dGPS Correction

Kinematic Post Processing (PPK) with RTKLIB

Written by Ben Purinton (purinton@uni-potsdam.de),

inspired by dGPS manual of Bodo Bookhagen

(bodo.bookhagen@uni-potsdam.de)

March 2019

Table of Contents

1 Installing RTKLIB **3**

2 Converting dGPS data to RINEX format **3**

3 Post-processing data with RTKLIB **4**

3.1 Preparation 4

3.1.1 Base Station 4

3.1.2 Precise GNSS Orbits: Navigation and Orbital Files 5

3.2 Files we now have for processing 6

3.3 Processing Example 6

4 Resulting Data **18**

4.1 Converting Data to .csv and .shp 18

1 Installing RTKLIB

We will use the free program *RTKLIB* to correct dGPS points using kinematic post-processing (PPK) with a base station. Download the file *rtklib_2.4.2.zip* from the "Full Package with Source Programs" area. The graphical user interface (GUI) is made for **Windows (and we use Windows in this manual)**, on Linux or Mac the command line interface (CUI) must be compiled by the user.

Note on Automation: Everything we do here can also be done at the command line (and therefore via programming in Python) on any operating system, but here we only describe the processing of one file using the GUI on Windows. The interested user is directed to the *RTKLIB* manual (included in the folder *rtklib_2.4.2 > docs*).

Unzip the *rtklib_2.4.2.zip*, then go to *rtklib_2.4.2 > bin* and you will see the GUI programs including *rtkpost_mkl.exe*, which we use for this exercise. **We use the *_mkl* version of *rtkpost*, which allows fast multi-core processing**. The command line equivalent program is *rtklib_2.4.2 > bin > rnx2rtkp* (see the *RTKLIB* manual for usage).

2 Converting dGPS data to RINEX format

Different dGPS units record data in different formats. For example, a Leica dGPS outputs files in *.m00* format. We need the data as *RINEX* to proceed with open source correction. This is the standard / universal GNSS format for data collection. You can read more about it here: https://en.wikipedia.org/wiki/RINEX.

The easiest way to convert data is with the command line program *teqc*. **Note: If you are using Trimble data in *.ssf* format, then you can only convert to *RINEX* with the *Trimble Pathfinder Office* program.** *Teqc* is included in *RTKLIB*. Open a command prompt window and input the following (for a *.m00* dGPS file from a Leica unit):

```
cd C:\path\to\dGPS_files
```

```
C:\path\to\rtklib_2.4.2\bin\teqc.exe +obs FileName.obs +nav \
    FileName.nav FileName.m00
```

Here *FileName.m00* is the *.m00* file you want to convert, and *FileName.obs/nav* are the same file converted to *RINEX* (with the same root name, but different extensions). This will create a new two part *RINEX* file: the *.obs* observation and *.nav* navigation. In the example below we use a dGPS rover file collected on 26 February 2019 using a Leica model:

```
C:\path\to\rtklib_2.4.2\bin\teqc.exe +obs BP01_leicaR2.obs \
```

```
    +nav BP01_leicaR2.nav BP01_6611_0226_174951.m00
```

Note on RINEX Formats: Recently *RINEX* v.3 files have become popular / standard. These files are capable of recording information from additional satellite signals. Base station data downloaded for correction will often be in this v.3 format. *Teqc* does not currently support *RINEX* v.3 conversion, so the *.obs* and *.nav RINEX* files output by *teqc* will be in v.2 format.

3 Post-processing data with RTKLIB

You can now use *RTKLIB* to post-process your *RINEX* files (*.obs* and *.nav*) using PPK. Before we begin, we convert the date of the collection (e.g., February 26, 2019) to the day of the year (e.g., *057*) with the calculator here: https://www-air.larc.nasa.gov/tools/jday.htm.

We also need the GPS week and day of the week to download the orbital clock files (see Sect. GNSS Satellite Orbital File), which we can look up here: https://www.ngs.noaa.gov/CORS/Gpscal.shtml. For dGPS data collected on 26 February 2019, the week is 2042 and the day of the week is 2 (Sunday = 0, Monday = 1 … Saturday = 6), giving a day identifier of *20422*.

All data that we require is hosted through the International GNSS Service (IGS) and the links can be found at: https://kb.igs.org/hc/en-us/articles/115003935351.

3.1 Preparation

Before you can correct your data with *RTKLIB*, you will need to get (1) the base-station data and (2) precise orbits. Some files from permanent base stations may take one to two days to become available on the IGS servers (ftp://igs.ensg.ign.fr/pub/igs/)

3.1.1 Base Station

The local base station *RINEX* data can be downloaded from the following FTP server for any permanent station around the world: ftp://igs.ensg.ign.fr/pub/igs/data/ The nearest permanent station can be found here: http://www.igs.org/network. For example, GNSS data from the Puna Plateau should be corrected with the Salta, Argentina permanent station with code *UNSA*. Permanent station data for every day are at ftp://igs.ensg.ign.fr/pub/igs/data/*yyyy*/*doy*/*code/*.crx.gz* with *yyyy* year, *doy* day of year, *code* permanent station code (*UNSA*), and * is a wildcard containing additional file information. We want the *crx.gz* file (not *rnx.gz* or *crx.sum.gz*), which is the *compact-RINEX* (*CRINEX*) base station data, decimated to 30 second observations. In the example, *UNSA00ARG_R_20190570000_01D_30S_MO.crx.gz* is used as the permanent base station.

NOTE: You can also use your own base station *RINEX* data, in which case you do NOT need to find or download the permanent station data, but you still need to get the file into the *.YYo RINEX* observations format as in the steps below.

Preparing the base station data

Before we can use the *CRINEX* file in *RTKLIB* we need to convert it to the standard, uncompressed *RINEX* format (so from extension *.crx.gz* to extension *.YYo*) in the following steps:

1. Unzip the *.gz* file to a *.crx* file (use the free program *7-zip* for this: https://www.7-zip.org/download.html
2. We also need to change the file extension to *.YYd* (where *YY* is the 2-digit year; 19 for 2019), this can be done at the command line with `copy` (on Windows) or `cp` (on Linux/Mac):

   ```
   copy FileName.crx FileName.YYd
   ```

   For our file:

   ```
   copy UNSA00ARG_R_20190570000_01D_30S_MO.crx \
   UNSA00ARG_R_20190570000_01D_30S_MO.19d
   ```

3. Now we can use the command-line program *crx2rnx.exe* included in *RTKLIB* to convert from *CRINEX* to *RINEX*:

   ```
   C:\path\to\rtklib_2.4.2\bin\crx2rnx.exe \
   FileName.YYd
   ```

   For our file

   ```
   C:\path\to\rtklib_2.4.2\bin\crx2rnx.exe \
   UNSA00ARG_R_20190570000_01D_30S_MO.19d
   ```

   This will create our final *RINEX* base file automatically with the new extension *.YYo* (e.g., *UNSA00ARG_R_20190570000_01D_30S_MO.19o*)

3.1.2 Precise GNSS Orbits: Navigation and Orbital Files

We also need some GNSS satellite navigation files and the satellite clock corrections to get a more exact dGPS correction. For additional information on these files see here: ftp://www.ngs.noaa.gov/cors/README.txt

GNSS Satellite Navigation File

The general navigation *RINEX* data (*brdc* file) can be downloaded from the same server location as the base station: ftp://igs.ensg.ign.fr/pub/igs/data/*yyyy*/*doy*/*brdcdoy*0*.YYn.Z*, again replacing *yyyy*, *doy*, and *YY* with your numbers. We want the *.YYn* file for GPS satellites and not the *.YYg*, which is for GLONASS satellites. Again, unzip this *.Z* file using *7-zip*. We should now have a navigation file like *brdc0570.19n*, where *057* is your day-of-year and *19* is your 2-digit year identifier.

