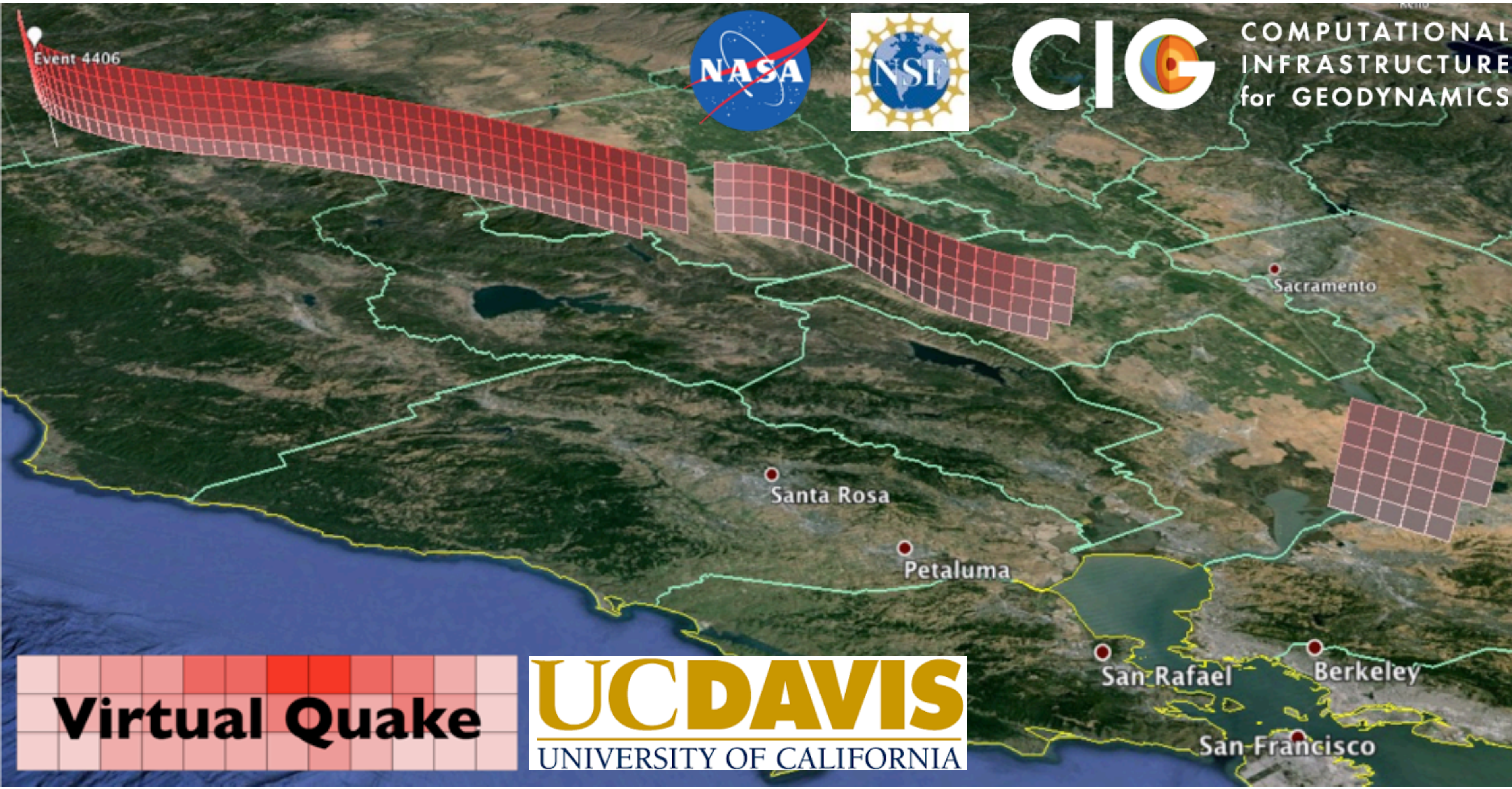
An Introduction to Virtual Quake

Kasey Schultz and John Wilson
University of California, Davis
Department of Physics - Rundle Group

CIG Webinar – December 3, 2015



Outline for Today

- What is Virtual Quake (VQ)?
- What physics does it use?
- Where to find the Virtual Quake code
- Setting up a simulation
  - Building a fault model
- Running Virtual Quake
- Analyzing Results with PyVQ (Python script)
- Featured Virtual Quake Papers
- Direct Simulation Data Analysis with PyVQ
- Other VQ projects underway
- Interesting future studies

History of Virtual Quake

- 1989
  - "Virtual California" slider-block model proposed by John Rundle
- 1990s-2000s
  - Virtual California simulator developed and continuously improved
- Early 2000s
  - Virtual California begins to simulate the California fault system (using only vertical strike-slip faults), Prof. Rundle produces simulation-based seismic hazard assessment for California [PNAS 2005]
- Early 2010s
  - Virtual California gets more complex, all fault geometries
- 2012
  - NASA Software of the Year Co-Winner (as part of QuakeSim group)
  - A Series of BSSA papers validate Virtual California simulations with other earthquake simulation groups (RSQSim, AllCal, Visco-Sim)
- 2013-Today
  - "Virtual Quake" is a greatly improved, open source, modularized, and modern C++ code.
  - Includes fault model-building tools, simulation analysis script, and Python library "quakelib"

Virtual Quake Validation: Earthquake Simulator Comparison

- 2012 series of papers in BSSA
- Compared California fault model simulations between four different earthquake simulators

M. K. Sachs, E. M. Heien, D. L. Turcotte, M. B. Yikilmaz, J. B. Rundle, and L.H. Kellogg. *Virtual California Earthquake Simulator*. Seismological Research Letters, *83*(6): 973–978, 2012.

Tullis, T. E., K. Richards-Dinger, M. Barall, J. H. Dieterich, E. H. Field, E. M. Heien, L. H. Kellogg, F. F. Pollitz, J. B. Rundle, M. K. Sachs, D. L. Turcotte, S. N. Ward, and M. B. Yikilmaz (2012a). Generic earthquake simulator, *Seismol. Res. Lett.* **83,** no. 6, 959–963.

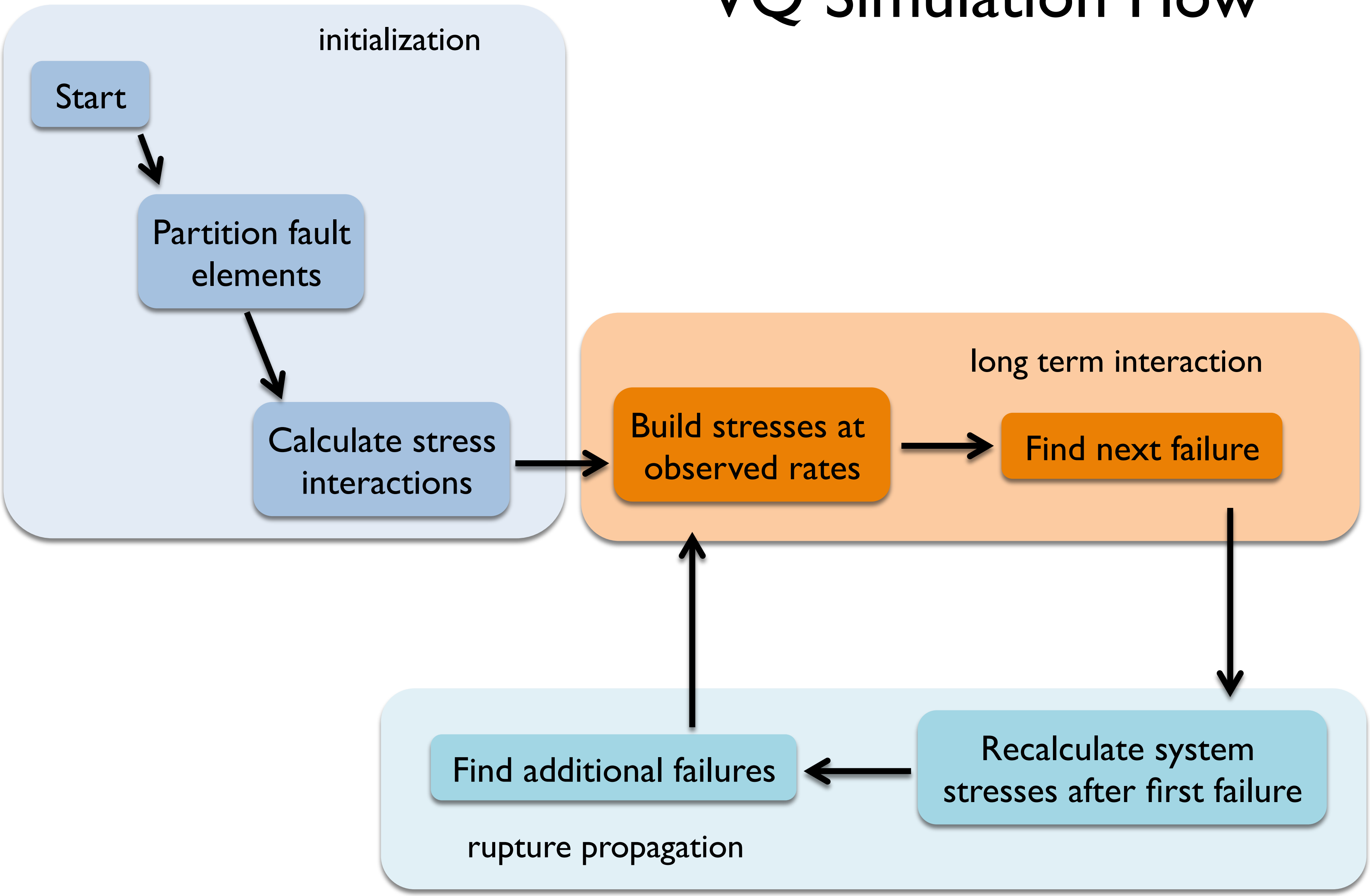
→ VQ physical assumptions and simplifications are reasonable

Tullis, T. E., K. Richards-Dinger, M. Barall, J. H. Dieterich, E. H. Field, E. M. Heien, L. H. Kellogg, F. F. Pollitz, J. B. Rundle, M. K. Sachs, D. L. Turcotte, S. N. Ward, and M. B. Yikilmaz (2012b). Comparison among observations and earthquake simulator results for allcal2 California fault model, *Seismol. Res. Lett.* **83,** no. 6, 994–1006.

→ VQ simulations consistent with observed EQ rates and with independent simulators

★ 2012 NASA Software of the Year as part of the QuakeSim team (HQ at NASA JPL) ★

VQ Simulation Flow



Physics of Virtual Quake - Summary

- Boundary element, elastic half-space code. Simulates earthquake sequences with stress interactions between faults over 10,000 – 100,000 years.
- Specify a fault model: Fault geometry, long term slip rates, aseismic fractions
- Faults are meshed by VQ at a desired resolution (1-3km typical for large systems)
- Each fault patch experiences "chaotic" elastic rebound from self-loading (via slip rate) and being stressed by other fault patches (stress Greens' functions for interactions – e.g. Okada 1992). Greens function matrices pre-computed
- Failure is governed by a Coulomb Failure Function (Stein 1999)
- Dynamic failure is also allowed at sub-critical stresses to approximate stress intensity at a crack tip during a rupture
- 2 free parameters:
  - Dynamic failure coefficient (fractional stress increase required during an earthquake for a fault patch to join the rupture, 0.5 = 50% increase from event start to join rupture)
  - Stress drop coefficient (complicated, think of it as a tuning parameter for the overall earthquake rates) (stress drops are used to determine amount of slip during event)
- Can incorporate slip-weakening (one of many optional parameters)

Interaction Model: Green's Functions

$$\sigma_{ij}(x,t) = \int dx'_k T^{kl}_{ij}(x-x')s_l(x',t)$$

Discrete: stress $\sigma(t)$ and slip s(t) only evaluated on fault elements

$$\sigma^A_{ij}(t) = \sum T^{AB}_{ij} s_B(t)$$

Shear stress along slip direction

$$\sigma^A_s(t) = \sum T^{AB}_s s_B(t)$$

Normal stress

$$\sigma^A_n(t) = \sum T^{AB}_n s_B(t)$$

★ Stress Green's functions T^{AB}_n and T^{AB}_s are complicated, in general they go like $1/r^3$, and positive slip s(t)>0 reduces local stress.

Earthquake Model

Static Failure

$$CFF^A(t) = \sigma_s^A(t) - \mu_s^A \sigma_n^A(t)$$

Earthquake at time t_{EQ} if $CFF(t_{EQ})=0$ (Coulomb Failure Function)

μ_s coefficient of static friction, derived from fault parameters

Dynamic Failure

$$\frac{CFF_{init} - CFF_{final}}{CFF_{init}} > \eta$$

Elements on same fault as the initial failed element can slip even if $CFF(t_{EQ}) \neq 0$

Dynamic triggering factor η controls rupture propagation

$\eta=0.5$ in simulation used later

Building Virtual Quake from Source

- CIG geodynamics.org/software/vq (stable releases)
- GitHub github.com/geodynamics/vq (bleeding edge updates)
- Continuously integrated on a CIG server
  - Ensures VQ will build on a variety of Linux machines
- Daily build tests on Mac OS X and Linux
- Windows not supported
- Build on command line with Cmake
  - See the MANUAL for full install instructions

```
$ cd vq
```

Make the build directory and navigate to it.

```
$ mkdir build
$ cd build
```

Use CMake to configure before compiling VQ.

