UNIT 7. Scaling the formal errors and computing velocities using *globk*

Ref. GLOBK manual
===

1. A STRATEGY TO SCALING FORMAL ERRORS

What follows is an example of how to scaling the solutions formal errors to obtain 1σ uncertainties for the GPS velocities. The example is based on the current work done at the seismo lab on the tectonics of the Adriatic region (central Mediterranean).

We process data from about 30 stations of the EUREF and ASI network to obtain regional deformation velocities. The data set is composed of continuous GPS observations from 3 different sources:

a) Solution (processed at the seismo lab) of EUREF and ASI networks
b) Global IGS solutions from SOPAC
c) Global EURA solutions from SOPAC

Before combining GPS observations and scale the formal errors, we

a) obtained daily solutions for the EUREF and ASI sites (we label these daily solutions ADRI) from 1999 to 2002; we decided to trim the data set by processing for every month 4 days (belonging to the same GPS week) of continuos GPS observations; in summary, for every year we have 13 weeks of observations for a total of 52 daily solutions; note that this is an approximation of the MIT strategy to combine data in monthly averages before computing deformation velocities;
b) identified outliers from the daily time series; because of the large number of observations available we decided to eliminate from our data set all the "weak" solutions; we did this by using the *rename* command in the *earthquake file*

The scaling strategy involves several steps.

<u>For every GPS week</u>, we compute the weights (given by the *globk* 'prefit chi**2') for each data set from an initial (unweighted) combination

Globk_adri_0995.prt	Globk_eura_0995.prt	Globk_igs_0995.prt
The prefit chi**2 for 1494 input parameters is 0.066	The prefit chi**2 for 1800 input parameters is 0.126	The prefit chi**2 for 5406 input parameters is 0.554

We use these weights to <u>combining daily ADRI, EURA and IGS solutions</u> (i.e., ADRI_99034.GLX and so on) and obtaining a weight for every daily combination.

globk_comb_99034.gdl	globk_comb_99034.prt
../glbf/h9902031200_igs1.glx 0.554 ../glbf/h9902031200_igs2.glx 0.554 ../glbf/h9902031200_igs3.glx 0.554 ../glbf/h9902031200_eura.glx 0.126 ../glbf/h9902031200_adri.glx 0.066	The prefit chi**2 for 2166 input parameters is 2.105

Now that we have a weight for every daily combination, we can compute <u>weekly combinations</u> (i.e., ADRI_99.0995.GLX and so on) and their weights

globk_comb_0995.gdl	Globk_comb_0995.prt
ADRI_99034.GLX 2.105 ADRI_99035.GLX 1.486 ADRI_99036.GLX 2.464 ADRI_99037.GLX 1.811	The prefit chi**2 for 2988 input parameters is 0.443

We re-run the combination with the old daily weights scaled by the prefit chi**2 (i.e., 0.443) until our chi**2 is of the order of 1 (in this case only a second run was necessary)

ADRI_99034.GLX 0.933 ADRI_99035.GLX 0.658 ADRI_99036.GLX 1.09 ADRI_99037.GLX 0.802	The prefit chi**2 for 2988 input parameters is 0.994

At this point we have 13 weekly observations (1 week for every month) for every year, for a total of 52 observations in the time span of our project. We remind that our approach was an approximation of the MIT strategy to combine data in monthly averages before computing deformation velocities
MIT found that monthly combinations essentially represent white noise. Monthly solutions provide sufficient observations to do good statistics on the residuals about the linear trends to properly weight the velocities.

2. COMPUTING THE DEFORMATION VELOCITIES

To compute the deformation velocities, we use *globk* and the command files (available in the templates/ directory) *globk_vel.cmd* and *glorg_vel.cmd*. As usual, we run *globk* from the directory gsoln/.

