# The CalcDeformation program

The CalcDeformation.py program is used to evaluate the NZGD2000 deformation model. For a given latitude, longitude and date it calculates the east, north and up correction (in metres) that is applied to an NZGD2000 reference coordinate to obtain an epoch coordinate at that date (equivalent to an ITRF96 coordinate at that date).

The program can be used to:

- calculate the deformation at a specified time and place
- apply the deformation model to longitude/latitude/ellipsoidal height coordinates
- calculate and compare the deformation for different versions of the model
- calculate the coordinate change corresponding to reverse patches
- list the components of a model
- check that a model is correctly formatted and self consistent

The program is a python command line program. It is run from the command line using the syntax

python CalcDeformation.py [options] [arguments]

The options select how the program will operate and the arguments are generally the input and output. Options start with a hyphen character. Most have a short and a long form. For example the date option can be entered as

*-d 2013-09-01*

or

*--date=2013-09-01*

The arguments are usually the names of the input and output file, but may be interpreted differently depending on the options selected. This is detailed further below.

### *Options for selecting the model*

| | |
|---|---|
| -m *dir*<br>--model-dir=*dir* | Select the directory in which the deformation model is located. The directory is the base directory of the model (which should be called 'model'). If this is not explicitly defined then it will use the environment variable 'NZGD2000_DEF_MODEL' to define the location, and if that is not defined it will try a sibling directory to the directory in which the program is located, and a directory just called model. |
| -v *version*<br>--version=*version* | Version of model to calculate. The default is the latest version available in the model. Each version is defined by an 8 digit value, for example 20000101, 20130801. The version parameter can also represent the difference between two versions using "20130801-20000101", or the negative of a version "-20130801". The negative may be used to transform ITRF96 epoch coordinates to NZGD2000 reference coordinates. |
| -r *version*<br>--base-version=*version* | An alternative way of specifying a version for calculating the difference between versions |
| -d *yyyy-mm-dd*<br>--date=*yyyy-mm-dd* | Specifies the date at which the model is to be evaluated. The default is the current date. When the program is using an input file this can also be entered as ":colname" where colname is the name of a column in the input file from which the date will be |

| | |
|---|---|
| | read. The dates can be entered as yyyy-mm-dd or yyyymmdd. |
| -b *yyyy-mm-dd*<br>--base-date=*yyyy-mm-dd* | This option is used to calculate the difference in deformation between the base date defined with this option and the date specified in with the --date option. The default is to caculate relative to the NZGD2000 reference coordinate. |
| -o *submodel*<br>--only=*submodel* | This is used to select the submodels of the model to use. The default is to use the complete model. This option can be used to evaluate one or more of the submodels for research use. Submodels are specified by their directory name. For example --only=ndm will just use the national velocity submodel. Patch directories are specified by the name after the "patch_" prefix, and only need match the beginning of the patch name. For example --only="c1" will match the patch_c1_20100904 patch. --only="c" will match all the Christchurch patches (c1, c2, c3, c4). Components can be combined with +, (eg "c1+c2"). Also the submodels selection can be prefixed with "-" to match everything except that component. For example --only=-ndm will use everything except the national velocity model. |

### *Options for selecting what to calculate*

| | |
|---|---|
| -l<br>--list | List the components in the model (as selected by the options above). |
| -k<br>--check | Check that the model is correctly formatted but do not calculate anything. This will load all data used by all versions of the model defined in the model.csv (including loading all grid and TIN files) |
| -a<br>--apply | This option is used to update the coordinates from the input file by adding the deformation model (typically converting reference coordinates to epoch coordinates). |
| -s<br>--subtract | This option is used to update the coordinates from the input file by subtracting the deformation model (typically converting epoch coordinates to reference coordinates) |
| -e *elements*<br>--elements=*elements* | Specifies which deformation elements to calculate. The required components are entered as a colon separated list and can include any of de, dn, du, eh, and ev. The default is de:dn:du |
| -p<br>--patch | Calculate the reverse patch corrections for a specified version. These corrections are the coordinate changes that should be applied when the new deformation model is implemented to match the reverse patches in the model (Generally this would be used with the --version option to calculated the reverse patch changes between two revisions of the model). |

### *Options for defining the input and output*

The main use of the CalcDeformation.py program is to calculate deformation and update coordinates. The coordinates at which the model is to be evaluated are generally provided in an input file, which can be either a space delimited or comma separated file. The output file is in the same format as the input file.

