Laboratory Project:
Solar Oscillation Spectrum Analysis

Prof. Ilídio Lopes
José Vargas Lopes
(josevlopes@ist.utl.pt)
Astrophysics Laboratory

October 2, 2018

PART II: Spectrum Analysis

In the second part of the astrophysics laboratory, you will compare the oscillation frequencies obtained in the first part with real data measured by the Global Oscillation Low Frequency experiment (GOLF) aboard the ESA-NASA's Solar and Heliospheric Observatory (SoHO), which is orbiting the earth at a distance of 1.5 million kilometres (about four times the distance to the moon). The experiment records variations in the Sun's luminosity, from which its surface velocity dispersion is deduced. Your task is to use the velocity dispersion time series to obtain the measured oscillation frequencies.

1. The time-series for this part of the project have been acquired by two different Photomultipliers (pm1 and pm2) of the experiment. The detectors have been further calibrated using two different methods (calib1 and calib2).

   (a) Plot the Velocity dispersions measured by the different satellite photo multipliers for each calibration method.
   CAUTION: Be sure to read the file `readme_data_golf.txt`.

   (b) Choose one of the data-sets. Justify your choice.

2. To obtain the frequency oscillation modes, you will have to perform a Fourier transform to the velocity dispersion.

   (a) Apply the Fourier transform to the chosen data-set, and plot the result.

   (b) What is the order of magnitude of the observed frequencies/periods? Can you explain why there are high intensity peaks near $f \approx 0$ Hz?

3. To identify each eigen-frequency, you will need to compare the observed frequencies (obtained from Fourier analysis of the data-series) with the theoretical frequencies obtained by modelling the Sun.

(a) Match the observed frequencies with the oscillation frequencies obtained in **Part I** by direct comparison between the acoustic modes.

(b) Compute both the predicted small and large separations

$$\Delta\nu \equiv \nu_{n+1,l} - \nu_{n,l}, \qquad \delta\nu = \nu_{n,l} - \nu_{n-1,l+2} \tag{1}$$

for the obtained acoustic modes in **Part I**. Discuss.

4. The eigen-frequencies of solar oscillations, and in consequence, the large and small separations, can be obtained from fitting an adequate function to the data.

(a) Fit a Lorentzian function

$$L(x) = \frac{1}{\pi}\frac{\frac{1}{2}\Gamma}{(x-x_0)^2 + \left(\frac{1}{2}\Gamma\right)^2} \tag{2}$$

to the obtained Fourier spectrum to obtain the observed eigen-frequencies (choose some of the observed peaks to fit).

(b) Compute the large and small separations for the fitted frequencies and compare them to the frequencies obtained in **Part I**.

Annex I: Useful commands MatLab

- Import data:

  importdata(`filename`,`delimiter`,`Header lines`)

  – The file name `filename` has to be inside quotes, and needs to be on the current working directory.

  – `delimiter` is a string and indicates to MatLab the delimiter between the data values.

  – `Header lines` is an integer, and indicates do MatLab the number of header lines in file `filename`.

- Interpolate data:

  `vq` = interp1(`x`,`v`,`xq`,`method`)

  – `vq` is the interpolated data from (`x`, `v`) at the query points `xq`.

  – `method` is the method of interpolation. It can be: `linear`, `cubic`, `spline`, etc.

- Plot data:

  plot(`x1`,`y1`,`LineSpec1`,`x2`,`y2`,`LineSpec2`,...)

  – Plot the curves (`x1`,`y1`), (`x2`,`y2`) etc.

  – `LineSpec` sets the line style, marker symbol, and colour. For example (...,`"--"`,`"r"`,...) will plot a red dashed line.

- Fourier Transform:

  `output` = DataFFT (`inputdata`)

  – `inputdata` is the data to be transformed. It needs to be in the format of an array with 2 columns.

  – `ouputdata` is the output data. It is in the format of an array with 2 columns.

- Fit data:

  `fitobject` = fit(`x`,`y`,`fit_fun`)

  – `fitobject` is an object containing the fit coeficients from fitting the function `fit_fun` to data (`x`,`y`).

  – `fit_fun` is the function to be fitted. It can be a built-in function, such as `"poly2"`, or it can be a function written by the user, using the function fittype(`"function"`).