

Uploading Data for TPV101 and TPV102

February 10, 2007

Contents:

1. Introduction.
2. On-Fault Time Series Data Files.
3. Off-Fault Time Series Data Files (TPV102 Only).
4. Contour Plot Data Files.
5. Using the Web Server to Upload Files.

1 Introduction

Files are uploaded to the SCEC code validation web server at this address:

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

You need to upload three kinds of files:

- On-fault time series files, which give slips, slip rates, stresses, and state variable for each on-fault station at each time step.
- Off-fault time series files, which give displacements and velocities for each off-fault station at each time step (for TPV102 only).
- Contour plot files, which give the locations and rupture times of each node on the fault surface.

Section 2 describes the on-fault time series file format, and gives an example of an on-fault time series file. Section 3 describes the off-fault time series file format, and gives an example of an off-fault time series file. Section 4 describes the contour plot file format, and gives an example of a contour plot file. Section 5 has instructions for uploading your files to the server.

Direct questions to Michael Barall.

2 On-Fault Time Series Data Files

Time series data is supplied as a set of ASCII files. You need to supply one file for each station. There are 9 stations, as follows. Refer to the benchmark problem description for a diagram of station locations.

On-Fault Stations for TPV101 and TPV102	
Station Name	Location
faultst-120dp030	On fault, -12.0 km along strike, 3.0 km down-dip.
faultst000dp030	On fault, 0 km along strike, 3.0 km down-dip.
faultst120dp030	On fault, 12.0 km along strike, 3.0 km down-dip.
faultst-090dp075	On fault, -9.0 km along strike, 7.5 km down-dip.
faultst000dp075	On fault, 0 km along strike, 7.5 km down-dip.
Faultst090dp075	On fault, 9.0 km along strike, 7.5 km down-dip.
faultst-120dp120	On fault, -12.0 km along strike, 12.0 km down-dip.
faultst000dp120	On fault, 0 km along strike, 12.0 km down-dip.
faultst120dp120	On fault, 12.0 km along strike, 12.0 km down-dip.

Each time series file contains 9 data fields, as follows. Refer to the benchmark problem description for a diagram illustrating sign conventions.

On-Fault Time Series Data Fields for TPV101 and TPV102	
Field Name	Description, Units, and Sign Convention
t	Time (s).
h-slip	Horizontal slip (m). Sign convention: Positive means right lateral slip.
h-slip-rate	Horizontal velocity (m/s). Sign convention: Positive means right lateral motion.
h-shear-stress	Horizontal shear stress (MPa). Sign convention: Positive means shear stress that tends to cause right-lateral slip.
v-slip	Vertical displacement (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 velocity (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 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). Sign convention: Positive means compression .
log-theta	Logarithm to base 10 of the state variable. Units: The state variable theta is in seconds.

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

On-Fault Time Series File Format for TPV101 and TPV102	
File Section	Description
File Header	<p>A series of lines, each beginning with a # symbol, that give the following information:</p> <ul style="list-style-type: none"> • Benchmark problem (TPV101 or TPV102) • 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 (9 lines) • Anything else you think is relevant
Field List	<p>A single line, which lists the names of the 9 data fields, in column order, separated by spaces. It should be:</p> <pre>t h-slip h-slip-rate h-shear-stress v-slip v-slip-rate v-shear-stress n-stress log-theta</pre> <p>(all on one line). The server examines this line to check that your file contains the correct data fields.</p>
Time History	<p>A series of lines. Each line contains 9 numbers, which give the data values for a single time step. The lines must appear in order of increasing time.</p> <p>C/C++ users: For all data fields <i>except</i> 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).</p> <p>Fortran users: For all data fields <i>except</i> 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).</p> <p>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.</p>

