Notes on using `surfacevel2strain.m`

A Matlab program to convert surface velocity fields into strain-rate maps

Carl Tape, Pablo Musé, Mark Simons

January 23, 2011

```
surfacevel2strain/USER_INFO/surfacevel2strain_manual.pdf -- this document
surfacevel2strain/USER_INFO/Tape2009gps.pdf              -- GJI2009
surfacevel2strain/USER_INFO/Tape2009gps_supplement.pdf   -- supplemental notes
surfacevel2strain/matlab/                                -- matlab scripts
```

Please email me (`carltape@gi.alaska.edu`) with suggestions or corrections for the code or these notes.

1 Introduction

The purpose of this program is to take an input set of discrete velocity vectors on the surface of a sphere, and then *estimate* a continuous velocity vector field on the surface. The estimated field is based on weighted and damped least squares, where the weights are determined from the errors associated with the measurements, but it does not use off-axis covariance values. The program considers 2- or 3-component velocity fields. Several real and synthetic examples were demonstrated in *Tape et al.* (2009).

The general procedure is this:

1. Specify a latitude-longitude box for your region of interest. For example, for Japan, we might use: `lonmin0 = 128.0 ; lonmax0 = 147.0 ; latmin0 = 30.0 ; latmax0 = 46.0` This box is used to get the possible wavelet center-points for the estimation, and also to describe the bounds for plotting.
2. Obtain a set of 2- or 3-component velocity field data: $\mathbf{v}(r_i,\, \theta_i,\, \phi_i)$, where i denotes the index of the observation. Observations outside the bounding region above will be excluded.
3. Save velocity field observations in a "standard" format.
4. Specify parameters in `surfacevel2strain.m` and compute the estimated continuous velocity field.
5. Plot output files in GMT.

1.1 Example with 1D data: `sphereinterp.m`

In order to demonstrate the basic features of `surfacevel2strain.m`, we will demonstrate the problem of estimating a continuous field on the sphere from a set of discrete 1D observations. The example data are Moho depths derived from receiver functions in southern California (*Yan and Clayton*, 2007), embedded within Moho depths from Crust2.0 (*Bassin et al.*, 2000). The data sets from these studies are available on-line.

The default example for `sphereinterp.m` should produce Figures 1–6, in addition to the text in Appendix A. We will review some of the key steps of the estimation below.

1. Obtain a set of observations and associated uncertainties (Figure 1).
2. Obtain the center-points for all basis functions (spherical wavelets) to be used in estimating the continuous field (Figure 2). A basis function is kept if a specified number of observation locations is within its spatial support.
3. Obtain a regularization parameter for the inverse problem (Figure 3).
4. Solve least-squares inverse problem for the model vector, which contains coefficients of basis functions. Then compute the estimated field, then analyze the residuals (Figures 4 and 5).
5. Using the same model vector, plot the estimated field on a uniform mesh (Figure 6). A fancier rendition of Figure 6 is shown in Figure 7.

2 Preparing the velocity field dataset (`get_gps_dataset.m`)

The most general velocity field dataset is one that contains the data covariances and standard errors. For a three-component field, the data covariance matrix will have six values per observation. Thus, for each new dataset, I save a version in a "standard" format. In `get_gps_dataset.m`, the command

```
[dlon,dlat,ve,vn,vu,se,sn,su,ren,reu,rnu,start_date,finish_date,name] = read_gps_3D(filename);
```

will load the pre-saved GPS dataset. For the Japan example, see `read_gps_japan.m` and `japan_gps_dat.m` as a guide.

The variables `start_date`, `finish_date`, and `name` are not used in `surfacevel2strain.m`. For now, the program does not use the covariance terms `ren`, `reu`, `rnu`.

3 Running `surfacevel2strain.m`

We now have simply a set of files, one for each grid q, that contain the center-points of what will be spherical wavelet basis functions.

I tested a run using the japan velocity field, considering the horizontal components only (ndim = 2), and performing the estimation for scales q=0 to q=8.

Here is the output to the Matlab command window:

```
OUTPUT HERE
>>
```

The program prompts the user for various input parameters, and more are needed prior to running the program. The first run will threshold the basis functions and compute the design matrix for the inverse problem. There are several ways to select the damping parameter. Figure ?? shows the output figure from `ridge_carl.m` showing three different parameter selection techniques.

At this point, the inverse problem is done. Now, the rest of the program is for plotting. Because computing the design matrix and the damping parameter can be computationally intensive, we do not clear these variables on subsequent runs when we are re-plotting different things. In other words, we now execute the program again:

```
>>
```

The output is a set of figures showing the estimated velocity fields, as well as scalar fields derived from the spatial velocity gradient tensor field, $\mathbf{L}(\phi, \theta)$.

Plotting scalar and vector fields in GMT

There are several `perl` scripts for plotting in GMT here: `surfacevel2strain/gmt/`. These should be thought of a a *guide* for plotting and will require several modifications for your own purposes. The primary script to run is `plot_strain.pl`.

Example output figures are shown in Figures ??–??.

