ASTR415: Problem Set #3

Curran D. Muhlberger
University of Maryland
(Dated: March 26, 2007)

Non-linear root finding and model fitting are two procedures that are essential to modern numerical sciences. Due to their non-linearity, iterative techniques are necessary in both cases. Fortunately, for 1-dimensional root finding, the algorithms are simple to implement, converge rapidly, and, by falling back to bisection, are robust for continuous functions. Non-linear least-squares model fitting is far less straight-forward to implement from scratch, but existing software packages, such as the routines in *Numerical Recipes* ([1]), are general enough to be used in most situations.

In this problem set, root-finding techniques are used to locate the five Lagrange points of a restricted 3-body system given the magnitudes of the two masses and their separation. The gravitational potential and vector field are also calculated on a grid for the purposes of plotting. Finally, Lorentzian and Gaussian models are fit to data representing line strength at various frequencies, and the Gaussian was found to be a significantly better fit with a chi-square probability of $Q(\chi^2|\nu) = 0.65$.

I. LAGRANGE POINTS

In the restricted 3-body problem, two masses m_1 and m_2 separated by a distance d orbit their center of mass. If a third body of negligible mass is introduced to the system, there are five equilibrium locations where the body will rotate the center of mass at the same frequency as the two massive bodies. These locations are known as Lagrange points. In a reference frame that co-rotates with the two massive bodies, the Lagrange points occur where the effective gravitational acceleration is zero. Here, the gravitational potential will exhibit either a maximum, a minimum, or a saddle point.

The two massive bodies will rotate their center of mass at an angular frequency $\Omega = \sqrt{G(m_1+m_2)/d^3}$. The effective potential in the rotating frame is given by

$$\Phi = -G\left(\frac{m_1}{r_1} + \frac{m_2}{r_2}\right) - \frac{1}{2}\Omega^2 r^2 \tag{1}$$

where r_1 is the distance from the first body, r_2 is the distance from the second body, and r is the distance from the center of gravity.

In Cartesian coordinates, the effective gravitational acceleration is given by the vector field

$$g_x = \Omega^2 x - Gm_1\frac{x-x_1}{r_1^3} - Gm_2\frac{x-x_2}{r_2^3} \tag{2}$$

$$g_y = y\left(\Omega^2 - \frac{Gm_1}{r_1^3} - \frac{Gm_2}{r_2^3}\right) \tag{3}$$

The class `LagrangePoints` represents such a restricted 3-body system in a rotating frame. Given the magnitudes of the two masses and their separation distance, an instance of this class can be constructed whose methods calculate the gravitational potential and acceleration vector at any point in the frame. By default, the main method prints x, y, Φ, g_x, and g_y for points in a 25×25 grid whose width and height are 4 times the separation of the bodies. Plots of the potential and acceleration field are shown in figure 1 for $m_1 = 3$, $m_2 = 1$, $d = 1$ and figure 2 for $m_1 = 100$. $m_2 = 1$, $d = 1$.

The Lagrange points occur at places where the acceleration vector is zero. If their location can be restricted to a horizontal or vertical line, then a 1-D root finding routine can be used to find their position numerically. A simple and effective root finder is the *secant method*, described in [2]. Given two distinct guesses r_1 and r_2 for a root of f, the location of the root is found iteratively using the relation

$$r_{n+1} = r_n - f(r_n)\frac{r_n - r_{n-1}}{f(r_n) - f(r_{n-1})} \tag{4}$$

The secant method converges with an order of $\phi = (1+\sqrt{5})/2$ and avoids the need to compute the function's derivative (required in Newton's method). Unlike bisection, however, it is not guaranteed to converge and does not keep the root bracketed. In the case of the 2-body gravitational field this is not an issue, and the simplicity of implementing the secant method justifies its use in place of a more robust technique like Brent's method.

