Part II Computational Physics Projects

David Buscher <db106@cam.ac.uk>

v2.6 — 2020/02/04

Contents

1 Introduction

You should choose one of the following projects. The aim is to develop and test *your own* `C++` or `Python` program, and use this to investigate the problem. You may (and indeed are encouraged) to use the functions of the GNU Science Library `GSL`, and the `FFTW` library or the `numpy` and `scipy` `Python` packages where needed.

Keep a record of the analysis you do before programming, details of the library routines you use, and the tests that you make, as well as the final commented program and results. The written up form of these, possibly together with an oral session with a head of class, will form the basis of the assessment.

The problems are to differing degrees **open-ended**: what is given below is just a starting point for each project. You should investigate the problem you have chosen as thoroughly as you can in the time allocated for this task. Bear in mind that the credit for this work is one unit of further work.

2 Write-up

Your report should not exceed 3000 words (excluding summary, captions, references and appendices — note that your code listing should be in an appendix). **Include a word count at the end of your summary.**

The following is intended as a guide to the structure of your report, and what should be included in it. You should follow the basic structure suggested in the booklet "Keeping Laboratory Notes and Writing Formal Reports" and the general advice given there on style, except as noted below.

The aim is to produce a report on your solution to one of the set problems. Background material will have been covered in lectures and in the exercises.

What you should concentrate on in your write up, therefore, are those aspects of computational physics relevant to the problem you have chosen. The physics of some of the problems may go beyond your current knowledge, and understanding the advanced ideas behind some of the problems is not part of this exercise. However, basic physical reasoning and the physics that you have already met in Part IB and this term's Part II courses will be assumed.

You should include the following:

1. A brief introduction to the problem.
2. The analysis of the computational aspects of the problem that were necessary for you to undertake its solution. This might include a brief description of relevant computational physics, choice of approach, choice of suitable routines, scaling of the problem, etc.
3. The implementation: that is, your program. The full program listing should be included in an appendix. In the main text your description of the program

should be relatively short, concentrating on your overall approach and the way you implemented the necessary algorithms to solve the problem.

4. Performance: you should include a brief section on the performance of your program, and discuss any steps taken to ensure the program ran efficiently. You might also like to include indicative run times for your code, i.e. how much CPU time was used to generate a given result.
5. Results and analysis. This should include discussion of the tests you performed to demonstrate that your program is working correctly. Your final results should also be presented together with a discussion of errors and any other computational problems. You should aim to present several clear plots which illustrate the results.

2.1 Structure

A possible structure might be the following (although details are problem specific):

Abstract; Introduction; Analysis; Implementation; Results and Discussion; Conclusions.

2.2 Creating your Report: Word or LaTeX

You can create your report using any word processing package of your choice. Many of you will use MS Word or similar applications, which is fine. However, an excellent alternative is LaTeX, a popular program for creating documents that has good support for scientific work (mathematics), and for including program listings; it is widely used in scientific research. All the lecture handouts and presentations were created as PDF files using LaTeX.

If you would like to try this system, there is an example document on the course web page and there is lots of support available on the web. You can use the **Overleaf** web-based service (`www.overleaf.com`), which allows LaTeX to be used without installing anything on your computer, and makes the process of learning and using LaTeX much more straightforward than it was in the past.

2.3 Submitting your report

As well as submitting a paper copy of your report to the Teaching Office in the usual way by the deadline, you must also deposit an electronic copy on the MCS system in your course pigeonhole. Specifically you should upload your report and associated files to

`/ux/PHYSICS/PART_2/pigeonhole/CRSID/project`

by the same deadline. `CRSID` is of course your own user identifier.