References

Bassin, C., G. Laske, and G. Masters (2000), The current limits of resolution for surface wave tomography in North America, in *EOS Trans. Am. Geophys. Un.*, vol. F897, p. 81.

Tape, C., P. Musé, M. Simons, D. Dong, and F. Webb (2009), Multiscale estimation of GPS velocity fields, *Geophys. J. Int.*, *179*, 945–971.

Yan, Z., and R. W. Clayton (2007), Regional mapping of the crustal structure in southern California from receiver functions, *J. Geophys. Res.*, *112*, B05311, doi:10.1029/2006JB004622.

A Output from `sphereinterp.m`

Below is the output generated to the Matlab command window in the process of generating Figures 1–6 (Section 1.1).

```
 Type an index corresponding to a region (1=socal): 1
 Type an index corresponding to a dataset (1=moho): 1
getsubset.m: 124 points in the subset out of 124
------------------------------------------------------------
entering sphereinterp_grid.m to obtain spherical wavelet basis functions
Support of the spherical wavelets:
   q      deg          km
   0     82.442     9167.1
   1     47.310     5260.7
   2     24.707     2747.3
   3     12.500     1389.9
   4      6.268      697.0
   5      3.136      348.8
   6      1.569      174.4
   7      0.784       87.2
   8      0.392       43.6
   9      0.196       21.8
  10      0.103       11.4
  11      0.052        5.8
  12      0.026        2.9

minimum allowable grid order is 3
   1.39e+06 meters (support of q = 3 wavelet) < 1.94e+06 meters (2*Lscale)
getspheregrid.m: lon-lat subregion of sphere

GRID ORDER q = 3
  dbase = 7.929e+00 deg
  lonmin, lonmax, latmin, latmax:
  -145.79, -89.21, 6.21, 61.79
```

```
  phmin, phmax, thmin, thmax:
  -2.5445, -1.5570, 0.4924, 1.4624
Patch occupies this fraction of the sphere: 6.074e-02
This corresponds to a square patch with side length 50.057 deg
getting the gridpoints for q = 3
  number of total possible gridpoints : 642
  maximum number of subset gridpoints : 39
q = 3, nf = 8, dbase = 7.929e+00 deg
  actual number of subset gridpoints : 42

GRID ORDER q = 4
  dbase = 3.965e+00 deg
  lonmin, lonmax, latmin, latmax:
  -133.89, -101.11, 18.11, 49.89
  phmin, phmax, thmin, thmax:
  -2.3369, -1.7646, 0.7000, 1.2548
Patch occupies this fraction of the sphere: 2.068e-02
This corresponds to a square patch with side length 29.207 deg
getting the gridpoints for q = 4
  number of total possible gridpoints : 2562
  maximum number of subset gridpoints : 53
q = 4, nf = 16, dbase = 3.965e+00 deg
  actual number of subset gridpoints : 53

GRID ORDER q = 5
  dbase = 1.982e+00 deg
  lonmin, lonmax, latmin, latmax:
  -127.95, -107.05, 24.05, 43.95
  phmin, phmax, thmin, thmax:
  -2.2331, -1.8684, 0.8038, 1.1510
Patch occupies this fraction of the sphere: 8.312e-03
This corresponds to a square patch with side length 18.517 deg
getting the gridpoints for q = 5
  number of total possible gridpoints : 10242
  maximum number of subset gridpoints : 85
q = 5, nf = 32, dbase = 1.982e+00 deg
  actual number of subset gridpoints : 78

GRID ORDER q = 6
  dbase = 9.912e-01 deg
  lonmin, lonmax, latmin, latmax:
  -124.97, -110.03, 27.03, 40.97
  phmin, phmax, thmin, thmax:
  -2.1812, -1.9203, 0.8557, 1.0991
Patch occupies this fraction of the sphere: 4.179e-03
This corresponds to a square patch with side length 13.130 deg
getting the gridpoints for q = 6
  number of total possible gridpoints : 40962
  maximum number of subset gridpoints : 171
q = 6, nf = 64, dbase = 9.912e-01 deg
  actual number of subset gridpoints : 158

GRID ORDER q = 7
  dbase = 4.956e-01 deg
  lonmin, lonmax, latmin, latmax:
  -123.49, -111.51, 28.51, 39.49
  phmin, phmax, thmin, thmax:
```

```
  -2.1553, -1.9463, 0.8816, 1.0731
Patch occupies this fraction of the sphere: 2.636e-03
This corresponds to a square patch with side length 10.429 deg
getting the gridpoints for q = 7
  number of total possible gridpoints : 163842
  maximum number of subset gridpoints : 432
q = 7, nf = 128, dbase = 4.956e-01 deg
  actual number of subset gridpoints : 401

GRID ORDER q = 8
  dbase = 2.478e-01 deg
  lonmin, lonmax, latmin, latmax:
  -122.74, -112.26, 29.26, 38.74
  phmin, phmax, thmin, thmax:
  -2.1423, -1.9592, 0.8946, 1.0602
Patch occupies this fraction of the sphere: 1.997e-03
This corresponds to a square patch with side length 9.077 deg
getting the gridpoints for q = 8
  number of total possible gridpoints : 655362
  maximum number of subset gridpoints : 1309
q = 8, nf = 256, dbase = 2.478e-01 deg
  actual number of subset gridpoints : 1216

 threshold the gridpoints

     q   num    id    i1    i2
     3    12    12     1    12
     4    14    26    13    26
     5    24    50    27    50
     6    45    95    51    95
     7    77   172    96   172
     8   102   274   173   274

GRIDPOINTS q = 3 to 8 (274):
274 wavelets / 1948 total with >= 3 stations inside their corresponding spatial supports

 q = 3 to 8, j = 1 to 274 (274)
 q = 3 to 6, j = 1 to 95 (95)
 q = 7 to 7, j = 96 to 172 (77)
 q = 8 to 8, j = 173 to 274 (102)
------------------------------------------------------------
entering sphereinterp_est.m to estimate smooth scalar field on the sphere
no input polygon provided --> full plotting grid will be used
choice of regularization parameter:
ordinary cross validation
input uncertainties will be used within inversion

Constructing the design matrix...
 creating the L-curve...
 ii = 1/40, lam = 0.001
 ii = 2/40, lam = 0.0017013
   :
 ii = 37/40, lam = 203091.7621
 ii = 38/40, lam = 345510.7295
 ii = 39/40, lam = 587801.6072
 ii = 40/40, lam = 1000000
```

```
Pick the regularization parameter:
L-curve lambda = 7.017e-02 (index 9)
    OCV lambda = 3.455e-01 (index 12)
    GCV lambda = 2.031e-01 (index 11)
your pick lam0 = 3.455e-01 (index 12)
 computing the model vector...

 computing values at the plotting points...
100 out of 2200
  :
2200 out of 2200
Elapsed time is 15.812080 seconds.

Number of observations, ndata = 124
Number of basis functions, ngrid = 274
For testing purposes, try decreasing one of these:
  qmax = 8, the densest grid for basis functions
  nx = 50, the grid density for plotting
  ndata = 124, the number of observations (or ax0)
>>
```

