + (\mu_d - \mu_s) \times d/d_0 \qquad \text{if } d \le d_0$$

$$\mu = \mu_d \qquad \text{if } d > d_0$$

While the node is slipping, the shear stress on the fault is

$$\tau = \mu \times \max(\sigma_n - P_f, 0) + c_0$$

Tension on the fault: You are likely to encounter tension on the fault at shallow depths. The procedure is to **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 $\max(\sigma_n - P_f,\ 0)$. In particular, notice that when the fault is in tension, the shear stress is equal to the frictional cohesion (and not equal to zero).

We recommend that you **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.

Backwards motion: If you encounter backwards motion, the slip distance d appearing in the formulas above is the total slip distance, not the net slip distance. For example, if the fault slips down 0.3 m and then slips up 0.1 m, the slip distance d is 0.4 m (and not 0.2 m).

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 5: Drucker-Prager Plasticity

Constitutive Law

For TPV13 we are using non-associative Drucker-Prager plasticity with yielding in shear. The material is characterized by six constitutive parameters:

Density $\rho = 2700\ \mathrm{kg/m^3}$

Shear-wave velocity $V_s = 3300\ \mathrm{m/s}$

Pressure-wave velocity $V_p = 5716\ \mathrm{m/s}$

Cohesion $c = 5$ MPa

Bulk friction $\nu = 0.85$

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

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 $F(\sigma) = \sqrt{J_2(\sigma)} - Y(\sigma)$

The Drucker-Prager material is required to satisfy the *yield equation*: $F(\sigma) \leq 0$

When $F(\sigma) < 0$, the material behaves like a linear elastic material, with Lame parameters λ and μ.

When $F(\sigma) = 0$, if the material is subjected to a strain that tends to cause in increase in $F(\sigma)$, then the material *yields*. For TPV13, we assume that the material yields in shear. *Yielding in shear* means that when the material yields, the stress tensor σ_{ij} changes by an amount proportional to the stress deviator s_{ij}, so as to preserve the condition $F(\sigma) = 0$ with no change in mean stress σ_m. This is explained further in the section on the return map algorithm, below.

Note for the mathematically inclined: Yielding in shear corresponds to using a plastic flow potential function $Q(\sigma) = \sqrt{J_2(\sigma)}$.

Return Map Algorithm

The return map algorithm is a popular algorithm for implementing plasticity, 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.

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:

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

Step 2. Calculate the following:

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

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

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

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

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

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

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

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

$$r = Y\left(\sigma^{\text{trial}}\right) / \sqrt{J_2(\sigma^{\text{trial}})}$$

$$\sigma_{ij} = \sigma_m^{\text{trial}}\delta_{ij} + rs_{ij}^{\text{trial}}$$

It is easy to check that in this last case, we obtain $F(\sigma) = 0$ and also $\sigma_m = \sigma_m^{\text{trial}}$.

Sample Fortran Code

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

```
c  ---[ Constitutive parameters ]---
c     Density
      density = 2700.0
c     Vs
      vs = 3300.0
c     Vp
      vp = 5716.0
c     Shear modulus
      mu = density*(vs**2)
c     Lame volumetric parameter
      lambda = density*(vp**2 - 2.0*(vs**2))
c     Cohesion is 5 MPa
      cohes = 5.0e6
c     Bulk friction
      blkfric = 0.85
c     Angle of friction
      angfric = atan(blkfric)
c     Fluid pressre
      fluidpresh = 0.0

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     Stress deviators are adjusted, no change to mean stress
      if (tau .gt. taulim) then
        yldfac = 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
```

Test Problem #1 (S-Wave Single-Element Test)

We are providing two test problems, that you can use to test your implementation of non-associative Drucker-Prager plasticity. The first problem tests the effect of an S-wave on a single element, and the second problem tests the effect of a P-wave on a single element. Both problems have analytic solutions.

This test is performed on a single element.

- For codes that use cubical elements, the single element is a cube oriented with edges parallel to the Cartesian axes. One face of the cube is at $x = 0$ and the opposite face of the cube is at $x = 1$ m.

