Notes on using `surfacevel2strain.m`

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

Carl Tape

March 25, 2009

Suggested reading: *Tape et al.* (2009)

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

WARNING: These instructions are out-dated and in need of revision.

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.

I suggest first reading *Tape et al.* (2009). I also have a set of supplemental notes:

`surfacevel2strain/USER_INFO/gps_paper_supplement.pdf`

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 all possible spherical wavelet center-points for your designated region using `getsubgrids.f90`.
3. 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.
4. Save velocity field observations in a "standard" format.
5. Specify parameters in `surfacevel2strain.m` and compute the estimated continuous velocity field.
6. Plot output files in GMT.

I will demonstrate these steps for the Japan cGPS velocity field (*Sagiya et al.*, 2000; *Sagiya*, 2004), provided by Takeo Ito.

2 Running getsubgrids.f90

In getsubgrids.f90, the important parameters are: qmin, qmax, as well as the latitudes and longitudes describing the boundaries of a "square" lat-lon patch that covers the region of your observation points. I suggest always using qmin = 0, which controls the longest-scalelength basis functions; qmax controls the shortest-scalelength basis functions.

I have run this program for a sample region covering Japan, using qmax = 12. The output files are in the directory

fortran/data_getgrids/subgrids_japan_01/.

The output file num_gridpoints.dat shows the number of gridpoints selected for each grid order q:

```
0  8
1  16
2  29
3  50
4  80
5  168
6  433
7  1316
8  4522
9  16702
10  64145
11  251269
12  994563
```

The output file subgrid_bounds.dat shows the bounding box used for each grid:

```
-0.62304846E+02   0.18000000E+03  -0.90000000E+02   0.90000000E+02
 0.32847577E+02   0.18000000E+03  -0.65152423E+02   0.90000000E+02
 0.80423788E+02   0.18000000E+03  -0.17576212E+02   0.90000000E+02
 0.10421189E+03   0.17078811E+03   0.62118942E+01   0.69788106E+02
 0.11610595E+03   0.15889405E+03   0.18105947E+02   0.57894053E+02
 0.12205297E+03   0.15294703E+03   0.24052974E+02   0.51947026E+02
 0.12502649E+03   0.14997351E+03   0.27026487E+02   0.48973513E+02
 0.12651324E+03   0.14848676E+03   0.28513243E+02   0.47486757E+02
 0.12725662E+03   0.14774338E+03   0.29256622E+02   0.46743378E+02
 0.12762831E+03   0.14737169E+03   0.29628311E+02   0.46371689E+02
 0.12781416E+03   0.14718584E+03   0.29814155E+02   0.46185845E+02
 0.12790708E+03   0.14709292E+03   0.29907078E+02   0.46092922E+02
 0.12795354E+03   0.14704646E+03   0.29953539E+02   0.46046461E+02
```

The format is lonmin, lonmax, latmin, latmax, with row 1 for $q = 0$ and so on. The program generates the global gridpoints for a particular order q grid, and then only writes the gridpoints to file that are within the specified lat-lon box. By allowing the size of the box to vary with each q, you are able to sufficiently cover the edge regions of the region of interest with all scalelength basis functions, in particular, the longer-scalelength basis functions. Run plot_points.m to see the locations of the gridpoints.

3 Preparing the velocity field dataset (see 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`.

4 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:

```
 Type 1 for new inversion or 0 otherwise: 1
 Type 1 to use spherical wavelets, 2 for spherical splines: 1
 Type 1 to use the DIAGONAL covariance matrix for weighting (0 otherwise): 1
 Type the number of components of the v-field for the inversion (1,2,3) : 2
 Type 1 to plot with the mask (0 otherwise) : 1
 Type 1 to write output to files for GMT plotting (0 otherwise) : 1
 Type an index corresponding to a region (1=us, 2=cal, 3=socal, ..., 8=parkfield, 9=japan): 9
 Type an index corresponding to a v-field dataset (1=REASON, 2=CCMMv1.0, 3=ASIA, 4=japan, etc): 4
 about to read /home/carltape/gmt/gps_data/ASIA/japan/japan_takeo_ito_subset_3D.dat
 Type 1 to remove a uniform rotation, 0 otherwise: 0
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

 Type 1 to automatically pick min grid order, 0 otherwise: 0
 Type max allowable grid order, qmax (1 to 10): 8

          q          num
          0           8
          1          16
```

```
          2          29
          3          50
          4          80
          5         168
          6         433
          7        1316
          8        4522

 load the gridpoints in this region
 q = 0 : num = 8
 q = 1 : num = 16
 q = 2 : num = 29
 q = 3 : num = 50
 q = 4 : num = 80
 q = 5 : num = 168
 q = 6 : num = 433
 q = 7 : num = 1316
 q = 8 : num = 4522

 threshold the gridpoints

     q   num    id    i1    i2
          0           4           4           1           4
          1           6          10           5          10
          2          11          21          11          21
          3          16          37          22          37
          4          27          64          38          64
          5          53         117          65         117
          6         114         231         118         231
          7         291         522         232         522
          8         765        1287         523        1287
 Enter max q grid for secular field (5 or 6): 5

Thresholding GRIDPOINTS q = 0 to 8 (1287):
1287 wavelets / 6622 total with >= 3 stations inside their corresponding spatial supports

 q = 0 to 8, j = 1 to 1287 (1287)
 q = 0 to 5, j = 1 to 117 (117)
 q = 6 to 6, j = 118 to 231 (114)
 q = 7 to 7, j = 232 to 522 (291)
 q = 8 to 8, j = 523 to 1287 (765)

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