GNSS Satellite Orbital File

The precise orbits need to downloaded from a separate directory at the same server: ftp://igs.ensg.ign.fr/pub/igs/products/*wwww*/ where *wwww* is the GPS week calculated above in Section Post-processing data with RTKLIB. There you will find the orbital files for every day-of-week (Sunday = 0, Monday = 1 … Saturday = 6) in the formats: *igp*, igr*,* or *igu**. In this example we use the ultra-rapid orbits: *igu20422_18.sp3.Z*. Again, unzip this *.Z* file using *7-zip*. We should now have an orbital file like *igu20422_18.sp3*, where *2042* is your GPS week, *2* is your day-of-week, and in this case for the ultra-rapid orbits *_18* is the hour of release (either *_00*, *_06*, *_12*, or *_18*). The *igr** and *igp** rapid and precise files do not have the hour-identifier and would look like: *igr20422.sp3.Z*.

NOTE: The final precise orbits (*igp** files) are only available 12-18 days after the observation data. The rapid orbits (*igr** files) are available 17-41 hours after the observation date and provide a similar accuracy. For faster processing you can also use the ultra-rapid orbits (*igu** files) produced every 6 hours.

3.2 Files we now have for processing

- A *RINEX .obs* observation file from your field measurements (e.g., *BP01_leicaR2.obs*)
- A *RINEX .YYo* observation file from your own base station OR a permanent station *RINEX* file (e.g., *UNSA00ARG_R_20190570000_01D_30S_MO.19o*)
- A broadcast navigation file from you date (e.g., *brdc0570.19n*)
- An orbital file from your GPS week and day-of-week (e.g., *igu20422_18.sp3* or *igr20422.sp3*)

3.3 Processing Example

Open the *rtkpost_mkl.exe* (not *rtkpost.exe*, that version is too slow) program in the *rtklib_2.4.2 > bin* directory. Follow carefully the steps in Figure 1 to Figure 12, where we correct a file collected on 26 February 2019 (*BP01_leicaR2.obs*).

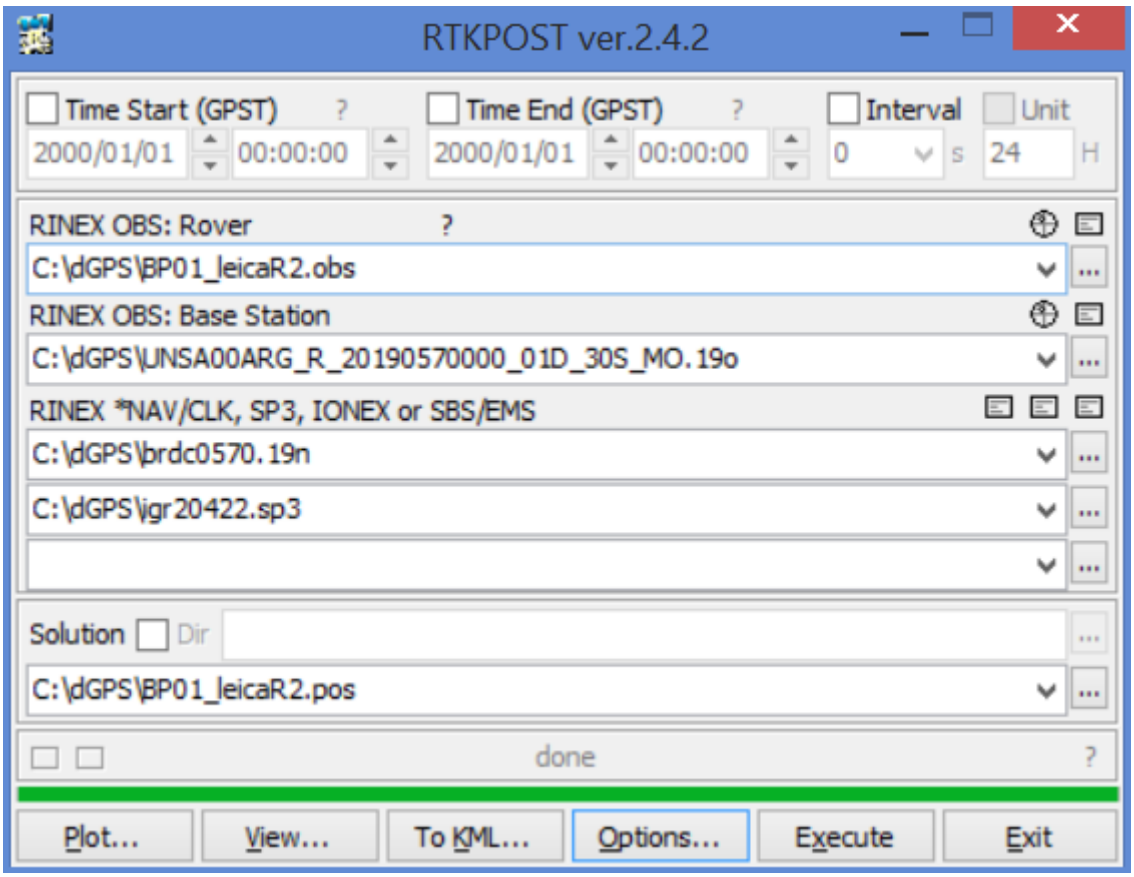

Figure 1: *RTKPOST* (*rtklib_2.4.2 > bin > rtkpost_mkl.exe* for post-processing dGPS data. The first input is uncorrected *RINEX* observation data, the second input is *RINEX* base station observation data, the third input is satellite broadcast navigation data, and the fourth input is precise orbit data. Output the corrected file as *.pos*. Now we need to set the processing options in the following steps.

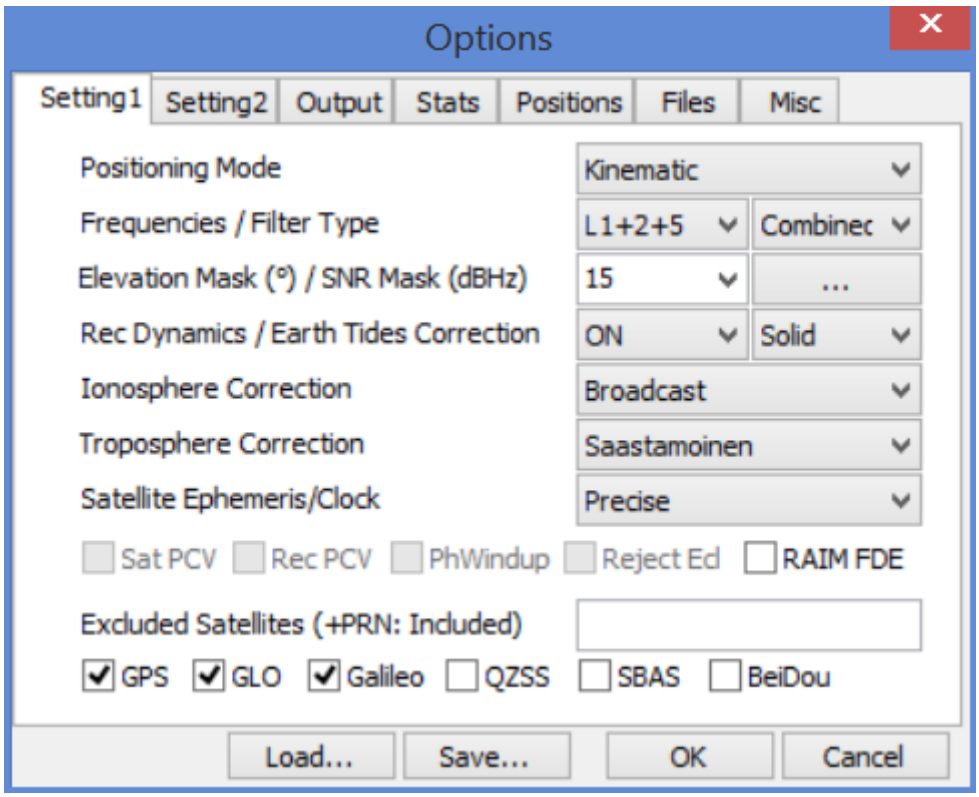

Figure 2: Tab 1 (Setting1) from the Options list (Parameter Window (1/7)). Make use to use the Positioning Mode Kinematic and also set all other parameters exactly as above. More advanced users can look in the *RTKLIB* manual and change some of these settings. Note that the calculation of the Kinematic solution will take some time, but will yield very good results. One can use different Ionospheric and Tropospheric correction and other options for the correction of the satellite ephemeris and clocks.