- For codes that use tetrahedral elements, the single element is a tetrahedron. One face of the tetrahedron is at $x = 0$, and the remaining node is at $x = 1$ m.

Velocity and displacement of all nodes are prescribed as boundary conditions.

- For nodes located at $x = 0$, all components of displacement and velocity remain at zero.

- For nodes located at $x = 1$ m, displacement is initially zero. The y component of velocity is constant at $1\ \mathrm{m/s}$. All other components of velocity are zero.

All six components of the stress tensor in the element are initially zero.

Calculate all six components of the stress tensor as a function of time.

We recommend using a time step of 5.0e−6 seconds, and calculating for times t from 0 to 5.0e−4 seconds.

Use the material properties for TPV13:

Density $\rho = 2700\ \mathrm{kg/m^3}$

Shear-wave velocity $V_s = 3300\ \mathrm{m/s}$

Pressure-wave velocity $V_p = 5716\ \mathrm{m/s}$

Cohesion $c = 5$ MPa

Bulk friction $\nu = 0.85$

Fluid pressure $P_f = 0$

Analytic Solution

For $t \leq 1.2956829537\mathrm{e}{-4}$ seconds, the element behaves elastically, and the stress tensor components are:

$$\sigma_{xy} = 2.9403000000\mathrm{e}{+10} * t$$
$$\sigma_{xx} = \sigma_{xz} = \sigma_{yy} = \sigma_{yz} = \sigma_{zz} = 0$$

For $t > 1.2956829537\mathrm{e}{-4}$ seconds, the element behaves plastically, and the stress components are:

$$\sigma_{xy} = 3.8096965888\mathrm{e}{+6}$$
$$\sigma_{xx} = \sigma_{xz} = \sigma_{yy} = \sigma_{yz} = \sigma_{zz} = 0$$

Test Problem #2 (P-Wave Single-Element Test)

This problem tests the effect of a P-wave on a single element. It is exactly the same as the first problem, above, except for the velocity that is applied to nodes at $x = 1$ m:

- For nodes located at $x = 1$ m, displacement is initially zero. The x component of velocity is constant at $-1\ \mathrm{m/s}$. All other components of velocity are zero.

We recommend using a time step of 5.0e−5 seconds, and calculating for times t from 0 to 5.0e−3 seconds.

Analytic Solution

For $t \leq 1.7246213245\text{e}{-3}$ seconds, the element behaves elastically, and the stress tensor components are:

$$\sigma_{xx} = -8.8216171200\text{e+}10 * t$$
$$\sigma_{yy} = \sigma_{zz} = -2.9410171200\text{e+}10 * t$$
$$\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$$

For $t > 1.7246213245\text{e}{-3}$ seconds, the element behaves plastically, and the stress components are:

$$\sigma_{xx} = -8.5665432299\text{e+}10 * t - 4.3990587021\text{e+}6$$
$$\sigma_{yy} = \sigma_{zz} = -3.0685540651\text{e+}10 * t + 2.1995293511\text{e+}6$$
$$\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$$

Part 6: 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. The formulas were given above, in *Part 2*. 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 TPV12 and TPV13, 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

$$\rho = 1000\ \mathrm{kg/m^3}$$
$$\sigma_1 = \sigma_2 = \sigma_3 = (1000\ \mathrm{kg/m^3})(9.8\ \mathrm{m/s^2})(\mathrm{depth\ in\ meters})$$

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.)