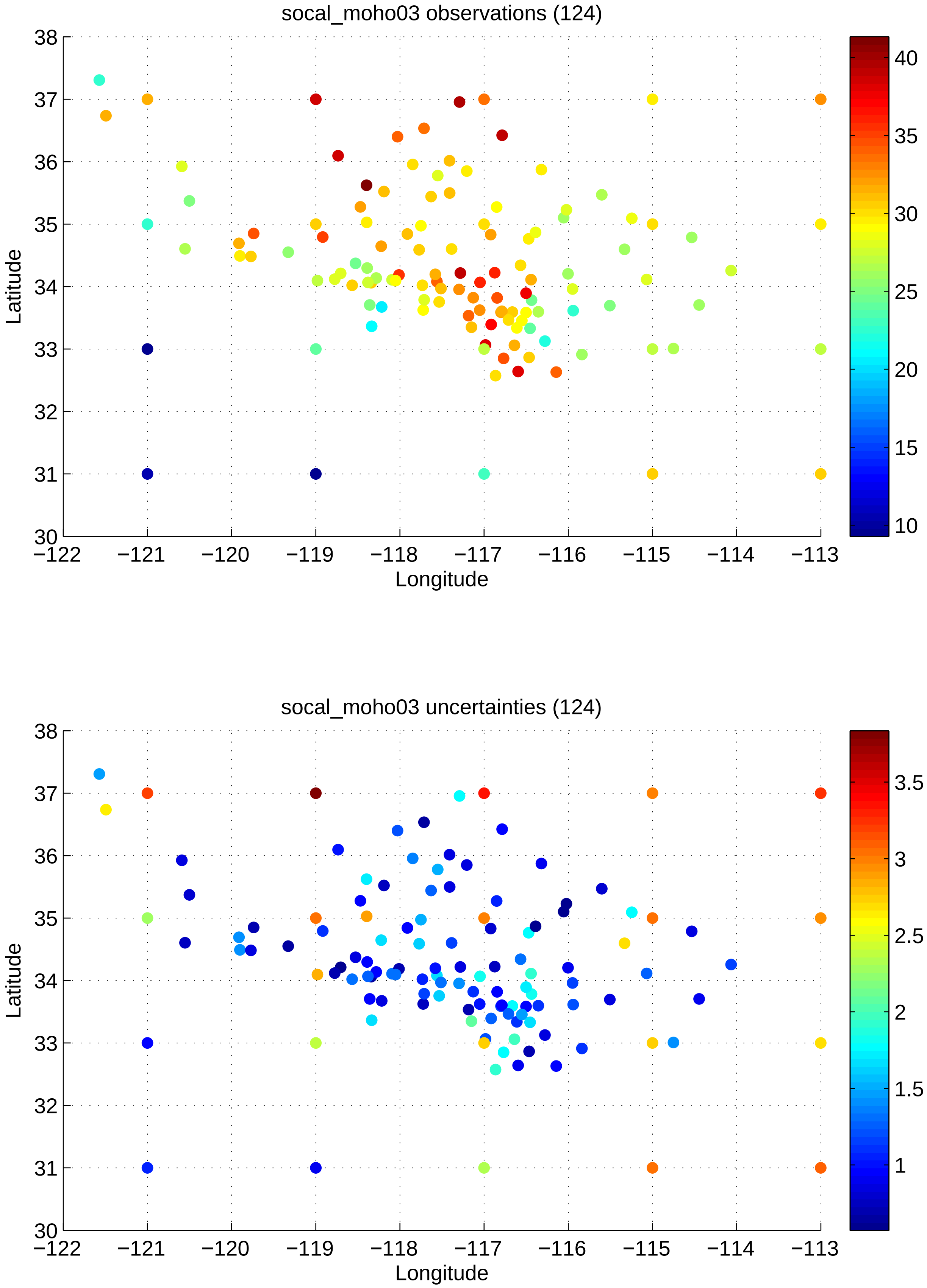

Figure 1: Example 1D data set for estimating a continuous function on the sphere (Section 1.1). The data set is comprised of Moho depth estimates at discrete locations. The top plot is for the Moho depth, and the bottom plot is for the estimated uncertainty associated with