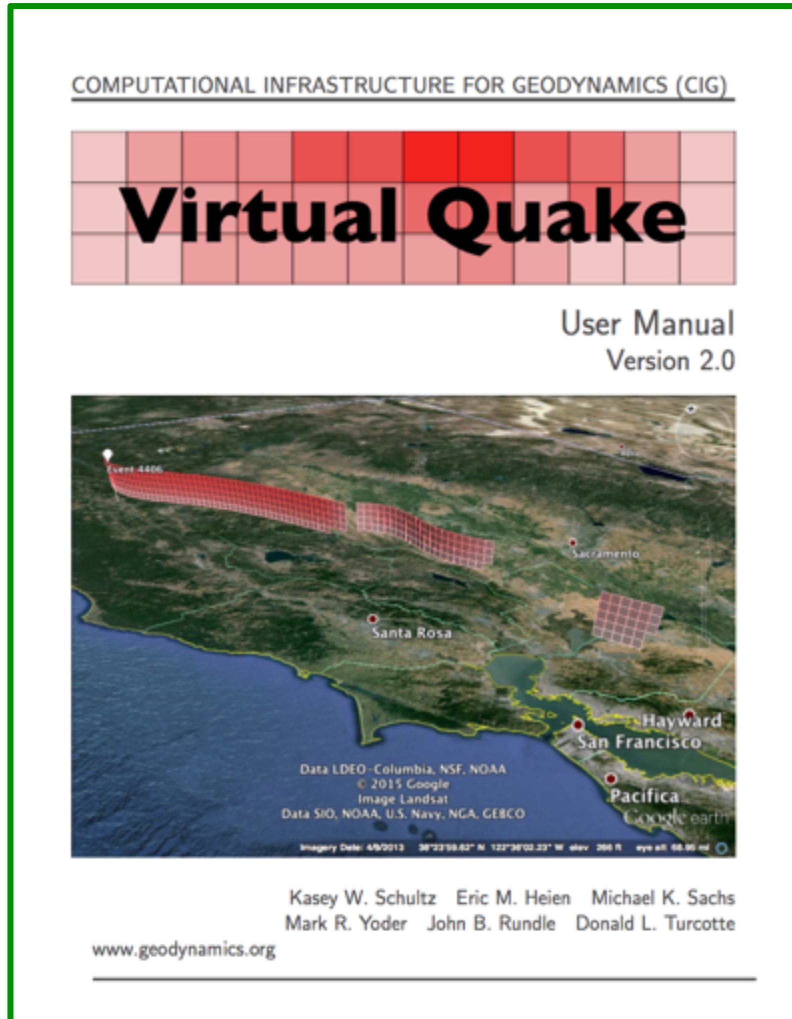
```
$ cmake ..
```

Use make to build QuakeLib and the VQ binaries.

```
$ make
```

The final step is required only if the user intends to

```
$ sudo make install
```



Extensive Unit-Testing Suite

- After installing Virtual Quake, use Cmake to run a suite of tests

```
$ cd build/
$ make test
Running tests...
Test project /.../build
        Start   1: CondUnitTest
  1/230 Test   #1: CondUnitTest .............................   Passed    0.08 sec
        Start   2: FricUnitTest
  2/230 Test   #2: FricUnitTest .............................   Passed    0.09 sec
        Start   3: GreenUnitTest
  3/230 Test   #3: GreenUnitTest ............................   Passed    0.09 sec
        Start   4: OctreeTest
  4/230 Test   #4: OctreeTest ...............................   Passed    0.06 sec
        Start   5: UtilUnitTest
  5/230 Test   #5: UtilUnitTest .............................   Passed    0.15 sec
        Start   6: EventUnitTest
  6/230 Test   #6: EventUnitTest ............................   Passed    0.05 sec
        Start   7: GeomUnitTest
  7/230 Test   #7: GeomUnitTest .............................   Passed    0.05 sec
        Start   8: MetadataUnitTest
  8/230 Test   #8: MetadataUnitTest .........................   Passed    0.04 sec
...
        Start 230: test_two_consistent_taper_renorm_3000
230/230 Test #230: test_two_consistent_taper_renorm_3000 ....   Passed    0.72 sec

100% tests passed, 0 tests failed out of 230

Total Test time (real) =  54.90 sec
```

Building a Fault Model

- We build fault models from trace files (one per fault)
- Trace file: At some resolution, specify the following parameters along the trace of the fault
  - lat/lon, dip, rake, slip rate, depth (along dip), aseismic fraction
- Our fault mesher will build the fault from the trace information, filling the space of the fault with elements (many mesher options)

Building a Fault Model: Example

- Given two trace files, build the fault model with the mesher
- Also output a Google Earth KML file of the model

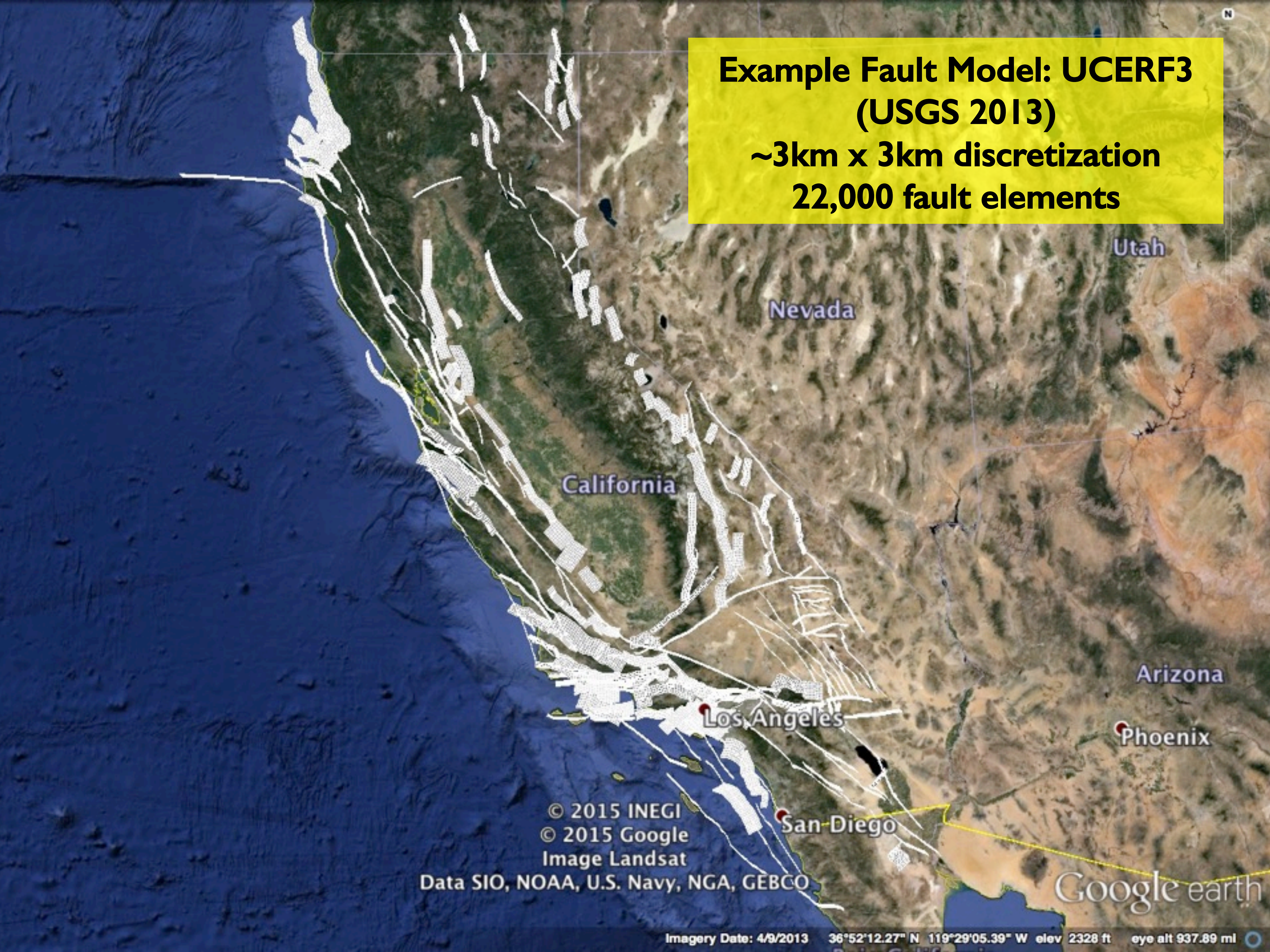
```
$ ~/vq/build/src/mesher  --import_file  trace_file1.txt  --import_file_type=trace
  --import_file  trace_file2.txt  --import_file_type=trace  --export_file=combined_model.txt
  --export_file_type=text  --export_file=combined_model.kml  --export_file_type=kml
```

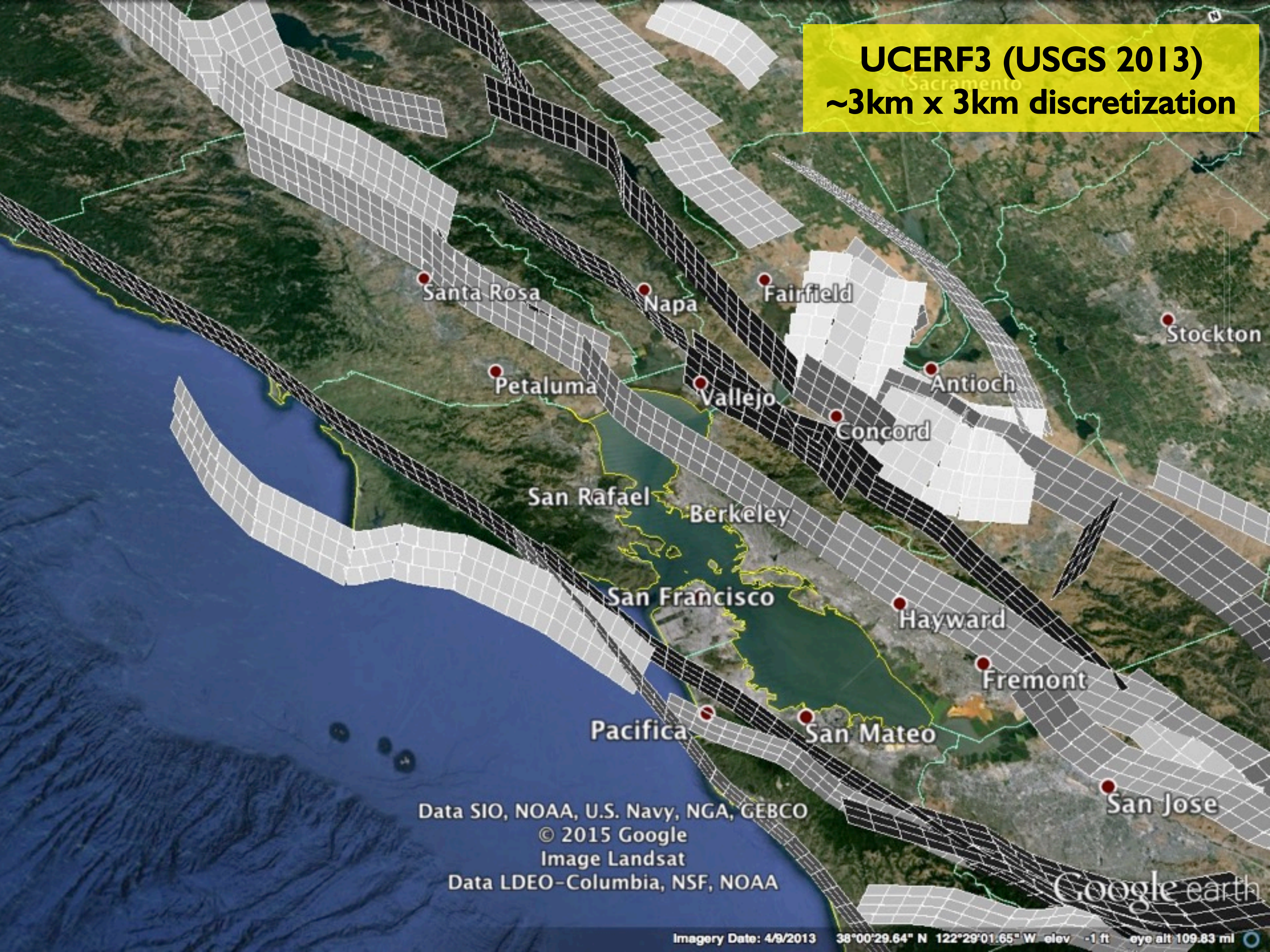
```
# fault_id: ID number of the parent fault of this section
# num_points: Number of trace points comprising this section
# section_name: Name of the section
105 59 Bartlett_Spring
# latitude: Latitude of trace point
# longitude: Longitude of trace point
# altitude: Altitude of trace point (meters)
# depth_along_dip: Depth along dip (meters)
# slip_rate: Slip rate at trace point (centimeters/year)
# aseismic: Fraction of slip that is aseismic at point
# rake: Fault rake at trace point (degrees)
# dip: Fault dip at trace point (degrees)
# lame_mu: Lame's mu parameter at trace point (Pascals)
# lame_lambda: Lame's lambda parameter at trace point (Pascals)
40.2622 -123.459 0 12000 0 0 180 90 3e+10 3.2e+10
40.2357 -123.465 0 12000 0.439 0 180 90 3e+10 3.2e+10
40.209 -123.47 0 12000 0.557 0 180 90 3e+10 3.2e+10
40.1823 -123.473 0 12000 0.615 0 180 90 3e+10 3.2e+10
```

Example Fault Model: UCERF3 (USGS 2013)

~3km x 3km discretization

22,000 fault elements




UCERF3 (USGS 2013)
~3km x 3km discretization
Sacramento
Santa Rosa
Napa
Fairfield
Stockton
Petaluma
Vallejo
Antioch
Concord
San Rafael
Berkeley
San Francisco
Hayward
Fremont
Pacifica
San Mateo
San Jose
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
© 2015 Google
Image Landsat
Data LDEO-Columbia, NSF, NOAA
Google earth
Imagery Date: 4/9/2013
38°00'29.64" N 122°29'01.65" W elev -1 ft
eye alt 109.83 mi