Part 7: On-Fault Stations, and Time-Series File Format

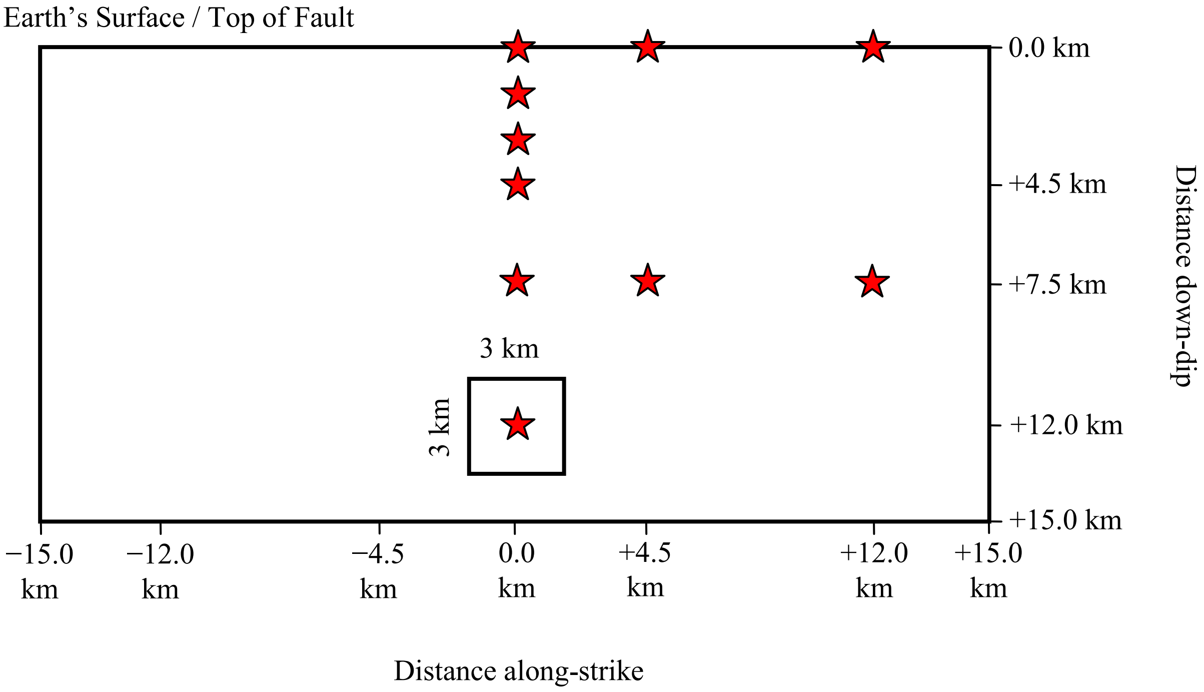
The 3D benchmarks use the 10 on-fault stations listed below. The 2D benchmarks use the 6 stations located at 0 km along strike. Refer to the next page for a diagram of station locations.

You need to supply one time-series file for each station.

On-Fault Stations for TPV12 and TPV13	
Station Name	**Location**
`faultst000dp000`	On fault, 0 km along strike, 0 km down-dip.
`faultst045dp000`	On fault, 4.5 km along strike, 0 km down-dip.
`faultst120dp000`	On fault, 12.0 km along strike, 0 km down-dip.
`faultst000dp015`	On fault, 0 km along strike, 1.5 km down-dip.
`faultst000dp030`	On fault, 0 km along strike, 3.0 km down-dip.
`faultst000dp045`	On fault, 0 km along strike, 4.5 km down-dip.
`faultst000dp075`	On fault, 0 km along strike, 7.5 km down-dip.
`faultst045dp075`	On fault, 4.5 km along strike, 7.5 km down-dip.
`faultst120dp075`	On fault, 12.0 km along strike, 7.5 km down-dip.
`faultst000dp120`	On fault, 0 km along strike, 12.0 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.

On-Fault Station Locations



There are 3 stations at the earth's surface:

- 0 km, +4.5 km, +12.0 km along-strike, and 0 km down-dip distance.

There are 7 deeper stations:

- 0 km along-strike, and 1.5 km, 3.0 km, 4.5 km, 7.5 km, 12.0 km down-dip distance.
- +4.5 km along-strike, and 7.5 km down-dip distance.
- +12.0 km along-strike, and 7.5 km down-dip distance.

Note that the vertical position of a station is specified as distance down-dip, *not* depth. For example, a station at 7.5 km down-dip is located at a depth of 6.495 km.