Plot of the deformation velocities can be obtained running *sh_plotvel* from the directory gplot/

In these notes we present two examples
- Deformation velocities in a global (ITRF00) reference frame
- Deformation velocities in a local (stable Europe) reference frame

<u>Deformation velocities in a global (ITRF00) reference frame</u>

==

FIRST STEP. Copy in gsoln/ and edit the command files. Below are the templates I used to compute a velocity field in the ITRF00 reference frame

<table>
<tr><th>globk_vel.cmd</th><th>glorg_vel.cmd</th></tr>
<tr><td><pre>
* Globk command file for EUREF velocities 1999-2001
* last modified by mb 051901
*
  eq_file ../templates/scec_eq.v2.1
  make_svs ../tables/sat1.apr
  com_file @.com
  srt_file @.srt
  srt_dir -1
  sol_file @.sol

apr site file(s)
  apr_file ../templates/itrf00.apr
</pre></td><td><pre>
*Glorg command file for velocity solution
*
apr site file(s)
ITRF96 for global stabilization
NNR frame
   apr_file ../templates/itrf00.apr

Define the stabilization frame
 source ../templates/stab_site.global

Set parameters to estimate in stabilization
  pos_org xrot yrot zrot xtran ytran ztran
  rate_org xrot yrot zrot xtran ytran ztran
</pre></td></tr>
</table>

```
  desc EUREF velocity 1999-2002

# Globk print output
x  minimal since using glorg
 prt_opt cmds psum vsum gdlf eras

# (1) Max chi**2, (2) Max prefit diff,
# (3) Max rotation
  max_chi 30 300 2000.0

* Apply the pole tide
  app_ptid ALL

* Allow the network to be loose
 apr_neu  all   100.0   100.0  100.0  1 1 1

* Estimate translation - .0005 m**2/yr = 15 mm/half-
# yr
 apr_tran .005 .005 .005 0 0 0
 mar_tran .0025 .0025 .0025 0 0 0

# GLORG commands
 org_cmd glorg_vel.cmd
org_opt CMDS PSUM VSUM GDLF ERAS
org_out adri_vel.org

# Set height ratios
x allow all selected stations through
  stab_min 1. 1.
  cnd_hgtv 1000 1000 2.0 1000.0

 # Iterations and editing
  stab_ite 4 0.5 4.

# Velocity equates (long list, shown first station
# only)
 equate yakz_gps ndot yakb_gps ndot
 equate yakz_gps edot yakb_gps edot
 equate yakz_gps udot yakb_gps udot
 equate yakz_gps npos yaka_gps npos
 equate yakz_gps epos yaka_gps epos
 equate yakz_gps upos yaka_gps upos
 equate yakz_gps ndot yaka_gps ndot
 equate yakz_gps edot yaka_gps edot
 equate yakz_gps udot yaka_gps udot
```

==

==

SECOND STEP. Copy (or link) your annual average solution in glbf/

==

==

THIRD STEP. Run *globk*

globk 6 adri_vel.prt adri_vel.log adri_GLX.gdl globk_vel.cmd

where fhe .gdl is the (unweighted) list of the weekly .GLX combinations

You use the output of *glorg* (.org file) to plot the velocity field

==

==

FOURTH STEP. Copy (or link) your *glorg* output (.org) in gplot/

==

==

FIFTH STEP. Run *sh_plotvel* - plot shown in FIGURE 1

Use the 'prefit chi' from the *glorg* output (.org) to scale the formal errors of your velocities

```
diablo% grep 'prefit chi' adri_vel.org
 The  prefit chi**2 for    2286 input parameters is     2.507
```

The runstrin for *sh_plotvel* is

sh_plotvel -f ../gsoln.global/adri_vel.org -D ../templates/ex_sites.eura -I ../templates/in_sites.eura -maprange europe -ps adri.global -factor 0.25 -u 2.5 -sitefont 2 -maxsigma 10 -color -arrowcolor1 255/0/0 -coast_res f -arrow_value 20

(see the Appendix for an explanation of *sh_plotvel* options)

==

Deformation velocities in a local (stable Europe) reference frame

The general approach is very similar, but NOTE the CHANGES (**bold**) in the definition of the reference frame in *glorg_vel.cmd* and *glorg_vel.cmd*. Minor changes in the runstring of *sh_plotvel* as well.