each depth. Values are from *Yan and Clayton* (2007) and *Bassin et al.* (2000).

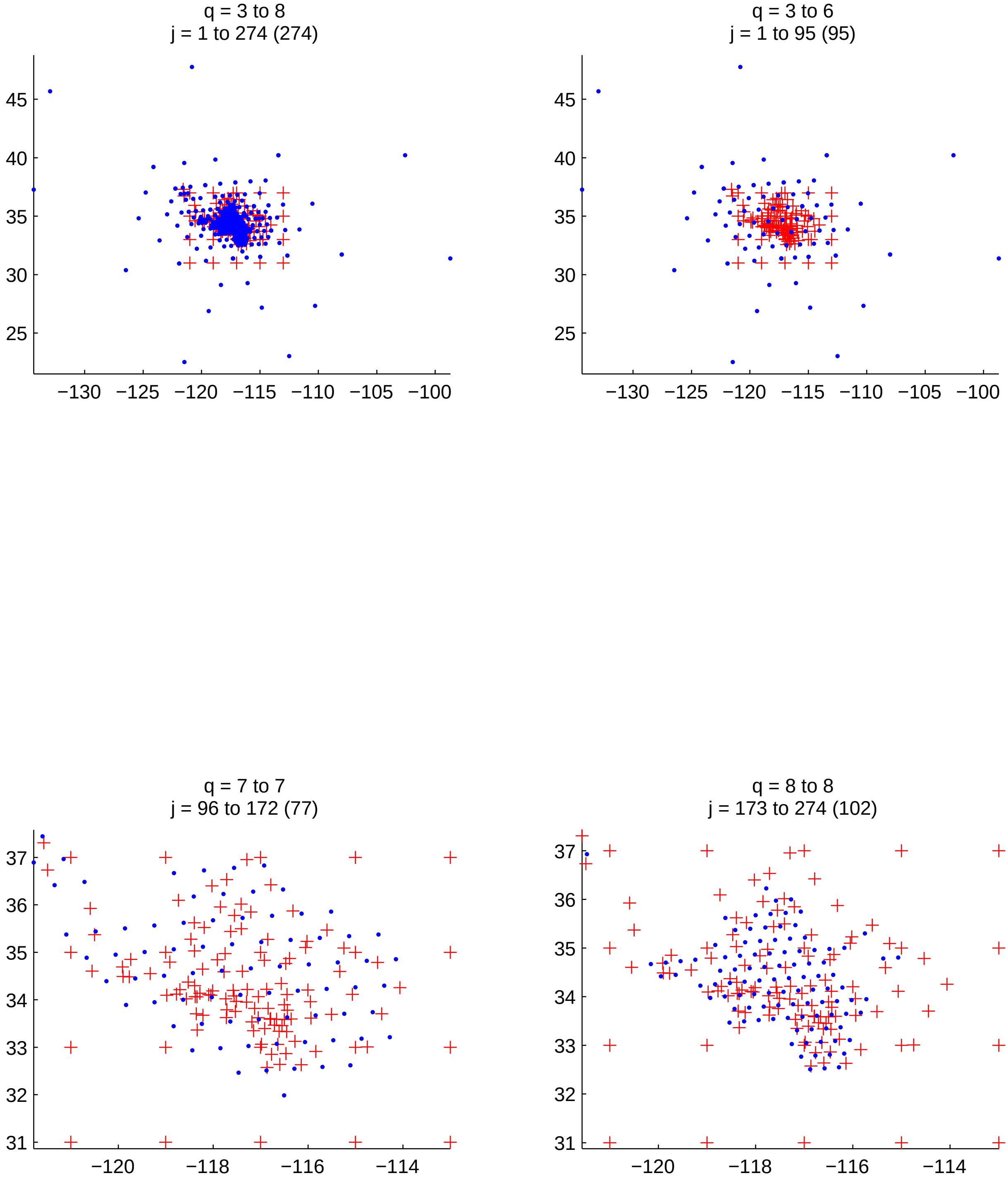

Figure 2: Basis function gridpoint centers for grids q=3-8, q=3-6, q=7, and q=8. Observation locations are plotted at red '+' markers (Figure 1). Note that the axes range varies for each subplot.

