The SCEC/USGS Spontaneous Rupture Code Verification Project

Benchmark Comparison Tool

Benchmark Descriptions

Newest Benchmarks

Metrics (Version 1)

Code Descriptions

Michael Barall

Invisible Software, Inc.

SCEC Rupture Dynamics Code Verification Workshop, March 14, 2014

Plan

- Demonstrate metrics on the website.

- Issues for computing metrics.

- Metric formulas.

- Computational steps.

- Conclusions.

Issues for Computing Metrics

Goal: Generate Comparison Metrics for the CVWS Website

The website is big! And getting bigger all the time.

- 8.6 GB of data.

- 39 benchmarks.

- 36 participating modelers.

- 10,000 uploaded files.

- Average of 14 submissions per benchmark.

- 560,000 data pairs available to be compared!
  (excluding comparisons between different stations, or between different benchmarks)

Metric generation must be (almost) fully automatic.

Metric values are absolute (that is, not depending on the range of results submitted), so they don't change as new results are uploaded.

Available Data

On-fault time series:

- Gives slip, slip rate, shear stress, and normal stress as a function of time.
- Given at a set of pre-selected on-fault station locations (average 10 per benchmark).
- Each modeler selects their own time step value.

Off-fault time series:

- Gives displacement and velocity as a function of time.
- Given at a set of pre-selected off-fault station locations (average 8 per benchmark).
- Each modeler selects their own time step value.

Rupture contour plot:

- Gives the time at which each point on the fault surface(s) begins to slip.
- Each modeler selects their own array of points on the fault surface(s).

Metrics must be computed only from this data.

Metric computation must deal with the fact that modeling results are sampled at different times and different locations.
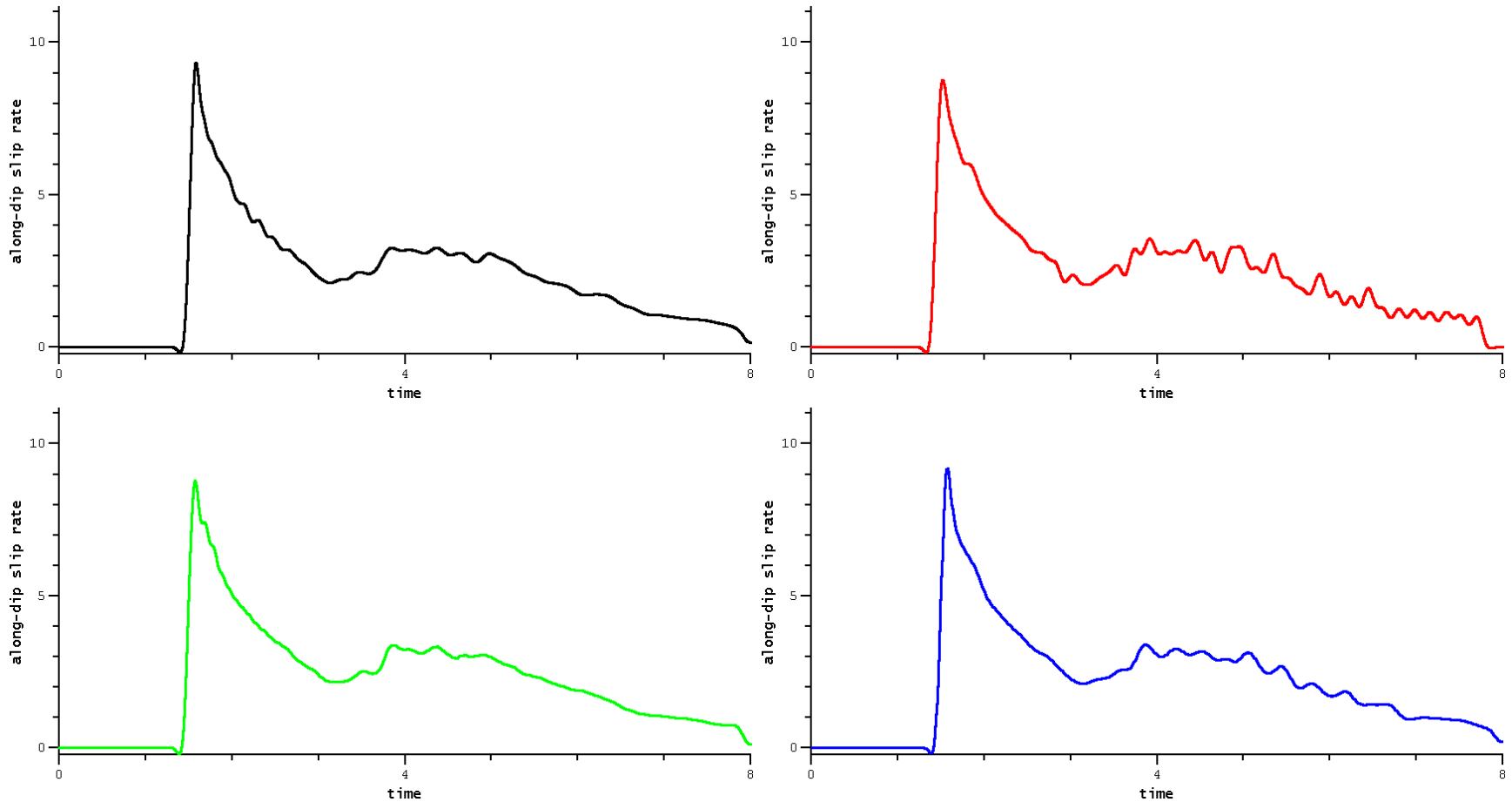
Filtering
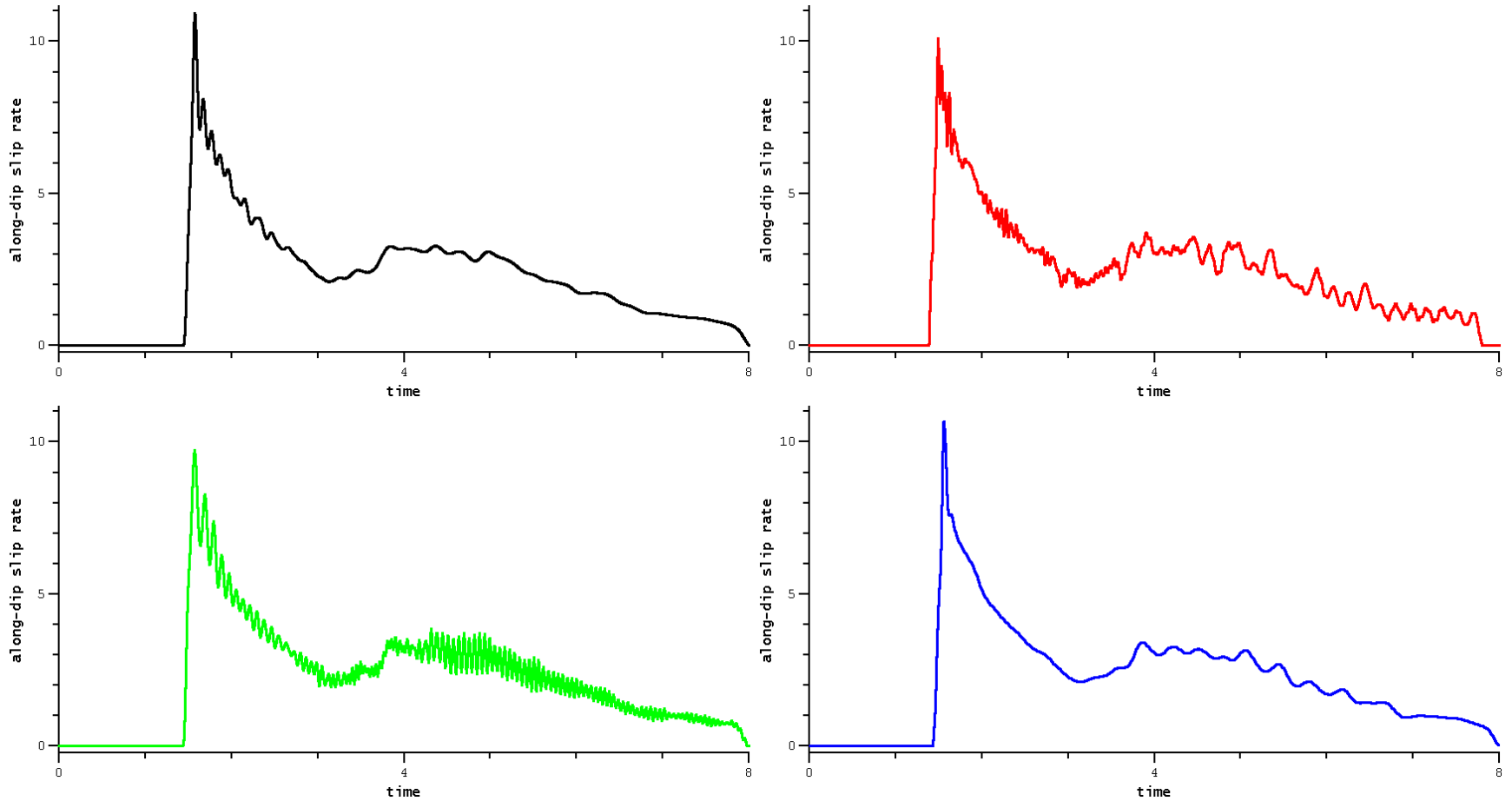
The above pictures show **unfiltered** data.