You should put there:

- A copy of your report as a PDF file (you should not include the Word or LaTeX source). Please give the file a sensible name, something like `project-CRSID.pdf` would be a good choice.
- The source code of your program, in one or multiple files
- Any associated scripts or `Makefiles` required to compile and link the code (not needed for `Python`);
- Any animation files you refer to in your report;

2.4 Marking

The credit for this project is one unit of further work. Approximately equal credit will be given to each of the following areas:

1. Analysis of the computational physics aspects of the problem;
2. Implementation of the algorithm, including code style, readability, quality;
3. Results, analysis of errors (if applicable), tests and discussion of the relevant computational physics;
4. Overall presentation of the report, structure, etc.

A number of candidates may be selected for Viva voce (i.e. oral) examination early in the Lent Term after submission, as a matter of routine, and therefore a summons to a Viva should not be taken to indicate that there is anything amiss. You will be asked some straightforward questions on your project work, and may be asked to elaborate on the extent of discussions you have had with other students. You may also be asked to give a live demonstration of your program. So long as you can demonstrate that your write-up is indeed your own, your answers will not alter your project marks.

3 Programming language

If you so wish, you may use other languages to help with scripting, graphing etc, but *the core numerical routines must be written in `C++` or `Python`* (you should use `python3` and not `python2`).

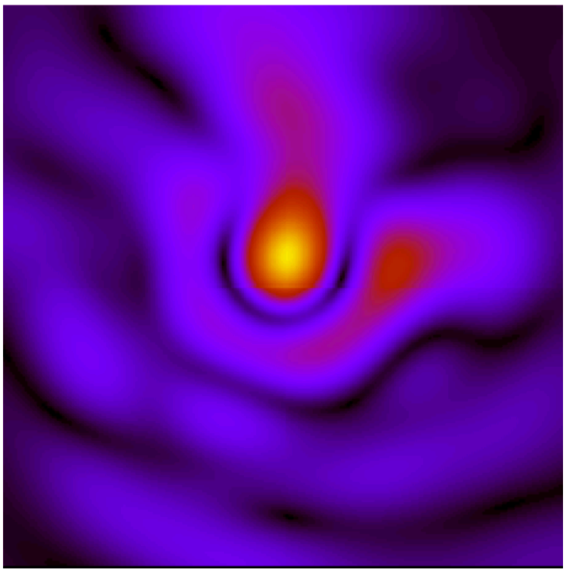
4 Collaboration

The remarks about collaboration and cheating which are contained in the course handbook, with reference to experimental work, also apply here. The project differs from the compulsory exercises in this respect. In the exercises, working together to promote your learning is encouraged. In this assessed project, you should work essentially on your own.

Discussion of the problems among students is encouraged, as this will often help your understanding. However, your program and your report should be written by you, and only results produced by your program should be presented in your report. It would be unacceptable for two students to submit near identical programs.

Any attempt to pass off the work of others as being produced by yourself will be regarded as cheating and may be referred to the examiners. When the report is handed in, you will be asked to sign a declaration about the degree of collaboration that went in to the work that is being submitted.

5 Project A: Optics with imperfect mirrors



This project uses 2-dimensional Fourier techniques to predict the properties of real mirrors which are imperfect i.e. whose surfaces are not perfectly smooth (or equivalently, perfect mirrors observing through a phase-distorting medium such as the atmosphere). A mirror is a two-dimensional aperture, whose far-field diffraction pattern is equal to the Fourier transform of the aperture function $A(x, y)$. For a perfect optical telescope, $A(x, y) = 1$, for $r < R$ and $A = 0$ otherwise, where R is the telescope radius and $r^2 = x^2 + y^2$. The diffraction pattern is then the well-known Airy pattern.

In practice, a telescope has a complex aperture function A. Its phase may deviate from zero because of errors in the manufacturing or the gravitational deformation of the surface — in other words, the mirror is not perfectly parabolic.

In addition, the aperture function may not have constant amplitude. Radio antennas will often use a *tapered aperture function*, such that $|A(x, y)| = \exp(-r^2/(2\sigma_T^2))$ where the length σ_T is a design parameter: this is called *Gaussian illumination* of the aperture. (Often the taper is quoted as the power at the edge of the dish with respect to that at the centre, in decibel units).