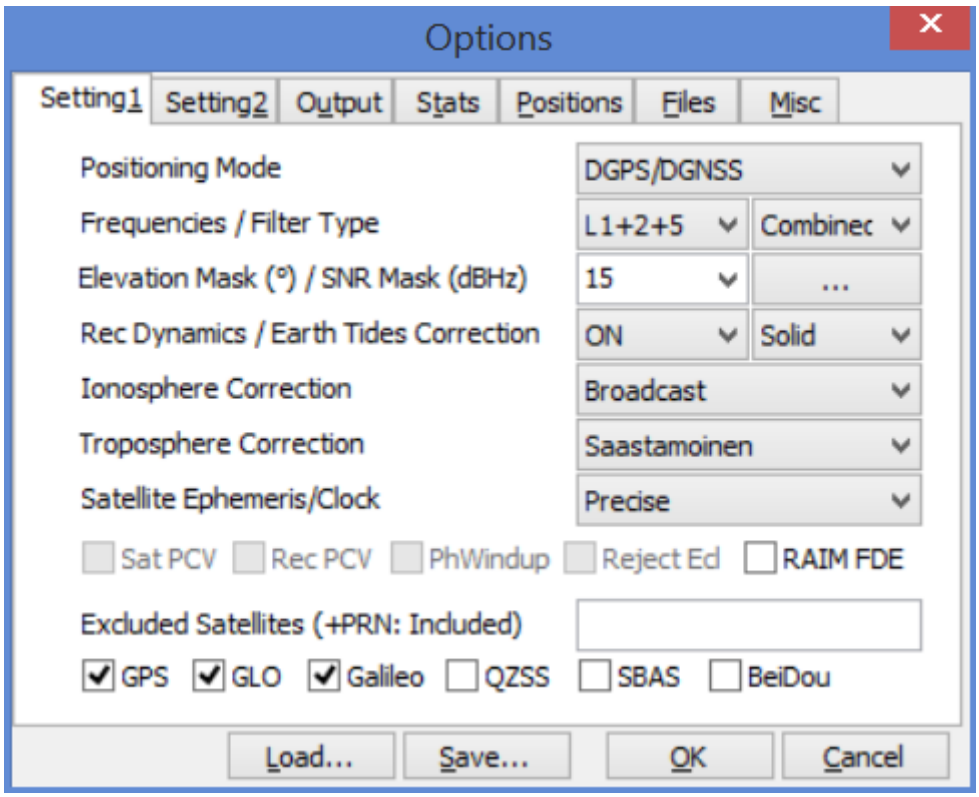

Figure 3: An alternative setting using the DGPS/DGNSS option (Parameter Window (1/7)). Uncertainties are larger, but calculation is faster. We advise using "Kinematic" positioning mode always.

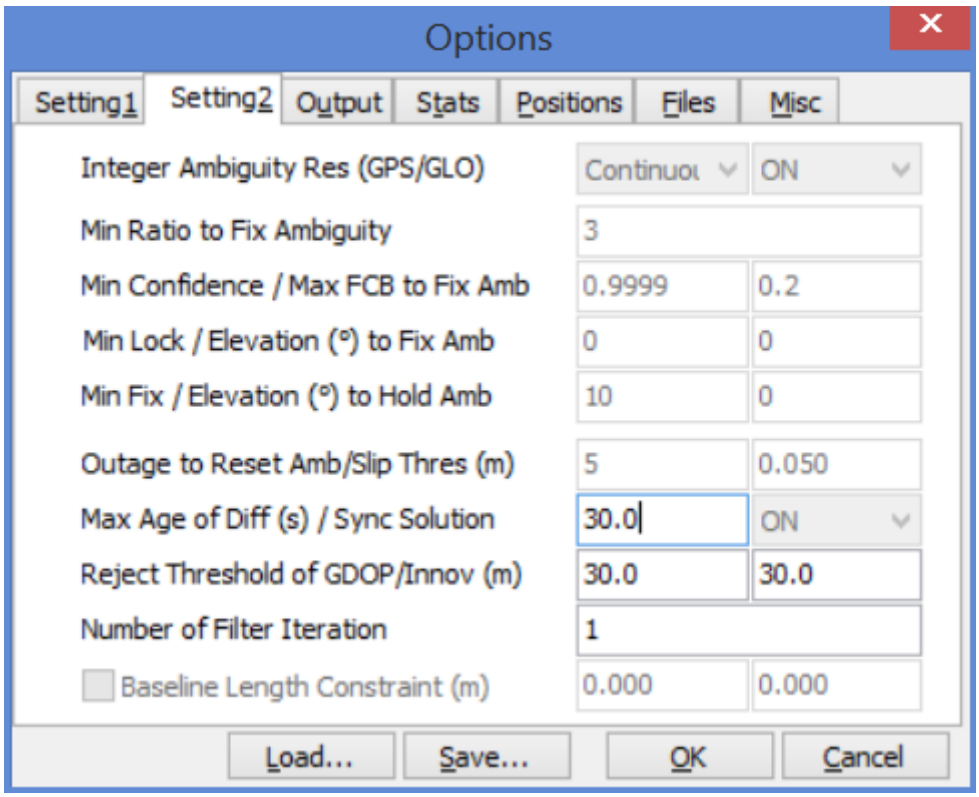

Figure 4: No changes are required here (Parameter Window (2/7)).

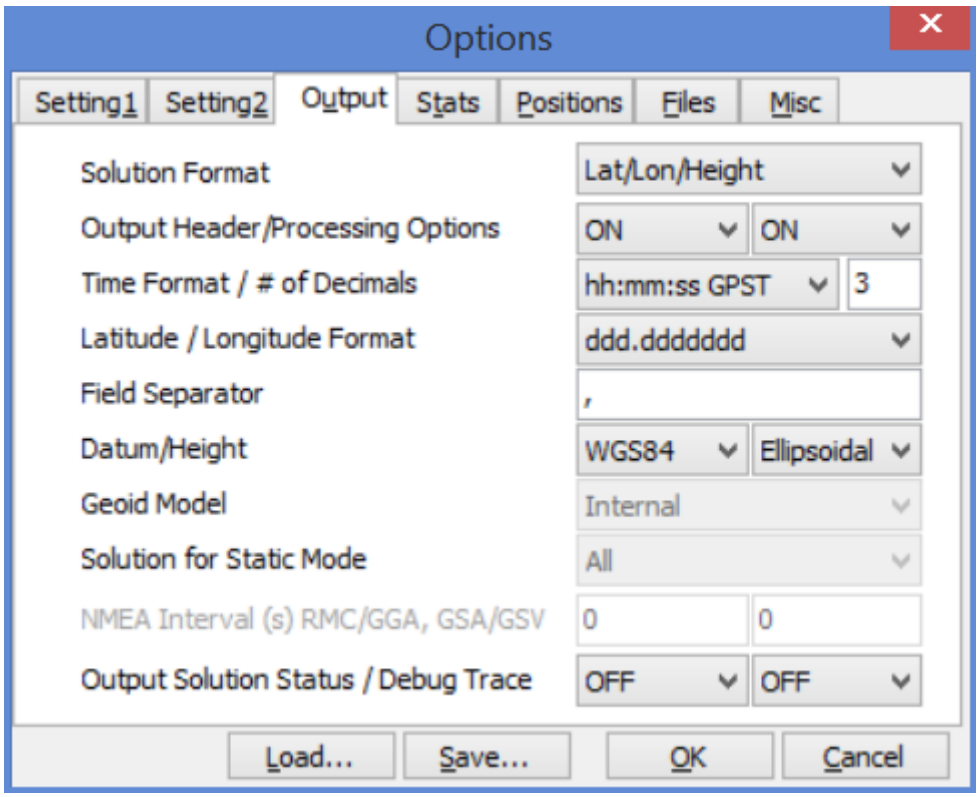

Figure 5: In the output tab, you can define the datum and height model (Parameter Window (3/7)). Be sure to add comma (,) as Field Separator. It is recommended to use the WGS84 ellipsoidal height model to obtain elevations that can be directly compared to the new SRTM NASADEM and the TanDEM-X DEM. Other appropriate Geoid Models (e.g., EGM96) can be selected.