Codes differ in the amount and type of oscillations they produce. Some codes seem to have built-in filtering, others seem to have none.
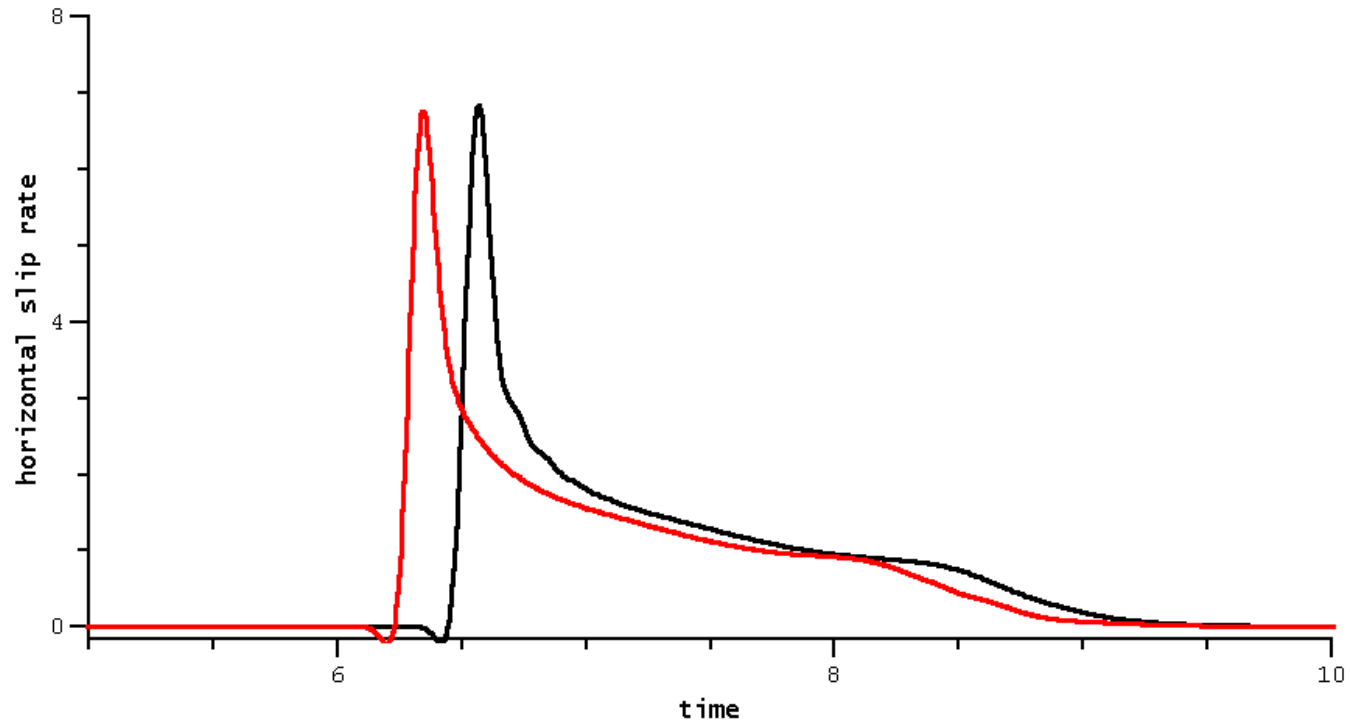
Filtering

The above pictures show the same data **filtered at 5Hz**.

