[1] Emcee: The MCMC Hammer
Alan Elliott, May 18, 2025.

[2] The package I am using is Emcee. Emcee is a python implementation of the Goodman and Weare's Monte Carlo Markov Chain sampler. Some examples of problems that Emcee can solve are numerical integration, estimating uncertainties given a set of data, and sampling from a distribution. Emcee uses a "walker" to draw it's samples from multi-dimensional spaces.

[3] I selected Emcee because it was suggested as a possible final project in class, and because I found Monte Carlo simulations interesting.

[4, 5] The version of emcee I am using is version 3.1.6. This could be accessed by running print(emcee.__version__). Emcee was originally released in 2012, and has been updated numerous times since. The latest update was version 3.1.6, which was released in April, 2024. Given this, it seems that Emcee is still being maintained. The package is being maintained by at least some of the original authors; Dan Foreman-Mackey is listed as an author on the original paper and has been listed as a contributor on most of the updates. However, I didn't see the other authors listed as contributors to recent updates, and there were numerous non-authors who have been listed as contributors to recent updates too. I was unable to find instructions on how to contribute to the project, though clearly it can happen as there are many contributors not amongst the original authors. A couple other programs that solve similar problems are PyMC, and mcmc-python.

[6, 7] Emcee was very easy to install and use; to install it, all that was needed was to run !pip install emcee, and the only additional library needed was Numpy. Installation was relatively quick, taking less than 5 minutes.

[8] The source code is accessible via github at https://github.com/dfm/emcee.

[9] Yes, Emcee is used in some other codes. One example I found was a code called CosmoHammer, which says that it embeds Emcee into its python framework. This paper is available at https://ui.adsabs.harvard.edu/abs/2013A%26C.....2...27A/abstract.

[10] The code can be used via a jupyterlab notebook.

[11] See the Jupyterlab Notebook accompanying this write-up for the codes. Within the notebook there is a test code, which was used initially to make sure Emcee ran. This test code was taken directly from the Emcee readthedocs main page. Also included are two examples of me using it, one to sample a multi-dimensional Gaussian Density function, and one to estimate the true uncertainties on a set of data if we discover that we had underestimated them. In both examples I

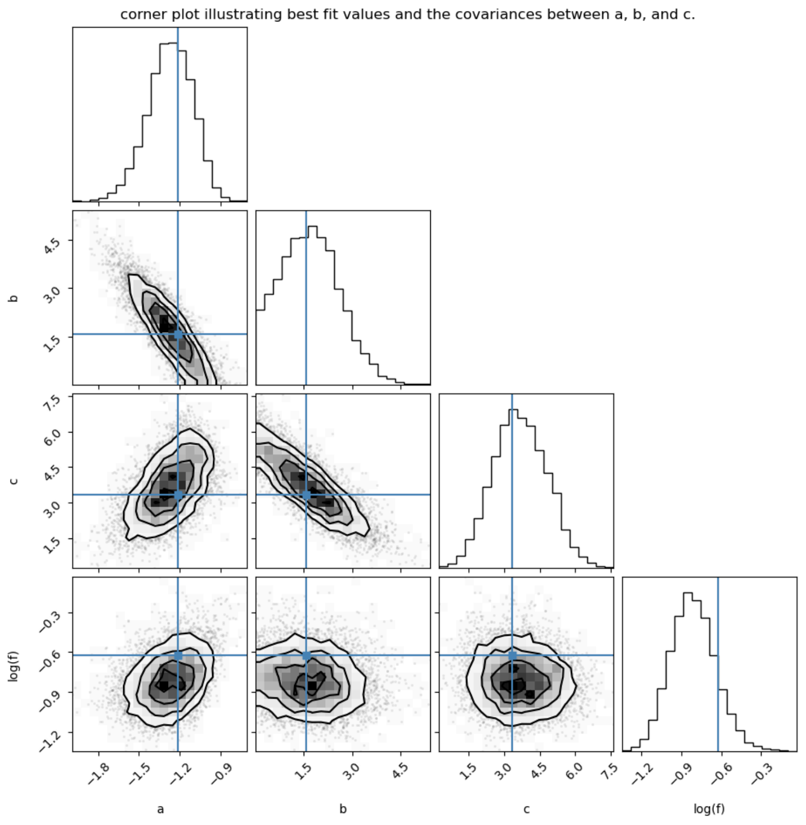
used readthedocs pages to figure out how to use Emcee, but I made sure to choose different numbers, and switched up the problems we were trying to solve (for example, the readthedocs page determined the uncertainties on a linear fit, while my code determined it for a quadratic fit).