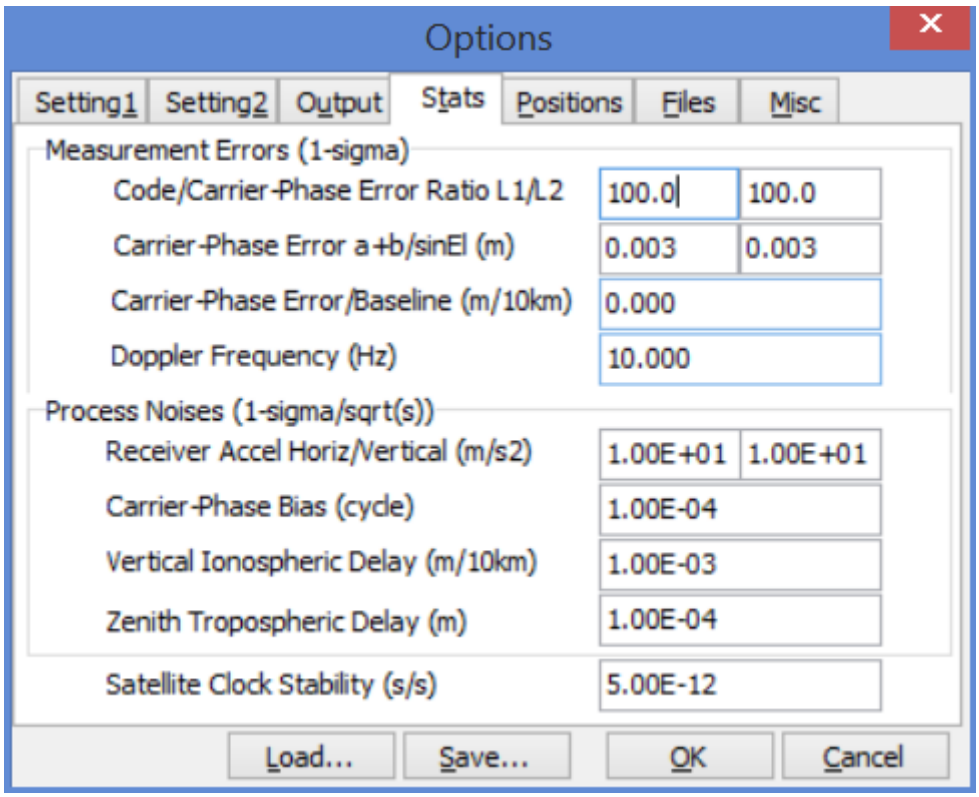

Figure 6: No changes are necessary in the Stats tab (Parameter Window (4/7)).

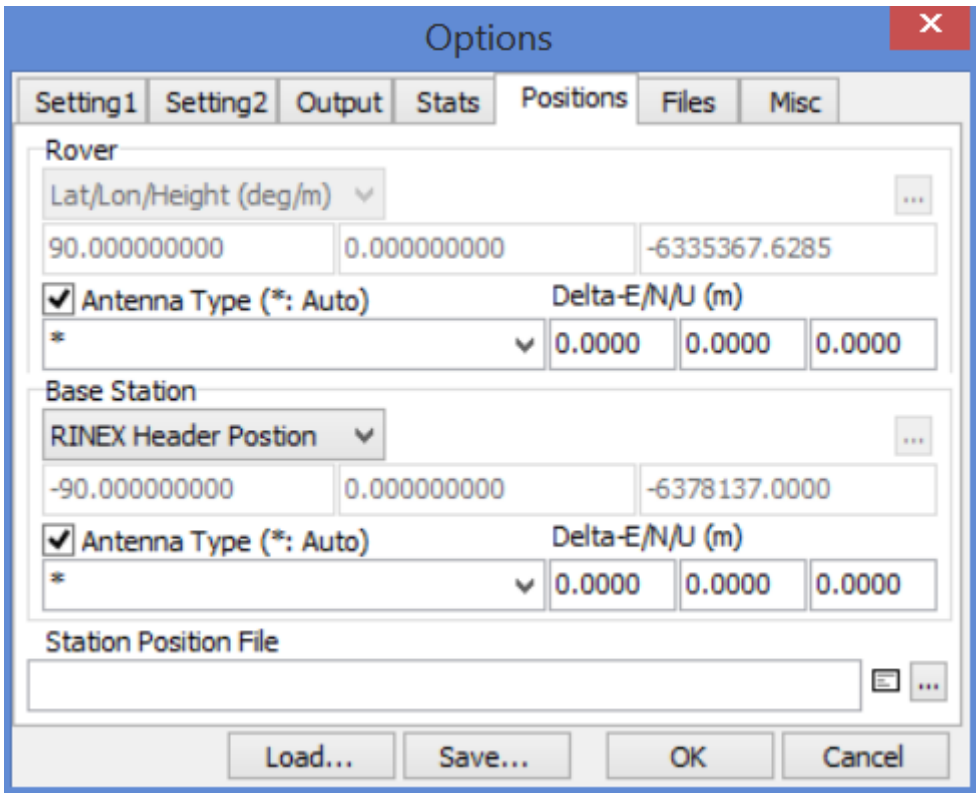

Figure 7: In "Rover" check on the "Antenna Type" option and put a *. Do not fill in the "Delta-E/N/U" section. Set the "Base Station" location to the *RINEX* Header Position and also check on the "Antenna Type" option and put a *. Do not fill in the "Delta-E/N/U" section. The * wildcards allow *RTKLIB* to read the antenna heights held in the *RINEX* header information (Parameter Window (5/7)).

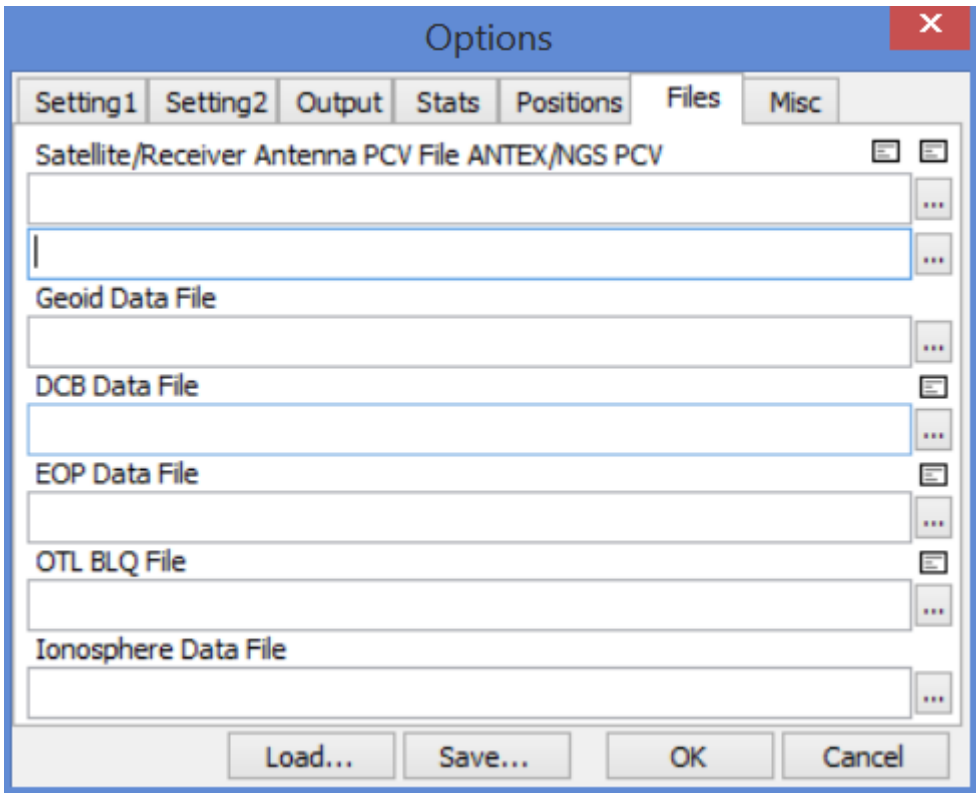

Figure 8: No changes necessary in the Files tab (Parameter Window (6/7)).