Running Virtual Quake: Parameter File

```
sim.time.start_year            = 0
sim.time.end_year              = 10000
```

simulation time range

```
sim.greens.method              = standard
sim.greens.output              = stress_your_fault_model_3km.h5
```

greens functions

```
sim.file.input                 = your_fault_model_3km.h5
sim.file.input_type            = hdf5
sim.file.output_event          = events_your_fault_model_3km_drops0-4_dyn0-5_10kyr.h5
sim.file.output_event_type     = hdf5
```

file I/O

```
sim.friction.stress_drop_factor    = 0.4
sim.friction.dynamic               = 0.5
```

tuning parameters

```
sim.file.output_stress             = test_stress.txt
sim.file.output_stress_index       = test_stress_index.txt
sim.file.output_stress_type        = text
sim.file.output_stress_num_events  = 5000
```

sim checkpoint saving

There are many more parameters, see the User's Manual for full list

Running Virtual Quake

```
terremoto:RUNNING kasey$ mpiexec -np 4 ./vq params_allcal.d
# ********************************
# *** Virtual Quake ***
# *** Version 2.0.0 ***
# *** Git revision ID 34c6c14ca5e4554c1358a1faf8b5102624356774 ***
# *** QuakeLib 1.2.0 Git revision 34c6c14ca5e4554c1358a1faf8b5102624356774 ***
# *** MPI process count       : 4 ***
# *** OpenMP not enabled        ***
# ********************************
# Initializing blocks.
# To gracefully quit, create the file quit_vq in the run directory or use a SIGINT (Control-C).
# Reading Greens function data from file greens_UCERF3_VQmeshed_3km.h5....0%....1%....3%....4%....5%....6%....7%....8%....9%....10%....1
1%....12%....13%....14%....15%....17%....18%....19%....20%....21%....22%....23%....24%....25%....26%....27%....28%....30%....31%....32%.
...33%....34%....35%....36%....37%....38%....39%....40%....41%....42%....44%....45%....46%....47%....48%....49%....50%....51%....52%....
53%....54%....55%....56%....57%....58%....59%....61%....62%....63%....64%....65%....66%....67%....68%....69%....70%....71%....72%....73%
....74%....75%....77%....78%....79%....80%....81%....82%....83%....84%....85%....86%....87%....88%....89%....91%....92%....93%....94%...
.95%....96%....97%....98%....99%..
# Greens function took 492.36 seconds.
# Greens shear matrix takes 911.485 megabytes
# Greens normal matrix takes 911.485 megabytes
# Greens Shear:
 max: 4.65309e+06
 min: -2.08774e+07
 mean: -2.97305e+06

# Greens Normal:
 max: 2.37615e+06
 min: -2.23179e+06
 mean: 259898

# Greens DiagShear:: -2.08774e+07 -- -9.10632e+06 (-1.53893e+07)
# Greens DiagNormal:: -2.23179e+06 -- 2.37615e+06 (38773)
# Greens offDiagShear:: -1.33614e+06 -- 4.65309e+06 (1.24163e+07)
# Greens offDiagNormal:: -1.91868e+06 -- 2.04839e+06 (221126)

# Global Greens shear matrix takes 3.56049 gigabytes.
# Global Greens normal matrix takes 3.56049 gigabytes.
# Displaying simulation progress every 10 seconds.
#
# ************************************************
# ***
# *** Blocks                    : 21833
# *** Faults                    : 313
# ***
# *** Present Time (years)      : 0
# *** Min  cff                  : -1.87866e+07
# *** Mean cff                  : -8.32438e+06
# *** Max  cff                  : -7.08086e+06
# ***
# To access the event output file during the simulation, pause
# by creating the file pause_vq.  Delete the file to resume.
# Writing events in format hdf5 to file UCERF3_VQmeshed_3km_10kyr_dyn0-8_stressDrops0-5_GreenLimits_dynDrops.h5
# events        year       minCff[index]      avrCff       maxCff[index]
      10        88.9  -1.877e+07[11572]  -7.889e+06   3.111e+06[ 2609]
```

Virtual Quake Output: Events

event_nu...	event_year	event_trig...	event_ma...	event_she...	event_she...	event_nor...	event_nor...	start_swe...	end_swee...
2607	1046.695...	6087	5.956098...	5436685....	-2.59680...	2.181858...	2.179609...	3636559	3636615
2608	1046.716...	5819	6.021254...	2774705....	-4.94128...	7.163904...	7.160898...	3636615	3636635
2609	1046.811...	11946	6.301970...	4029808....	-1.08651...	2.469329...	2.451347...	3636635	3636727
2610	1046.866...	7848	6.539181...	-4.54143...	-1.34861...	5.517006...	5.469131...	3636727	3636880
2611	1047.955...	13827	6.505863...	-244304...	-9.94745...	2.150692...	2.150705...	3636880	3638243
2612	1048.732...	3879	6.931241...	2.307990...	-6.52989...	2.824318...	2.819658...	3638243	3640852
2613	1048.956...	1147	6.095512...	4.508588...	-1.99687...	3.809657...	3.810502...	3640852	3640912
2614	1050.029...	4773	5.608864...	-238111...	-1.97361...	9.072139...	9.071898...	3640912	3640921
2615	1050.138...	9975	6.557674...	4.484858...	-2.38021...	8.101862...	8.141672...	3640921	3641327
2616	1051.737...	1219	5.709503...	-332517...	-2.47707...	1.598515...	1.598960...	3641327	3641344
2617	1052.092...	3411	5.483343...	-927742...	-1.80262...	6.420186...	6.420175...	3641344	3641359
2618	1052.960...	11	5.413527...	-1.26828...	-2.21256...	7.472676...	7.472672...	3641359	3641368
2619	1054.207...	7	5.700531...	-575329...	-2.93711...	8.568940...	8.571740...	3641368	3641383
2620	1055.389...	234	5.853259...	-1.09007...	-4.41516...	9.774805...	9.796425...	3641383	3641404
2621	1055.783...	1618	6.883576...	2.845782...	-5.99145...	1.826339...	1.827452...	3641404	3643065
2622	1056.472...	2780	5.903702...	1.411938...	-2.60536...	2.155342...	2.158261...	3643065	3643090
2623	1056.667...	14025	6.553990...	1.692082...	-5.50267...	7.095451...	7.151653...	3643090	3643927
2624	1056.892...	15	5.729011...	-264316....	-2.34920...	1.085080...	1.086250...	3643927	3643940
2625	1057.376...	76	6.824152...	4.003585...	-1.65340...	2.101754...	2.119302...	3643940	3645407
2626	1058.046...	159	5.680672...	1.442671...	-707938...	1.204290...	1.205646...	3645407	3645421
2627	1058.066...	2519	6.102326...	2.756317...	-1.98748...	1.761913...	1.761059...	3645421	3645481
2628	1058.897...	6367	7.179519...	5.782284...	-2.03671...	3.389651...	3.384227...	3645481	3649746
2629	1059.124...	14072	5.768294...	1296098....	-1.87837...	1.400909...	1.399145...	3649746	3649762
2630	1059.642...	3120	5.921011...	7419263....	-2.99668...	1.541562...	1.538850...	3649762	3649790
2631	1059.926...	167	6.272923...	4.605299...	-819437...	5.810756...	5.796892...	3649790	3650027
2632	1060.083...	219	5.255504...	-467569....	-582630...	4.281084...	4.281083...	3650027	3650032
2633	1060.299...	11801	6.717671...	1.474351...	-887481...	9.779089...	9.824407...	3650032	3651110
2634	1060.819...	2968	5.611284...	8692057....	-967259...	9.459485...	9.457330...	3651110	3651122
2635	1061.557...	3133	6.563689...	1.305315...	-3.58960...	7.625865...	7.687646...	3651122	3651448
2636	1062.212...	3397	6.738846...	2.083049...	-3.88056...	1.586436...	1.597113...	3651448	3653344
2637	1062.446...	3831	6.157829...	3.017735...	-2.99324...	2.618902...	2.611564...	3653344	3653390

Virtual Quake Output: Sweeps

event_nu...	sweep_nu...	block_id	block_slip	block_area	block_mu	shear_init	shear_final	normal_init	normal_fi...
3	1	1	0.091324...	2249995....	3.000000...	5970497....	4713737....	5.959574...	5.959574...
3	1	8	0.063723...	2250011....	3.000000...	5253309....	5364172....	1.986525...	1.986525...
3	1	120	0.770756...	2250045....	3.000000...	-480740...	-1.00392...	1.986525...	1.986525...
3	2	112	0.062629...	2250045....	3.000000...	5367573....	3544792....	1.986525...	1.986525...
3	2	121	0.087574...	2250045....	3.000000...	6026826....	3899285....	5.959575...	5.959575...
3	2	0	-0.34458...	2249995....	3.000000...	-172410...	-364001...	1.986525...	1.986525...
3	2	1	0.622398...	2249995....	3.000000...	-403463...	-826121...	5.959574...	5.959575...
3	2	8	0.774877...	2250011....	3.000000...	-488897...	-997007...	1.986525...	1.986525...
3	2	120	-0.17731...	2250045....	3.000000...	-172410...	-628520...	1.986525...	1.986525...
3	3	2	0.075903...	2249995....	3.000000...	5654536....	3864808....	9.932624...	9.932624...
3	3	9	0.124672...	2250011....	3.000000...	6667470....	3199696....	5.959575...	5.959575...
3	3	16	0.085067...	2249995....	3.000000...	5652420....	3282901....	1.986524...	1.986524...
3	3	0	-0.10048...	2249995....	3.000000...	-902158....	-267881...	1.986525...	1.986525...
3	3	1	-0.14372...	2249995....	3.000000...	-137329...	-502642...	5.959575...	5.959575...
3	3	8	-0.19523...	2250011....	3.000000...	-174683...	-625111...	1.986525...	1.986525...
3	3	112	0.778916...	2250045....	3.000000...	-488570...	-1.00073...	1.986525...	1.986525...
3	3	120	0.101412...	2250045....	3.000000...	-172156...	-367852...	1.986525...	1.986525...
3	3	121	0.628517...	2250045....	3.000000...	-403188...	-829930...	5.959575...	5.959575...
3	4	104	0.086013...	2250045....	3.000000...	5670070....	3258286....	1.986525...	1.986525...
3	4	113	0.125644...	2250045....	3.000000...	6687750....	3169351....	5.959575...	5.959575...
3	4	122	0.076274...	2250045....	3.000000...	5662280....	3856945....	9.932625...	9.932625...
3	4	0	0.063209...	2249995....	3.000000...	-899616....	-199799...	1.986525...	1.986525...
3	4	1	0.084247...	2249995....	3.000000...	-137057...	-294006...	5.959575...	5.959575...
3	4	2	0.575244...	2249995....	3.000000...	-359082...	-738058...	9.932625...	9.932625...
3	4	8	0.128265...	2250011....	3.000000...	-174361...	-368653...	1.986525...	1.986525...
3	4	9	0.642181...	2250011....	3.000000...	-408747...	-837442...	5.959575...	5.959574...
3	4	16	0.776779...	2249995....	3.000000...	-489043...	-998093...	1.986525...	1.986525...
3	4	112	-0.20096...	2250045....	3.000000...	-174362...	-620207...	1.986525...	1.986525...
3	4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Recent Virtual Quake Publications

K. W. Schultz, M. K. Sachs, E. M. Heien, M. R. Yoder, J. B. Rundle, D. L. Turcotte, and A. Donnellan, **Virtual Quake: Statistics, Co-Seismic Deformations and Gravity Changes for Driven Earthquake Fault Systems**, *International Association of Geodesy Symposia*, accepted May 2015

"The State of VQ", and EQ probabilities for California

K.W. Schultz, M.K. Sachs, J.B. Rundle, D.L. Turcotte, **Simulating Gravity Changes in Topologically Realistic Driven Earthquake Fault Systems**, *Pure and Applied Geophysics*, DOI: 10.1007/s00024-014-0926-4, in press (2015)

Co-seismic Gravity Changes

M. R. Yoder, K. W. Schultz, E. M. Heien, J. B. Rundle, D. L. Turcotte, J. W. Parker and A. Donnellan. **The Virtual Quake earthquake simulator: A simulation based forecast of the El Mayor-Cucapah region and evidence of earthquake predictability**, *Geophysical Journal International*,

accepted June 2015

EQ Probabilities, Activation vs. Quiescence

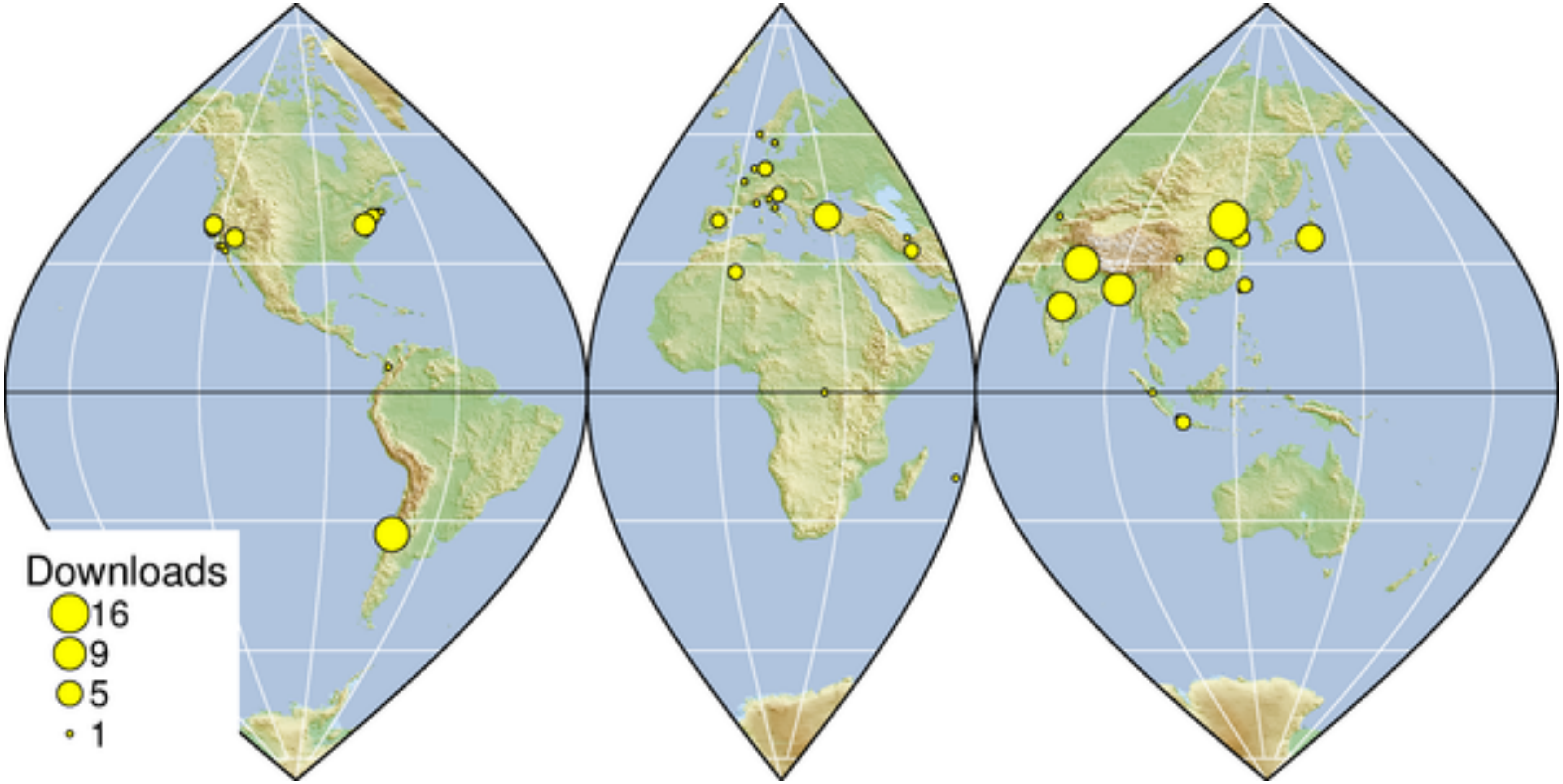
M. R. Yoder, K. W. Schultz, E. M. Heien, J. B. Rundle, D. L. Turcotte, J. W. Parker and A. Donnellan. **Forecasting earthquakes with the Virtual Quake simulator: Regional and fault-partitioned catalogs**, *International Association of Geodesy Symposia*, under review (2015)