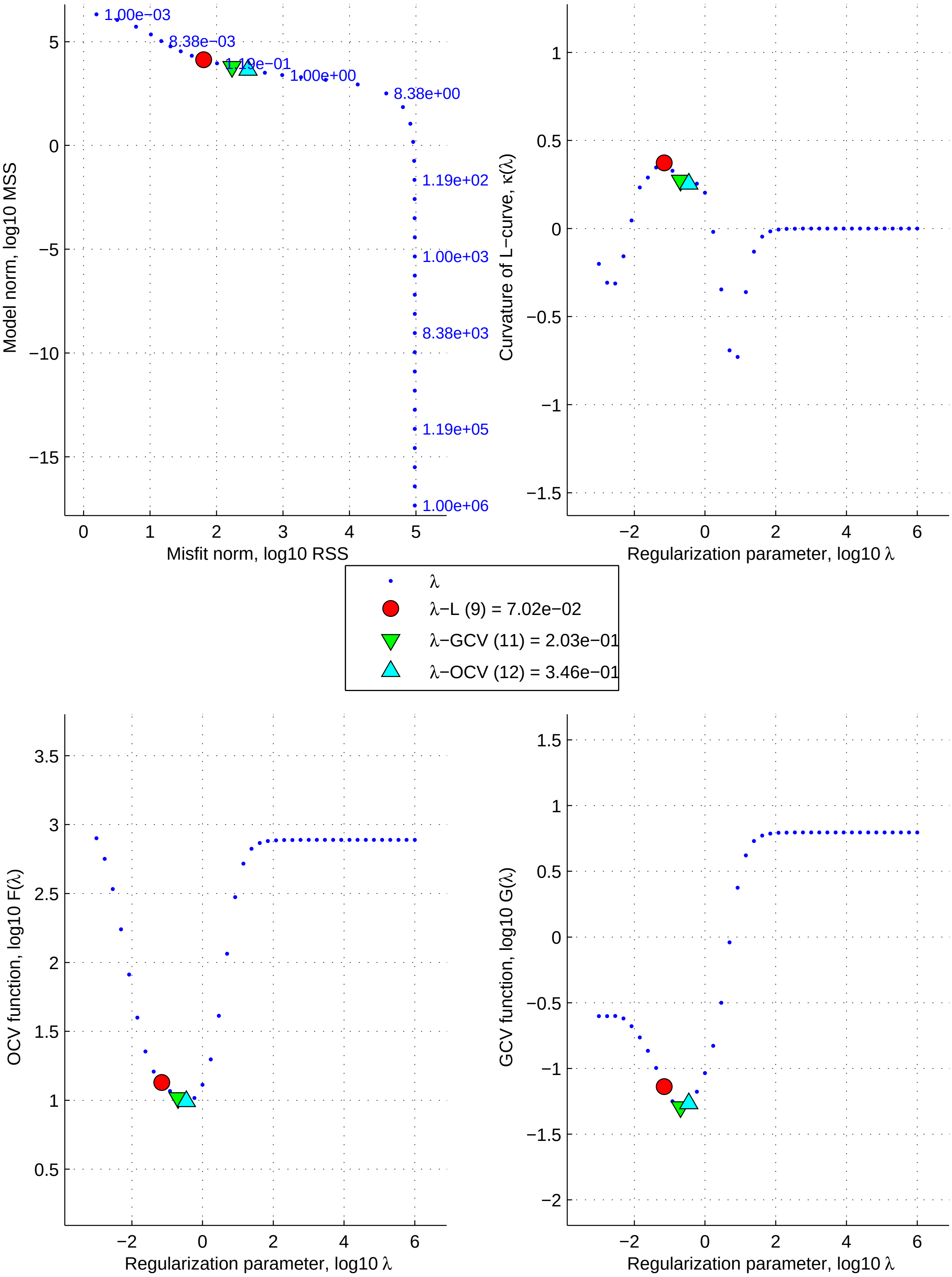

Figure 3: Selection of regularization parameter, λ, considering three different approaches: L-curve, ordinary cross-validation, general cross-validation. See *Tape et al.* (2009) for details and references. For this example, all three techniques select a similar regularization paramter.