The investigations to be performed are:

1. No surface errors. The phase is zero in this case. Write a program to calculate the far-field amplitude and phase diffraction pattern for a mirror using a 2-dimensional FFT. Use this to quantitatively assess the accuracy of your numerical simulation. For this problem use $R = 6\,\mathrm{m}$, wavelength $\lambda = 1\,\mathrm{mm}$, and $\sigma_T = R/2$.
2. Investigate how the on-axis intensity of the diffraction pattern or “beam pattern”, i.e. squared modulus of the diffraction pattern amplitude at the centre

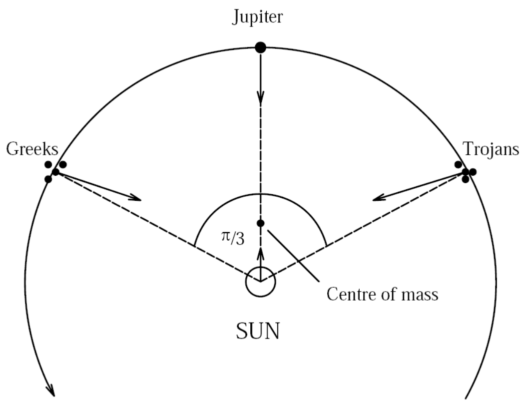
of the pattern, depends on σ_T/R. Attempt to quantify how both the size of the beam pattern (for example the full width at half maxmimum extent) and the on-axis intensity vary with the taper σ_T/R.

3. Now add some random errors to your surface, which introduce phase errors (but not amplitude errors) into the aperture function. A physical error of ε in metres in part of the aperture introduces a phase error $4\pi\varepsilon/\lambda$ (can you see why?).
   Add random Gaussian-distributed errors with RMS value σ_ε to your surface and see how they affect the beam as the ratio $\sigma_\varepsilon/\lambda$ varies. In particular plot the on-axis power and see how it falls off with $\sigma_\varepsilon/\lambda$.
4. Investigate how the beam pattern varies according to the spatial distribution of errors in the surface. Rather than purely uncorrelated random errors from point to point, introduce correlated deformations into the surface with a correlation length l_c and see how l_c affects the beam patterns.
   You can generate a correlated deformation by convolving (blurring/smoothing) an uncorrelated random surface error with a 2-D spatial Gaussian with a standard deviation l_c. Scale the amplitude of the Gaussian inversely with its width so that changing l_c does not change the RMS wavefront error (and check that this works). Be careful about what happens to the convolution at the edges of your data.
   Explore the behaviour of the beam pattern as a function of l_c for values of the RMS wavefront error ranging from 0.1 to 2 radians (you can of course explore a larger range). It is instructive to look at *statistical* properties of the beam by generating a large number of random wavefront patterns with the same RMS wavefront and correlation length. Example statistical properties include the shape and size of the *mean* beam intensity pattern, and the fluctuation in the intensity at the centre of the beam.
   Another statistical property of the beam pattern you can investigate is the mean *Modulation Transfer Function* (MTF), that is to say the the mean modulus of the Fourier transform of the beam intensity pattern, normalised so that the MTF at zero spatial frequency is unity.

Using logarithmic intensity scaling in the 2D plots of the diffraction pattern can be helpful to see the details of the pattern.

You can check some of your results against results from the literature, including radio astronomy textbooks (e.g. Rohlfs and Wilson, “Tools of Radio Astronomy”) and the literature on “speckle” from Gaussian-correlated surfaces.

6 Project B: The Trojan asteroids



Accompanying the planet Jupiter in its orbit around the Sun are two groups of asteroids, named individually after the main protagonists in the Trojan wars. The two groups are constrained at points of stable equilibrium (the Lagrange points) preceding and trailing the planet at an angular distance of $\pi/3$ radians. The combination of the gravitational attraction of the Sun and Jupiter gives a resultant force on a Trojan asteroid towards the centre of mass of the two bodies such that the centripetal acceleration causes the asteroid to orbit with the same period as the planet.