Presented Fault-wise EQ Probabilities

12 Virtual Quake Publications listed on CIG:

geodynamics.org/cig/news/publications/#vq

Virtual Quake Users

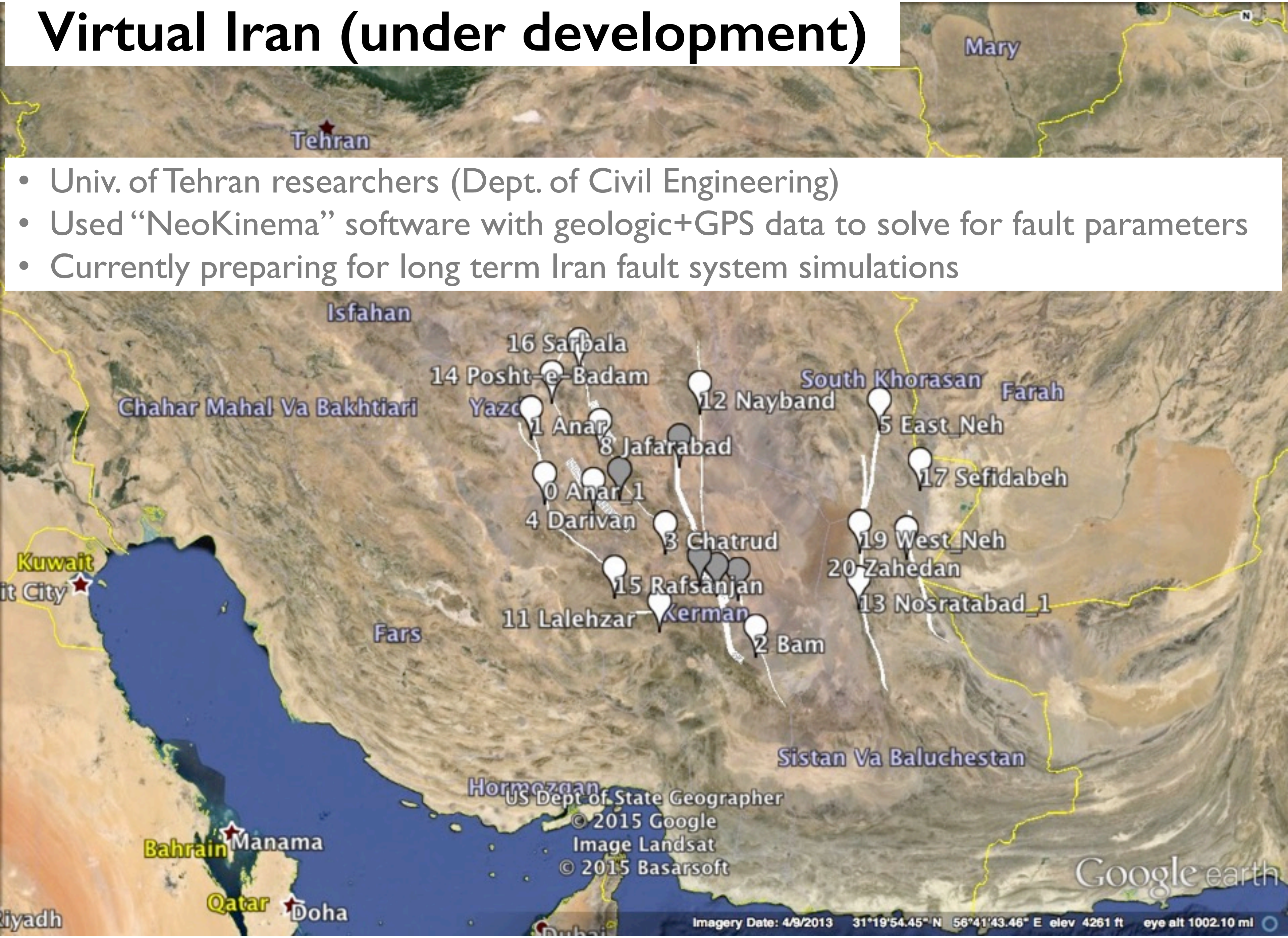


Currently Collaborating with:

- Univ. of Tehran researchers (Dept. of Civil Engineering)
- Universidad Católica del Norte; Antofagasta, Chile

Virtual Iran (under development)

- Univ. of Tehran researchers (Dept. of Civil Engineering)
- Used “NeoKinema” software with geologic+GPS data to solve for fault parameters
- Currently preparing for long term Iran fault system simulations



The quakelib module and PyVQ Analysis Script

- We use SWIG to wrap the QuakeLib C++ libraries into "quakelib"
- **quakelib** is an object-oriented Python library (example script below)

```
import quakelib

# ------ Initialize model object ----------
model = quakelib.ModelWorld()

# ------ Read model file ------------------
model.read_file_hdf5("path/to/model_file.h5")
# or
model.read_file_ascii("path/to/model_file.txt")

# ------ Read simulation (event) file -----
events = quakelib.ModelEventSet()
events.read_file_hdf5("path/to/event_file.h5")

# Create a KML file showing the co-seismic slip distribution for event #1415
model.write_event_kml("my_event_1415.kml", events[1415])
```

- **PyVQ** is a Python script that utilizes the quakelib module
- Creating the same file as above with PyVQ on the command line:

```
$ python ~/vq/pyvq/pyvq/pyvq.py --event_kml --model_file path/to/model_file.h5
     --event_file path/to/event_file.h5 --event_id 1415
```

PyVQ Analysis Script

- Well-commented
- In Python, so easy to read through the code
- Automatic file naming: utilizes sub-setting parameters and model and event file names where applicable
- Easily adaptable, write whatever functions you want

[Show the file]

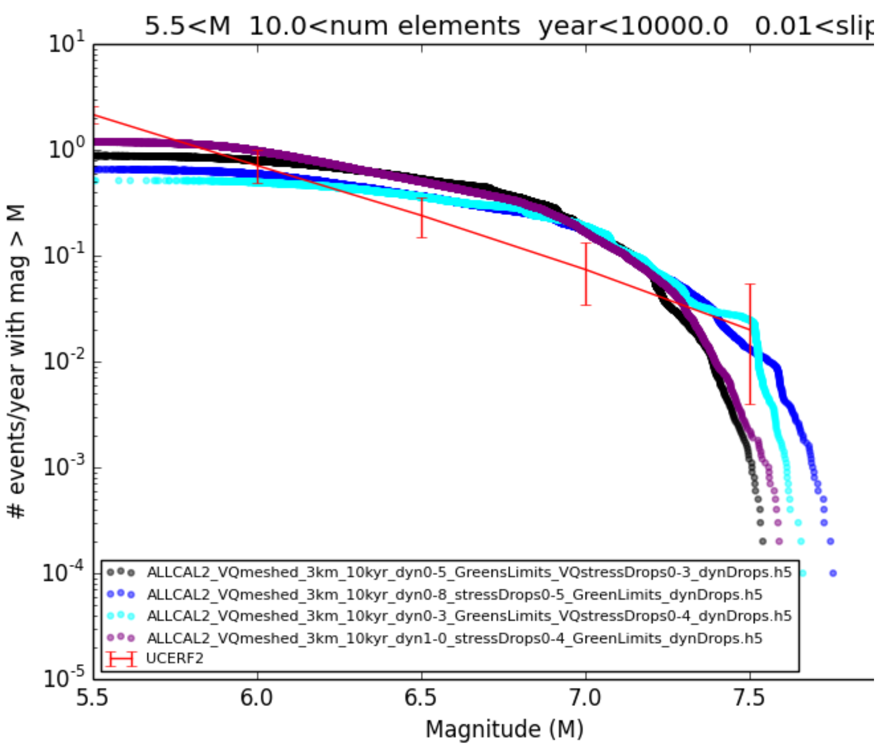
- **Many examples to follow** illustrate the utility of PyVQ and show the commands used to make the plots

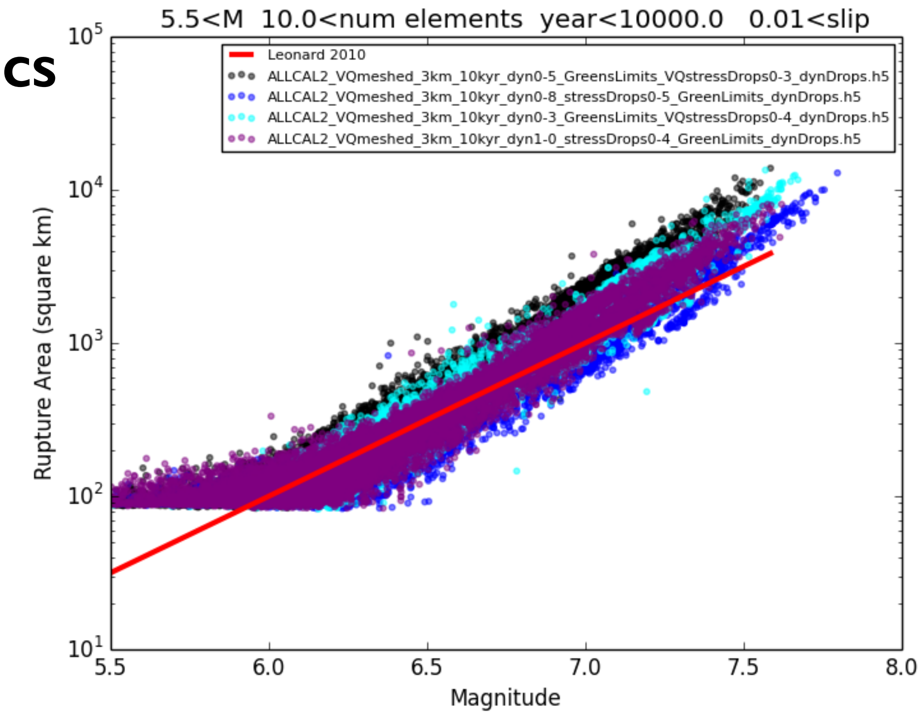
PyVQ - Direct Simulation Data Analysis

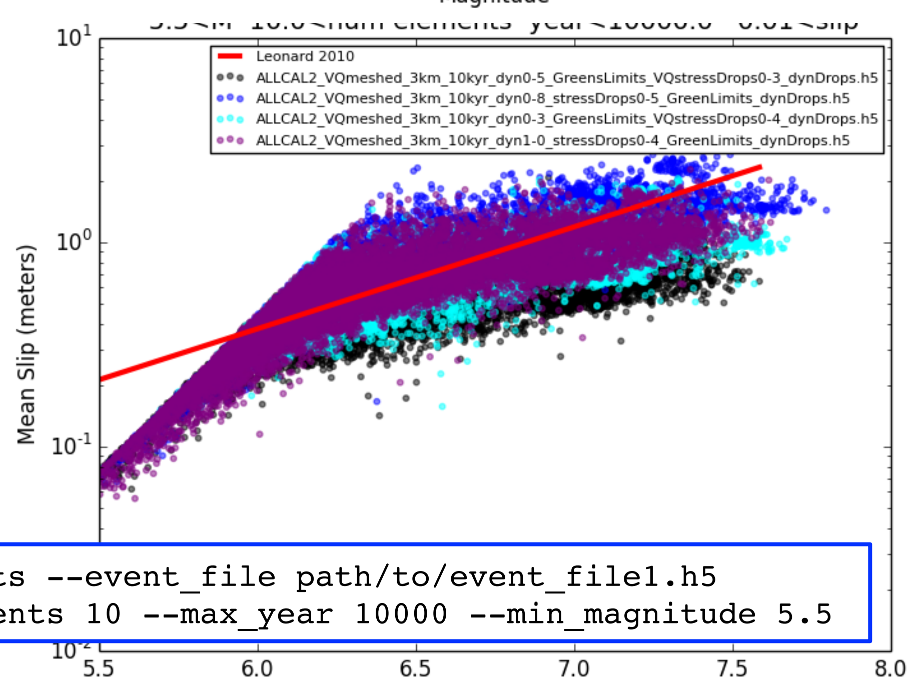
- Computing conditional probabilities for large EQs given the simulated EQ distribution
- Visualizing simulated earthquake statistics
- Visualizing spatial patterns for simulated earthquakes
  - Co-seismic gravity changes
  - Co-seismic InSAR interferograms & surface deformation
  - Co-seismic Geoid height changes
- Visualizing slip during simulated earthquakes