```
 got the regularization parameters: lam0 = NaN 1.43e+04 8.38e+03
>>
```

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 3 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:

```
 Type 1 for new inversion or 0 otherwise: 0

 computing the model vector...

Constructing the base design matrix for plotting...
 computing a mask for plotting...
 compute L : assume isotropic linear elasticity, Poisson solid, and surface condition
 Enter minimum value of omega for euler vectors (default = 0 rad/yr): 1e-7
 computing the max eigenvalue at each point...
 plotting velocities at different scales ...
 plotting strain maps...
    1.0000    0.0275
    2.0000    0.0204
    3.0000    0.9858
    4.0000    1.8736
    5.0000    2.0316
    6.0000    1.4753
 plotting components of the scalar quantity STRAIN...
    1.0000    0.0000
    2.0000    0.0000
    3.0000    0.0000
    5.0000    0.0000
    6.0000    0.0000
1  2.3156e-07
2  1.9341e-07
3  1.5831e-07
4  2.2591e-07
5  1.6457e-07
6  3.5766e-07
1  -1.5831e-07
2  3.5766e-07
3  1.6457e-07
4  3.1818e-07
 qmax = 5
 qmax = 6
 qmax = 7
 qmax = 8
 qmax = 5
 computing a mask for plotting...
 qmax = 6
 computing a mask for plotting...
```