We apply a 5 Hz low-pass filter to velocity, slip rate, and shear and normal stress.

Time Shifts

Codes may produce very similar waveforms, which differ just by a shift in time.

A metric that just subtracts one time series from the other would understate the agreement.

When comparing two time-series, we shift the relative time to produce an optimum fit.

Vector Components

Many time series include components that are close to zero, and may be dominated by noise.

Example: Vertical slip rate, in a strike-slip benchmark.

Comparisons of near-zero components are not meaningful. But we don't want to just throw them away, because, for example, some strike-slip benchmarks have significant vertical slip at some stations.

> In a time series, components are combined to form a *vector-valued* time series, and all metrics are computed using the *vector* values.

At on-fault stations:

- Horizontal and vertical slip are combined to form a 2D slip vector.
- Horizontal and vertical slip rate are combined to form a 2D slip rate vector.
- Horizontal and vertical shear stress are combined to form a 2D shear stress vector.

At off-fault stations:

- Horizontal, vertical, and normal velocity are combined to form a 3D velocity vector.
- Horizontal, vertical, and normal displacement are combined to form a 3D displacement vector.

This also reduces the number of metric values.

Rupture Boundaries
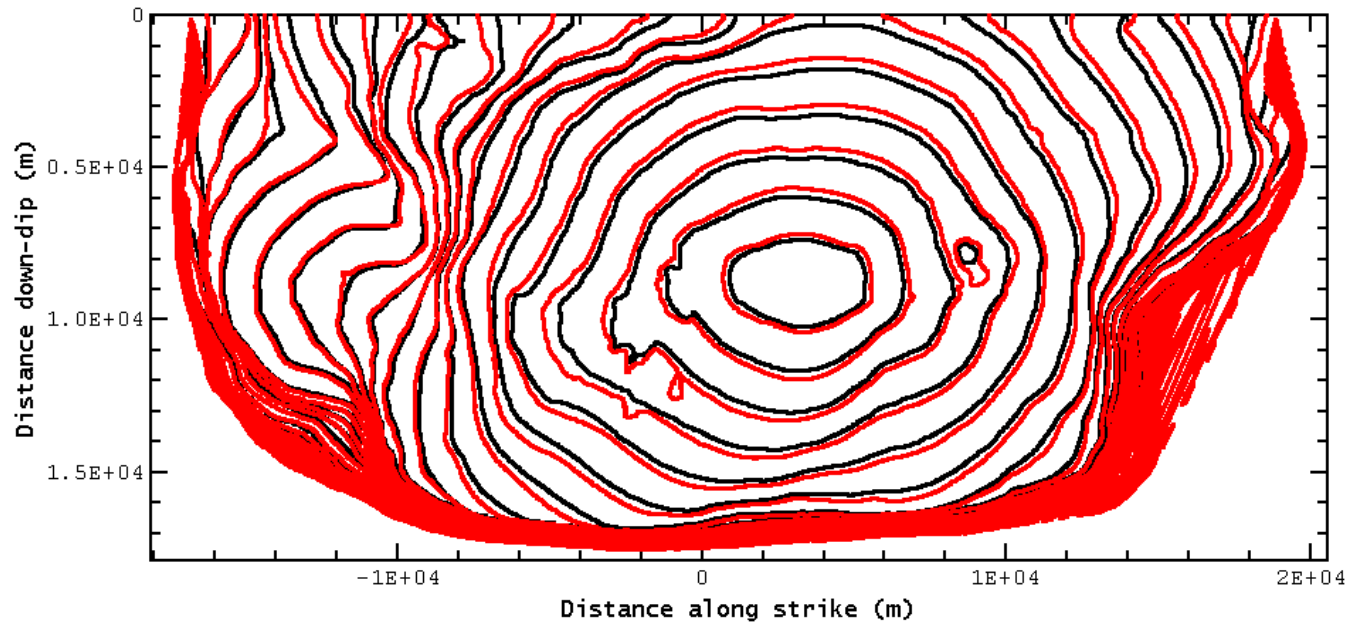
Some benchmarks have ruptures that slow down and stop.

A metric that compares rupture times would see a large difference in areas where the rupture propagates slowly.

When comparing rupture times, we exclude areas where the rupture velocity is less than 1 km/sec.

Software Design

Existing website code:

- Written in C.
- Manages the database of modeling results.
- Computes everything on the fly, as requested.
- Generated web page output in HTML 3.2.

New metric code:

- Written in C++.
- Needs access to the modeling database.
- Runs off-line because metrics take a long time to compute.
- Generates web page output in HTML 5.

It is not good design to have two independent pieces of code accessing the same database.

Before starting this project, the website code was revised so that:

- It generates output in HTML 5.
- It can be compiled as a library, so that external programs can call it.
- There is a C++ "wrapper" that provides access to the C library from C++.

The metric code calls the website code, both to obtain input from the database, and to generate HTML 5 output.

Metric Formulas

Metric Formula for Time Series

For Metrics Version 1, we use a normalized root-mean-square difference, with a time shift, expressed as a percentage.