- The next few slides give examples of direct simulation data analysis

Example: Simulated EQ Statistics

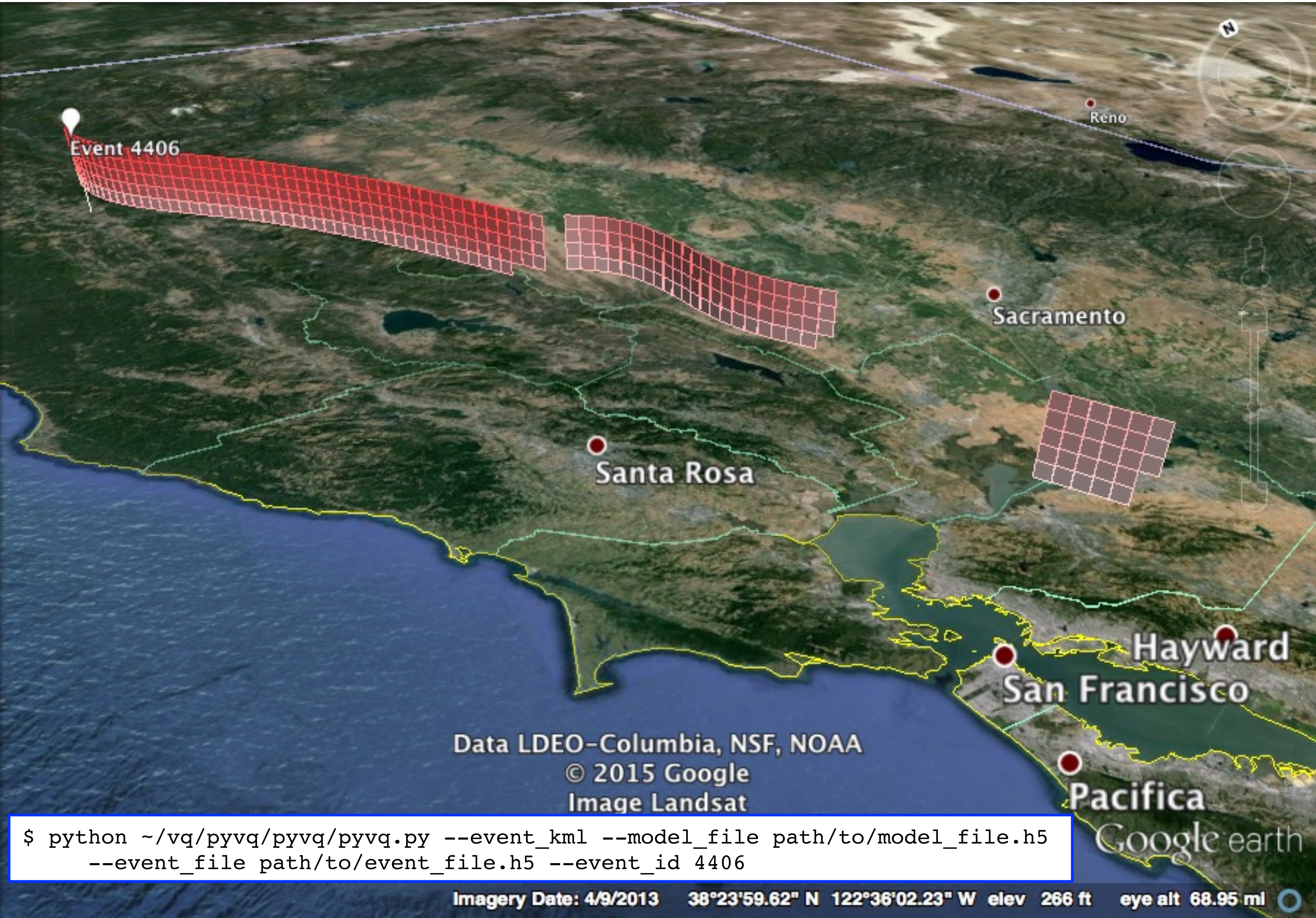






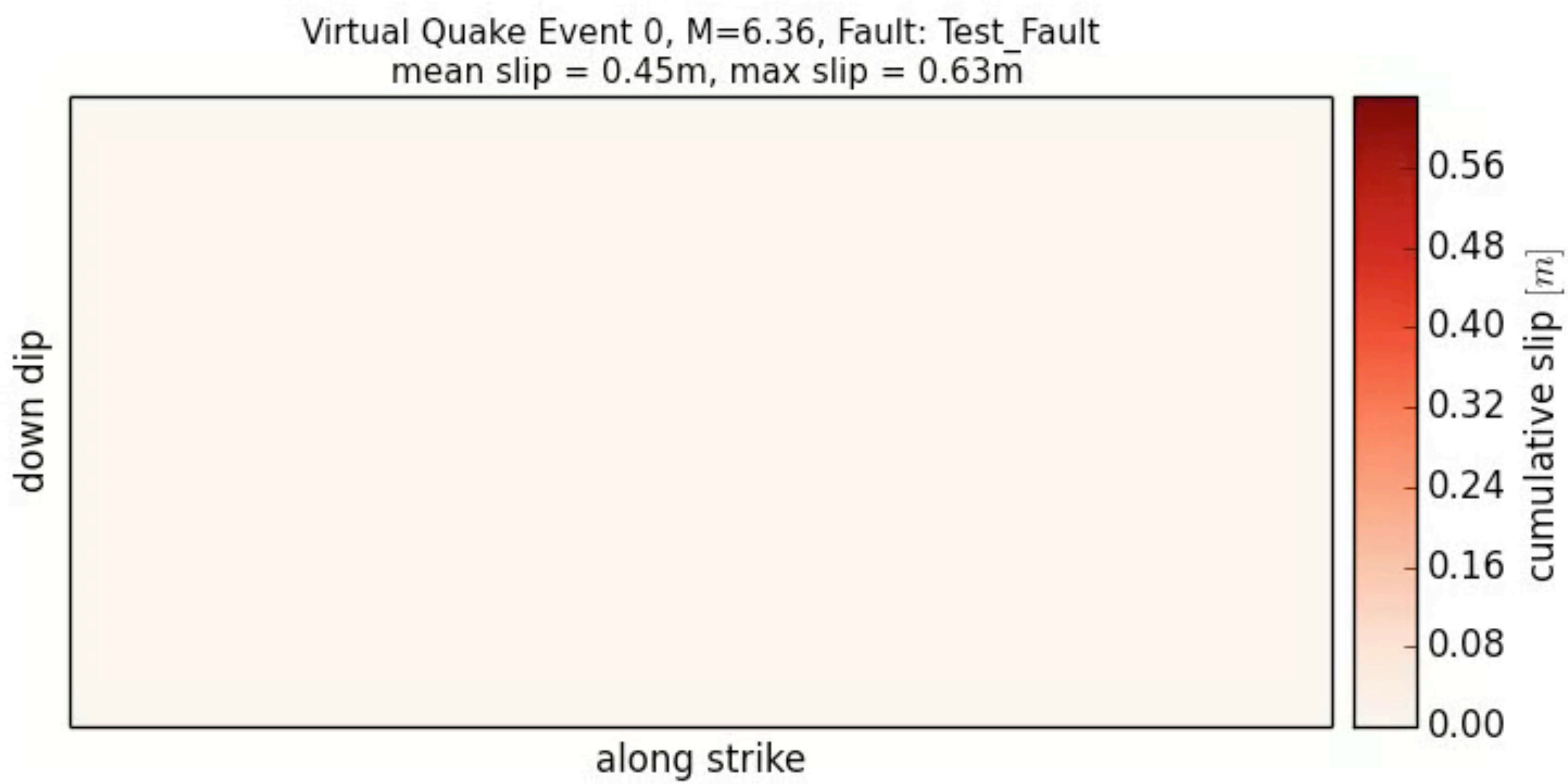

```
$ python ~/vq/pyvq/pyvq/pyvq.py --all_stat_plots --event_file path/to/event_file1.h5
    path/to/event_file2.h5 ... --min_num_elements 10 --max_year 10000 --min_magnitude 5.5
```

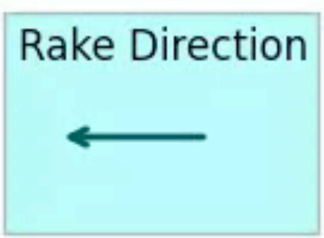
Example Event KML [show Google Earth]




```
$ python ~/vq/pyvq/pyvq/pyvq.py --event_kml --model_file path/to/model_file.h5
    --event_file path/to/event_file.h5 --event_id 4406
```

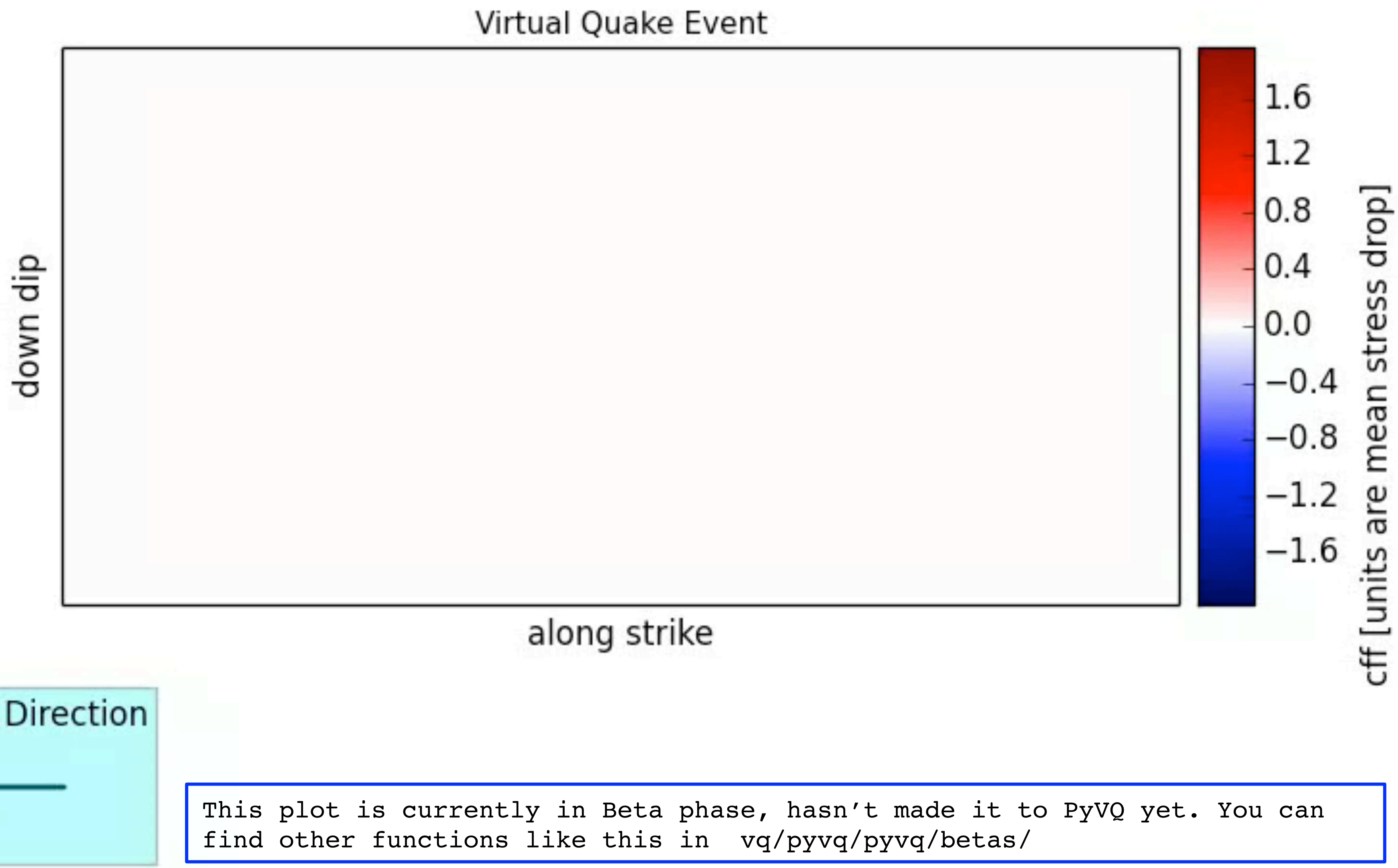
PyVQ Example: Slip "during" simulated earthquake





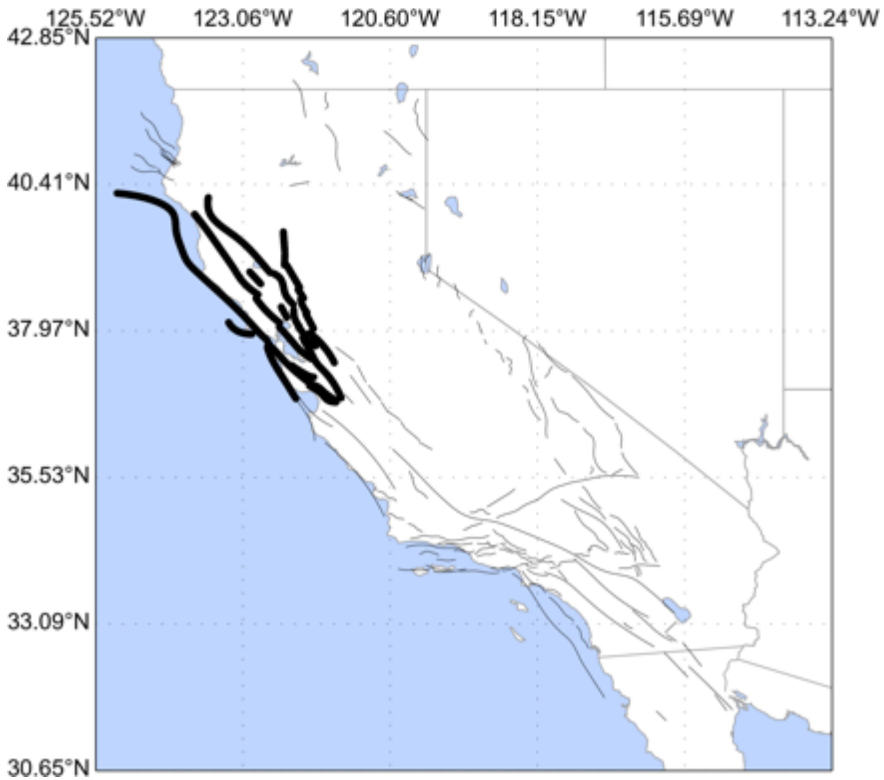

```
$ python ~/vq/pyvq/pyvq/pyvq.py --event_movie --event_id 0
    --model_file path/to/model_file.h5 --event_file path/to/event_file.h5
```