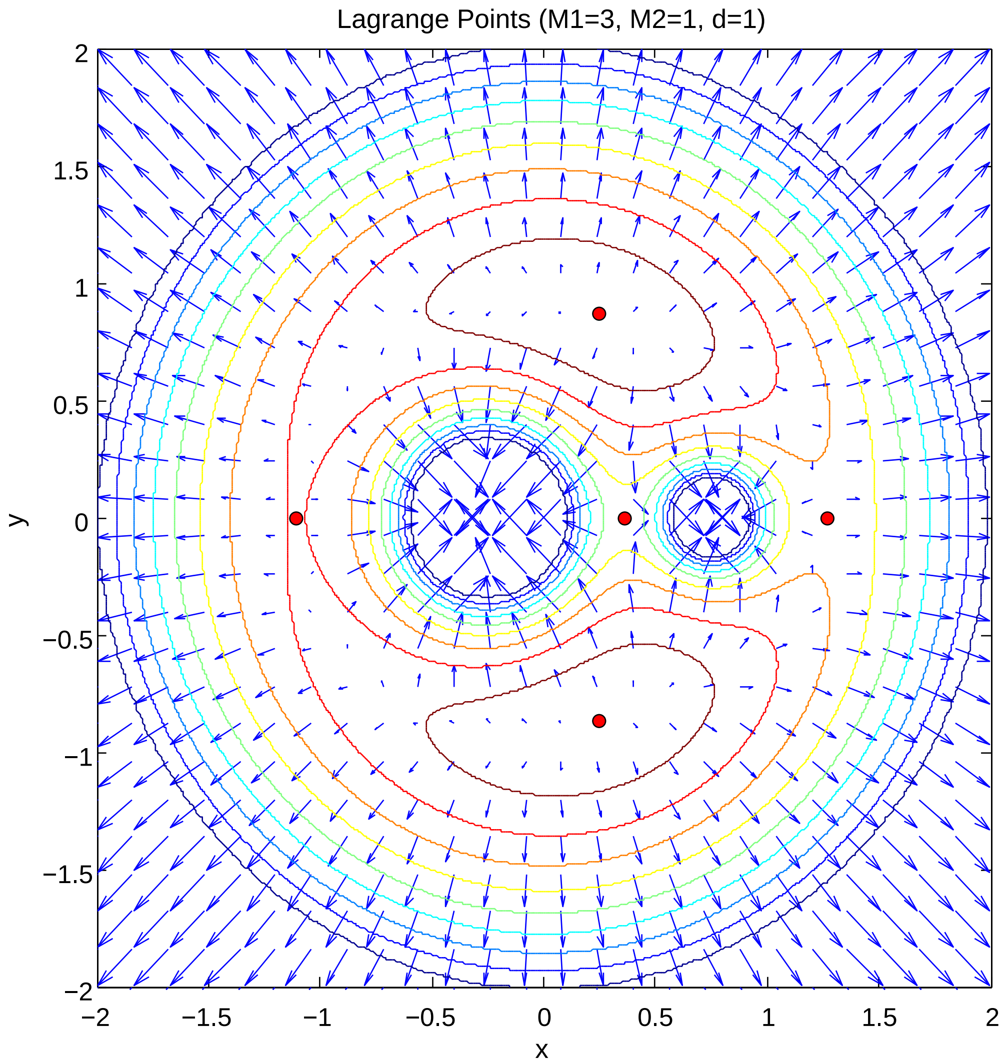

FIG. 1: Contour plot of the effective gravitational potential overlayed with a vector plot of the effective gravitational acceleration. Both quantities have been clipped near their median for better display. The frame is co-rotating with the 2-body system given by $m_1 = 3$, $m_2 = 1$, $d = 1$. The five Lagrange points are shown as red circles.

In `LagrangePoints`, a static method `findRoot` is provided for finding the root of a general 1-D function using the secant method. For L_1, L_2, and L_3, the Lagrange points lie on the x-axis, so roots are found for the function $f(r) = g_x(r, 0)$. The initial guesses used are $0.2d$ and $0.3d$ for L_1, $1.2d$ and $1.3d$ for L_2, and $-1.4d$ and $-1.3d$ for L_3. To locate L_4 and L_5, roots are found for the function $f(r) = g_y((x_1 + x_2)/2, r)$, since these points are known to lie on the equilateral triangles connecting the two masses. The initial guesses used are $\pm 0.5d$ and $\pm 1.0d$. In figures 1 & 2, the Lagrange points are plotted as red circles. Their locations in the rotating frame are given in table I.