Write a program to solve the equations of motion for this system numerically. Remember to test your program: one test is to demonstrate that the asteroids will stay fixed at the Lagrange points over many orbits of Jupiter around the Sun.

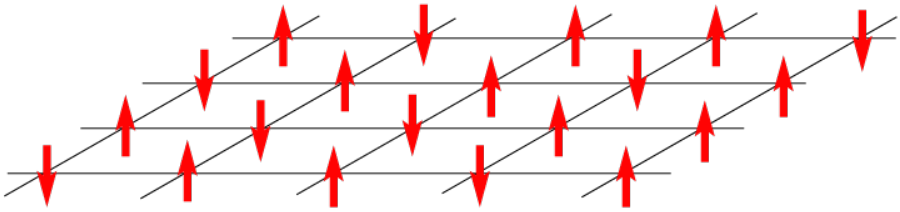
Vary the initial positions of the asteroids "slightly" to investigate the stability of the equilibrium positions. Plot the positions of the asteroids through a few hundred orbits, and show that they oscillate about the Lagrange points, but do not escape from the stable position.

Derive quantitative measures of how far the orbits wander from the Lagrange points. Explore how the wander of the orbits varies with respect to the initial conditions in both displacement and velocity — you could try selecting random points in the initial (position, velocity) phase space and see how they evolve. Use the results from a large number of initial conditions to develop quantitative relationships where possible. Explore the effects of evolution over many orbits — do initially stable orbits become unstable?

Trojan asteroids are also present in the orbit of Mars. Run your program again for a range of other planetary masses, within one or two orders of magnitude of that of Jupiter, and derive a relationship linking the range of wander and the planetary mass.

Suitably scaled units for this problem are solar system units, where the mass of the Sun ($M_{\odot}$) is taken as unit mass, unit distance is the astronomical unit (AU) and unit time is one year. For such a system the gravitational constant G is $4\pi^2$. The mass of Jupiter can be taken to be $0.001\,M_\odot$ and the average distance of Jupiter from the Sun is $5.2\ \mathrm{AU}$. The masses of the asteroids can be regarded as negligible in this system.

7 Project C: The Ising Model of a Ferromagnet



In this project you will use the Metropolis algorithm to investigate ferromagnetism using a simplified model of the spin-spin interactions known as the Ising model.

The basics of the Ising model were covered in lectures. We treat the magnet as a set of spins on fixed lattice sites in two dimensions. Suppose we have an N by N lattices, for a total of N^2 sites. The energy of the system is written

$$E = -J \sum_{<ij>} s_i s_j - \mu H \sum_{i=1}^{N^2} s_i$$

where the sum $< ij >$ is over nearest neighbour pairs of atoms only. In 2-D, there are thus 4 nearest neighbour interactions. J is the exchange energy, μ the magnetic moment, and H the field. It is easiest to use *periodic boundary conditions*, so that every spin has 4 neighbours.

The basic algorithm was described in lectures: starting from some initial set of spin orientations, perhaps a random or uniform set, the model is evolved in time by a Monte-Carlo technique: pick a spin, and calculate the energy ΔE required to flip it. If $\Delta E < 0$, flip the spin; if $\exp(-\Delta E/(k_B T)) > p$, where p is a random number drawn from a uniform distribution in the range $[0, 1]$, then flip the spin; else do nothing.

We now repeat this step for many lattice sites. One can either step systematically through the lattice doing N^2 possible flips, or pick N^2 random points. You can think of these N^2 operations together as a single *time step*.

Now repeat many steps to evolve the spin system, sampling relevant quantities and averaging them. The physical goal is to measure the thermodynamic and magnetic properties of the system as a function of time and as a function of the temperature and magnetic field.