==

FIRST STEP. Copy in gsoln/ and edit the command files. Below are the templates I used to compute a velocity field in the stable Europe reference frame

globk_vel.cmd

```
* Globk command file for EUREF velocities 1999-2001
*  last modified by mb 051901
*
  eq_file ../templates/scec_eq.v2.1
  make_svs ../tables/sat1.apr
  com_file  @.com
  srt_file  @.srt
  srt_dir -1
  sol_file  @.sol

# apr site file(s)
  apr_file ../templates/itrf00_eura.apr

  desc EUREF velocity 1999-2002

# Globk print output
x  minimal since using glorg
 prt_opt cmds psum vsum gdlf eras

# (1) Max chi**2, (2) Max prefit diff,
# (3) Max rotation
  max_chi 30 300 2000.0

* Apply the pole tide
  app_ptid ALL

* Allow the network to be loose
 apr_neu  all   100.0   100.0  100.0  1 1 1

* Estimate translation - .0005 m**2/yr = 15 mm/half-
# yr
 apr_tran .005 .005 .005 0 0 0
 mar_tran .0025 .0025 .0025 0 0 0

# GLORG commands
 org_cmd glorg_vel.cmd
 org_opt CMDS PSUM VSUM GDLF ERAS
org_out adri_vel.eurasia.org
```

glorg_vel.cmd

```
* Glorg command file for velocity solution
*
# apr site file(s)
  apr_file ../templates/itrf00_eura.apr

#  Define the stabilization frame
  source ../templates/stab_site.eurasia

# Set parameters to estimate in stabilization
  pos_org  xrot yrot zrot xtran ytran ztran
  rate_org  xrot yrot zrot xtran ytran ztran

# Set height ratios
x allow all selected stations through
  stab_min 1. 1.
  cnd_hgtv 1000 1000 2.0 1000.0

 # Iterations and editing
  stab_ite 4 0.5 4.

# Velocity equates
 equate yakz_gps ndot yakb_gps ndot
 equate yakz_gps edot yakb_gps edot
 equate yakz_gps udot yakb_gps udot
 equate yakz_gps npos yaka_gps npos
 equate yakz_gps epos yaka_gps epos
 equate yakz_gps upos yaka_gps upos
 equate yakz_gps ndot yaka_gps ndot
 equate yakz_gps edot yaka_gps edot
 equate yakz_gps udot yaka_gps udot
```

==

==

SECOND, THIRD and FOURTH STEP are identical.

==

==

FIFTH STEP. Run *sh_plotvel* - plot shown in FIGURE 2

Use the 'prefit chi' from the *glorg* output (.org) to scale the formal errors of your velocities

```
diablo% grep 'prefit chi' adri_vel.eurasia.org
 The  prefit chi**2 for    2286 input parameters is     2.507
```

The runstrin for *sh_plotvel* is

```
sh_plotvel -f1 ../gsoln.eurasia/adri_vel.eurasia.org -u1 2.5 -D
../templates/ex_sites.eura -I ../templates/in_sites.eura -maprange europe -ps
adri.eurasia -sitefont 2 -maxsigma 10 -color -arrowcolor1 255/0/0 -coast_res f -
arrow_value 5
```