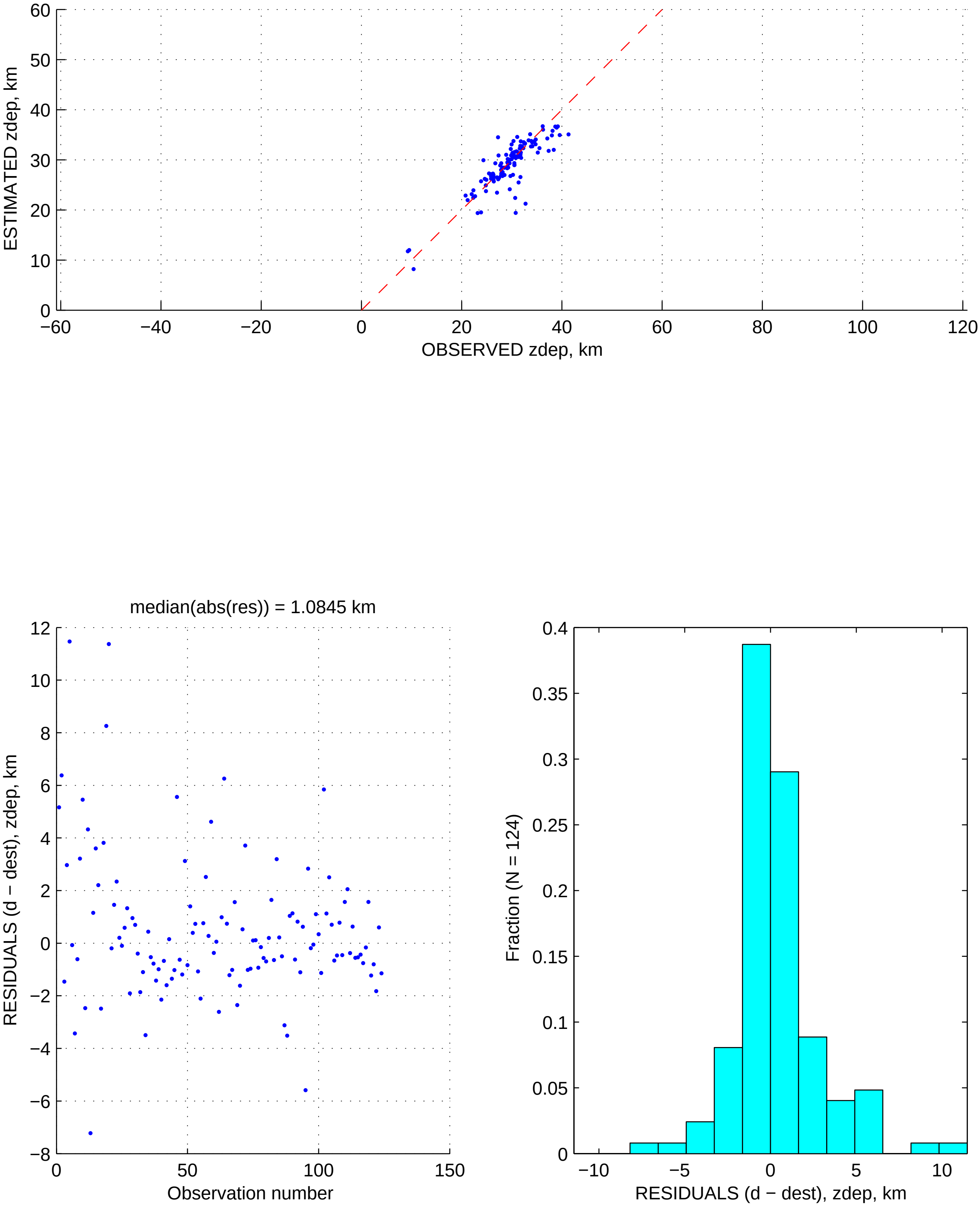

Figure 4: Comparison between observations and predictions for the 124 Moho depths.

