Lab #1 – Additional remarks

Drift correction

The goals of this lab are:

1) Load the data from field file into a matlab data matrix (matrix `data` provided). Understand its structure (meanings of the twelve columns)

2) Transform the LaCoste G267 readings to mGal. This procedure is called *calibration*. Add a column to the matrix `data` containing the calibrated readings.

3) Apply corrections for diurnal gravity variation due to the tides.

4) Derive the drift curve for the instrument. Put the drift value into the data table and subtract it from the data. In this way, you will obtain readings in mGal as if recorded by an instrument without drift.

5) Measure the standard deviations for each station for which you have repeated measurements. This will be the estimate of measurement error at each site.

How to perform these steps:

2) Calibration

a) Add an empty column #13 to the data matrix for the corrected gravity. At every step, you will update this column by applying the various corrections.

b) Add an additional column #14 for the raw calibrated values. Write a code to fill this column with calibrated values using the table and procedure 1)-6) at the end of the document lab1.pdf. Copy this column into column #13.

3) Tidal correction

a) Execute function `tides.m` as suggested in the provided template. This will return a vector of times of the measurements (in hours) and the corresponding tidal gravity.

b) Replace the column containing the observation times in `data` (#2) with the times returned by `tides.m`

c) Add a column to the matrix `data` and place the values of tidal gravity in it. Subtract this column from the column #13.

4) Drift correction

a) Extract a subset of the `data` matrix containing all base station readings (use `extract.m`).

b) For the base station, plot both the uncalibrated (column #12) and calibrated and tide-corrected gravity (#13) vs. time, ascertain that they are correct. Observe the "drift" – this is the strong continuous decrease of readings with time.

Here is the model for the drift: the drift is a time-only dependent value added to each gravity reading g_s at station s:

$$g_s\left(t_{\text{obs}}\right)=g_s+d\left(t_{\text{obs}}\right).$$

It is convenient to separate the long-term ("weekly") and short-term ("daily") drifts:

$$g_s\left(t_{\text{obs}}\right)=g_s+d_{\text{week}}\left(t_{\text{obs}}\right)+d_{\text{day}}\left(t_{\text{obs}}\right).$$

Our task here is to construct **smooth** functions $d_{\text{day}}(t)$ and $d_{\text{week}}(t)$ by using the plots, so that the drifts line $d_{\text{week}}\left(t_{\text{obs}}\right)+d_{\text{day}}\left(t_{\text{obs}}\right)$ passes through the middles of data points (for the base station) for each day and within about 0.01 mGal (instrument error) of most of them.

c) You will simply pick the drift values from the plots. Create two matrices with two columns each, containing: 1) the times at which the drifts are picked, 2) the drift value. The first matrix will contain your picked weekly drift, and the second – daily drift picked later.

```
Drift1  = [ time1,  drift_weekly1,

            time2,  drift_weekly2,

            …

            ];

Drift2  = [ time1,  drift_daily1,

            time2,  drift_daily2,

            …

            ];
```

d) To begin, take the first `time` slightly before the first point and the last time later than the last point. Plot the `drift1` curve vs. `time` in the same plot as in b). Adjust the values of `drift1` for these times so that the drift curves passes roughly through your gravity points in the plot. Set `drift2 = 0` for now. Add additional points to matrix `drift1` as needed (no more than 1–2 points).

e) Once the `drift1` curve is reasonably close to the data points, subtract it from the data. This subtraction will look like this (why?):

```
base_data[:,13] – interp1(drift1(:,1),drift1(:,2),base_data[:,2])
```

Plot this quantity vs. time. It will be near-constant, with much smaller variation than before.

f) Using function `axis()`, change the plot area to zoom in onto the first day of acquisition. Similarly to d), pick and plot the `drift2` curve so that it passes within ~0.01 mGal of the data points and is smooth. If this cannot be done for some points, you probably have a tare. Function `polyfit` may sometimes be *judiciously* used to make smooth curve (I still do not think this is necessary or justified).

g) Repeat step f) for each day, filling the `drif2` matrix.

h) Remove the measured drift from the base station data (why this formula?):

```
base_corrected = base_data[:,13] - …
      interp1(drift1(:,1),drift1(:,2),base_data[:,2]) - …
      interp1(drift2(:,1),drift2(:,2),base_data[:,2])
```

This value should be approximately constant. Use its mean to evaluate the shift required to place the base-station reading at the specified absolute gravity (g_{abs} = 981.128705 mGal this year):

```
base_shift = mean(base_corrected) -  g_abs
```

i) Now the total drift can be obtained **for all data points** as (why?):

```
total_drift = interp1(drift(:,1),drift(:,2)+drift(:,3),data[:,2])
      – base_shift
```

j) Put this value in a separate column of matrix `data` and subtract from column #13. Drift correction is done!

5) Standard deviations

After the drift correction, all readings (column #13) **for any given station** should be near-constant but not zero mean.

The scatter (square root of the variance) in these points can be measured by using the following formula for the standard deviation:

$$\sigma \approx s_{N-1} \equiv \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(g_i - \bar{g}_N\right)^2} \text{, where } \bar{g}_N \equiv \frac{1}{N}\sum_{i=1}^{N} g_i \text{ is the mean value.}$$

Another measure of the scatter between sequential points is:

$$\tilde{s}_{N-1} \equiv \sqrt{\frac{\sum_{j=1}^{N-1}\left(g_{j+1} - g_j\right)^2}{N-1}}.$$

a) Write a loop in your code to evaluate these expressions for each station **with repeats**. Compare:

   i. The values of $s_{N\text{-}1}$ for the base and other stations. Are they close to 0.01 mGal?

   ii. The two measures of $s_{N\text{-}1}$ for the base station.