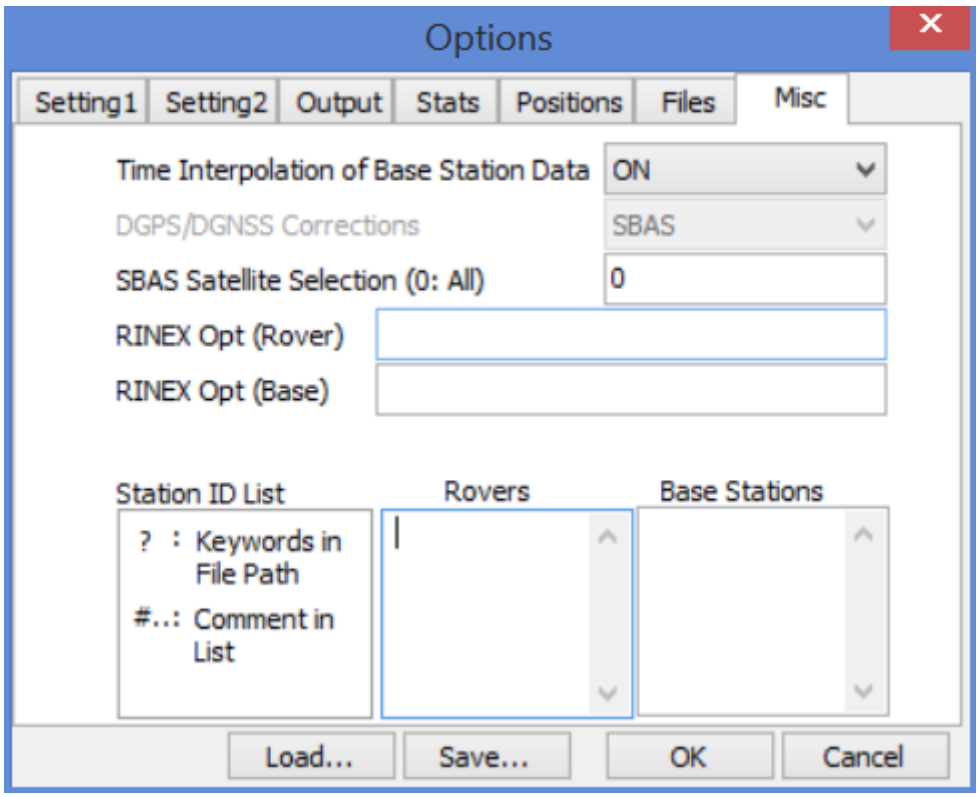

Figure 9: Set the Time Interpolation to ON (Parameter Window (7/7)).

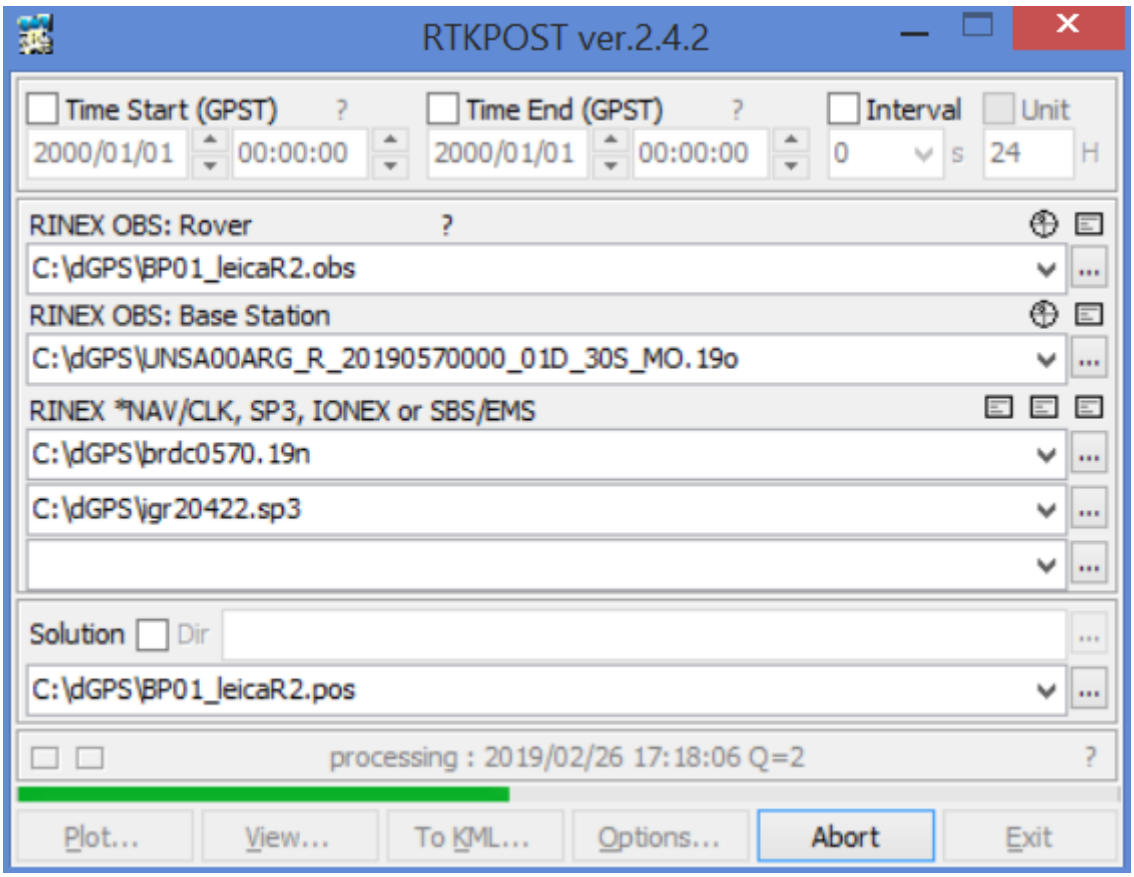

Figure 10: After we set the options we click Execute and wait. This may take a long time to process, especially with many points. The resulting data can be viewed by clicking on View. We can also output the results as a KML file to view in GoogleEarth.