For a function of time $f(t)$, which may be vector-valued, define the L^2 norm:

$$\|f(t)\|_2 \equiv \left[\int |f(t)|^2 \, dt \right]^{1/2}$$

Our main misfit metric is a normalized RMS difference. Given two functions $f(t)$ and $g(t)$, and a time shift t_s, the metric value is:

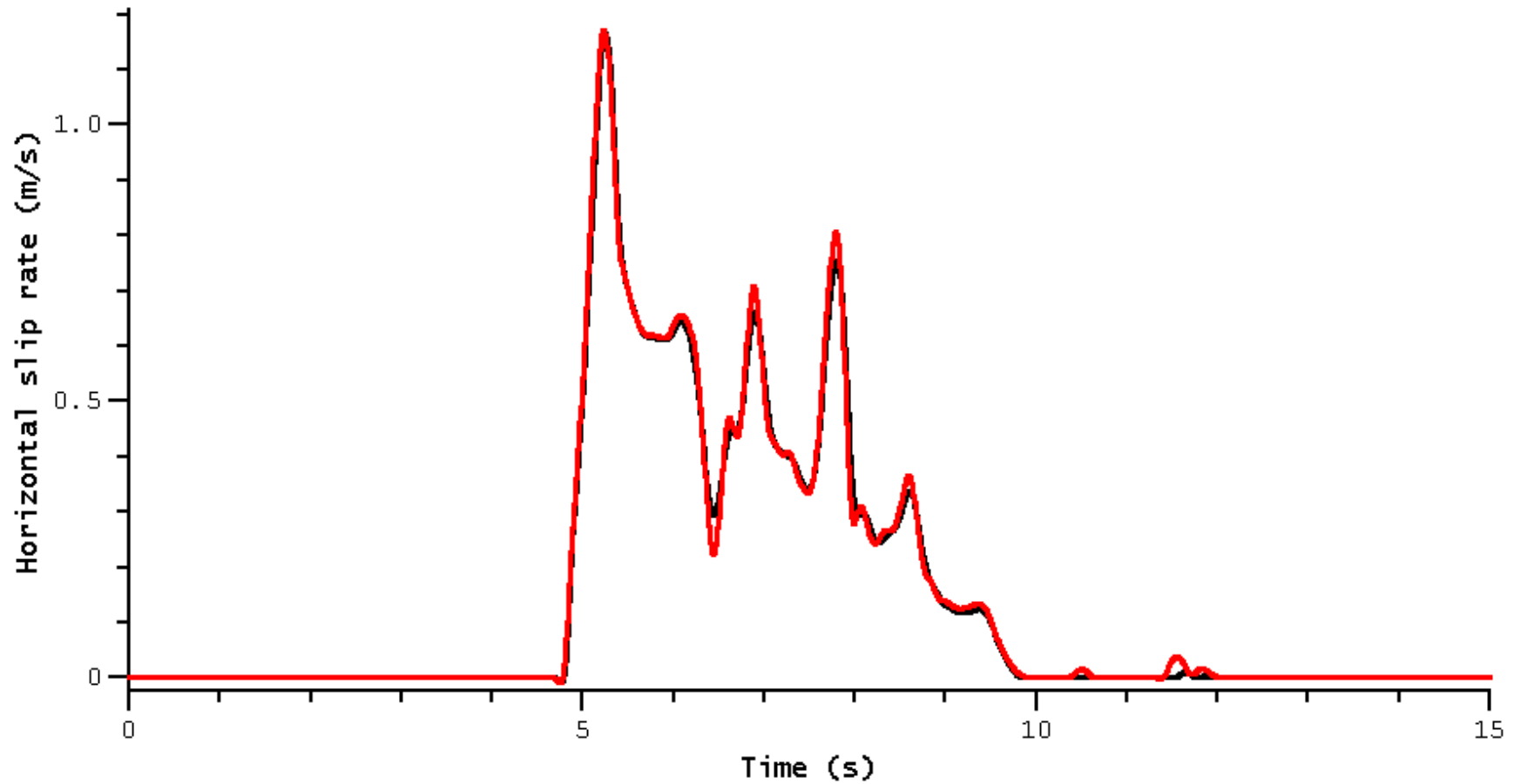$$Q \equiv \frac{\|f(t) - g(t - t_s)\|_2}{\|f(t)\|_2 + \|g(t - t_s)\|_2}$$

The integrals are computed over the range of times t where both $f(t)$ and $g(t - t_s)$ are defined.