The input and output file names are specified on the command line as arguments. These can be "-" to select the standard input and standard output streams. The program can also use a calculated grid of points instead of an input file by using the --grid option, or a single point with the --atpoint option.

The first line of the input file must contain the names of the columns of data - longitude, latitude, and optionally ellipsoidal height and calculation date. By default the column names are "lon", "lat", and "hgt" (the date comes from the --date parameter, but can be specified as a column name).

The output file is structured in the same way as the input file. It will include extra columns for the calculated deformation elements unless the --update option is selected, in which case the longitude, latitude, and height will be replaced with the updated values.
An example of a comma separated input file could be

```
name,lon,lat,hgt
P1,174.8343856,-36.6028445,132.711
P2,174.9328604,-41.19644089,39.585
P3,168.2920876,-46.58506458,124.663
```

The options relating to the file contents and format are

| | |
|---|---|
| -f *format*<br>--format=*format* | Format of the input and output files - *format* can be one of "csv" (excel compatible csv), "tab" (tab delimited), or "whitespace" (whitespace delimited). The default is csv format. |
| -c *colnames*<br>--columns=*colnames* | Specifies the names of the longitude, latitude, and optionally ellipsoidal height and date columns. The column names are listed separated by colon characters. For example<br>--columns=pt_longitude:pt_latitude::obs_date<br>The date column can also be specified with the --date option.<br>The default column names are "lon", "lat", and "hgt". |
| -g gridspec<br>--grid=gridspec | Specifies that a grid of points will be calculated instead of calculating using an input file. The grid specification consists of the values of the minimum longitude, minimum latitude, maximum longitude, maximum latitude, and number of longitude values (columns) and number of latitude values (rows). For example<br>--grid=170.0:-41.0:175.0:-38.0:26:16 |
| -x<br>--atpoint | Calculate the deformation at a single point. The arguments will be a longitude, latitude, and ellipoidal height in place of an input and output file name. |

### *Miscellaneous options*

| | |
|---|---|
| -q<br>--quiet | Reduces the outputs generated by the program |
| --cache=*action* | Manages the cached data. If the pytables python module is installed, then by default the program will build binary cached grid and TIN data when it loads the corresponding files for the first time. The --cache option specifies how this will be managed. The actions available are:<br>• ignore: the cache is not used and is left unchanged<br>• clear: the cache is cleared when the program starts and not used. |

| | |
|---|---|
| | • reset: the cache is cleared when the program is started and loaded with any data required as the program runs<br>• use: the data is used and added to as data is accessed when the program runs. This is the default option. |
| --logging | If present then the program generates additional logging output that may be useful for debugging |

## *Program requirements*

The CalcDeformation program requires python version 2 (tested on version 2.7) with the numpy module installed (tested with version 1.6.2). It also benefits from having the pytables module installed for caching grid models (tested with pytables version 2.3.1 and HDF5 version 1.8.4).

## *Examples of using the CalcDeformation program*

To check that the model is valid

```
python CalcDeformation.py --check
```

To list the contents of the model

```
python CalcDeformation.py -l
```

To calculate the deformation east, north and up at a set of points in the example file above which has filename testin.csv

```
python CalcDeformation.py testin.csv testout.csv
```

To calculate the deformation east, north, and up that applied at date 10 March 2012 at the points in the file

```
python CalcDeformation.py --date=2012-03-10 testin.csv testout.csv
```

To calculate the NZGD2000 epoch coordinates (ie ITRF96 coordinates) on 10 March 2012 from NZGD2000 reference coordinates in testin.csv (Note: it is recommended to use the ITRF_NZGD2000.py script for this calculation)

```
python CalcDeformation.py --date=2012-03-10 --apply testin.csv testout.csv
```

To calculate the NZGD2000 reference coordinates from NZGD2000 epoch coordinates on 10 March 2012 from in testin.csv (Note: it is recommended to use the ITRF_NZGD2000.py script for this calculation)

```
python CalcDeformation.py --date=2012-03-10 --subtract testin.csv testout.csv
```

To calculate the coordinate adjustments corresponding to the implementation of the reverse patch model at the test points

```
python CalcDeformation.py --patch testin.csv testout.csv
```

To calculate the difference between the old and new velocity models at a set of points

```
python CalcDeformation.py --version=20130801-20000101 --only=ndm --date=2001-01-01 testin.csv testout.csv
```