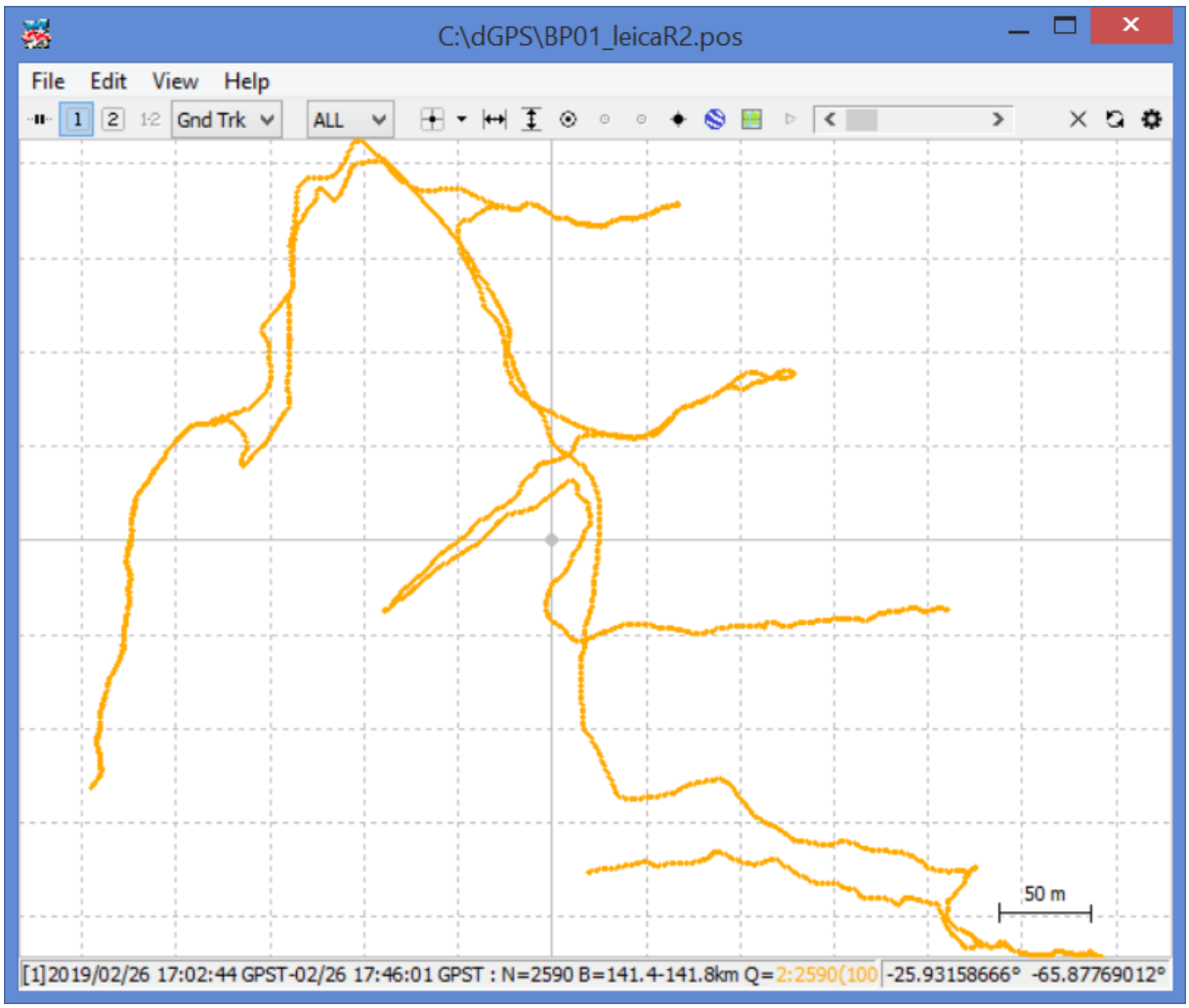

Figure 11: Example output ("Plot" button) from a dGPS corrected dataset using a kinematic solution.

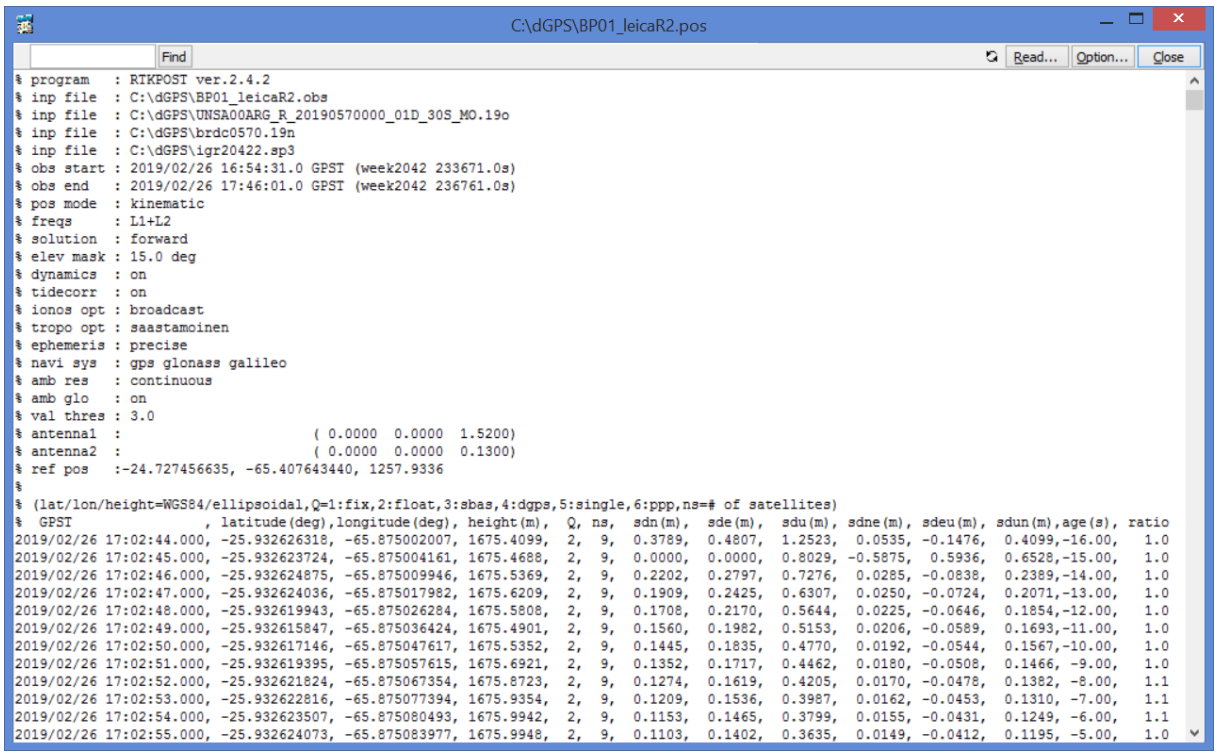

Figure 12: Example output ("View" button) *.pos* file output by the correction.

4 Resulting Data

The *.pos* file output by *RTKLIB* (Fig. 12) can be opened in any text-editor. For details of the column names, refer to the *RTKLIB* manual. The most important columns for us are the latitude, longitude, height, and the uncertainty in x, y, and z directions (sdn, sde, and sdu; provided in meters). Be careful to only use points with low (< 0.5 m or even less) vertical uncertainty in further analysis. The file can be converted to a *.csv* file and imported into QGIS then saved as a shapefile. Alternatively, the *.pos* file can be read in directly to Python or MATLAB and manipulated from there (e.g., as a Python pandas dataframe).

4.1 Converting Data to .csv and .shp

The first step is to open the *.pos* file in a text editor (notepad++ is a pretty good one). Delete all lines that begin with "%" and add as a new first line (Figure 13-14):

GPST,latitude(deg),longitude(deg),height(m),Q,ns,sdn(m),sde(m),sdu(m),sdne(m),sdeu(m),sdun(m),age(s),ratio

Next save the file with a new extension *.csv* (for comma separated values). This *.csv* can now be opened in QGIS by clicking the “Add Delimited Text Layer” option (Figure 15). The file can be opened and saved out as a shapefile as shown in Figure 16-18.

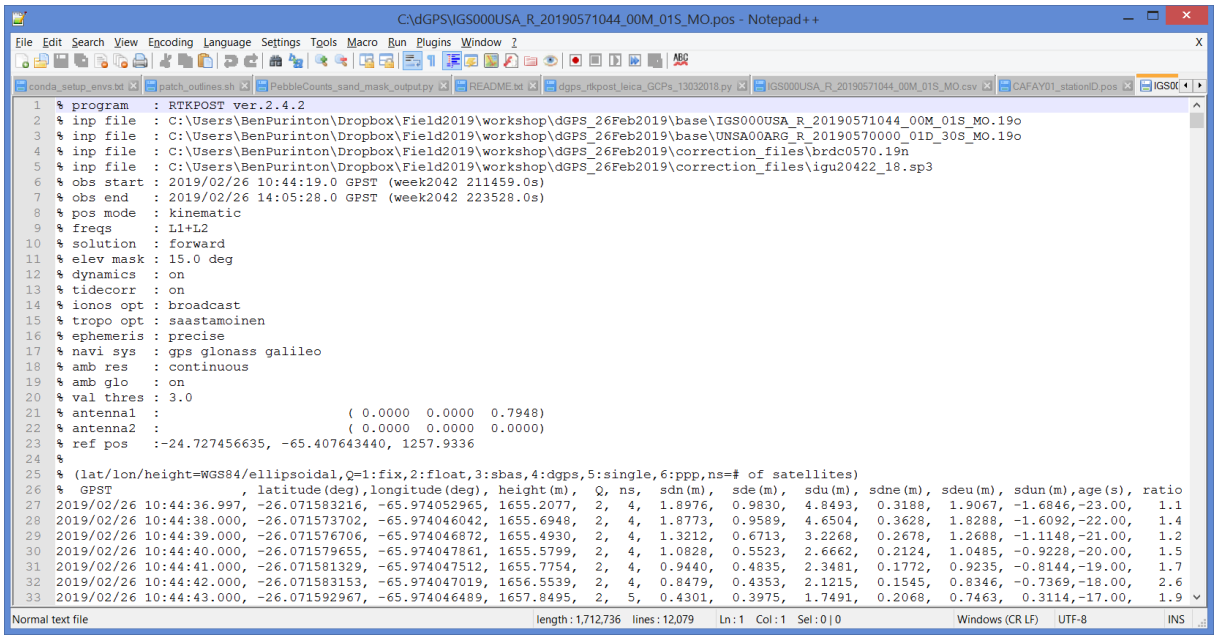

Figure 13: Open the *.pos* file in your favorite text editor. Here I’m using Notepad++.