```
 qmax = 7
 computing a mask for plotting...
 qmax = 8
 computing a mask for plotting...
 writing the multiscale estimated velocity field...
 writing the CUMULATIVE and INCREMENTAL multiscale estimated strain fields...
 compute L : assume isotropic linear elasticity, Poisson solid, and surface condition
 compute L : assume isotropic linear elasticity, Poisson solid, and surface condition
 compute L : assume isotropic linear elasticity, Poisson solid, and surface condition
 compute L : assume isotropic linear elasticity, Poisson solid, and surface condition
 compute L : assume isotropic linear elasticity, Poisson solid, and surface condition
 compute L : assume isotropic linear elasticity, Poisson solid, and surface condition
 compute L : assume isotropic linear elasticity, Poisson solid, and surface condition
 compute L : assume isotropic linear elasticity, Poisson solid, and surface condition
 writing the multiscale estimated RESIDUAL velocity field...
DONE with surfacevel2strain.m
>>
```

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 4–7.

References

Bird, P. (2003), An updated digital model of plate boundaries, *Geochem. Geophy. Geosyst.*, *4*, 1027, doi:10.1029/2001GC000252.

Sagiya, T. (2004), A decade of GEONET: 1994–2003—The continuous GPS observation in Japan and its impact on earthquake studies, *Earth Planets Space*, *56*, xxix–xli.

Sagiya, T., S. Miyazaki, and T. Tada (2000), Continuous GPS array and present-day crustal deformation of Japan, *Pure App. Geophys.*, *157*, 2303–2322.

