

# var\_res\_py : Python-based variable/ quad-tree resampler

Piyush Agram  
Romain Jolivet  
Seismological Laboratory  
Caltech

February 28, 2012

**var\_res\_py** is a python implementation of the variable sampler originally written in MATLAB by Mark Simons and Yuri Fialko. Few new improvements include an approximate covariance matrix for the resampled data.

## 1 Usage

**premodel.py** represents the driver script to the package and can be used with the following options. All the inputs included in `[.]` are optional.

```
> premodel.py data.unw los.unw outfile thresh [method] [nseg]  
[scl] [plot] [covar] [maxsize]
```

The inputs to the program are

**data.unw** The geo-coded unwrapped file in RMG format produced by ROI-PAC. The corresponding resource file (`data.unw.rsc`) should also be located in the same directory.

**los.unw** The geometry file in two possible formats - (Look Angle, Azimuth) in RMG format or (E,N,U) look vectors in inter-leaved format. More formats can be easily included by modifying the **load\_S** module in

**loaddata.py**. Optionally, if one is not interested in the geometry outputs this can be set to **N**. If the detailed geometry is not available and one is interested in using an estimate of the LOS vector, this input can be set to **D**. If using azimuth offsets, set this flag to **A**.

**outfile** This represents the prefix of the output files. The resampled data will be written to **outfile.txt** and the estimate of the covariance matrix to a binary float-32 file called **outfile.cov**.

**thresh** This number represents the threshold for the standard deviation of pixels in each box. If the standard deviation of pixels in a box does not exceed the threshold, it is no longer decomposed into smaller data units. Threshold represents the standard deviation of the observations in **cm**.

**method** This is the method used to estimate the standard deviation in a box. **CURV** takes into account the local curvature and slope in each box before estimating the standard deviation and **VAR** directly computes the standard deviation. If not set, **CURV** is assumed.

**nseg** The code has been designed to ideally work on images that are almost square. If multiple frames are stitched together for a combined analysis, **nseg** should be set to the number of frames. It represents the number of vertical frames, the data will be segmented into before processing. If not set, assumed to be 1.

**scl** Decides the scaling of the input data set. Can be **Y** for regular unwrapped phase data, **N** if the data is already in cm and **M** if the input data is in meters. Note that the threshold will be applied after scaling the input data. If not set, assumed to be **Y**.

**plot** Plotflag to be set to either Y or N. Use pyplot utilities to display the discrepancy between the original and resampled data set. If the covariance estimation flag is also set, the fit for covariance estimation and the estimate of covariance matrix are also displayed. If not set, assumed to be **Y**.

**covar** Covarflag to be set to either Y or N. An estimate of the covariance matrix is computed by randomly sampling data and using a `scipy curve_fit`

module. The atmospheric covariance function is assumed to be

$$C_{ij} = \sigma^2 \cdot \exp(-d_{ij}/\lambda) \quad (1)$$

where  $C_{ij}$  represents the covariance of  $i^{th}$  and  $j^{th}$  pixels and  $d_{ij}$  represents the distance between them. If not set assumed to be N.

**maxsize** maxsize sets the maximum size of the box. This way in areas where phase does not vary a lot, one can have a regular sampling and a fine resolution where the curvature sampling tells you to.

## 2 Outputs

### 2.1 outfile.txt

The primary output of the package is the resampled data points in **outfile.txt**. The columns of this ASCII file includes the following header:

```
Number xind yind east north data err wgt Elos Nlos Ulos
*****
```

**Number** Index number of the sample.

**xind** Column index of the sample in the image.

**yind** Row index of the sample in the image (Flipped for increasing lat values).

**east** East co-ordinate corresponding to **xind**.

**north** North co-ordinate corresponding to **yind**.

**data** Data corresponding to the sample in cm.

**err** Std. deviation of data in the box corresponding to the sample.

**wgt** Number of pixels included in the box corresponding to the sample.

**Elos** Component of LOS unit vector in East direction. Set to zero if **los.unw** == N

**Nlos** Component of LOS unit vector in North direction.

**Ulos** Component of LOS unit vector in Up direction.

## 2.2 outfile.rsp

This file lists the coordinates of the upper-left and down-right corners of the box containing the pixels used to compute each value given in the output.txt file. This file is needed to run SameResampAs.py. The columns of this ASCII file includes the following header:

```
xind yind UpperLeft-x,y DownRight-x,y  
*****
```

**xind** Column index of the sample in the image.

**yind** Row index of the sample in the image (Flipped for increasing lat values).

**Upper-left x** Upper-left corner column index.

**Upper-left y** Upper-left corner row index.

**Down-right x** Down-right corner column index.

**Down-right y** Down-right corner row index.

## 3 Hard-coded parameters

There are a couple of hard coded parameters in the code.

**minreslevel** Represents the largest box of data points to considered. Currently set to 2, implying the largest box is half the size of the original image. The default smallest size is  $2 \times 2$  pixels.

**nfrac** Currently set to 10% in **covaraps.py**. Sets the number of random data pairs as a function of non-zero data points in the original image to use for covariance parameter estimation.

**flipvert** Currently set to **N**. Set it to **Y** if you desire the image to be vertically flipped (For compatibility with old matlab code).

**scale** Distance scaling for better covariance curve fitting. Currently set to 0.001.

## 4 Additional Routines

We added a few routines that you might want to check, in case you are looking for something in particular.

---

**SameResampAs.py** is a routine that has been created to resample numerous interferograms in a similar way. The user re-samples an interferogram using `prepmodel.py`, which creates a `.rsp` file. This file lists the coordinates of the pixels averaged during the re-sampling process and is used as an input to `SameResampAs.py` to resample any interferogram that has the same geometry.

```
> SameResampAs.py file.unw file.rsp file.txt output plot(y/n)
```

The inputs to the program are

**file.unw** Unw file you want to re-sample. It has to be in the exact same geometry as the file used to generate the `rsp` file (no warning are displayed if it is not the case).

**file.rsp** Rsp file given by `prepmodel.py`.

**file.txt** Txt file given by `prepmodel.py`.

**output** Prefix of the output file.

**plot(y/n)** Plots a few things Yes or No.

---

**CheckResamp.m** is a matlab routine that provides the same plot as `prepmodel.py`, but the interpolation is different. You might want to modify it and check your resampling with matlab (more convenient).