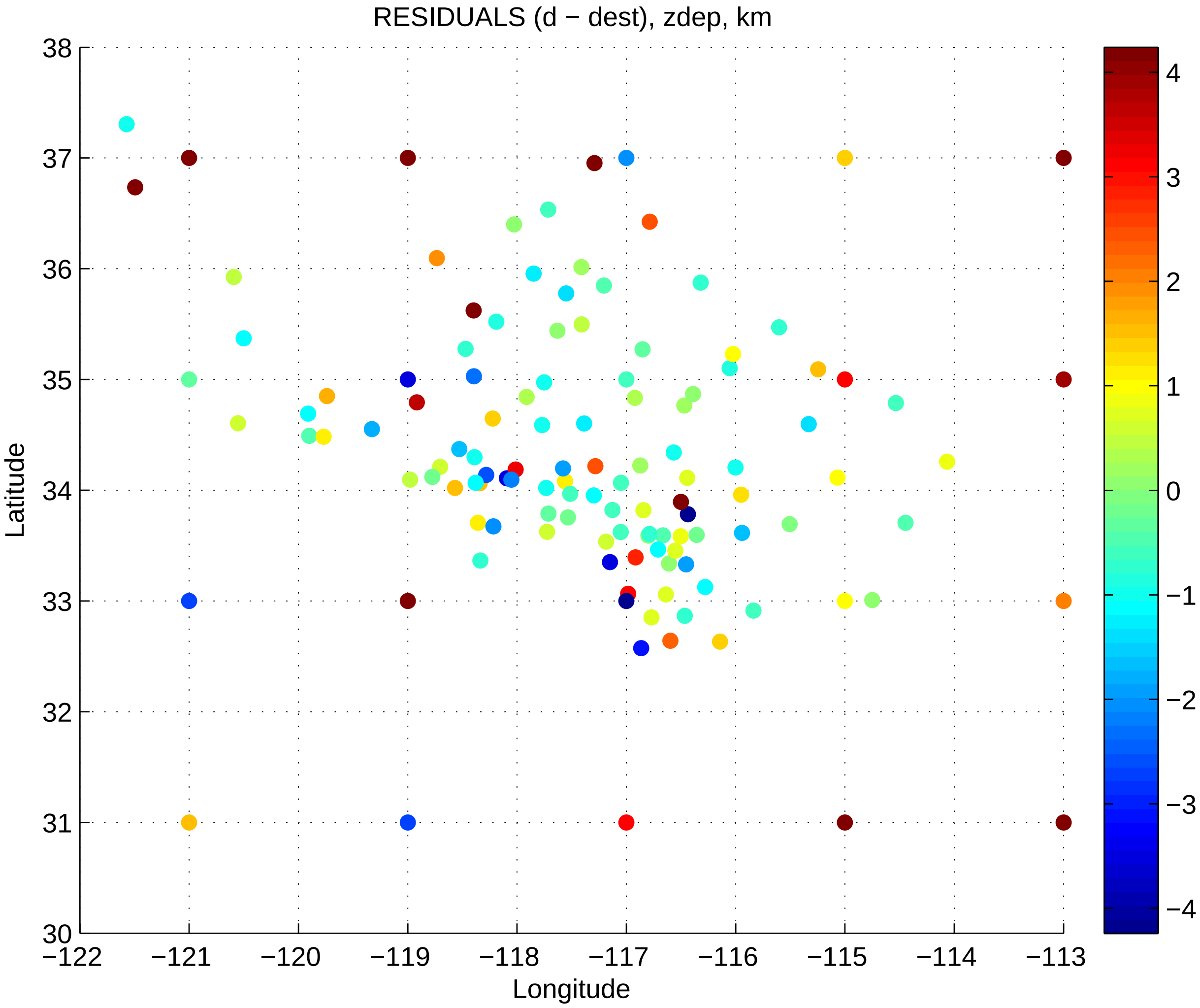

Figure 5: Spatial plot of residuals between observed and estimated Moho depths.