==

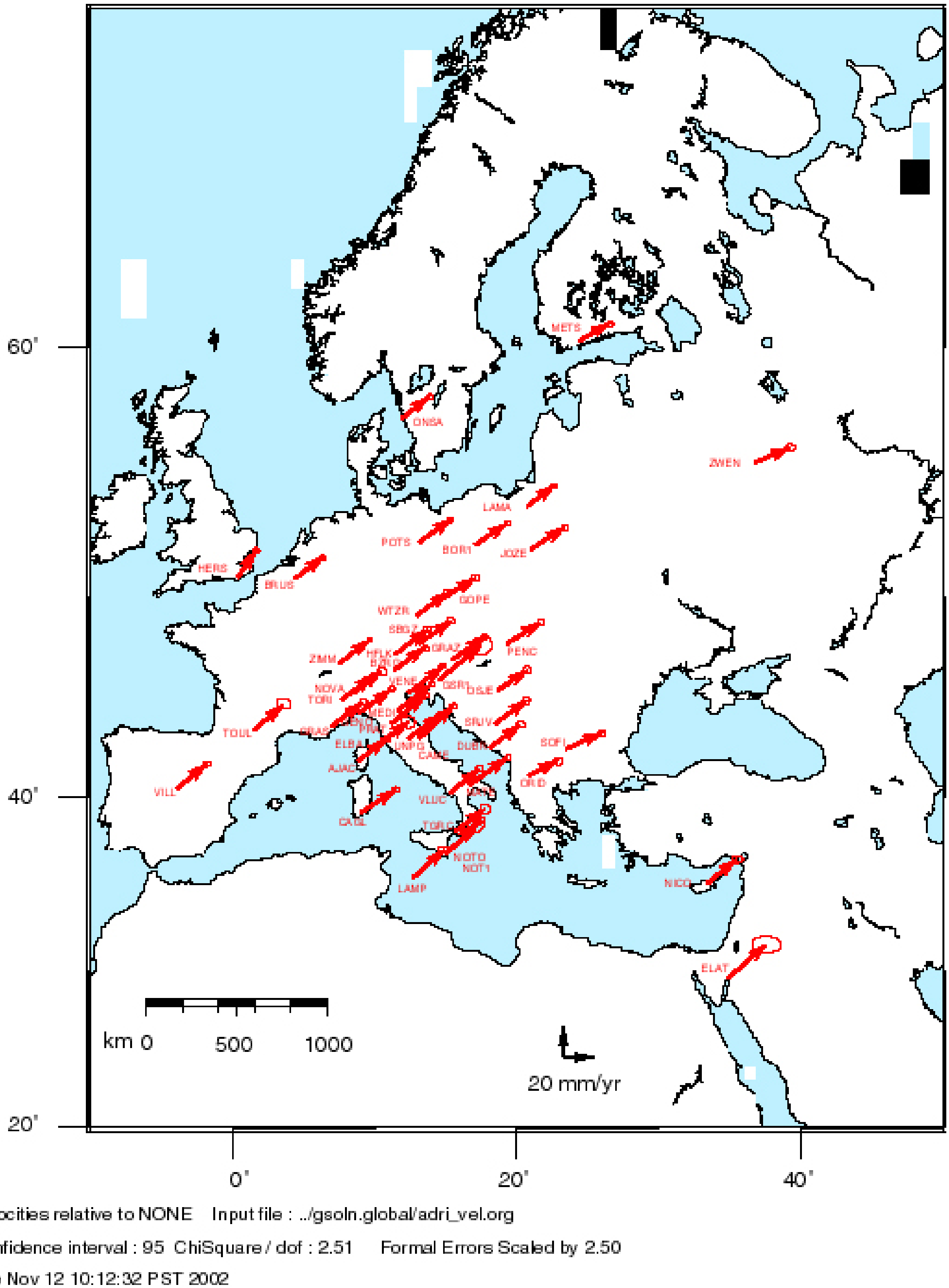
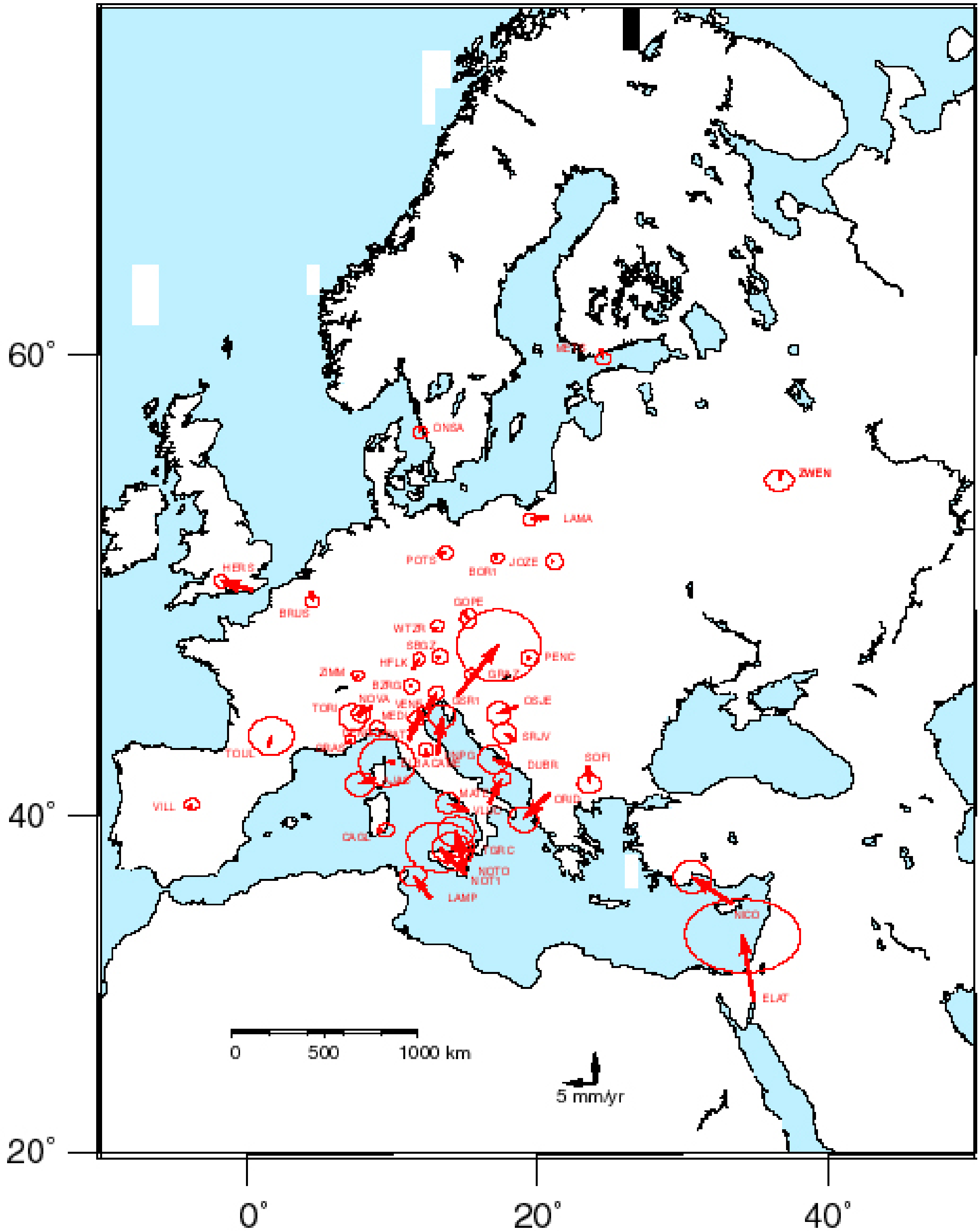