You will find it easiest to measure T in units of J/k_B (where J is the exchange energy). Use your model to investigate, as quantitatively and accurately as possible, the physical properties of the system.

The following are the **minimum** set of properties to investigate:

1. Set $H = 0$. Try to quantify how long (i.e. number of "steps") it takes to get from a given starting state, either completely random or all spins in the same direction, to a state which appears to be in equilibrium. Repeat this for a range of different temperatures T.
2. Quantify how the total magnetisation M fluctuates with "time" when the system is in equilibrium. The *autocovariance* of the magnetisation $M(t)$ for a time lag τ is given by
$$A(\tau) = \langle M'(t)M'(t+\tau)\rangle\,,$$
where $\langle\rangle$ denotes averaging over a "long" time and M' is the deviation from the mean $M' = M - \langle M\rangle$. The *autocorrelation* is given by $a(\tau) = A(\tau)/A(0)$. Determine the time lag τ_e over which the autcorrelation falls to $1/e$ at different temperatures, especially near the critical temperature T_c. Do this for different values of the lattice size N, and consider how this impacts the averaging time needed to get accurate values in the other investigations.
3. Determine how the mean magnetisation $\langle M\rangle$ varies with temperature (it is acceptable to determine the properties of $\langle |M|\rangle$ instead.)
4. Measure the heat capacity C of the system as a function of temperature. The fluctuation-dissipation theorem that says that $C = \sigma_E^2/(k_B T^2)$ where σ_E is the standard deviation of fluctuations in the system energy. Derive a critical temperature T_C for different values of N. Compare your value of T_C with Onsager's analytical result for an infinite lattice: $T_c = 2/\ln(1+\sqrt{2})$.
5. Investigate "finite-size scaling", which suggests that the critical temperature $T_C(N)$ for a lattice size of size N varies as
$$T_C(N) = T_C(\infty) + aN^{-1/v}$$
where a and v are constants. Derive estimates of $T_C(\infty)$ from your data and compare them to Onsager's result, remembering that you need to quantify your uncertainties.
6. Investigate what happens when $H \neq 0$: in particular, examine hysteresis effects when H is cycled at different temperatures.

8 Project D: Tidal Tails and Saturn's Rings

The aim of the problem is to develop a simple *N*-body simulation of interacting galaxies, and optionally develop it to investigate Saturn's rings. The problem will be simplified by considering two heavy masses and a number of light test masses. You should approach the problem in two stages.

Stage 1: creation of a single model galaxy

The model galaxy consists of a central heavy mass and five rings of test particles. Consider carefully the scaling of the problem. One possibility is to use a set of units such that the central mass has a mass of 1 unit and G=1. Arrange the test particles in rings about the central mass at radii 2, 3, 4, 5 and 6 units. An appropriate density of particles is 12, 18, 24, 30 and 36 on the five rings respectively. Since they are to be treated as test particles the only force felt by each particle is from the central mass alone. Determine appropriate velocities for the particles and set up an N-body simulation so that the particles correctly remain on circular orbits.

Stage 2: interaction

Stage 2 involves the introduction of a second "heavy" mass. This represents a perturbing galaxy with similar mass interacting gravitationally with the first. Determine the initial conditions of this second galaxy so that it will pass, approximately, a distance of 9 to 12 units from the first central mass at closest approach; the orbit should be chosen to be parabolic in the first instance (i.e. $E_{\text{total}} = 0$).

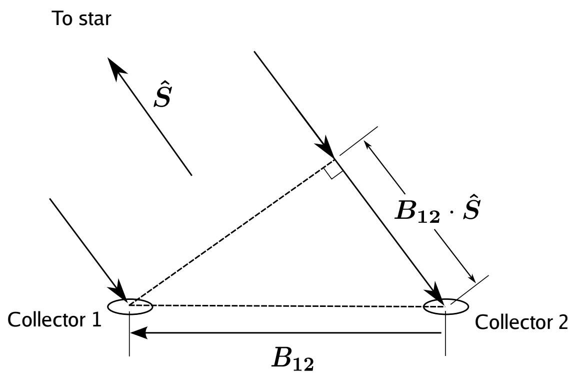
Test the motions of the two masses without the test masses. Re-introduce the test masses which now see a force due to the two “heavy” masses. Follow the evolution of the system by plotting the positions of all of the masses as a function of time (experiment with choosing the most appropriate timescale on which to plot snapshots of the system).