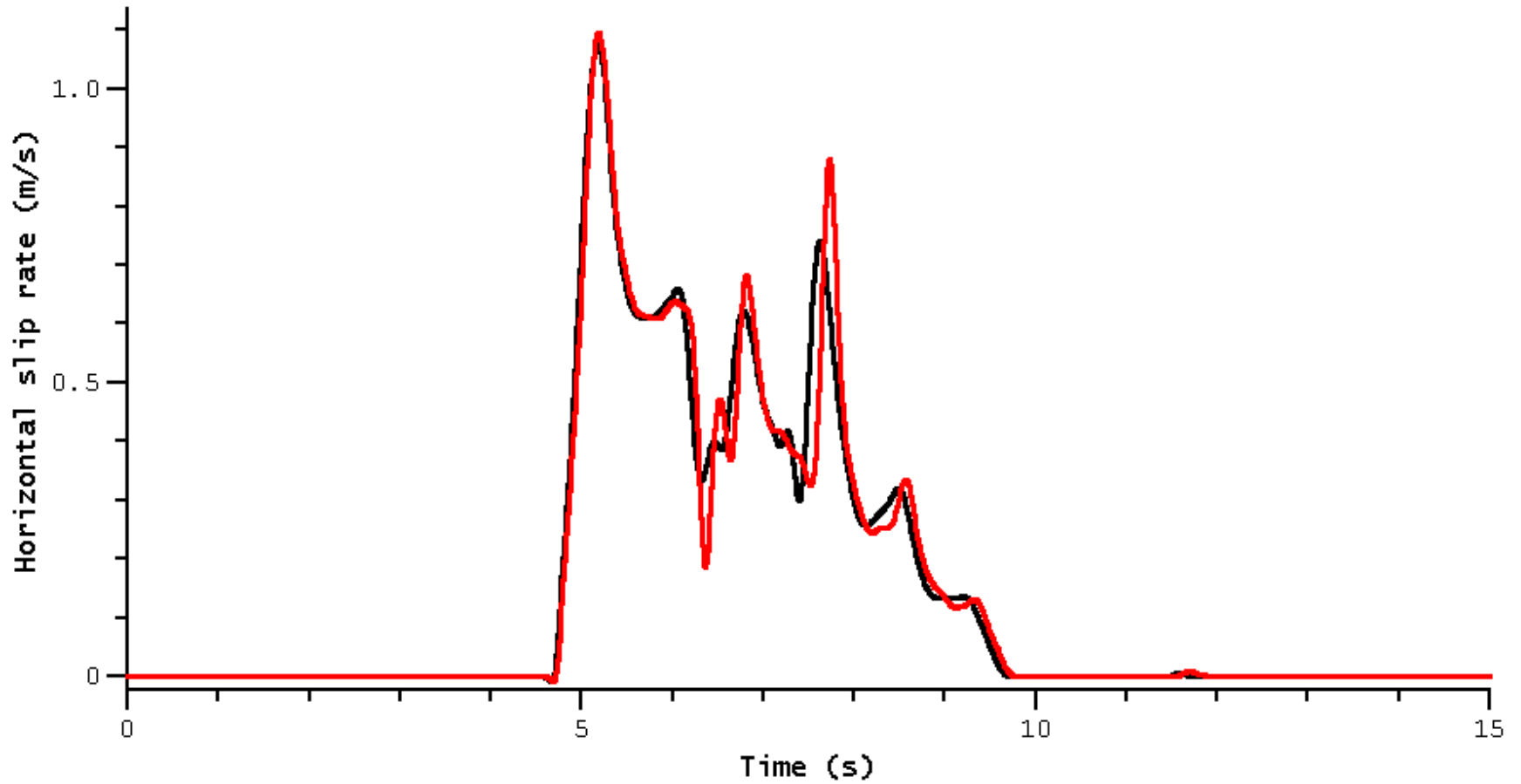It can be proved that

$$0 \leq Q \leq 1$$

The value $Q = 0$ indicates a perfect match. To convert Q into a percentage, we multiply by 200.
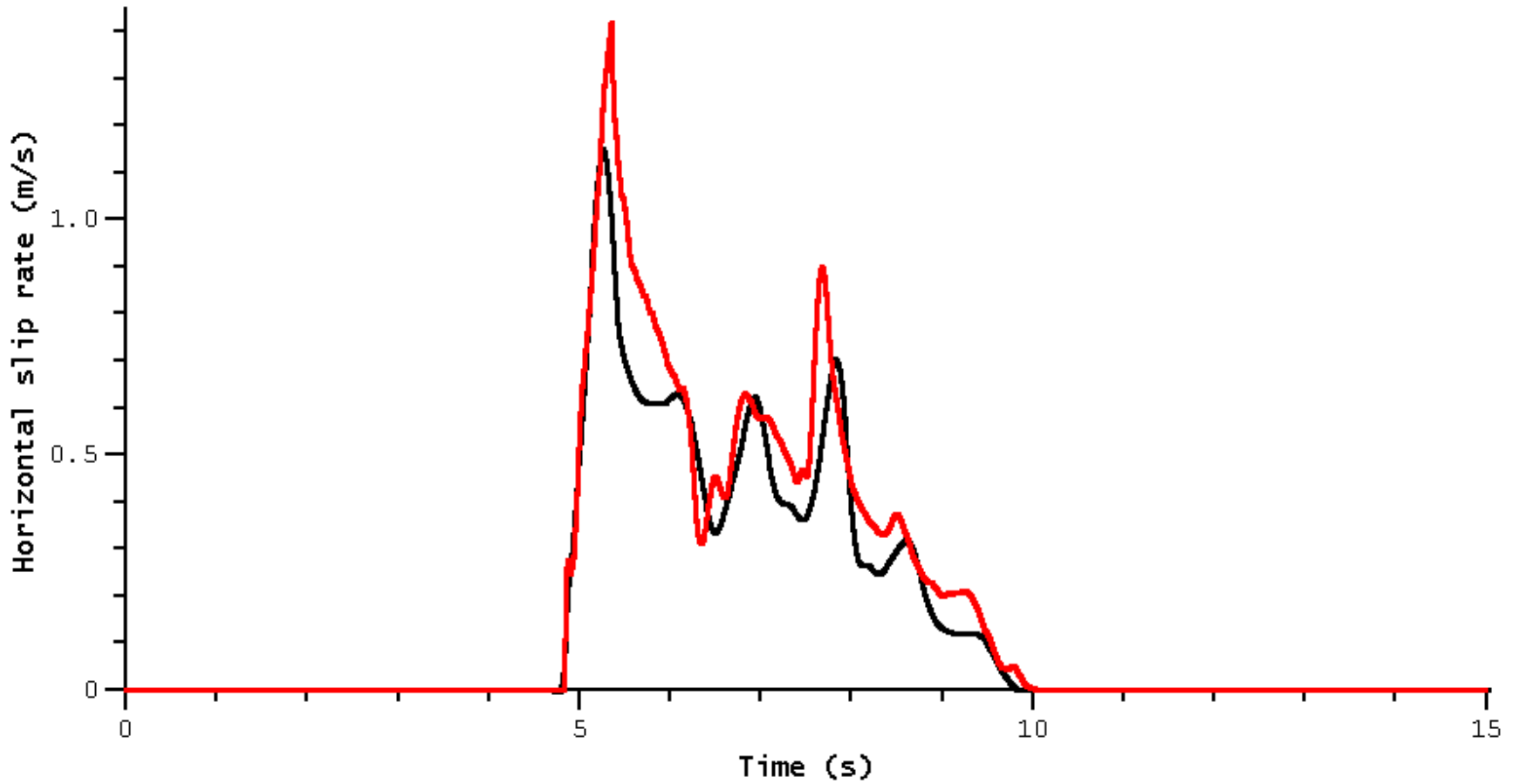
Time Series Metric Examples

Q = 4.0 percent

Time Series Metric Examples

Q = 10.7 percent

Time Series Metric Examples

Q = 24.1 percent

Metric Formula for Contour Plot

For Metrics Version 1, we use the root-mean-square difference in rupture time, expressed in milliseconds.

Let (x, y) be position on the fault plane. The contour plot data gives the rupture time as a function of x and y.