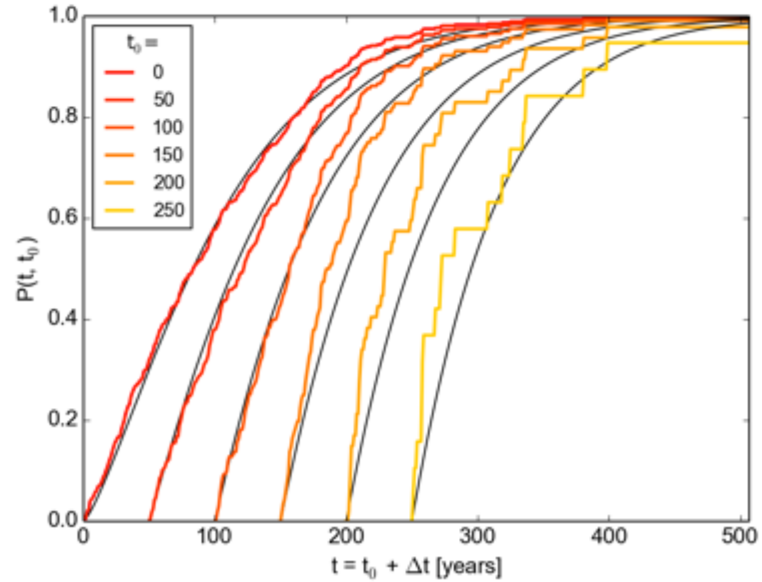
CFF (stress) during simulated EQ

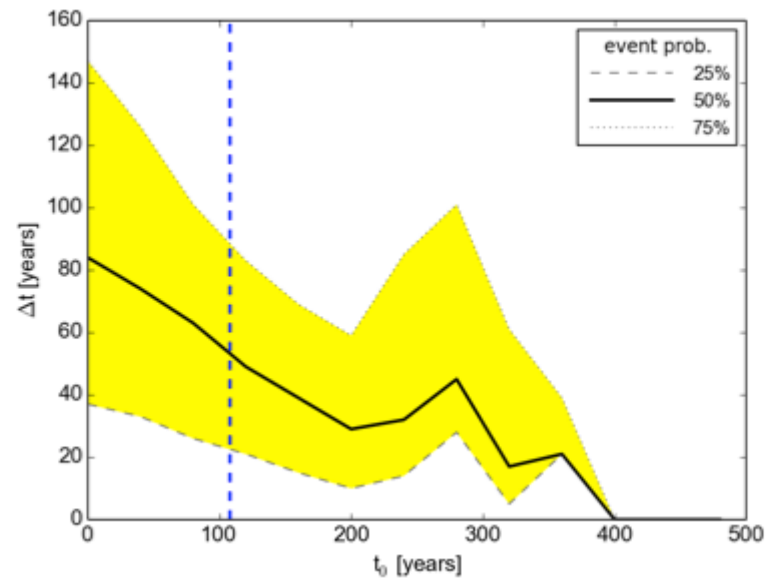



```
This plot is currently in Beta phase, hasn't made it to PyVQ yet. You can
find other functions like this in  vq/pyvq/pyvq/betas/
```

Northern California Earthquake Probabilities, M ≥ 7.5 (Schultz et al. 2015)





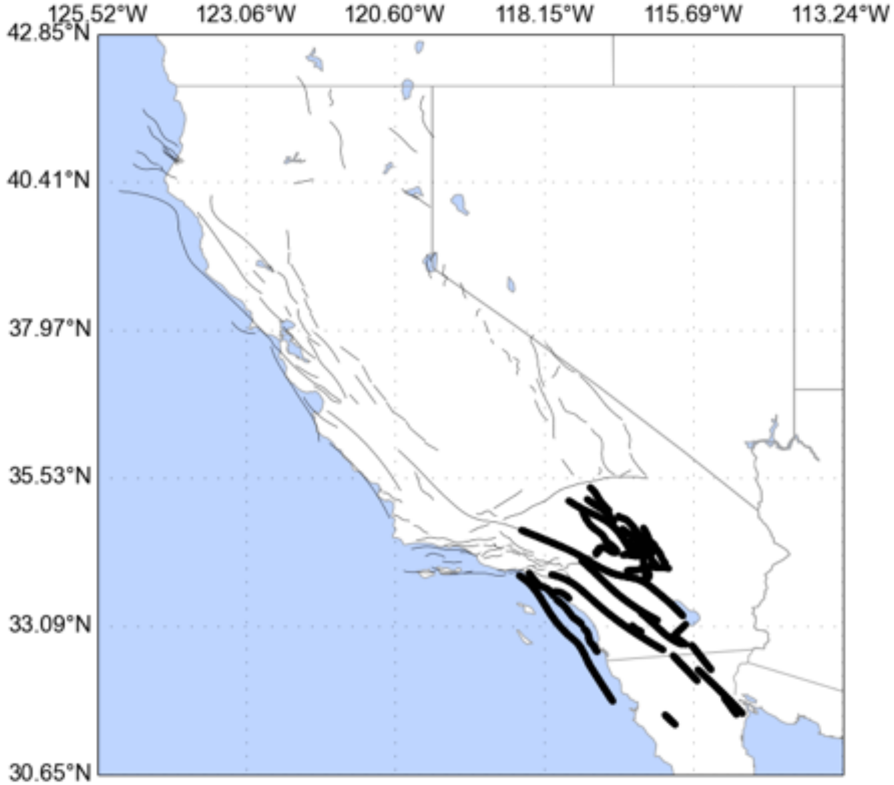


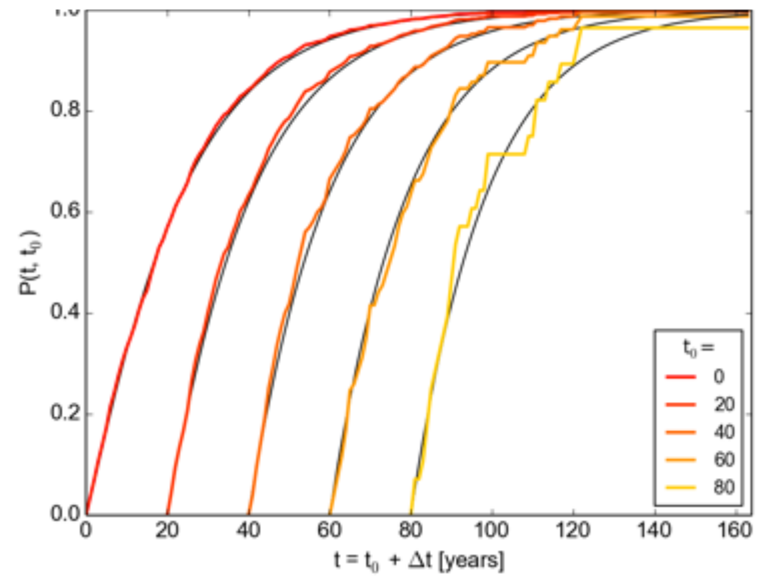
- ★ 50,000 year simulation of UCERF2 model
- ★ 482 earthquakes, M ≥ 7.5
- ★ Mean recurrence 98.4 years
- ★ 109 years since 1906 San Francisco M=7.9

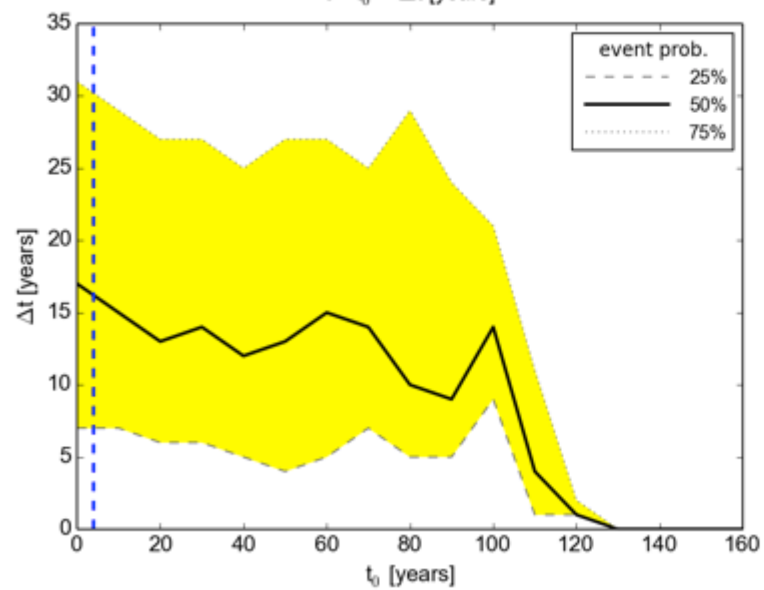
50% prob. of M ≥ 7.5 in next 55 years
75% prob. of M ≥ 7.5 in next 94 years

```
$ python ~/vq/pyvq/pyvq.py --model_file model_file.h5 --event_file event_file.h5
    --traces --plot_cond_prob_vs_t --plot_waiting_times --min_magnitude 7.5
    --use_sections 1 2 3 4 5 6 7 8 ... 26 27 ... 37 40 42
```

Southern California Earthquake Probabilities, M ≥ 7.0 (Schultz et al. 2015)





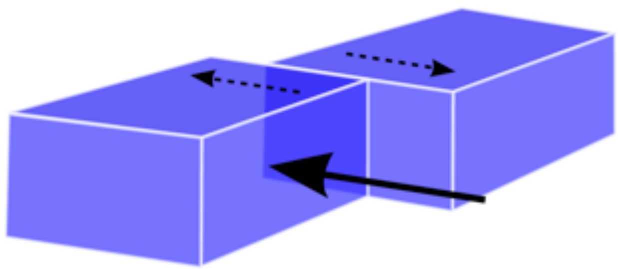


- ★ 50,000 year simulation of UCERF2 model
- ★ 1454 earthquakes, M ≥ 7.0
- ★ Mean recurrence 22.1 years
- ★ 5 years since 2010 El Mayor-Cucapah, M 7.2

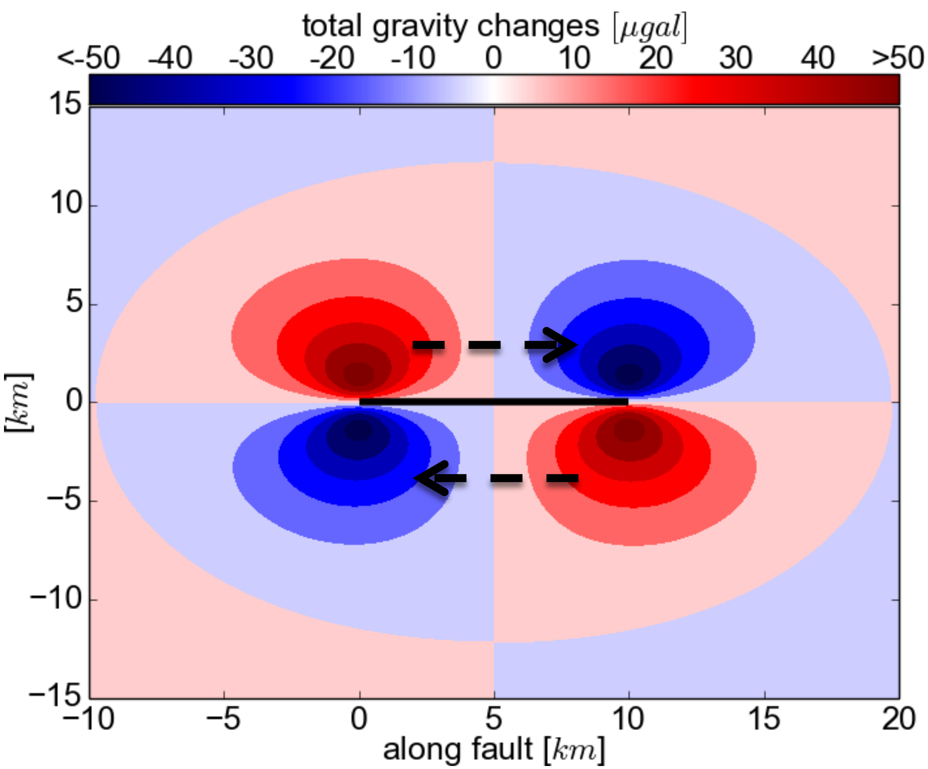
50% prob. of M ≥ 7.0 in next 16 years
75% prob. of M ≥ 7.0 in next 30 years

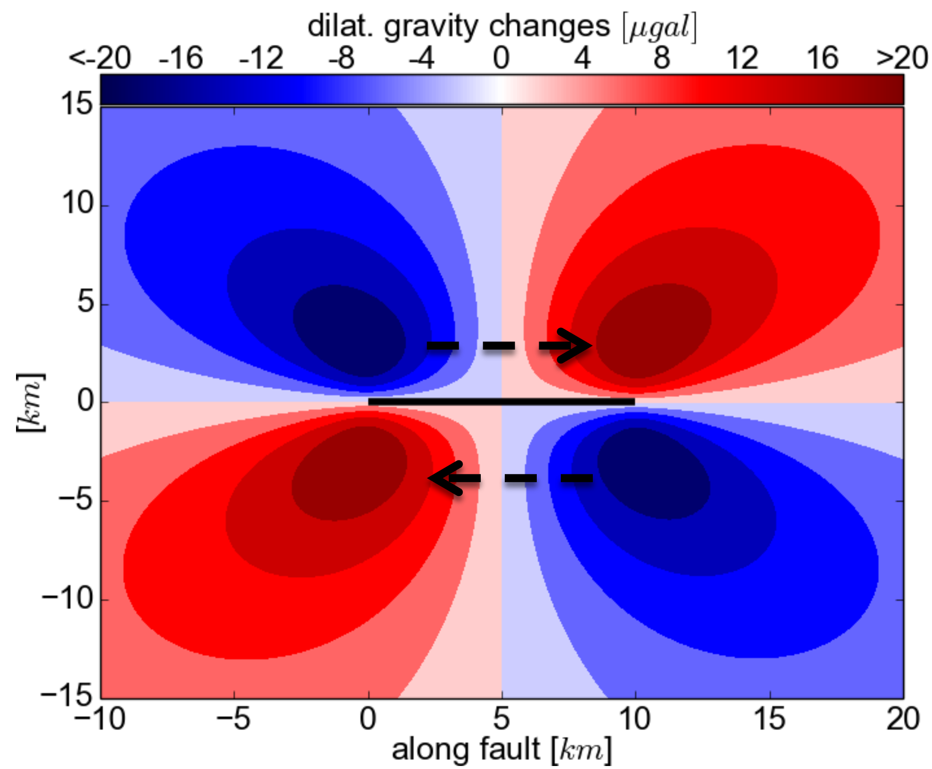
```
$ python ~/vq/pyvq/pyvq/pyvq.py --model_file model_file.h5 --event_file event_file.h5
    --traces --plot_cond_prob_vs_t --plot_waiting_times --min_magnitude 7.0
    --use_sections 126 127 128 129 130 ... 136 137 138 139 140 ... 2181
```