Use this model to explore the effect of reversing the direction of approach from clockwise to anticlockwise on the formation of tidal tails.

At this stage you can explore one of two problems:

1. Explore tidal tails in more depth, including the effects of varying the mass of the disrupting galaxy and the distance of closest approach. Try to derive as many quantitative results as possible, for example counting the fraction of test masses in different initial orbits which are disrupted in one sense or another. Increasing the number of test masses (by several orders of magnitude) and adding orbits at intermediate radii will help to increase the accuracy of these results.
2. Use the framework developed so far to study the dynamics of Saturn's rings, with the test masses representing light “boulders” in the rings. You are likely to need to have a large number of boulders in a large number of circular orbits to adequately represent the rings. Include the gravitational effects of Saturn itself (mass 5.7×10^{26} kg) when setting up the initial orbits and then add one or more moons: you can start by looking at the gravitational effect of the moon Mimas, assuming a circular orbit of radius 185,000 km and a mass of 3.7×10^{19} kg.
   Many of the features seen in Saturn's rings are thought to arise in part from the effects of collisions between the boulders. These effects are time-consuming to model accurately in this framework, so you can start by assuming collisionless test particles and see which effects are still observable. You could then try to approximate the effects of collisions by adding some kind of “viscosity” to the system, for example by exchanging a chosen (small) fraction of their momentum between test particles which pass within some radius of one another. Note that some of these phenomena in reality take many thousands of orbital cycles to show an appreciable effect. Try to develop quantitative measures of any qualitative phenomena you observe.

9 Project E: Surveying using stars



In this project you will analyse data from an optical interferometer to determine the 3-dimensional locations of the telescopes. The data are a complete year (2012) of optical path delay observations from the CHARA interferometer in California (see `http://www.chara.gsu.edu` for a general description). The interferometer observes the interference between light arriving at different telescopes, separated by distances of up to 330 metres, in order to make images of stars at very high angular resolution.

To generate interference fringes, the optical path delays from the star to the point of interference must be matched to a few microns. The optical path difference between two telescopes is given by $\hat{\mathsf{S}} \cdot (\mathbf{x}_1 - \mathbf{x}_2) + d_1 - d_2$ where $\hat{\mathsf{S}}$ is the unit vector pointing towards the star, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the vector locations of the telescopes with respect to some origin and d_1 and d_2 are internal delays inside the interferometer associated with the optical paths from telescopes 1 and 2 respectively to the point of interference. Note that the vector between two telescopes is known as the "baseline" $\mathbf{B}_{12}$.

The "internal" delays consist of (a) a continuously-variable optical path length equaliser (OPLE) whose path delay is measured in real time by a laser interferometer to sub-micron accuracy (b) a set of discrete "Pipes of Pan" (POP) delays which can be switched in to access different parts of the sky (c) an quasi-fixed component dependent on optics in the beam path from each telescope to the beam combination point (d) unknown atmospheric optical path delays above each telescope, which vary on timescales of less than a second but have a maximum amplitude of order 10–100 microns.

The aim of this project is to determine the positions of the telescopes as precisely as possible from a sequence of delay measurements, made when the total delays have been equalised to such an extent that interference fringes have been observed.

The data for this project are on the MCS system in the directory `/ux/PHYSICS/PART_2/data`. There are two CSV (comma-separated-variable) files, both in the same format. They contain tables of delay measurements and ancilliary data. The file `2012_all.csv` contains all the data for all pairs of telescopes, while `2012_S1W2.csv` contains the data for only one pair of telescopes, S1 and W2. In addition, the delays in `2012_S1W2.csv` have been adjusted to include estimates for the "POP" delays.