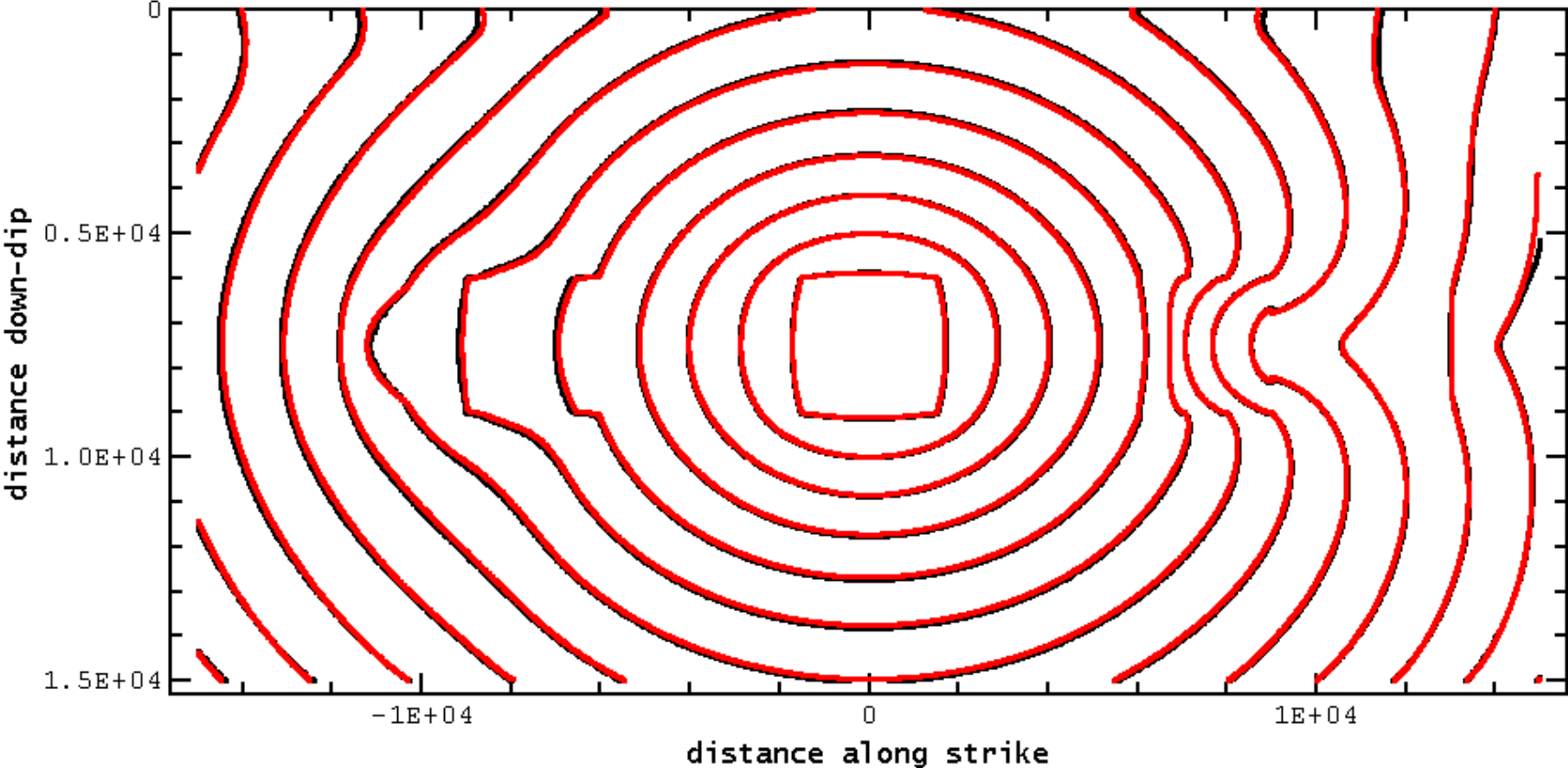For a function of position $f(x, y)$, , define the L^2 norm:

$$\|f(x, y)\|_2 \equiv \left[\int \int |f(x, y)|^2 \, dx \, dy \right]^{1/2}$$

Our main misfit metric is the RMS difference. If $f(x, y)$ and $g(x, y)$ are two rupture time functions, the metric value is:

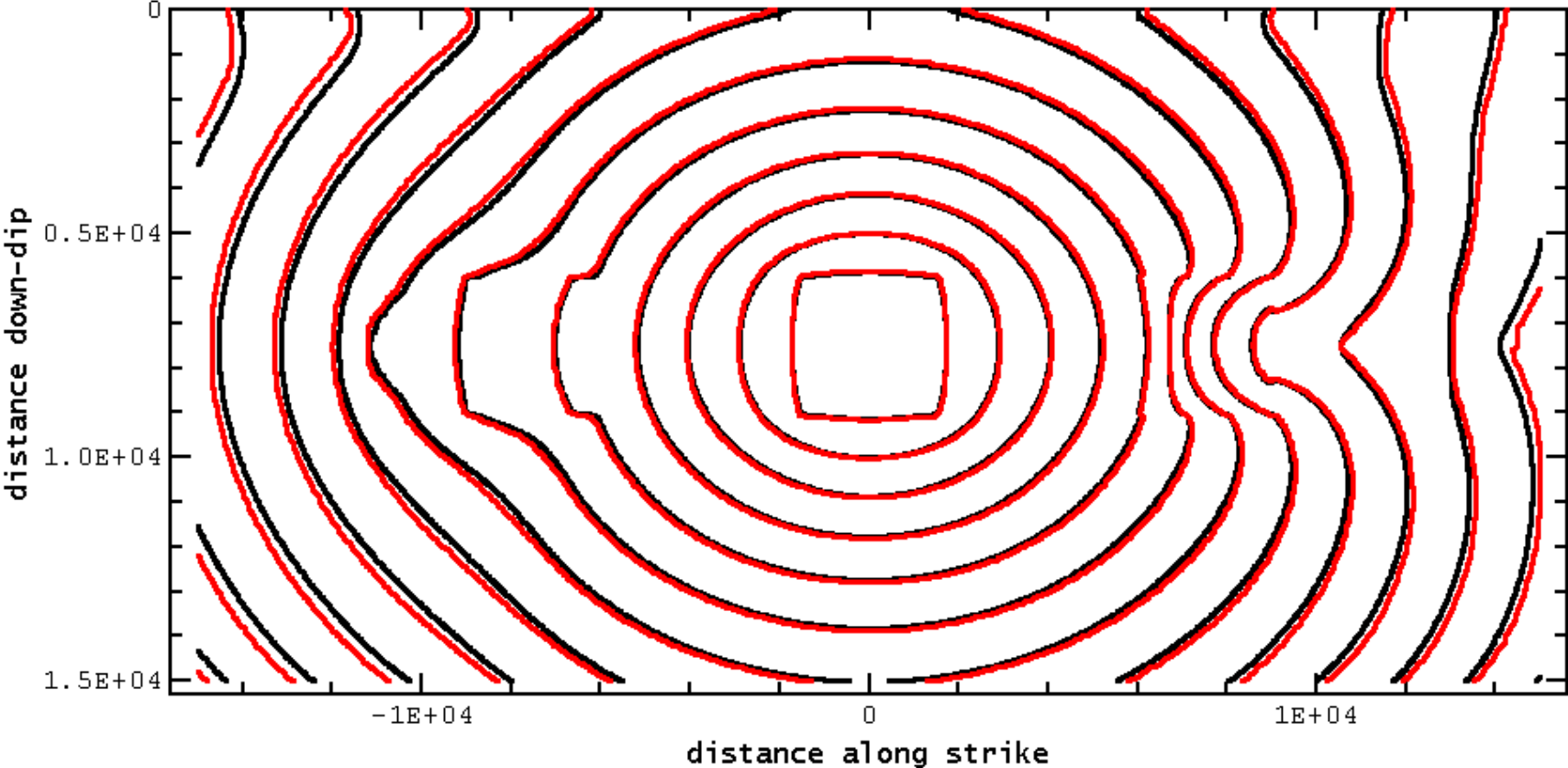$$T \equiv \frac{1}{\sqrt{A}} \|f(x, y) - g(x, y)\|_2$$