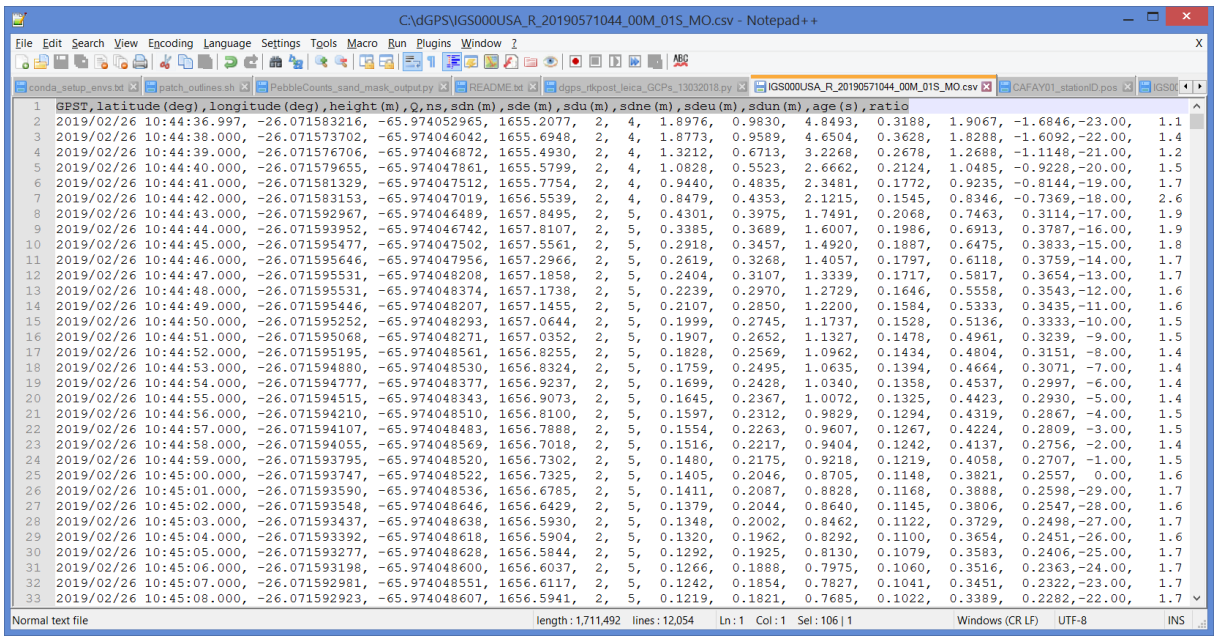

Figure 14: Delete all the lines beginning with % from the *.pos* file. Add a new first line: *GPST,latitude(deg),longitude(deg),height(m),Q,ns,sdn(m),sde(m),sdu(m),sdne(m),sdeu(m),sdun(m),age(s),ratio*. Save the file as a *.csv*.

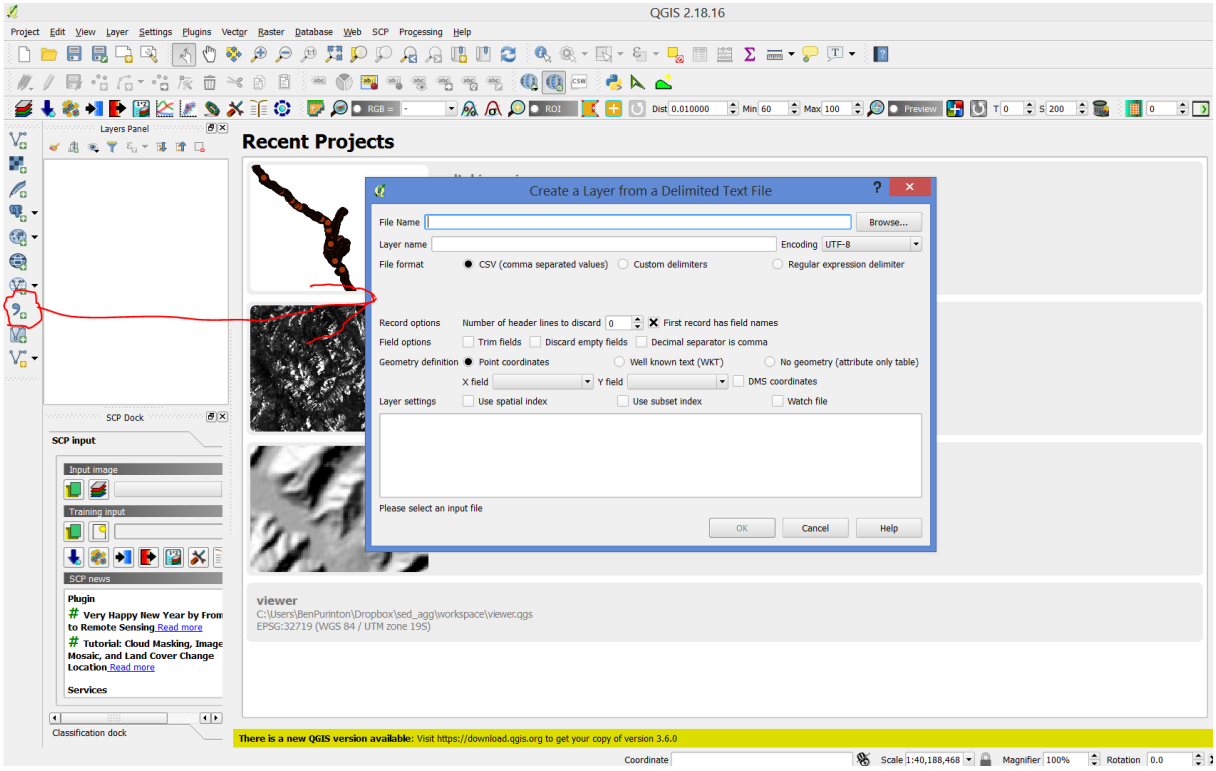

Figure 15: Open QGIS and add the *.csv* file with “Create Layer from Delimited Text File Option”.