Each row in the table represents the delay measured on a given star with a given pair of telescopes at a given time of night. The columns are:

bas1 : a baseline identifier. In the case of the file `2012_S1W2.csv` the identifier is always 'D/S1W2' but in the case of `2012_all.csv`, the identifier is of the form 'D/S1E2P14B23', where S1 and E2 are the two telescopes involved and P14B23 identifies a particular POP delay setup. Each POP setup can be assumed to introduce a different delay.

h,m,s : the hour minute and second UTC (Universal Coordinated Time, the successor to Greenwich Mean Time) when the observation was made.

star : identifier for the star. More about the particular star can be found in the catalogue `starlist.gz` in the same directory. This is a large compressed file which can be uncompressed using the `gunzip` program (or use `gzcat` to pipe it into a program like `less`).

az,el : The azimuth and elevation of the star, in degrees, at the instant the delay was measured. Azimuth and elevation are angular coordinates defined relative to the local north and local horizontal plane. See `https://en.wikipedia.org/wiki/Horizontal_coordinate_system` for more information.

delay1,delay2 : delays measured in metres on the two OPLEs (one per telescope) at the same instant as fringes were observed.

You can assume that the delay introduced by any given POP setting is unknown but stable and that the delay from the telescope to the beam combination point (item (c) above) varies from night to night. You can tell when it is a new night by the fact that the hour stops increasing (note that observations are at night and that sunrise in California is usually before midnight UTC). Alternatively you can use the star catalogue in `starlist.gz`, together with the time and the azimuth and elevation to directly determine the day of the year; the time a given star is at any given azimuth and elevation on any given night changes by about 4 minutes every night, due to the difference between sidereal time and solar time.

You can start by solving for the relative positions of the S1 and W2 telescopes using the `2012_S1W2.csv` file. You can check your solution against the solution in Table 1. Note that the POP delay corrections in the `2012_S1W2.csv` file are based on estimates with unknown levels of systematic error, and so should not be relied on in the analysis of the `2012_all.csv` file.

Now analyse the data from the `2012_all.csv` file. First analyse the whole year, under the assumption that the telescope locations and POP delays are fixed over the year. Construct a "design matrix" for the complete year of data. Note that it is normally a good idea to use the telescope positions rather than the baseline vectors

coordinate	value (μm)
S1.X	+0.000
S1.Y	+0.000
S1.Z	+0.000
W2.X	-69091868.860
W2.Y	+199335700.190
W2.Z	+460698.107

Table 1: Nominal telescope position solution for the data in the `2012_S1W2.csv` file. The Z coordinate is vertical, Y is North and X is East. Note that the S1 telescope has been set to be the origin of coordinates.

as model parameters, to take advantage of the redundancies arising from the fact there are 15 possible baseline vectors but we need only 5 independent vectors to represent the relative telescope locations. Look for any degeneracies in the design matrix, for example by using Singular Value Decomposition (SVD) and looking for particularly small singular values. The corresponding column of the $\mathbf{V}$ matrix from SVD can be used to understand the linear combination of model parameters which are involved in the degeneracy (see "Numerical Recipes" for more information on how to interpret and deal with singular values which are close to zero). Use this to interpret the form of the degeneracies and whether/how they affect the model parameters we are interested in, namely the telescope locations. Derive a solution for the telescope locations.

Look for outliers in the data, and evidence for whether there are systematic errors in any data e.g. the positions of some of the stars, which can affect the fit. Make sure to present visualisations of the data which back up your conclusions and to present uncertainties (error bars) with any quantitative results.

Now break the data into non-overlapping subsets in time, each perhaps a month or two in length. Use independent fits to the subsets to see if there is any evidence for changes during the year in the locations of any of the telescopes (California is a seismically-active zone!).