FIGURE 1 - GPS horizontal velocities and their 95% confidence ellipses in a global (ITRF00) reference frame



Velocities relative to NONE Input file : ../gsoln.eurasia/adri_vel.eurasia.org

Confidence interval : 95 ChiSquare / dof : 2.51 Formal Errors Scaled by 2.50

Tue Nov 12 10:32:34 PST 2002

FIGURE 2 - GPS horizontal velocities and their 95% confidence ellipses in a stable Europe reference frame

APPENDIX - Arguments for sh_plotvel: -help

Note: this a quite reduced version of the help including only the commands used in the examples

```
Script to plot station velocities on a map, using a GLOBK prt file as input.
Options allow selection of stations, multiple velocities, and enhanced maps.

Basic usage : sh_plotvel -f <file> -s <site>
    -file              : Input GLOBK prt file or GETREL file
    -site              : 4-char station id of reference site for velocities
                         If site omitted, plot absolute velocities

Additional options
-maprange type : Calls sh_map_elements with the keyword <type> to select a region.
                 Among those allowed are <world> <europe> <turkey> <cal>
                 See sh_map_elements for a complete list or to add your own
-ps file       : postscript save file. Default is FILE.SITE.ps

===================
 Extended features
===================
-D[ELETE]  file : Remove sites listed in the file from tables. Case insensitive
-I[NCLUDE] file : List of sites to be included.
-factor  value      : Scale the internally defined arrow value. by this factor
-maxsigma  value    : Maximum sigmas/error ellipses to be plotted
-arrow_value  value : define the size of the velocity scale arrow in mm. Default =
20mm/yr
-------
<< Multiple velocity solutions >>
-f[ile] file        : Input GLOBK org-file
-f1
-u[ncscale] value  : scale uncertainties with value. Default is 1.
-u1                  if issued automatic scaling with chisqr of the experiment.
-arrowcolor1       : rgb numbers. Default is black 0/0/0. -color default is black
255/0/0.
-------
<< Enhanced mapping options >>
-sitefont          : Size of site names. Default is 07.
-coast_res  value  : Selects the resolution of coastline to be plotted (f)ull,
(h)igh, (i)intermediate, (l)ow, or (c)crude. Default is: i
```