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

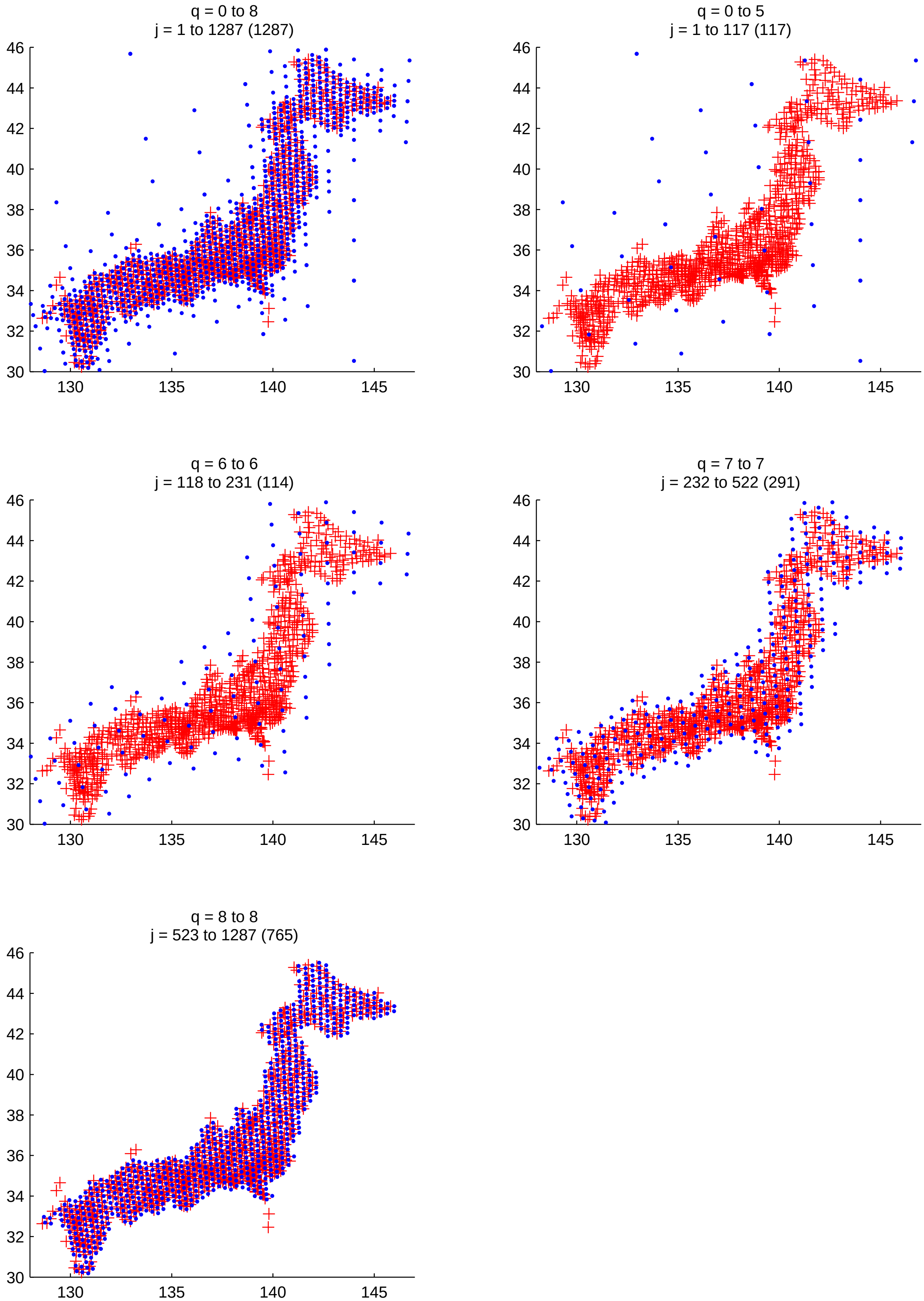

Figure 1: Basis function gridpoint centers for grids q=0-8, q=0-5, q=6, q=7, and q=8.

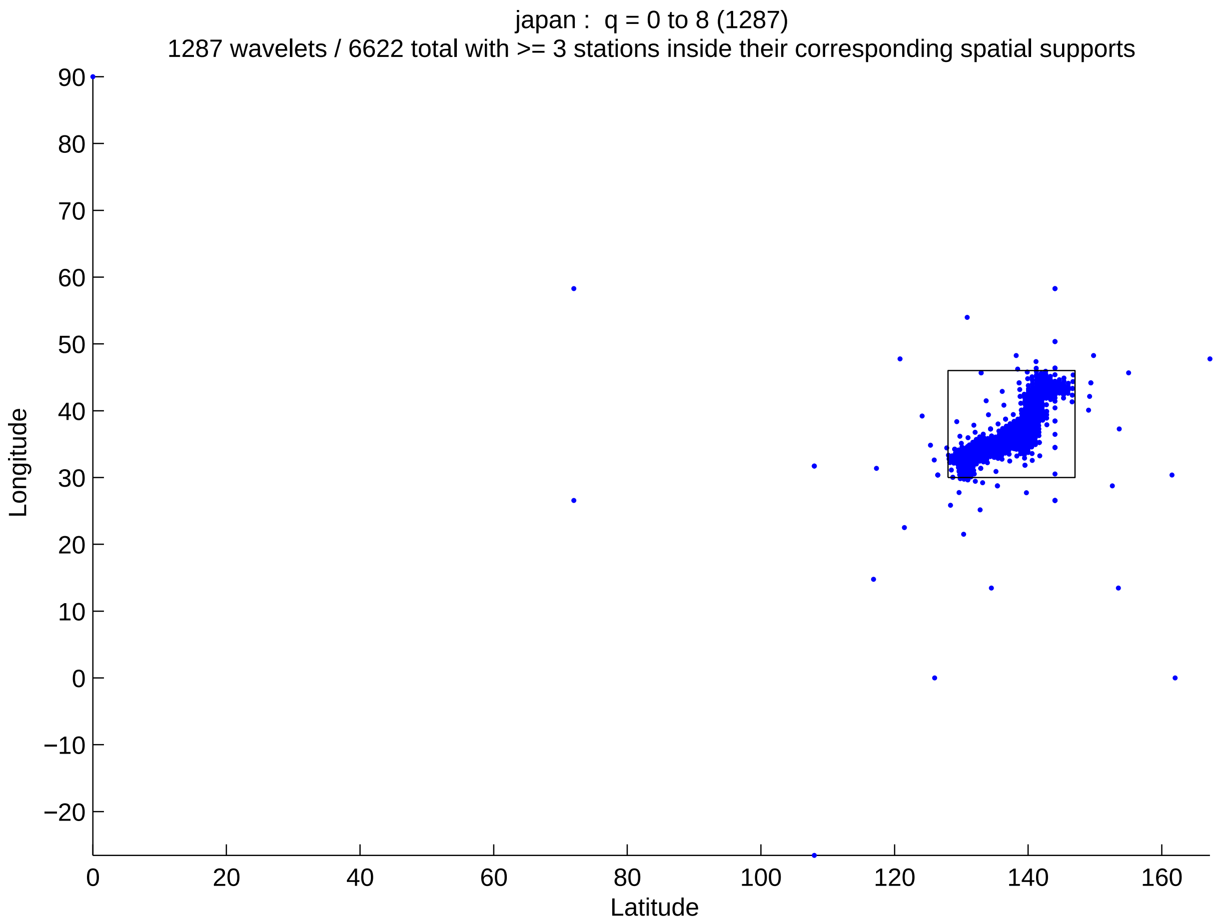

Figure 2: All basis function gridpoint centers for grids q=0-8. (See Figure 1).

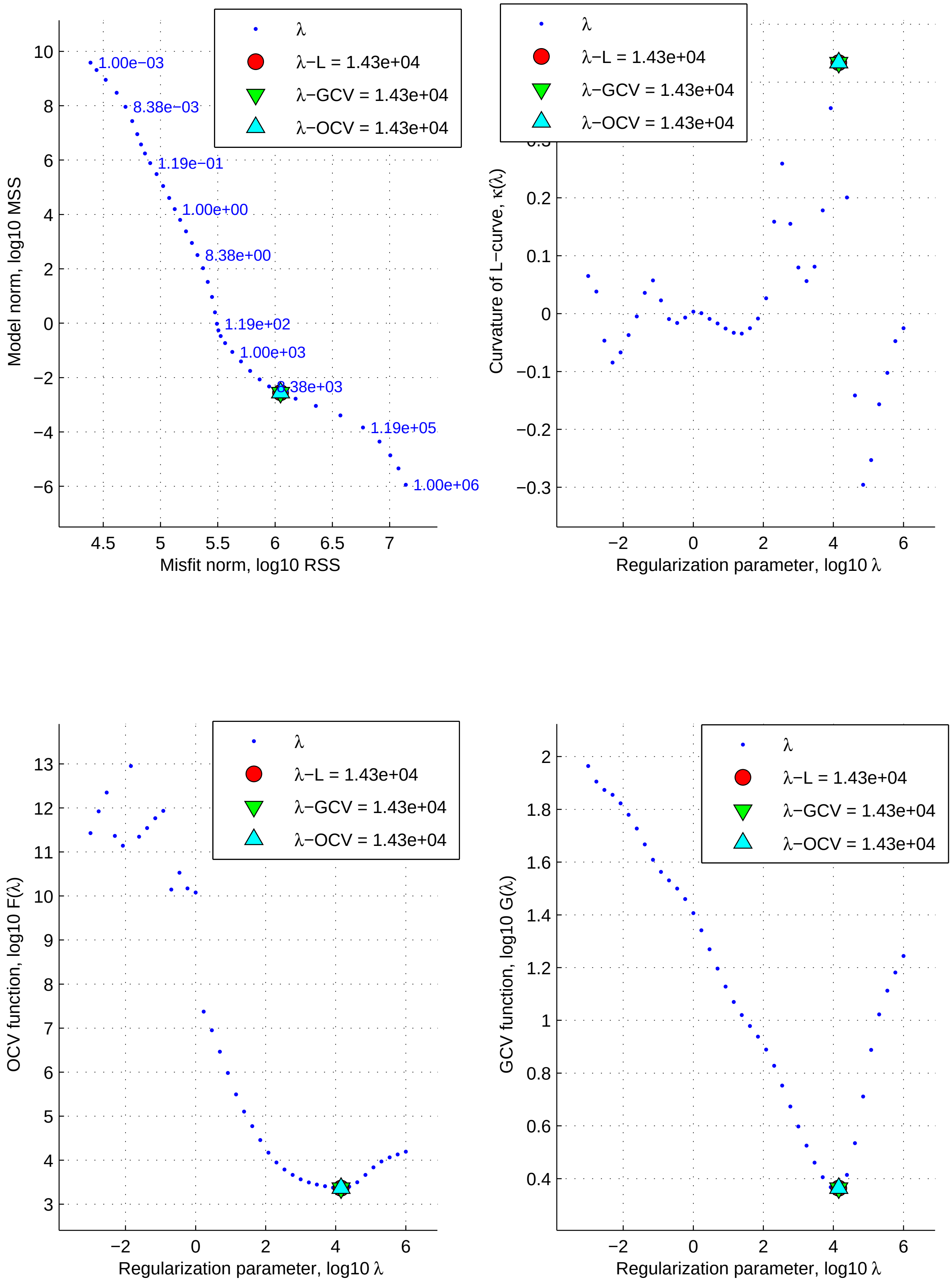

Figure 3: Damping parameter selection. Upper left shows the classical L-curve, as well as the λ value selected via three