The integral runs over the part of the fault surface where both $f(x, y)$ and $g(x, y)$ are defined, and A is the area of that part of the fault surface.
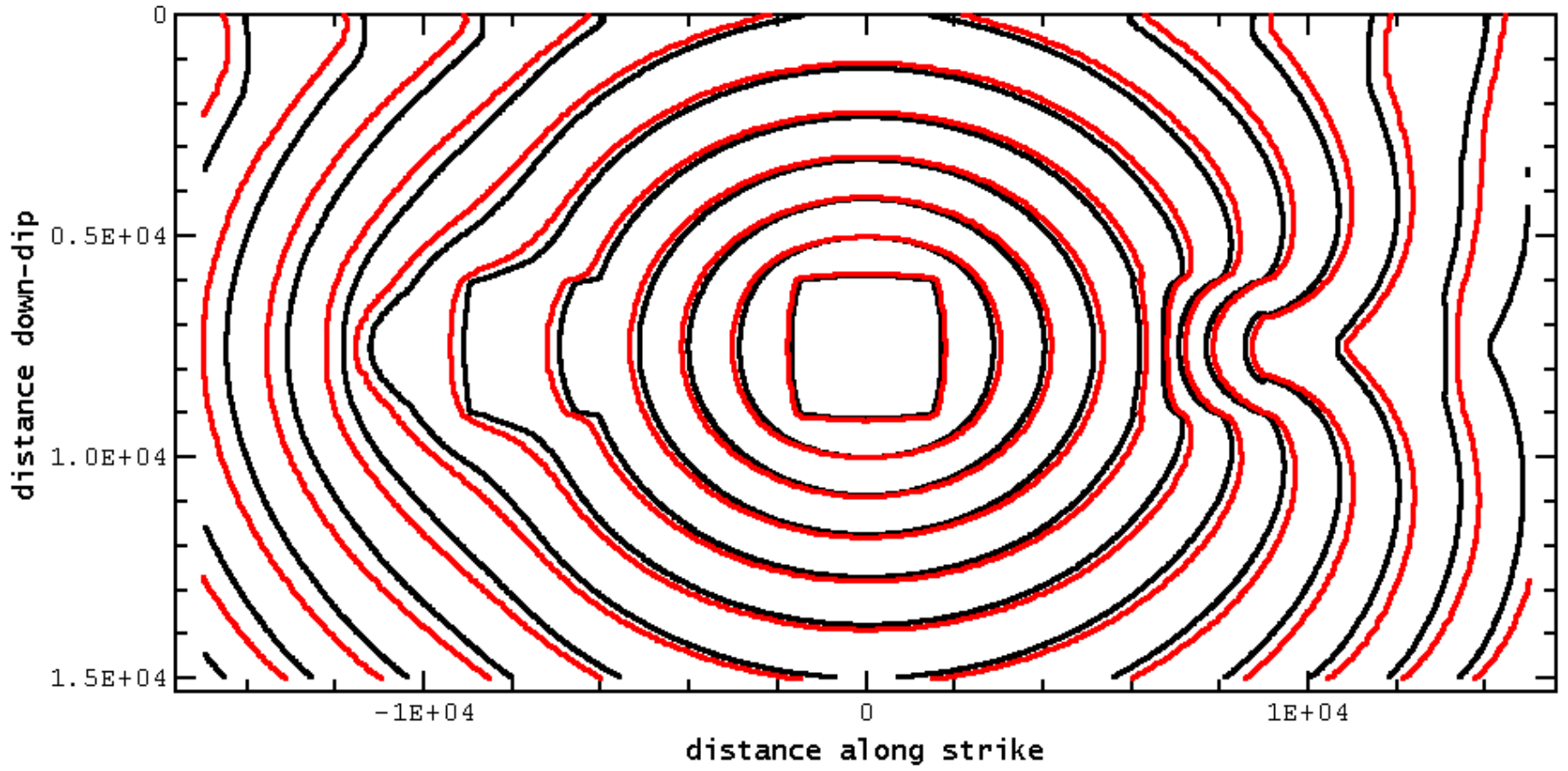
Contour Plot Metric Examples

T = 18.8 milliseconds

Contour Plot Metric Examples

T = 47.8 milliseconds

Contour Plot Metric Examples

T = 106.0 milliseconds

Computational Steps

Time Series — Preprocessing

Each time-series file is preprocessed as follows.

1. Remove the first time step.

2. Remove any time steps after the requested duration.

3. Apply a 5 Hz low-pass filter to slip rate, shear stress, normal stress, and velocity.

4. If it's a vertical planar fault in an elastic medium, remove the normal stress.

5. Remove the DC component from shear stress, normal stress, and rate-state variables.

6. Form components into vectors, to produce a vector-valued time series.

    - For an on-fault time series: 2d slip, 2d slip rate, and 2d shear stress.

    - For an off-fault time series: 3d displacement and 3d velocity.

7. Extend the time series to a continuous function of time, by linearly interpolating between the points supplied by the modeler.