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

On-Fault Time Series Data Fields for TPV12 and TPV13	
Field Name	**Description, Units, and Sign Convention**
`t`	Time (s).
`h-slip`	Horizontal slip along-strike (m). Sign convention: Positive means **right lateral** slip. Note: For the 2D benchmarks, this field should be zero (or very small).
`h-slip-rate`	Horizontal slip rate along-strike (m/s). Sign convention: Positive means **right lateral** motion. Note: For the 2D benchmarks, this field should be zero (or very small).
`h-shear-stress`	Horizontal shear stress along-strike (MPa). Sign convention: Positive means shear stress that tends to cause **right-lateral** slip. Note: For the 2D benchmarks, this field should be zero (or very small).
`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). In other words, positive means **normal faulting**.
`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). In other words, positive means **normal faulting**.
`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). In other words, positive means stress that tends to cause **normal faulting**.
`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 the fault is the **footwall**.
The **far side** of the fault is the **hanging wall**.

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

On-Fault Time Series File Format for TPV12 and TPV13	
File Section	**Description**
File Header	A series of lines, each beginning with a # symbol, that gives the following information: • Benchmark problem (TPV12 or TPV13) • Author • Date • Code • Code version (if desired) • 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 floating-point format. For the time field, we recommend using E21.13 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=TPV12
# author=A.Modeler
# date=2009/10/29
# code=MyCode
# code_version=3.7
# element_size=100 m
# time_step=0.005
# num_time_steps=1600
# location= on fault, 12 km along strike, 7.5km 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 8: Off-Fault Stations, and Time-Series File Format

The 3D benchmarks use the 12 off-fault stations listed below. The 2D benchmarks use the 10 stations located at 0 km along strike. 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 TPV12 and TPV13

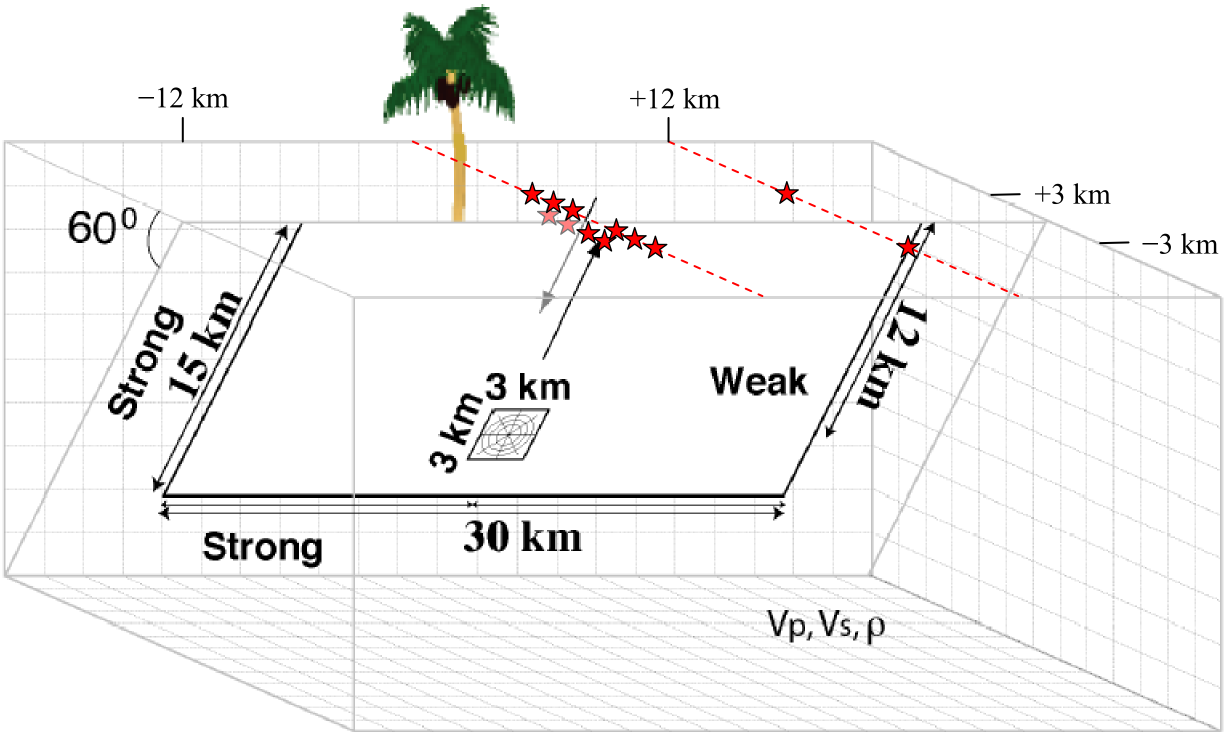
Station Name	Location
`body-030st000dp000`	-3.0 km off fault (near side), 0 km along strike, 0 km depth.
`body-020st000dp000`	-2.0 km off fault (near side), 0 km along strike, 0 km depth.
`body-010st000dp000`	-1.0 km off fault (near side), 0 km along strike, 0 km depth.
`body010st000dp000`	1.0 km off fault (far side), 0 km along strike, 0 km depth.
`body020st000dp000`	2.0 km off fault (far side), 0 km along strike, 0 km depth.
`body030st000dp000`	3.0 km off fault (far side), 0 km along strike, 0 km depth.
`body-010st000dp003`	-1.0 km off fault (near side), 0 km along strike, 0.3 km depth.
`body-005st000dp003`	-0.5 km off fault (near side), 0 km along strike, 0.3 km depth.
`body005st000dp003`	0.5 km off fault (far side), 0 km along strike, 0.3 km depth.
`body010st000dp003`	1.0 km off fault (far side), 0 km along strike, 0.3 km depth.
`body-030st120dp000`	-3.0 km off fault (near side), 12.0 km along strike, 0 km depth.
`body030st120dp000`	3.0 km off fault (far side), 12.0 km along strike, 0 km depth.