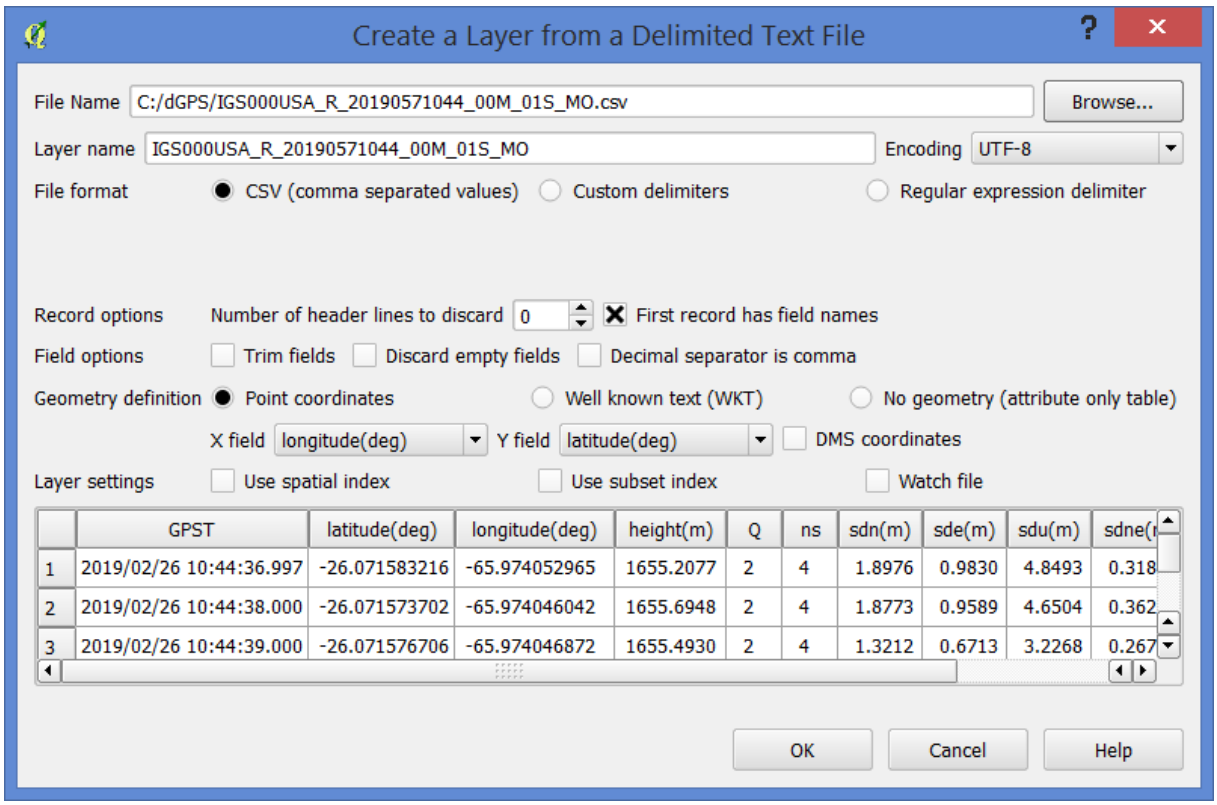

Figure 16: Browse to the new *.csv* file in "File Name". The information should be automatically filled out by the tool and should look like the above (longitude as the X field and latitude as the Y field).

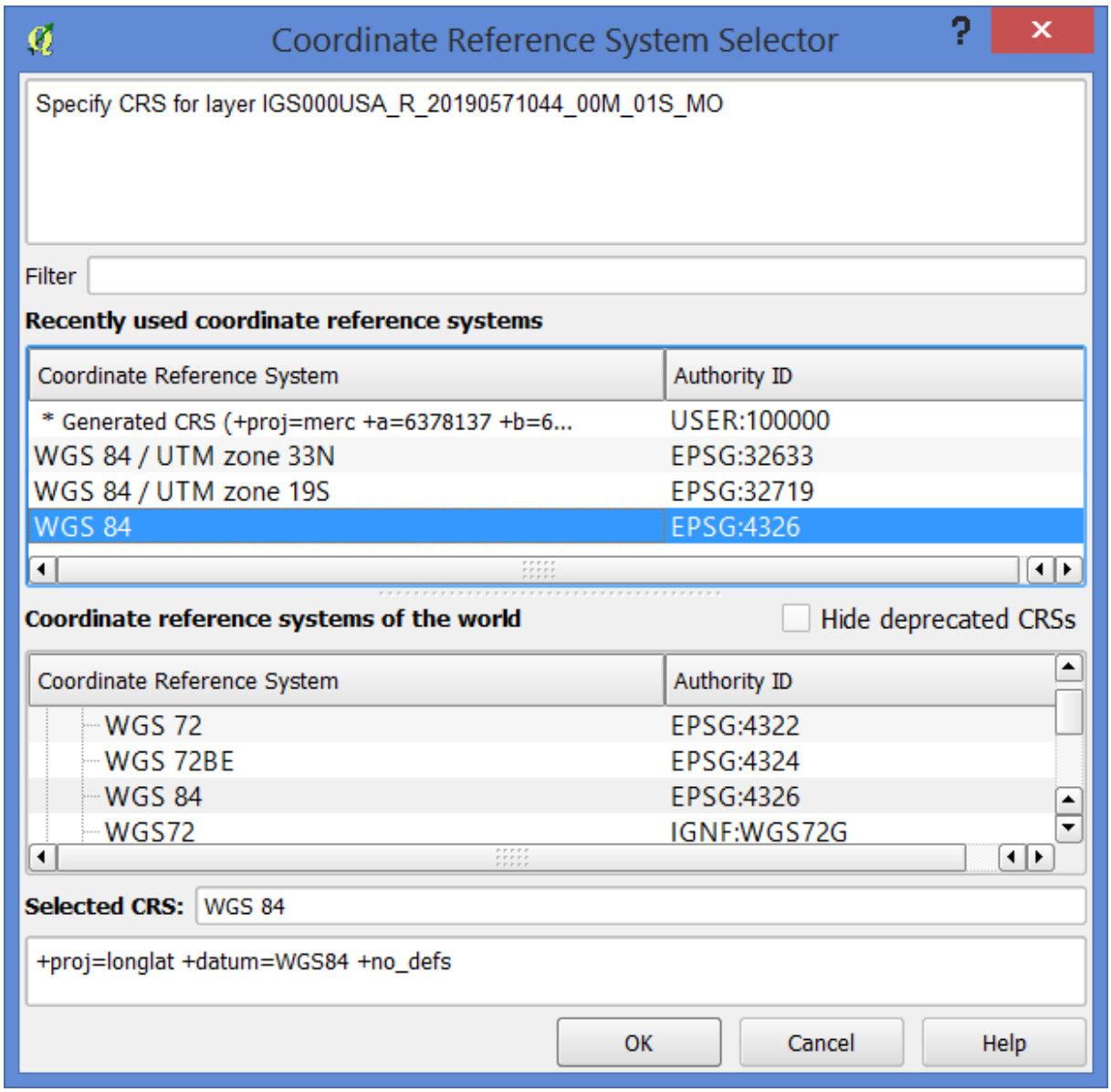

Figure 17: When prompted, give the coordinate system of the file as WGS84.

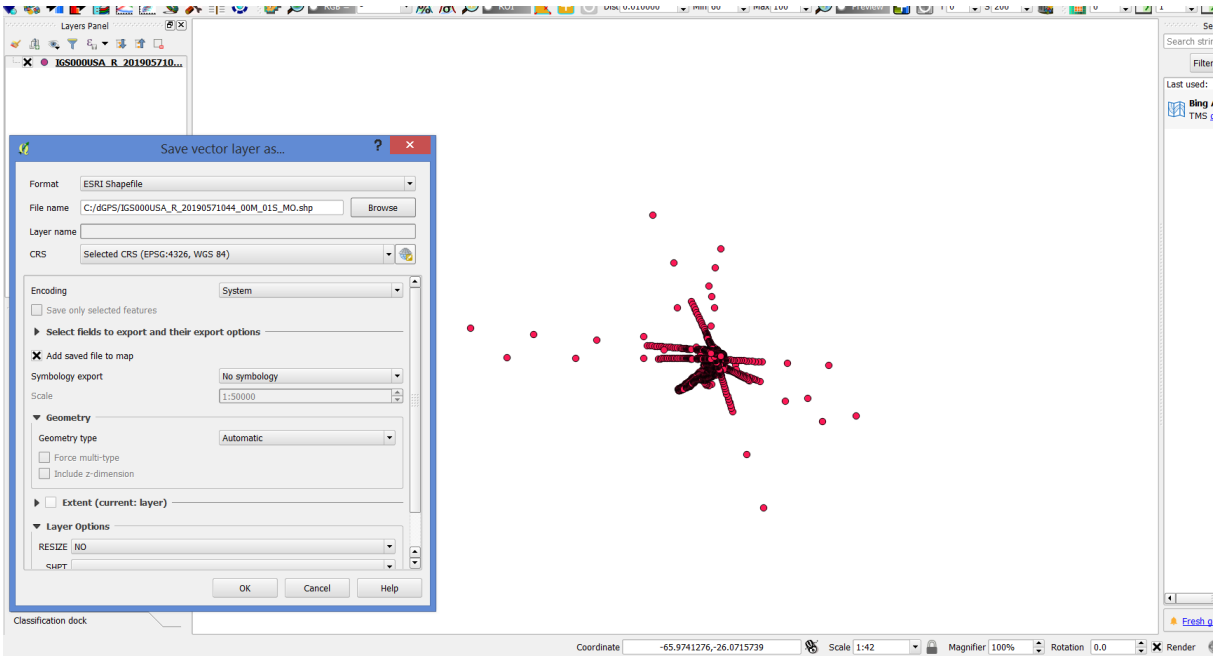

Figure 18: Right click on the file and go to “Save As”. Save the file out as “ESRI Shapefile” format, and provide a file name with the extension *.shp*. Press “OK” and the data is now converted to a *.shp* file for further use. It may be good to delete points from this shapefile with uncertainties in *XYZ* (*sdn, sde, sdu*) > 0.5 m, or even more precise (> 0.1 m), depending on your application and desired precision.