methods: (1) the maximum curvature of the L-curve; (2) the minimum of the ordinary cross-validation (OCV) function; (3) the minimum of the generalized cross-validation (GCV) function.

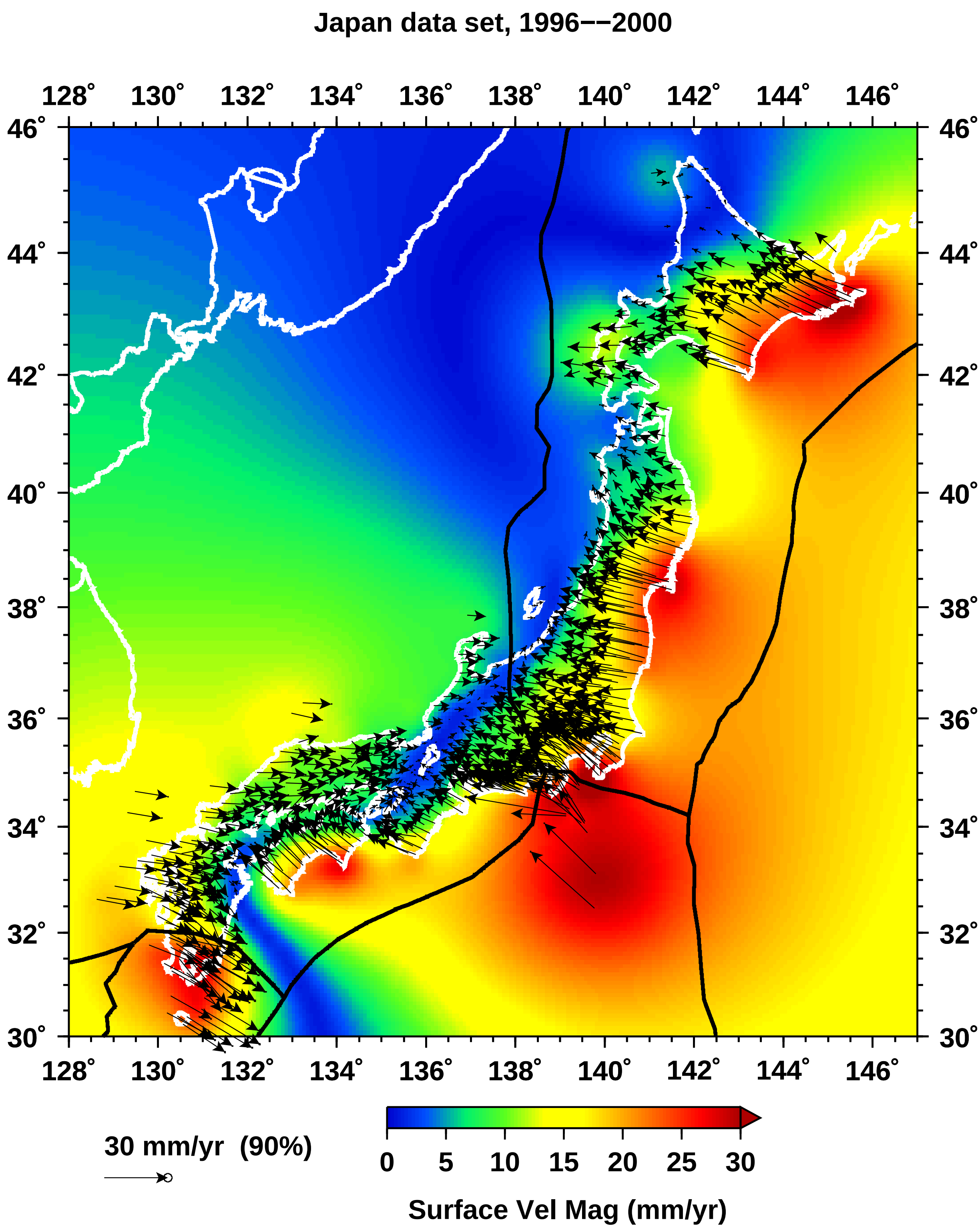

Figure 4: Estimated velocity field for Japan using spherical wavelets with scales q=0 to q=8. The cGPS observations (*Sagiya et al.*, 2000; *Sagiya*, 2004) were provided by Takeo Ito. Plate boundaries from *Bird* (2003).

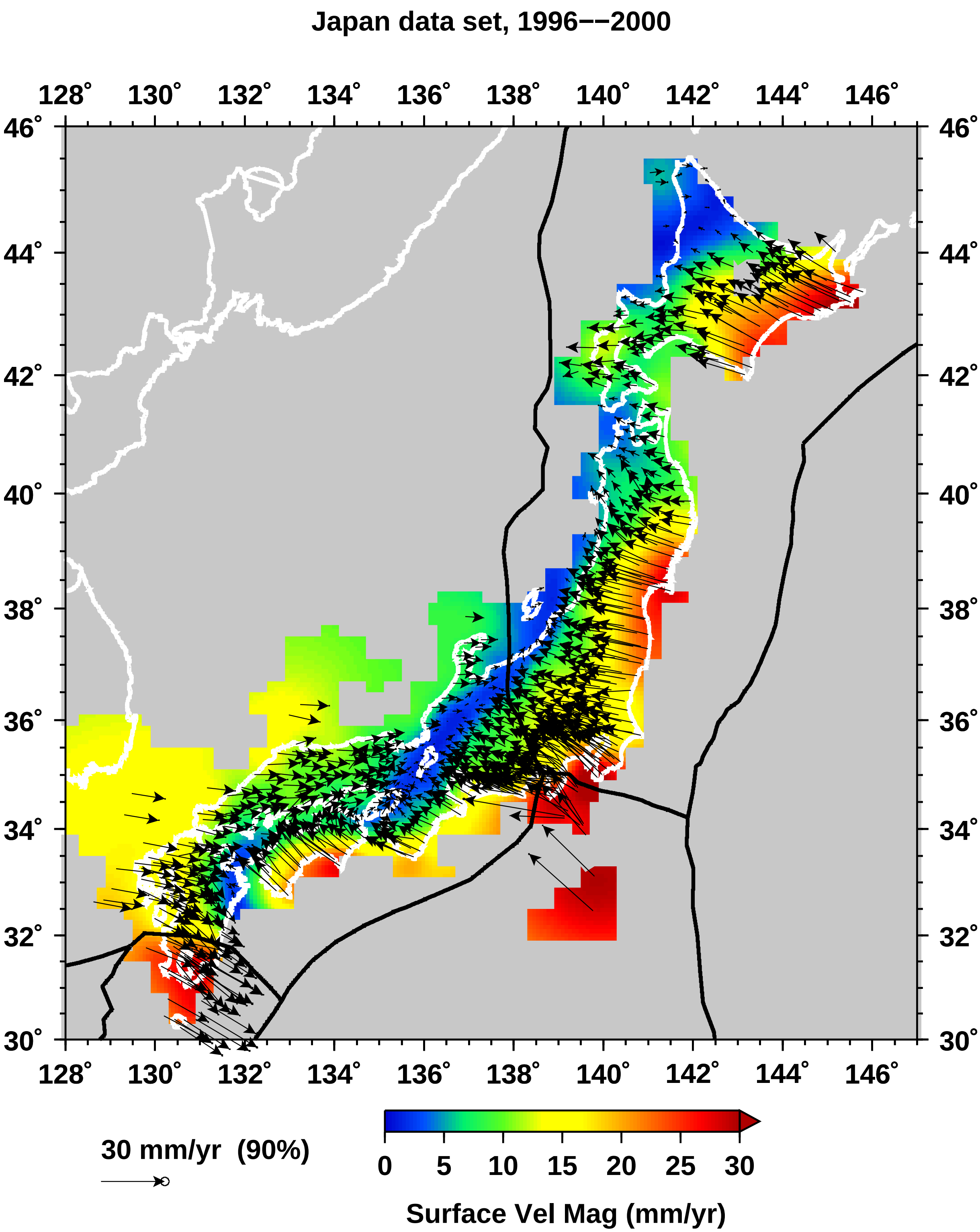