Time Series — Comparison

Each pair of time-series files at a given station is compared as follows.

1. Search for the optimum time shift. Choose the time shift to minimize a comparison metric:

   - For an on-fault station, where the peak slip rate is at least 0.050 m/s, find the best match for 2d slip rate.

   - For an on-fault station, where the peak slip rate is less than 0.050 m/s, find the best match for 2d shear stress.

   - For an off-fault station, find the best match for 3d velocity.

2. Compute the comparison metric for each pair of fields.

   - For on-fault stations: 2d slip, 2d slip rate, 2d shear stress, normal stress, rate-state variables.

   - For off-fault stations: 3d displacement, 3d velocity.

3. For on-fault stations, do not compare slip or slip rate if the peak slip rate is less than 0.01 m/s.

All integrals are performed analytically, treating each time series as a piecewise-linear function.

Time Series — Summaries

After computing the full set of comparison metrics, they are summarized in two ways:

- For each pair of users, compute the average comparison metric value over all stations and fields.

- For each station and field, compute the average comparison metric over all pairs of users.

Contour Plot — Preprocessing

Each contour-plot file is preprocessed as follows.

1. Form the Delaunay triangulation of the points supplied by the modeler.

2. Extend the rupture time to a continuous function of position on the fault surface, by interpolating linearly within each Delaunay triangle.

3. Compute the area inside each rupture contour, by integrating the unit function within the region enclosed by the contour.

4. Compute the rupture velocity at each point on the fault surface, as follows:
   a. Draw a circle of radius 300 m around a point.
   b. Perform a two-dimensional linear least squares regression within the circle.
   c. The gradient of the resulting linear function is taken to be the slowness.
   d. Rupture velocity is the reciprocal of slowness.

5. Compute the process zone width at each on-fault station.
   a. Examine the time series data to determine how long it takes for slip to reach the critical slip distance.
   b. The process zone width is that amount of time multiplied by the rupture velocity.

Contour Plot — Comparison

Each pair of contour plots is compared by computing the RMS difference in rupture time.

The integral runs over the part of the fault surface where both contour plots show a rupture occurring, and have a rupture velocity of at least 1 km/sec.

To perform integrals over the fault surface:

- Each contour plot file is resampled onto a rectangular grid with 12.5 m node spacing.

- Rupture-time values are assigned to each node by interpolation within the Delaunay triangles.

- Then, integrals are evaluated as Riemann sums on the resampled grid.

Colors
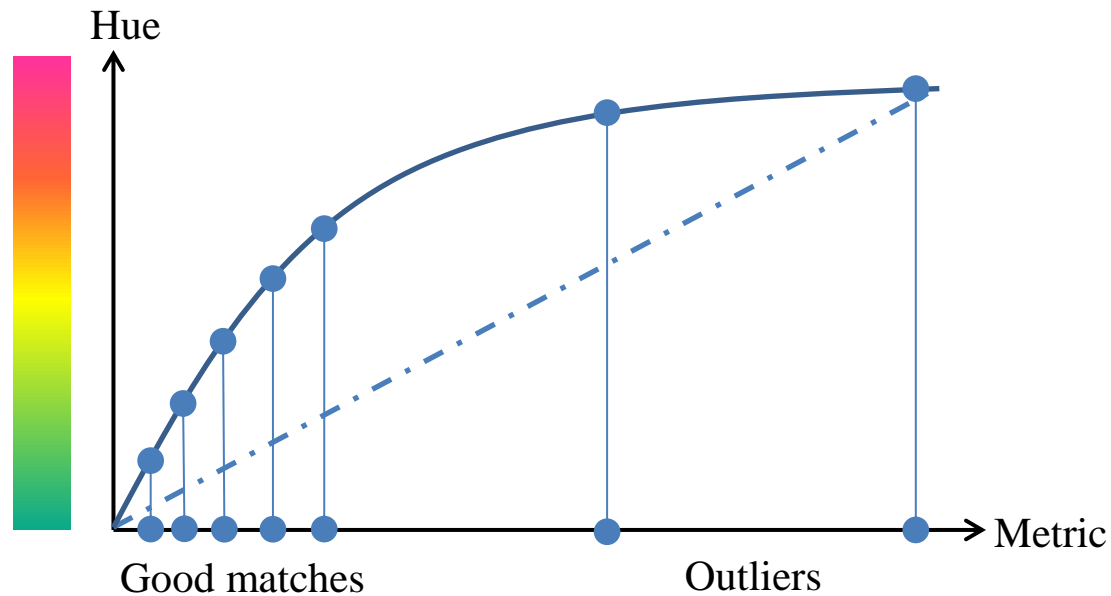
Colors help guide the eye to distinguish "better" from "worse" metric results. They are *not* intended to signify "acceptable" or "unacceptable" results.

A linear map of metric to hue produces bad results. Outliers stretch the color scale, leaving all the good matches colored green.

Instead, we use a non-linear mapping, to spread out the colors of the good matches, and leave the outliers clustered near the red end of the scale.

Color scale is adjusted to fit the range of values in each set of results.

Conclusions

Conclusions

- We have posted on our website "version 1" of metrics, for our entire set of benchmarks.

- We have developed procedures to generate all the metrics using an (almost) fully automatic process, with modeling data extracted directly from the website database.

- Numerical metric values are absolute, and do not depend on the range of results submitted. But colors are scaled to fit the range of results.

- Time series metrics are normalized RMS differences of vector-valued series, with an optimized relative time shift.

- Contour plot metrics are RMS differences of rupture time.

- Contour plot metrics also include the area of each rupture contour, and the process zone width at each on-fault station.

Observations and Questions

- For "version 1" the metrics seem successful, but we hope for improvement in future versions.

- There is simultaneously too many and too few metrics. Comparing every pair of time series produces more metrics than anyone can read, and yet a single number cannot fully describe how one time series compares to another.

- Are there more metrics, or different metrics, that would be useful to compute? Possible examples: L^1 norm, L^3 norm, peak amplitude, pulse duration, arrival time, L^2 norm within a time window surrounding the peak, others?

- Should we introduce an absolute scale of what metric values are "acceptable" results? If so, should we introduce an absolute color scale to match?

- How does the performance of these (or some other) metrics compare to a person performing a visual examination of modeling results?