In the station names, the first number is the horizontal perpendicular distance from the station to the slanting fault surface. A positive number means that the station is located on the **far side** of the fault.

The **near side** of the fault is the **footwall**.
The **far side** of the fault is the **hanging wall**.

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



There are 8 stations at the earth's surface:

- 0 km along strike, 0 km depth, and ±1.0 km, ±2.0 km, ±3.0 km perpendicular distance **from the fault trace**.
- +12 km along strike, 0 km depth, and ±3.0 km perpendicular distance **from the fault trace**.

There are 4 deeper stations:

- 0 km along strike, 0.3 km depth, and ±0.5 km, ±1.0 km perpendicular distance **from the fault surface at a depth of 0.3 km**.

The **near side** of the fault is in the front of the diagram.
The **far side** of the fault is in the back of the diagram.
Positive perpendicular distance from the fault 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 TPV12 and TPV13	
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. Note: For the 2D benchmarks, this field should be zero (or very small).
`h-vel`	Horizontal velocity, parallel to the fault strike (m/s). Sign convention: Positive means motion **to the right**. Note: For the 2D benchmarks, this field should be zero (or very small).
`v-disp`	Vertical displacement (m). Sign convention: Positive means displacement **downward** relative to the station's initial position.
`v-vel`	Vertical velocity (m/s). Sign convention: Positive means motion **downward**.
`n-disp`	Horizontal displacement, perpendicular to the fault strike (m). Sign convention: Positive means displacement **away from the viewer, into the paper** (that is, away from near side of the fault and toward the far side of the fault) relative to the station's initial position.
`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).

The **near side** of the fault is the **footwall**.
The **far side** of the fault is the **hanging wall**.