Figure 5: Same as Figure 4, but using the q=8 mask.

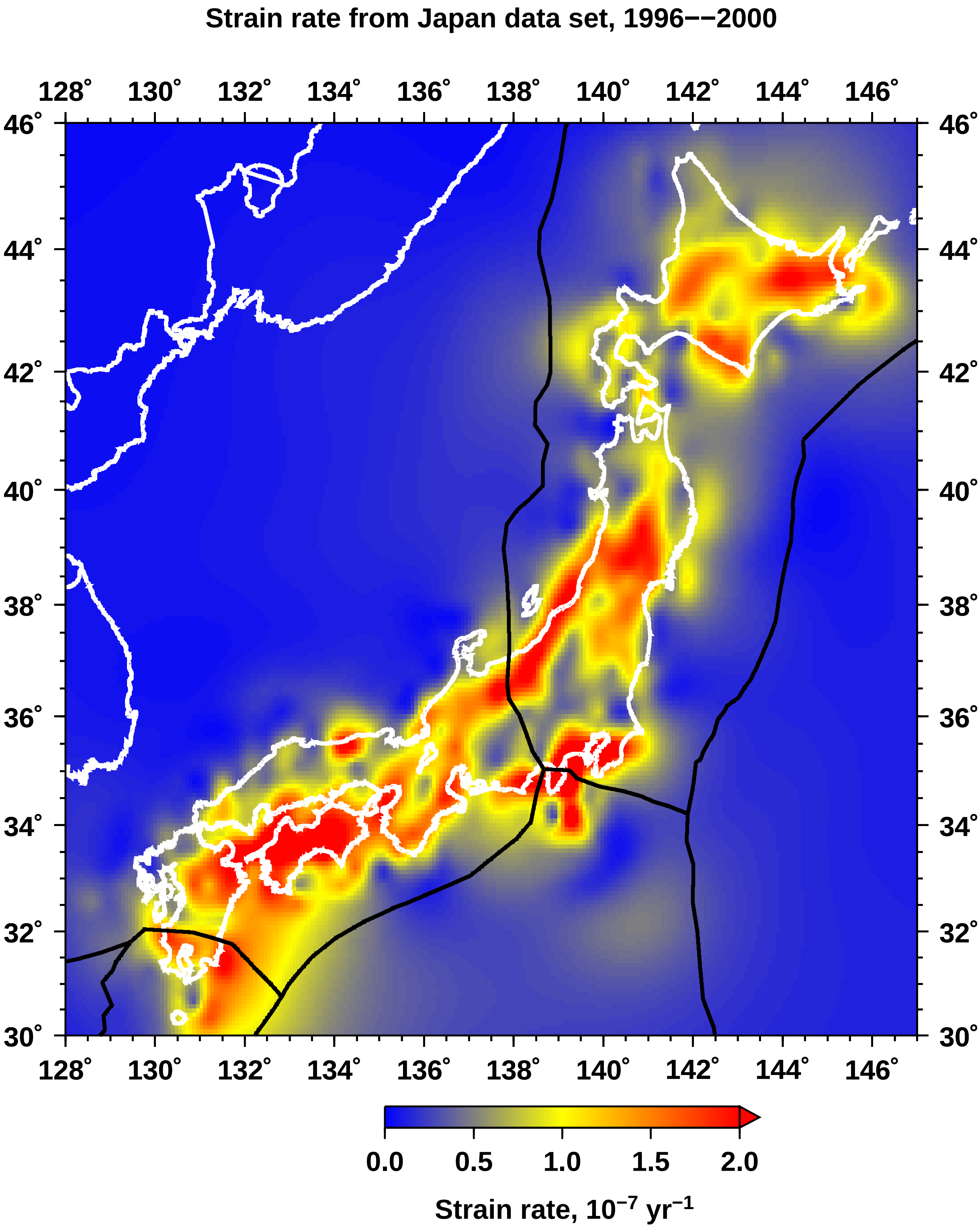

Figure 6: Strain-rate field for Japan, computed directly from the estimated horizontal velocity field in Figure 4. (We could have considered the vertical component as well, since it is available for the Japan dataset.) The strain rate is computed as the matrix norm of the strain-rate tensor at each point on the surface.

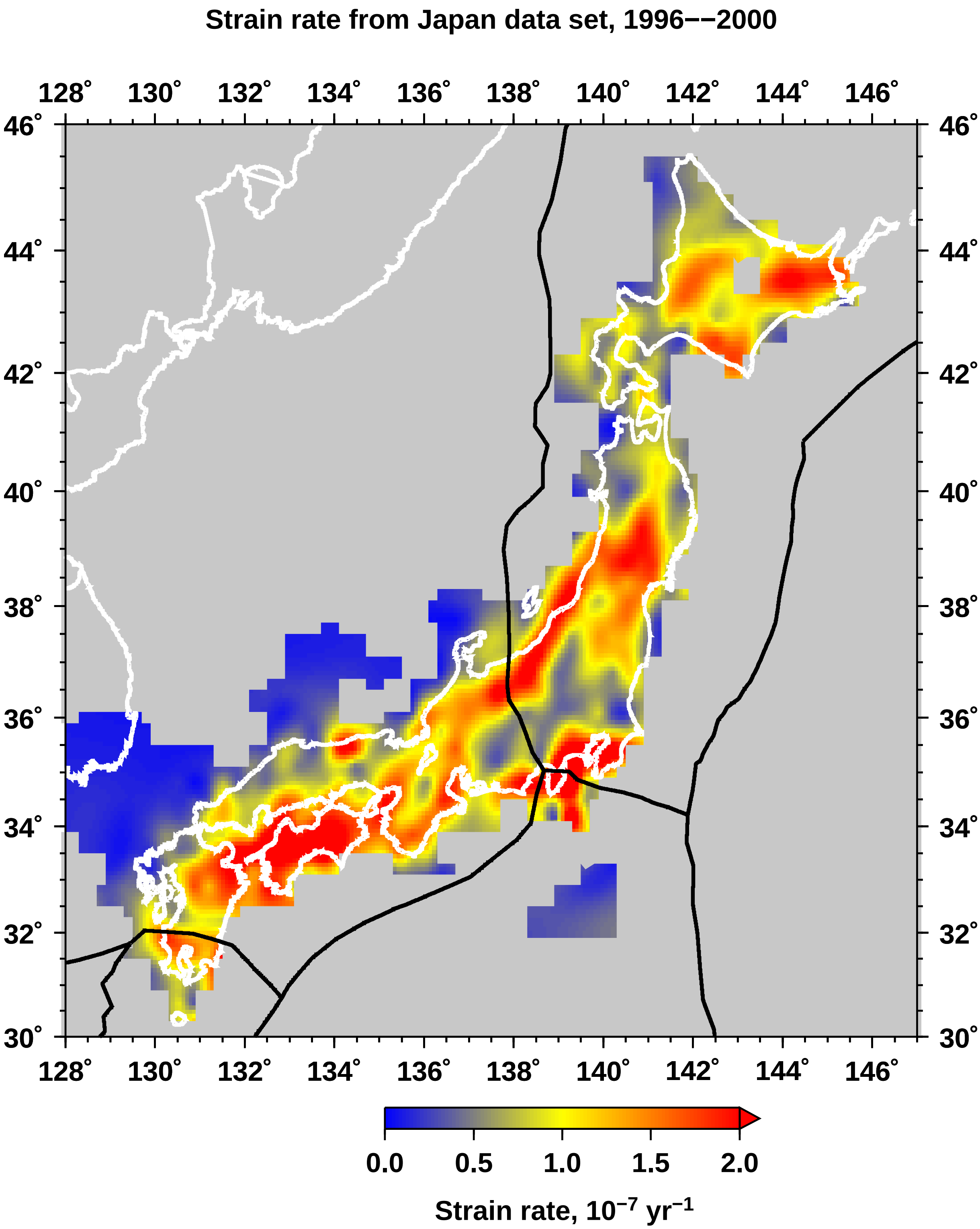

Figure 7: Same as Figure 6, but using the q=8 mask.