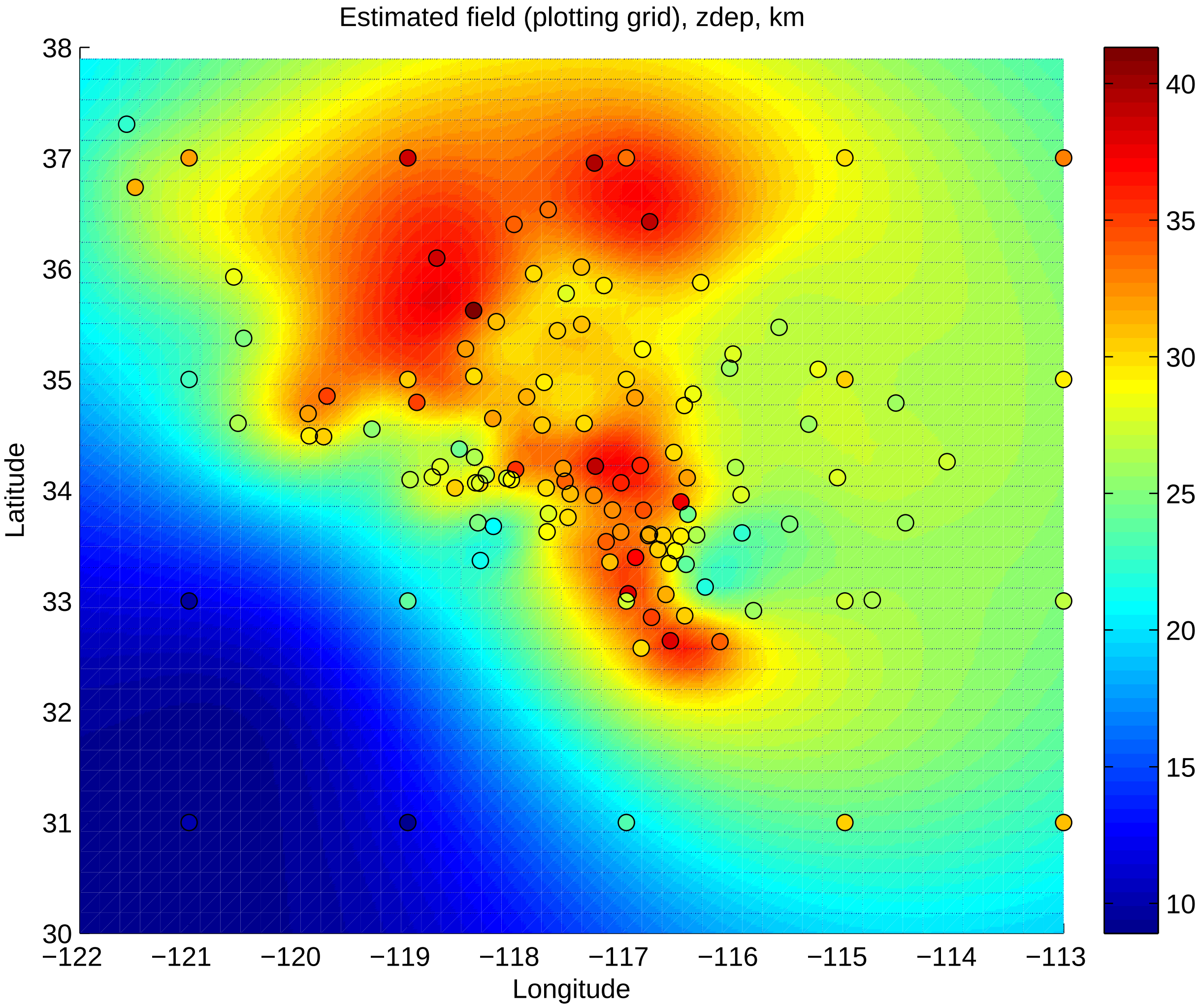

Figure 6: Estimated field, with observed values plotted as circles.

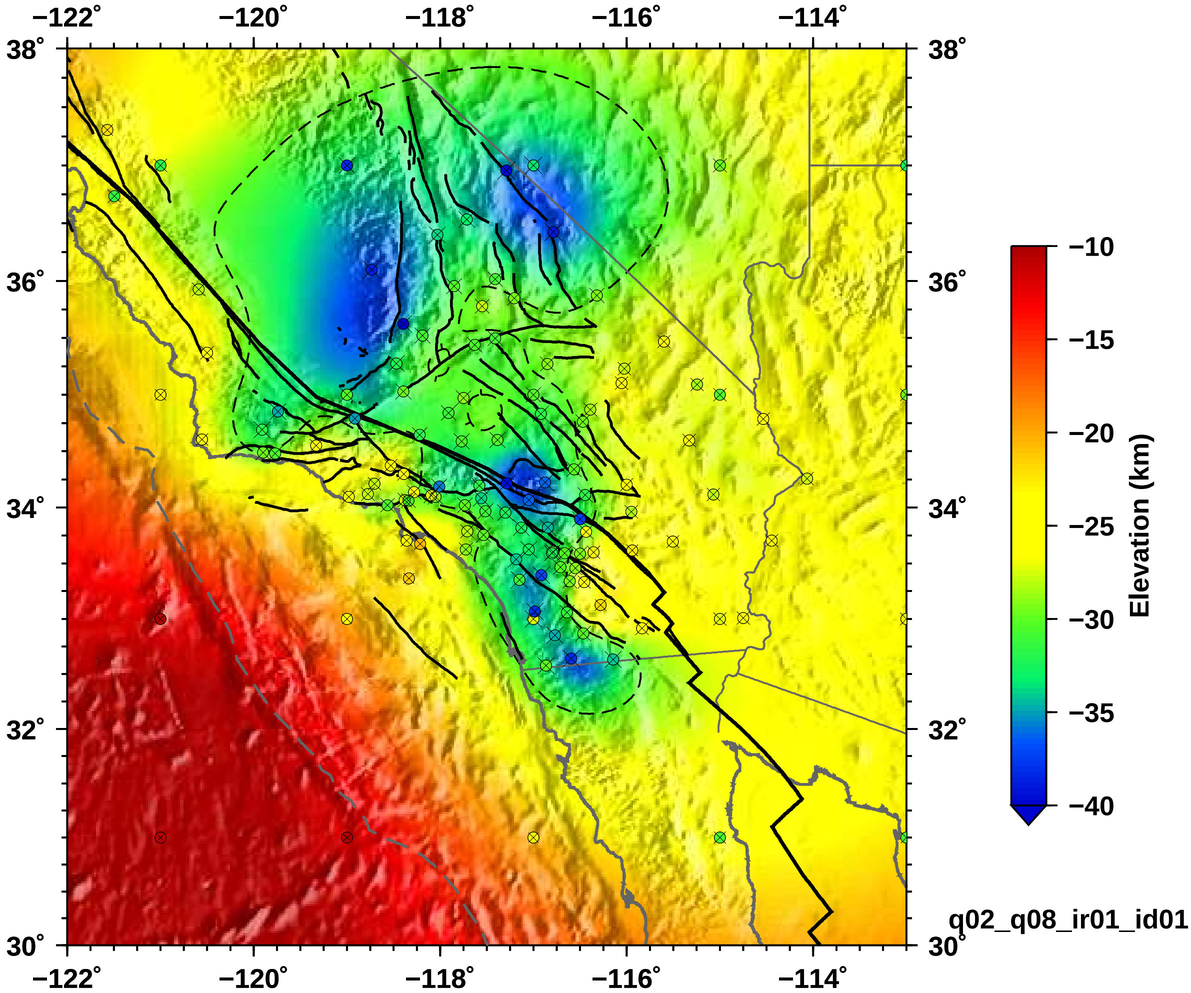

Figure 7: A fancier rendition of Figure 6. The dashed line shows the −30 km contour. Each filled circle is an observed values, with the 'x' marker proportional to the uncertainty.