II. DATA FITTING

The basic idea behind data fitting is to design a function of the fit parameters that yields a measure of the goodness of the fit. A common example is χ^2. This function is then minimized in parameter space by stepping along the gradient. In order to do this effectively, it is helpful to know the derivatives of the model analytically. Then, routines such as `mrqmin` from [1] can be used to perform the minimization and estimate the uncertainty in the final parameter values.

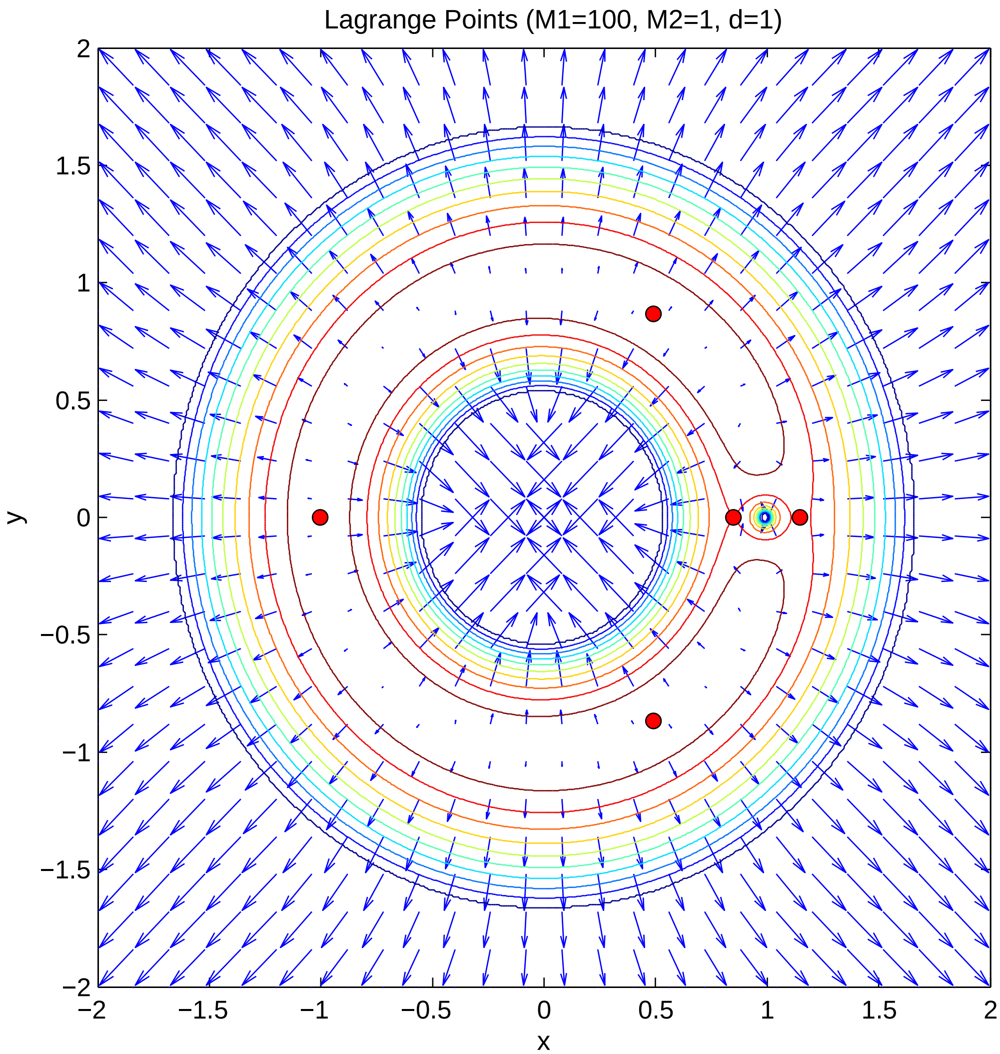

FIG. 2: Contour plot of the effective gravitational potential overlayed with a vector plot of the effective gravitational acceleration. Both quantities have been clipped near their median for better display. The frame is co-rotating with the 2-body system given by $m_1 = 100$, $m_2 = 1$, $d = 1$. The five Lagrange points are shown as red circles.