Gravity Change Patterns for Strike-slip Faults (Schultz et al. 2015)



- Greens Functions from Okubo [1992]
- Input: fault geometry + co-seismic slips
- unit: microgal = 10^{-8} m/s^2






```
$ python ~/vq/pyvq/pyvq.pyython ../vq/pyvq/pyvq.py --greens --field_type gravity
    --plot_name "strikeslip_dip90_LL" --uniform_slip 5 --colorbar_max 50
    --levels -50 -40 -30 -20 -10 0 10 20 30 40 50 --rake 0 --dip 90 --DTTF 1000
```

Surface gravity changes for VQ earthquake similar to 1906 San Francisco

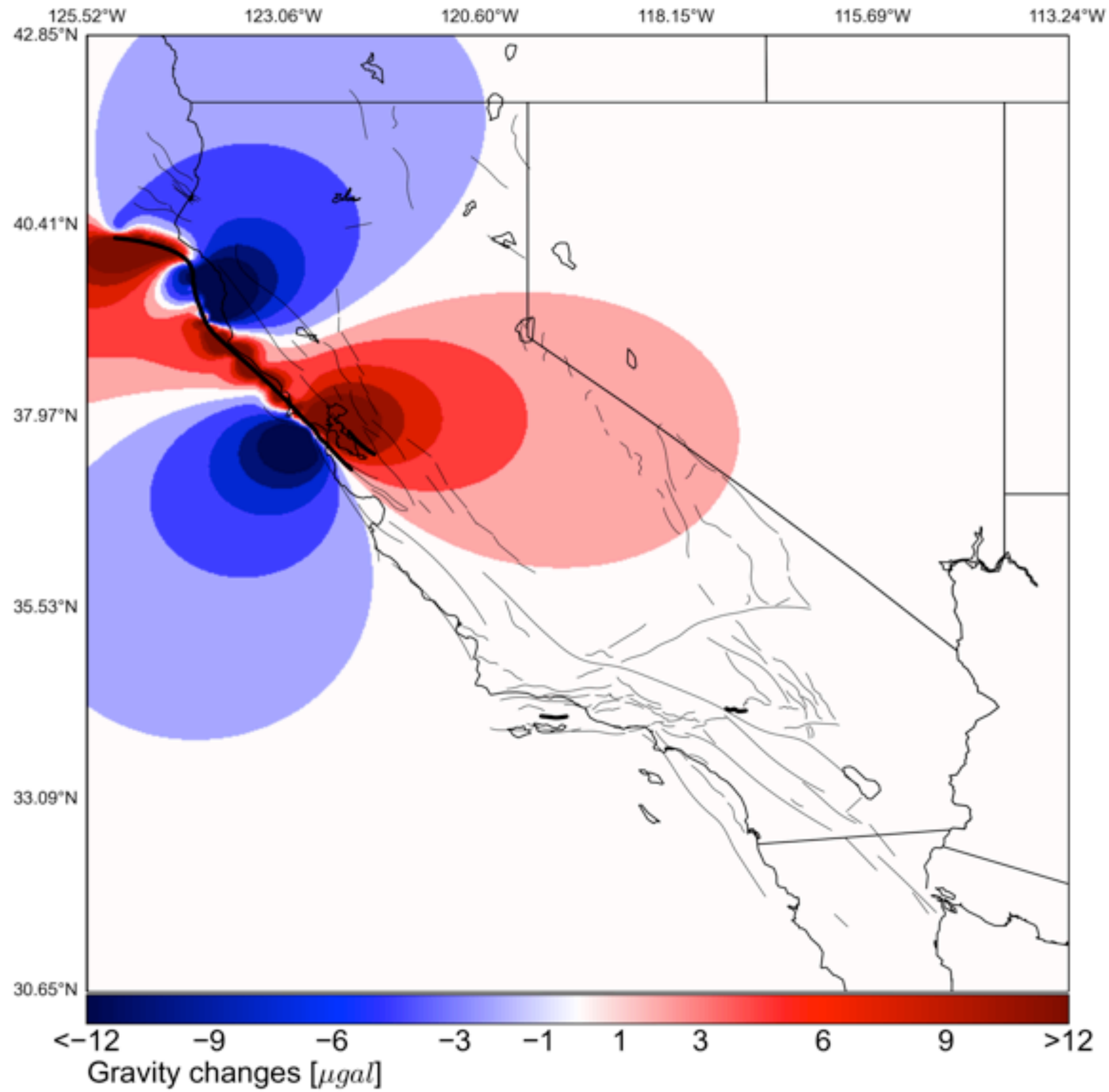
★ The signal you see measuring gravity changes on land with a dense network of sensors

Simulated earthquake

★ Strike-slip, M = 7.88

★ Mean slip 2.2m

★ Surface Rupture Length 712km




```
$ python ~/vq/pyvq/pyvq.py -model_file ../VQModels/allcal_fault_5000.h5
    --event_file events_norcal_fault_5km_5kyr_dyn0-5_BASSgen0.h5
    --field_plot --field_type gravity --event_id 1414 --colorbar_max 20
```

Other VQ Projects Currently Underway

- Combining VQ with the cellular automata tsunami propagation code "Tsunami Squares" [Steve Ward, UCSC] to produce simulated tsunami catalogs
- Developing a spatial verification method to compare simulated earthquake location/magnitude to observed earthquakes
- Developing a method to add fractal "roughness" to our over-simplified and smooth fault planes

- The next few slides give descriptions of these project

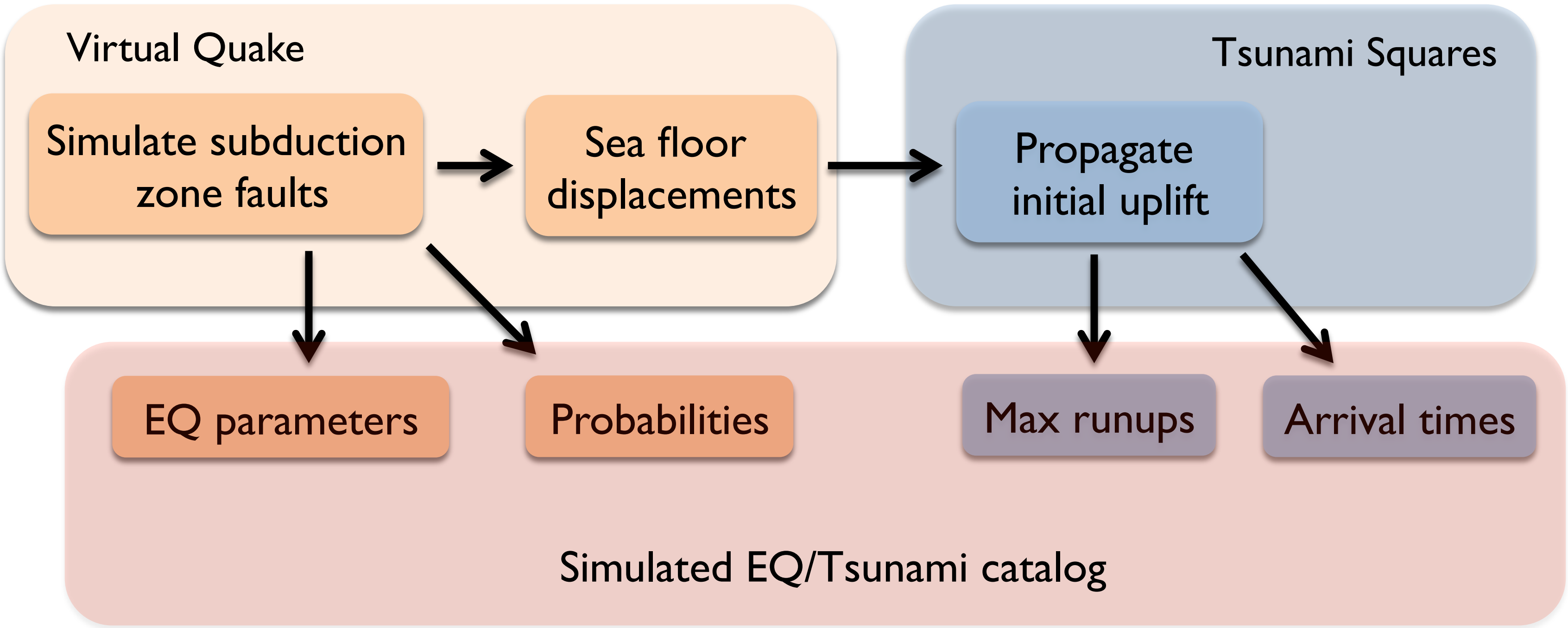
Tsunami Squares [Open-source version under development]

- ★ Cellular automata algorithm for tsunami propagation
- ★ Consistent with observed landslide-generated tsunamis

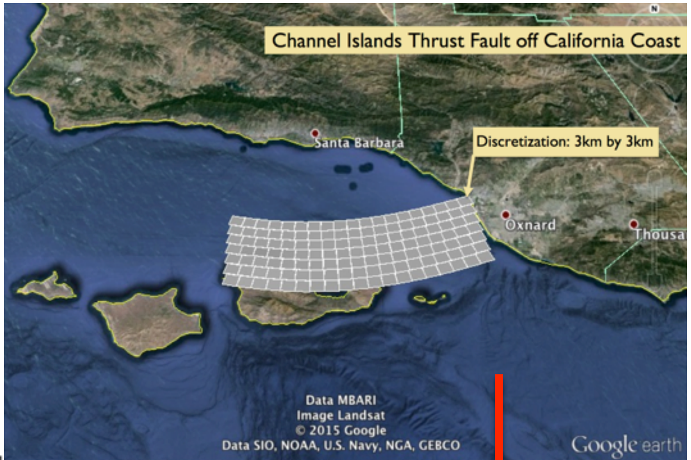
J. Wang, S. N. Ward, and L. Xiao. Numerical simulation of the december 4, 2007 landslide-generated tsunami in chehalis lake, canada. *Geophysical Journal International*, 201(1):372–376, 2015a.

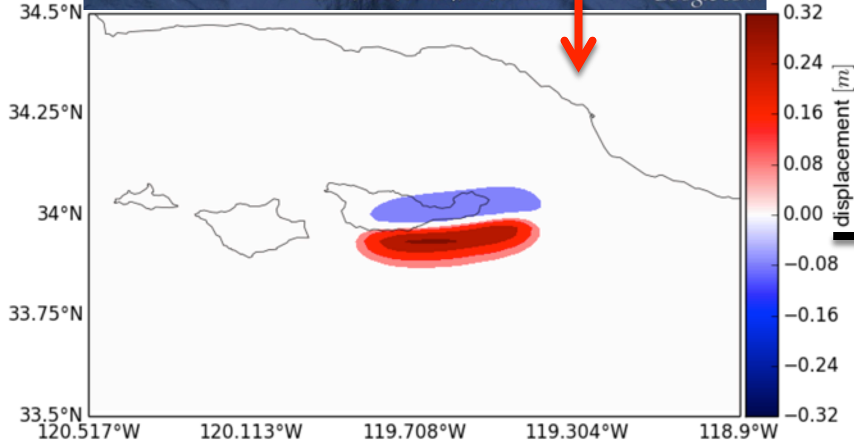
J. Wang, S. N. Ward, and L. Xiao. Numerical modelling of rapid, flow-like landslides across 3-d terrains: a tsunami squares approach to el picacho landslide, el salvador, september 19, 1982. *Geophysical Journal International*, 201(3):1534–1544, 2015b.

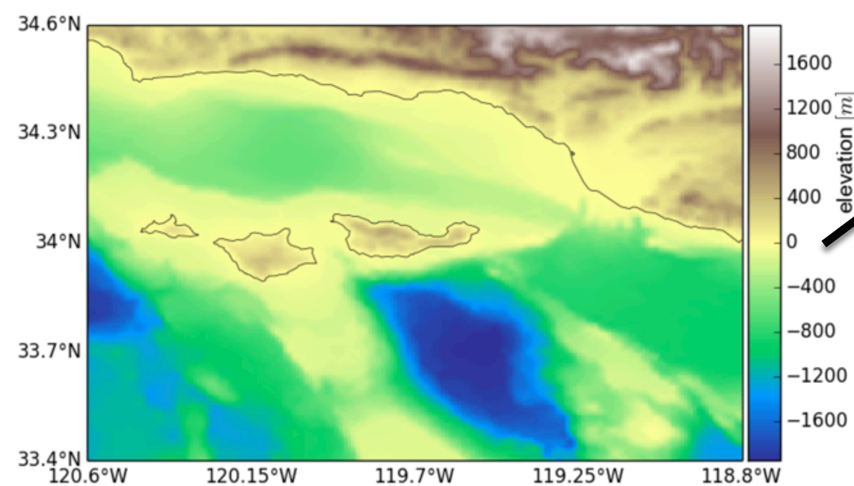
L. Xiao, S. N. Ward, and J. Wang. Tsunami squares approach to landslide-generated waves: Application to gongjiafang landslide, three gorges reservoir, china. *Pure and Applied Geophysics*, pages 1–16, 2015.