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

Off-Fault Time Series File Format for TPV12 and TPV13	
File Section	**Description**
File Header	A series of lines, each beginning with a # symbol, that gives the following information: • Benchmark problem (TPV12 or TPV13) • Author • Date • Code • Code version (if desired) • 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 7 data fields, in column order, separated by spaces. It should be: `t  h-disp  h-vel  v-disp  v-vel  n-disp  n-vel` (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 7 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 floating-point format. For the time field, we recommend using E21.13 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 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=TPV12
# author=A.Modeler
# date=2009/10/29
# code=MyCode
# code_version=3.7
# element_size=100 m
# time_step=0.005
# num_time_steps=1600
# location= 3.0 km off fault, 12 km along strike, 0.0km 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 9: Contour-Plot File Format

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

There is no contour-plot file for the 2D benchmarks.

The contour plot file is an ASCII file that contains three data fields, as follows. See `Signconvention3d.pdf` for a diagram illustrating coordinates and sign conventions.

Contour Plot Data Fields for TPV12 and TPV13	
Field Name	**Description, Units, and Sign Convention**
`j`	Distance along strike (m). Sign convention: Positive means a location **to the right** of the hypocenter. For TPV12 and TPV13, the value of `j` can range from -15000 to 15000.
`k`	Distance down-dip (m). Sign convention: Zero is the earth's surface, and positive means **underground**. Note that `k` is distance along-dip, and *not* depth below the earth's surface. For TPV12 and TPV13, the value of `k` can range from 0 to 15000.
`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.

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

Contour Plot File Format for TPV12 and TPV13

File Section	**Description**
File Header	A series of lines, each beginning with a # symbol, that gives the following information: • Benchmark problem (TPV12 or TPV13) • Author • Date • Code • Code version (if desired) • 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 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=TPV12
# author=A.Modeler
# date=2009/01/23
# 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.
```

Part 10: Using the Web Server to Upload Files

The web server lets you upload your modeling results. Once uploaded, you and other modelers can view the data in various ways.

Logging in and Selecting a Problem

To log in, start your web browser and go to the home page at:

```
http://scecdata.usc.edu/cvws/cgi-bin/cvws.cgi
```

Click on “Upload Files,” and then log in using your user name and password. Remember that passwords are case-sensitive.

You are then presented with a list of code validation problems. Find the problem you want, and click the “Select” button. You will then see a list of files for the problem.

Navigating the Site

You navigate through the site by clicking buttons on the web pages. Avoid using your browser’s Back or Forward buttons. If you use the Back or Forward buttons, you may get error messages from your browser.

Uploading a File

To upload a file, do the following:

1. Find the file you want to upload, and click “Select.” The server displays a page for you to upload the file.

2. Select the file on your computer that contains the modeling data. The exact method you use to select the file varies depending on which operating system and web browser you are using.

3. Click on “Click Once to Upload.” The file you selected is uploaded to the server.

When you upload a file, the web server immediately checks it to see that it is correctly formatted. There are three possible results:

- If the file is correctly formatted, the server displays a page saying that the upload succeeded.

- If the file contains errors, the server displays an error log. The error log lists the errors that were detected in the file, each identified as specifically as possible.

- If the file is correctly formatted, but is questionable in some way (for example, a missing time step), then the server displays a warning log, which describes the problem.

After uploading a file, the file list shows the date and time that you uploaded the file.

Remember that any file you upload will be visible to anyone who has access to the web site.

After uploading a file, additional functions become available. These functions let you graph, view, or delete the uploaded file.

Graphing a File

To graph a file, find the file you want and click “Graph.”

For a time-series file, the server displays graphs of all the data fields in the file. For a contour-plot file, the server displays the contour plot.

At the bottom of each graph page, there is a box you can use to adjust graphing preferences.

For a time-series file, you can apply a low-pass filter to remove high-frequency noise and oscillations. To apply a low-pass filter, scroll to the bottom of the page, enter the desired cutoff frequency, and click “Apply Changes.”

Graphing a file is a good way to check that the server is interpreting your data as you intended.

Viewing a File

To view the text of a file, find the file you want and click “View.”

Deleting a File

To delete a file from the server, find the file you want and click “Delete.” The server displays a page asking you to confirm the deletion.

If you want to replace a file, it is not necessary to delete the old file. Just upload your new file, and it automatically replaces the old file.