TABLE I: Locations of the Lagrange points found numerically for the restricted 3-body systems described by $m_1 = 3$, $m_2 = 1$, $d = 1$ and $m_1 = 100$, $m_2 = 1$, $d = 1$.

Lagrange Point	Location for $m_1 = 3$	Location for $m_1 = 100$
L_1	(+0.360743, +0.000000)	(+0.848624, +0.000000)
L_2	(+1.265858, +0.000000)	(+1.146320, +0.000000)
L_3	(-1.103167, +0.000000)	(-1.004125, +0.000000)
L_4	(+0.250000, +0.866025)	(+0.490099, +0.866025)
L_5	(+0.250000, -0.866025)	(+0.490099, -0.866025)

A Lorentzian is given by

$$\phi_L(\nu) = \frac{1}{\pi}\frac{\alpha_L}{(\nu-\nu_0)^2+\alpha_L^2} \tag{5}$$

Its first partial derivatives are given by

$$\frac{\partial \phi_L}{\partial \alpha_L} = \frac{1}{\pi}\frac{(\nu-\nu_0)^2-\alpha_L^2}{((\nu-\nu_0)^2+\alpha_L^2)^2} \tag{6}$$

$$\frac{\partial \phi_L}{\partial \nu_0} = \frac{2\alpha}{\pi}\frac{\nu-\nu_0}{((\nu-\nu_0)^2+\alpha_L^2)^2} \tag{7}$$

A Gaussian, on the other hand, is given by

$$\phi_G(\nu) = \frac{1}{\alpha_D}\sqrt{\frac{\ln(2)}{\pi}}e^{-\ln(2)(\nu-\nu_0)^2/\alpha_D^2} \tag{8}$$

Its first partial derivatives are given by

$$\frac{\partial \phi_G}{\partial \alpha_D} = \frac{1}{\alpha_D^2}\sqrt{\frac{\ln(2)}{\pi}}\left(\frac{2\ln(2)}{\alpha_D^2}(\nu-\nu_0)^2-1\right)e^{-\ln(2)(\nu-\nu_0)^2/\alpha_D^2} \tag{9}$$

$$\frac{\partial \phi_G}{\partial \nu_0} = \frac{2\ln(2)}{\alpha_D^3}\sqrt{\frac{\ln(2)}{\pi}}(\nu-\nu_0)e^{-\ln(2)(\nu-\nu_0)^2/\alpha_D^2} \tag{10}$$

The program `datafit` iteratively calls `mrqmin` until χ^2 stabilizes at a minimum. Given a set of data points with errors, it fits a Lorentzian and a Gaussian and reports the best-fit parameters, their uncertainties, the resulting χ^2, and the χ^2 probability value Q. Because `mrqmin` calls a single function for the model and all of its derivatives, this function can be highly optimized by re-using terms common in the above expressions. Unfortunately, this makes the function difficult to read.

When processing the file `ps3.dat` with 100 data points, `datafit` found that the best-fit parameters for a Lorentzian are $\alpha_L = 8.049 \pm 0.014$ and $\nu_0 = 47.502 \pm 0.042$ with $\chi^2 = 2692$. For $\nu = 98$ degrees of freedom, $Q(\chi^2|\nu) = 0.000$, so the probability that truly Lorentzian data with normal errors deviates to the extent that this data does from the model is effectively zero. In other words, a Lorentzian model can be disproved for this data set. The Lorentzian fit is plotted with the data in figure 3.

When fitting the Gaussian model, `datafit` found the best-fit parameters to be $\alpha_D = 12.013 \pm 0.017$ and $\nu_0 = 47.978 \pm 0.064$ with $\chi^2 = 92$. Here, $Q(\chi^2|\nu) = 0.648$. Therefore, deviations from the model at least as large as those exhibited by this data are expected 69% of the time. A Gaussian model therefore cannot be disproved as the true source of this data and is in fact quite consistent with it. The Gaussian fit is plotted with the data in figure 4.

---

[1] Press, William H., Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical Recipes in FORTRAN 77: The Art of Scientific Computing (Volume 1 of Fortran Numerical Recipes).* Second Edition. Cambridge University Press, 2001.

[2] Atkinson, Kendall E. *An Introduction to Numerical Analysis.* Second Edition. Wiley, 1989.

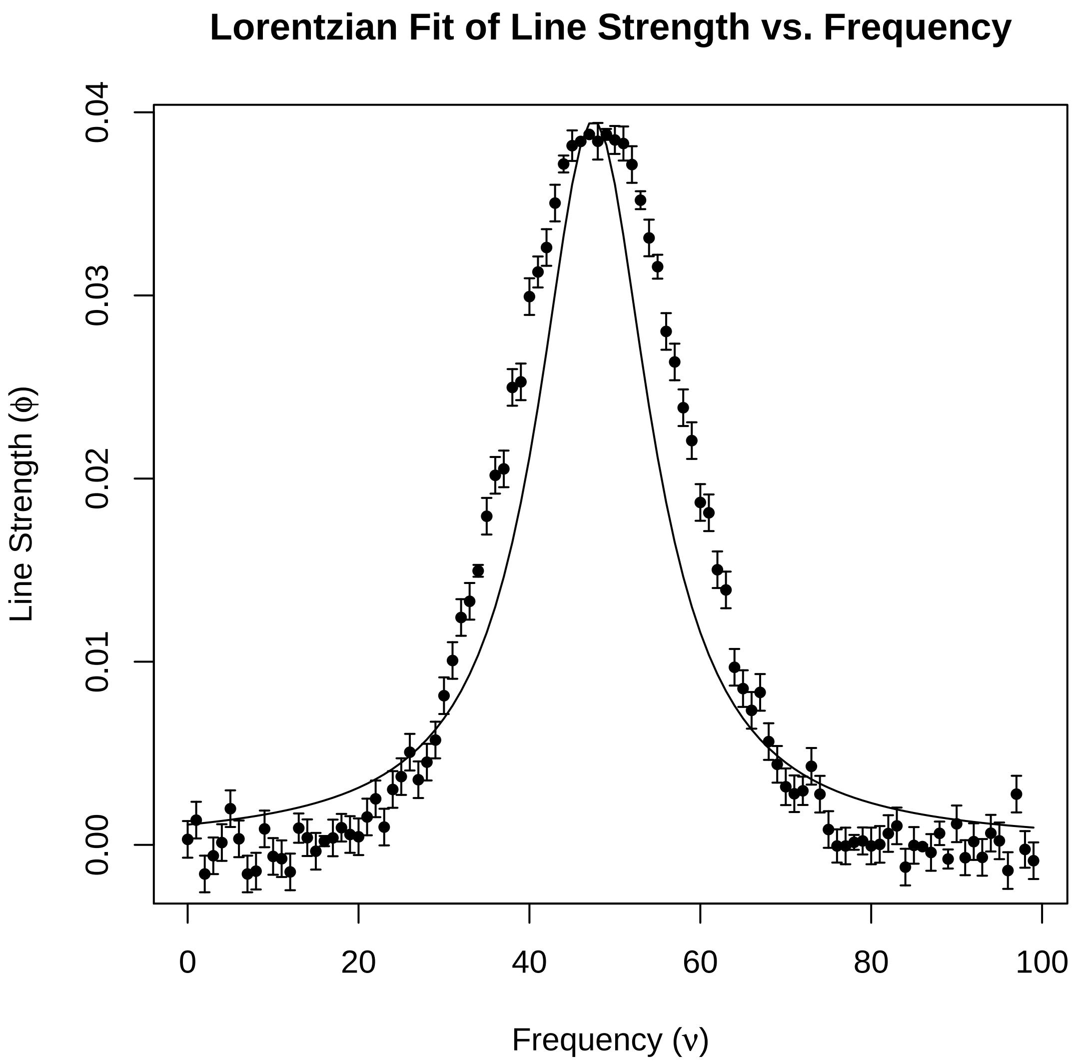

FIG. 3: Lorentzian fit to the data in `ps3.dat`. Parameters: $\alpha_L = 8.05$, $\nu_0 = 47.50$.