[12] Emcee does not generate its own plots. It suggests that you can use matplotlib, or a Corner plot. Corner is a module which allows for easy plotting of best fit values and covariances found by Emcee. In my code, I used both, though the figure seen in this report is from Cormer.

[13] Below is a Corner plot obtained from the second process mentioned in [11]. First, a set of data points was fit to y = ax^2 + bx + c. Then, after realizing that the reported uncertainties were too small, Emcee was used to estimate true errors. Then, Corner took the best fit values and the Emcee-generated errors and covariances, and put them into a compact plot.

Within the plot, the blue lines and blue squares show the best fit values for a parameter. The histograms along the long diagonal show the probability density function for each individual variable, while the rest of the plots show covariances between corresponding variables labelled along the x and y axes. a, b, and c are the fitting parameters, while log(f) is a parameter related to the fraction we underestimated the true uncertainties.

So, we can see that in this dataset, a and b, as well as b and c have high covariances, a and c have a more moderate covariance, and all the parameters have a low covariance with log(f).



[14] The package itself is purely python. However, it requires Numpy, which does pass off a lot of its work to C. However all the user interactions with both Emcee and Numpy are via Python.

[15] In the ensemble sampler portion of Emcee (the primary function of the package), you need to input a few parameters. Some of these parameters are “nwalkers” for the number of “walkers”, ndim for the number of dimensions, an initial state for the walkers to start in, the number of steps for the walkers to take, and a user-defined probability function. This is all you need to run the package.

[16] The output from Emcee is an array of state vectors for the system, an array describing the probability distribution, and an array describing the random state of the system.

[17, 18] The readthedocs page for emcee provided a test code you could run to make sure emcee was working, which is isolated from any other parts of code you may write in a program. Also, the readthedocs page frequently mentions how some parameters of commands were moved around or renamed following the 3.0.0 update for Emcee, and that people updating from older versions need to keep this in mind. This is in a way a form of regression. Also, since Emcee uses a Monte Carlo Method, its results have some inherent randomness to them. Thus, to standardize your results and ensure reproducibility, you can set a seed for the random number generator.

[19] Emcee requires Numpy to use. I found this out through the Emcee quickstart tutorial (available at https://emcee.readthedocs.io/en/v3.1.6/tutorials/quickstart/#quickstart), which states that you need to import numpy. Also, it doesn’t generate figures itself, so Matplotlib or Corner are suggested to generate figures.

[20] The documentation on Emcee was robust. The help(emcee.EnsembleSampler) command provided some helpful details on the different parameters of the function, but the bigger help was the readthedocs pages on emcee. There were numerous pages up which provided examples of use of Emcee and detailed explanations on how to use it’s functions and solve some problems with it.

[21] Yes, they give a preferred citation method, which is to site their paper titled “Emcee: The MCMC Hammer”, available at https://ui.adsabs.harvard.edu/abs/2013PASP..125..306F/abstract.

[22] Resources used for this project:
Readthedocs home page for Emcee: https://emcee.readthedocs.io/en/v3.1.3/
Corner readthedocs page: https://corner.readthedocs.io/en/latest/pages/quickstart/
Emcee quickstart tutorial: https://emcee.readthedocs.io/en/v3.1.6/tutorials/quickstart/#quickstart
Emcee fitting and uncertainty estimation: https://emcee.readthedocs.io/en/v3.1.6/tutorials/line/

Emcee source code: https://github.com/dfm/emcee
Paper introducing Emcee: https://ui.adsabs.harvard.edu/abs/2013PASP..125..306F/abstract
PyMC https://www.pymc.io/welcome.html
Mcmc-python https://github.com/pmocz/mcmc-python

Two papers citing Emcee:
https://ui.adsabs.harvard.edu/abs/2025NewA..11802374N/abstract
https://ui.adsabs.harvard.edu/abs/2025NewA..11902390P/abstract

AstroHammer: https://ui.adsabs.harvard.edu/abs/2013A%26C.....2...27A/abstract

[23] This package has been used extensively in literature since its release, with the ADS page showing over 10000 citations of emcee. A couple recent papers citing Emcee are:
Transit timing variations of the sub-Saturn exoplanet HAT-P-12b (available at https://ui.adsabs.harvard.edu/abs/2025NewA..11902390P/abstract), and
Studying orbital period variations of XY Leo through updated eclipse times and multi-model analysis (available at https://ui.adsabs.harvard.edu/abs/2025NewA..11802374N/abstract).

[24] For the most part, this class was good preparation for this package. I had to learn how Emcee's functions worked, and I had to use a couple other new functions (ex. Np.triu and np.vander), but for the most part the general methods used were pretty similar to stuff we did in class.

[25] I have no prior experience with emcee, and I did not work in a group on this project.