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 time-series file.
#
# This is the file header:
# problem=TPV102
# author=A.Modeler
# date=2008/01/23
# code=MyCode
# code_version=3.7
# element_size=100 m
# time_step=0.005
# num_time_steps=2400
# location= on fault, 9 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)
# Column #9 = log10 of state variable (log-seconds)
#
# The line below lists the names of the data fields:
# (Although rendered as two lines on this printed page, it must be one
# single line in the actual file.)
t h-slip h-slip-rate h-shear-stress v-slip v-slip-rate v-shear-stress
  n-stress log-theta
#
# Here is the time-series data.
# There should be 9 numbers on each line, but this page is not wide enough
# to show 9 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.
```

3 Off-Fault Time Series Data Files (TPV102 Only)

Off-fault time series data is supplied as a set of ASCII files. You need to supply one file for each station. There are 6 stations, as follows. Refer to the benchmark problem description for a diagram of station locations.

Only TPV102 has off-fault stations.

Off-Fault Stations for TPV102	
Station Name	Location
body-060s-120dp000	-6.0 km off fault (near side), -12.0 km along strike, 0 km depth.
body-090st000dp000	-9.0 km off fault (near side), 0 km along strike, 0 km depth.
body-060st120dp000	-6.0 km off fault (near side), 12.0 km along strike, 0 km depth.
body060s-120dp000	6.0 km off fault (far side), -12.0 km along strike, 0 km depth.
body090st000dp000	9.0 km off fault (far side), 0 km along strike, 0 km depth.
body060st120dp000	6.0 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 fault. A positive number means that the station is located on the **far side** of the fault.

Each time series file contains 7 data fields, as follows. Refer to the benchmark problem description for a diagram illustrating sign conventions.

Off-Fault Time Series Data Fields for TPV102	
Field Name	Description, Units, and Sign Convention
t	Time (s).
h-disp	Horizontal displacement, parallel to fault (m). Sign convention: Positive means displacement to the right relative to the station's initial position.
h-vel	Horizontal velocity, parallel to fault (m/s). Sign convention: Positive means motion to the right .
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 fault (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 fault (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 off-fault time series file consists of three sections, as follows:

Off-Fault Time Series File Format for TPV102	
File Section	Description
File Header	<p>A series of lines, each beginning with a # symbol, that give the following information:</p> <ul style="list-style-type: none"> • Benchmark problem (TPV102) • 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	<p>A single line, which lists the names of the 7 data fields, in column order, separated by spaces. It should be:</p> <pre>t h-disp h-vel v-disp v-vel n-disp n-vel</pre> <p>(all on one line). The server examines this line to check that your file contains the correct data fields.</p>
Time History	<p>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.</p> <p>C/C++ users: For all data fields <i>except</i> 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).</p> <p>Fortran users: For all data fields <i>except</i> 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).</p> <p>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.</p>

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 time-series file.
#
# This is the file header:
# problem=TPV102
# author=A.Modeler
# date=2008/01/23
# code=MyCode
# code_version=3.7
# element_size=100 m
# time_step=0.005
# num_time_steps=2400
# location= 6.0 km off fault, 12 km along strike, 0.0 km down-dip
# 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.
```

4 Contour Plot Data Files

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.

Each file contains three data fields, as follows. See the benchmark problem description for a diagram illustrating coordinates and sign conventions.

Contour Plot Data Fields for TPV101 and TPV102	
Field Name	Description, Units, and Sign Convention
j	<p>Distance along strike (m).</p> <p>Sign convention: Positive means a location to the right of the hypocenter.</p> <p>For TPV101 and TPV102, the value of j can range from -15000 to 15000.</p>
k	<p>Distance down-dip (m).</p> <p>Sign convention: Zero is the earth's surface, and positive means underground.</p> <p>(For TPV101, zero is the top of the fault surface, and points inside the fault surface have positive values of k.)</p> <p>For TPV101 and TPV102, the value of k can range from 0 to 15000.</p>
t	<p>Rupture time (s).</p> <p>This is the time at which fault slip-rate first changes from zero to greater than 1 mm/s.</p> <p>If this node never ruptures, use the value 1.0E+09.</p>

A pair of numbers (j, k) denotes a point on the fault surface.

The new contour plot file consists of three sections, as follows:

Contour Plot File Format for TPV101 and TPV102	
File Section	Description
File Header	<p>A series of lines, each beginning with a # symbol, that give::</p> <ul style="list-style-type: none"> • Benchmark problem (TPV101 or TPV102) • Author • Date • Code • Code version (if desired) • Node spacing or element size • Descriptions of data columns (3 lines) • Anything else you think is relevant
Field List	<p>A single line, which lists the names of the three data fields, in column order, separated by spaces. It should be:</p> <p style="text-align: center;">j k t</p> <p>(on one line).</p>
Rupture History	<p>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.</p> <p>C/C++ users: We recommend using 14.6E or 14.6e floating-point format.</p> <p>Fortran users: We recommend using E15.7 floating-point format.</p> <p>If a node never ruptures, the time should be given as 1.0E+09.</p> <p>Nodes may be listed in any order.</p>

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 new contour-plot file.
#
# This is the file header:
# problem=TPV102
# author=A.Modeler
# date=2008/01/23
# 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.
```

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.

4.1 Logging in and Selecting a Problem

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

```
http://sceccdata.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.

4.2 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.

4.3 Uploading a File

To upload a file, do the following:

Step 1. Find the file you want to upload, and click “Select.”

The server displays a page for you to upload the file.

Step 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.

Step 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.

4.4 Graphing, Viewing, and Deleting Files

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

4.4.1 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.

4.4.2 Viewing a File

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

4.4.3 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.