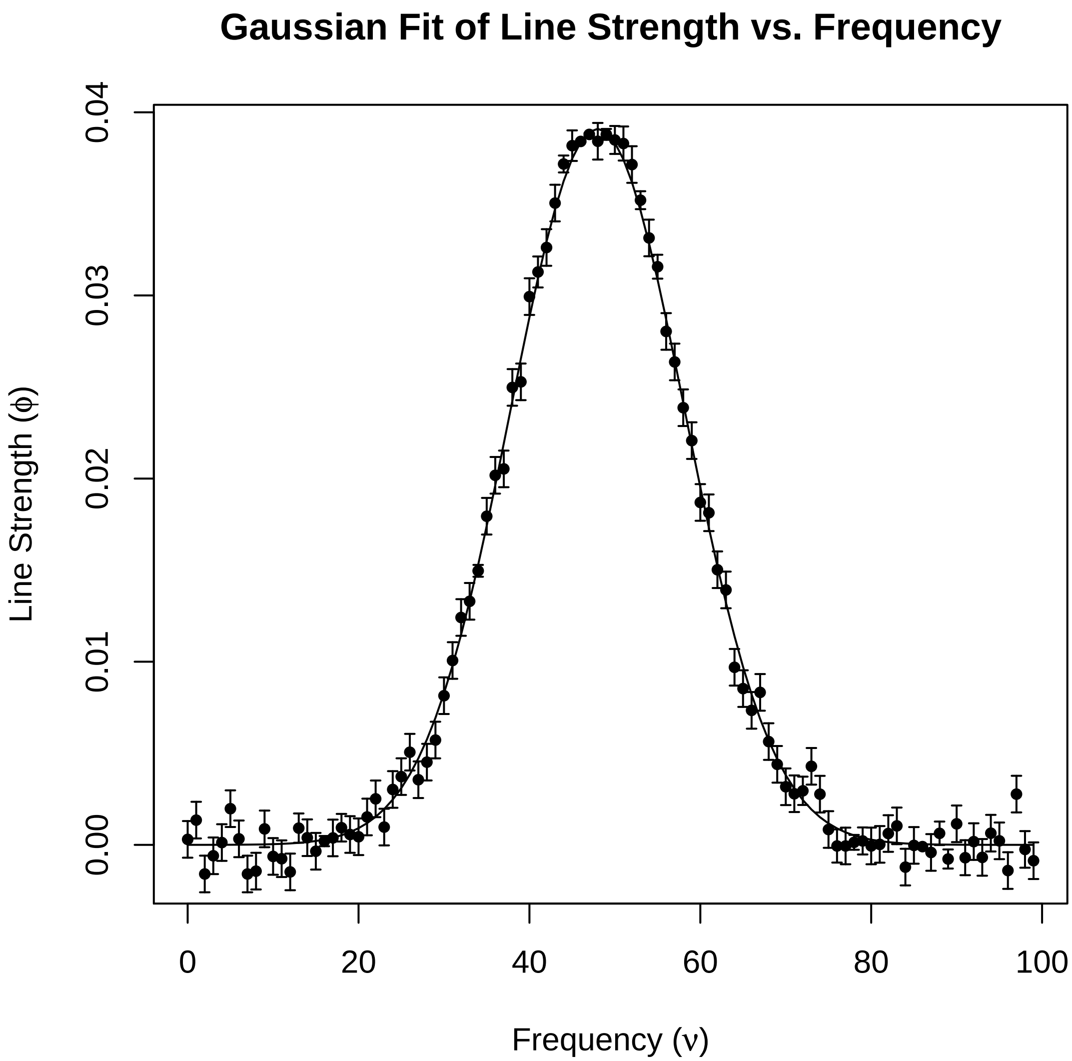

FIG. 4: Gaussian fit to the data in `ps3.dat`. Parameters: $\alpha_D = 12.01$, $\nu_0 = 47.98$.