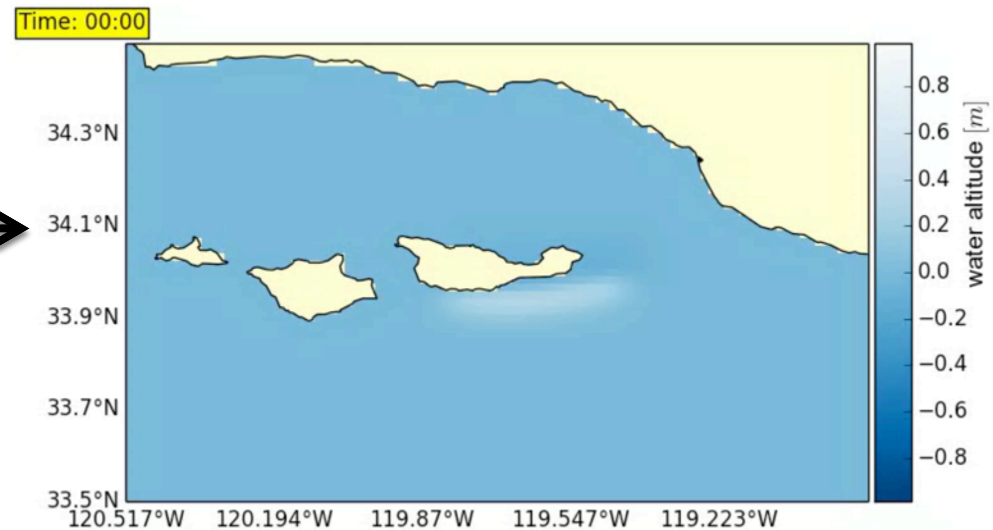


Virtual Quake + Tsunami Squares
[under development]



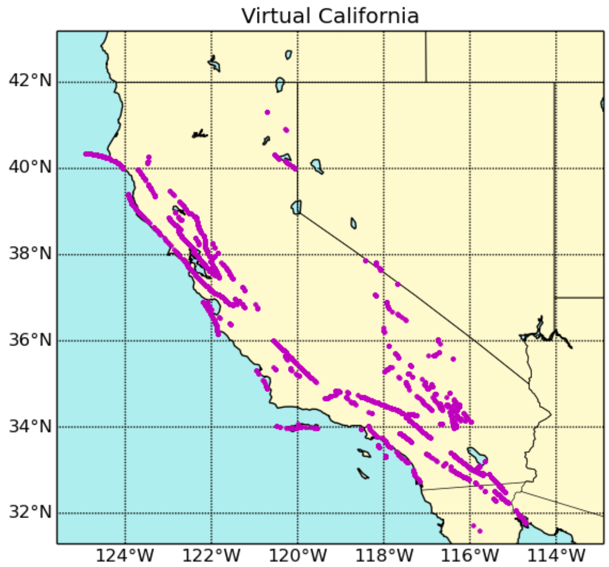


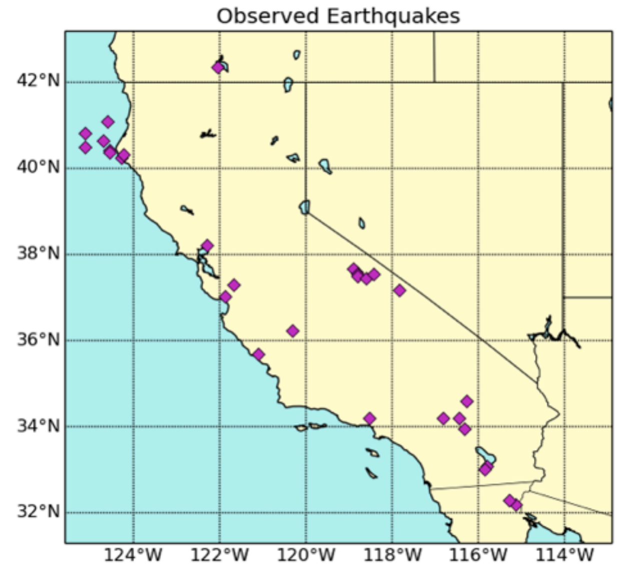




Spatial Verification Using ETAS

- In addition to statistics such as frequency-magnitude relations, slip-magnitude, etc, we wish to verify the spatial distribution of earthquakes from simulators
- Simulators yield earthquakes directly on top of modeled faults
- Most observed earthquakes occur off of known major faults





Spatial Verification Using ETAS

- Use Epidemic-Type AfterShock (ETAS) model to smear simulated earthquake seismicity through the all of space
- Omori modification to Gutenberg-Richter statistics, for an earthquake of magnitude *m* produces a sequence of aftershocks, down to magnitude m_c, numbering

$$N(> m_c)_{Omori} = 10^{b(m-\Delta m-m_c)}$$

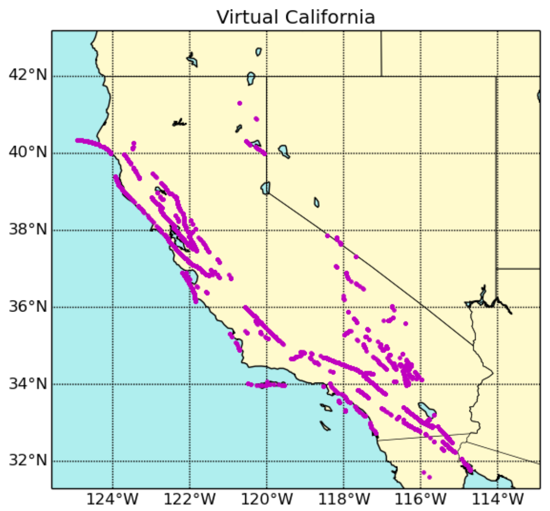
where Δm is 1.0. This number of aftershocks is then distributed over a spatial variant of an Omori power law distribution

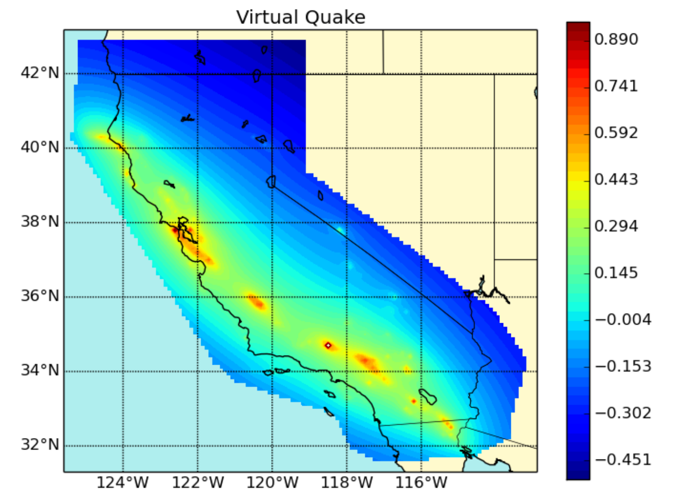
$$\frac{d\,N}{d\,r} = \frac{1}{\chi(r_0+r)^q}$$

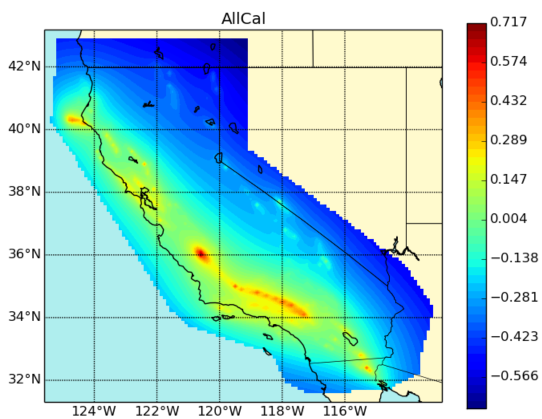
The parameter q is determined observationally to be 1.5. r_0 is related to the rupture length of the mainshock, and χ is determined from the normalization condition that all aftershocks must sum to N_{Omori}

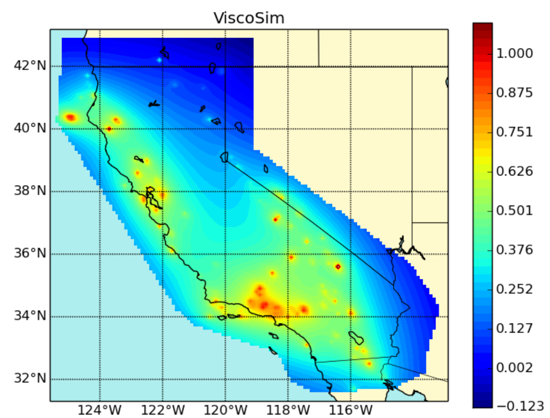
Spatial Verification Using ETAS

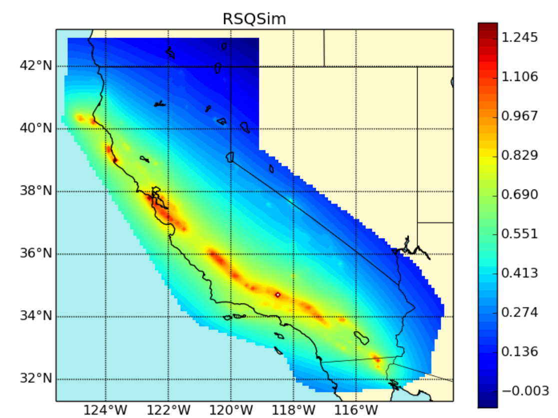
- Divide test region around California into 0.1° x 0.1° degree bins





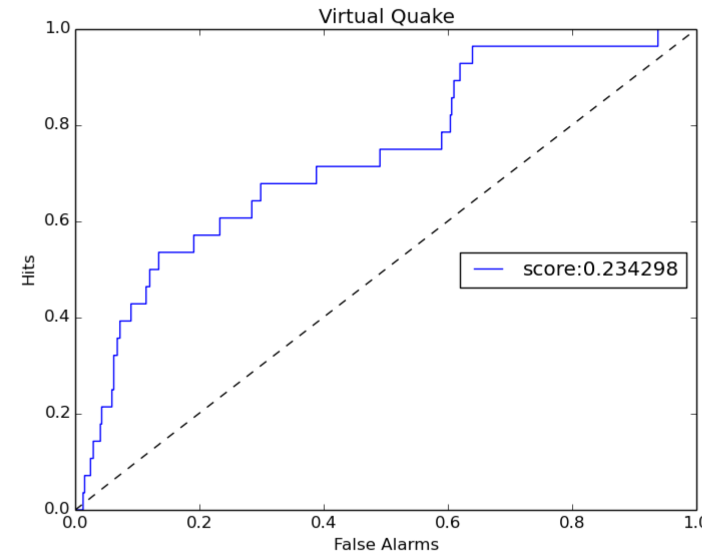


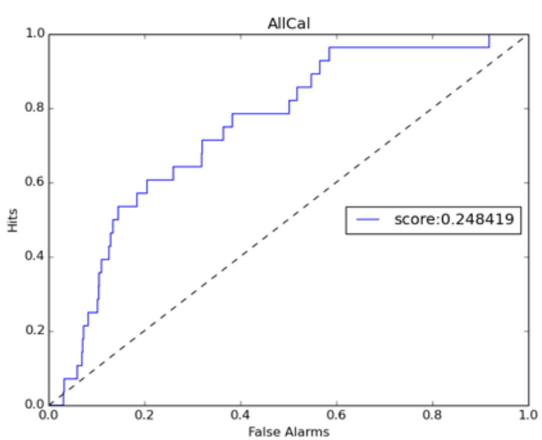


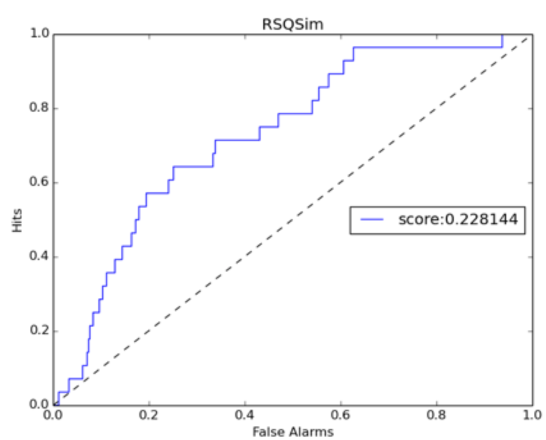


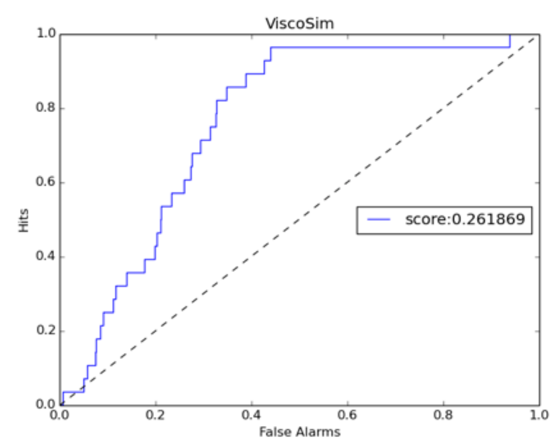
Spatial Verification: ROC method

- For a given rate map, spatial bins are ranked by their rate, and only bins ranked higher than a set threshold are considered positive forecasts (initially, only highest-ranked bin)
- If a real earthquake occurred in this bin, a "Hit" is recorded. If not, it is a "False Alarm"
- The threshold is then lowered, and the top two bins are considered in the same manner. Hits and False Alarms are for lower thresholds, until the whole map is considered a positive forecast.
- For all threshold values, Hits are plotted against False Alarms. The area under this curve is a measure of the skill of the forecast









Future Research with Virtual Quake: An Example

- Test the effect of modifying the input faults to include fractal distribution of "roughness"
  - Perturb the lat/lon of fault trace points along the strike according to a fractal distribution
  - Fix the end points and control the amount of perturbation to keep large scale fault geometry constant
  - Will this lead to aftershocks and foreshocks? Introduces anisotropy in shear and normal stress interactions

Future Research with Virtual Quake: An Example

Smooth Fault, 192 km in length

Future Research with Virtual Quake: An Example

Fractalized! Power spectrum power law exponent $\beta = 1.0$

Thank You for Tuning In

- Virtual Quake is funded by NASA, currently managed by Prof. Rundle's group at UC Davis

- Please read the users' manual on the VQ page at CIG

  www.geodynamics.org/software/vq

- We are open to collaboration, contact us:

  kwschultz@ucdavis.edu & jhnwilson@ucdavis.edu

UCDAVIS UNIVERSITY OF CALIFORNIA

NASA

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS