# ASTR4004/ASTR8004
# Astronomical Computing
# Assignment 2

Christoph Federrath

**due Thursday, August 22, 2024**

## 1 Plotting and fitting with gnuplot

1. Plot the turbulent density PDF data file from `http://www.mso.anu.edu.au/~chfeder/teaching/astr_4004_8004/material/mM4_10048_pdfs/EXTREME_hdf5_plt_cnt_0050_dens.pdf_ln_data` using column 1 (as $x$-axis) and column 3 (as $y$-axis). You can use the gnuplot template from `http://www.mso.anu.edu.au/~chfeder/teaching/astr_4004_8004/material/gnuplot.p`. Now add a Gaussian fit. To make the fit, use the gnuplot 'fit' command with the Gaussian model function,

$$f(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left[-\frac{(x-x_0)^2}{2\sigma_x^2}\right]. \tag{1}$$

2. Generate a script that does the fit and plots it on top of the data from $x_{\mathrm{min}} = -10$ to $x_{\mathrm{max}} = 10$. Plot the data as crosses and the Gaussian fit as a solid thin black line. Annotate the plot nicely (axis labels) and change the key (legend) text to give a reasonable description of what is plotted. Note that the data in columns 1 and 3 (which you should plot and fit) represent a probability distribution function (PDF) of the log-normalised gas density $s \equiv \ln(\rho/\rho_0)$ in a simulation of driven supersonic turbulence. Also change the key position, such that it is in the top right corner of the plot frame. Finally, let the script write out a postscript (.eps) file with the finished plot.

(25%)

## 2 Plotting multiple datasets and data manipulation with gnuplot

Here we apply data manipulation within gnuplot. Much more advanced methods are possible, but require some reading through the documentation and searching on the internet.

1. Make a copy of the previous script (without the fit), but now instead of plotting only one data file, plot the times/files with (0020, 0030, 0040) contained in the

tarball http://www.mso.anu.edu.au/~chfeder/teaching/astr_4004_8004/material/mM4_10048_pdfs/EXTREME_pdfs.tar.gz, all in one plot using different line styles or plot symbols and colours (so we can easily distinguish the three data sets from one another). Plot the data on a logarithmic $y$-axis from $y_{\mathrm{min}} = 10^{-5}$ to $y_{\mathrm{max}} = 2$. Use the $x$-axis range as in Section 1. Now shift the 0030 data up by a factor of 2 and the 0040 data up by a factor of 4. This should offset the curves, so they can be more easily distinguished.

2. Now make another copy of the script from Section 1 and plot $x$ (column 1) versus $\exp(x) \times y$ (the exponential of column 1 times column 3) as the new ordinate of the 0050 data file from above. This will generate a mass-weighted (or density-weighted) PDF ($P_M$) instead of the previous volume-weighted PDF ($P_V$). Note that mass-weighted and volume-weighted PDFs are always related by $P_M = \rho P_V$ with the density $\rho = M/V$. To see this, consider that we have $s \propto \ln(\rho)$ and thus $\exp(s) \propto \rho$, hence the multiplication with $\exp(x)$ to obtain the mass-weighted PDF. In order to do this, you have to gain access to the data in column 1 of the data file and exponentiate it via '`(exp($1))`' within the gnuplot `using` construction.

3. Also fit the new mass-weighted PDF as in Section 1 and let the script print the mean and standard deviation of the fitted data to the gnuplot shell (the fitted mean should now be positive and has a value of 0.71 and the fitted standard deviation is 1.14).

(25%)

# 3 The stellar initial mass function

Take the functional form of the Initial Mass Function (IMF) for the (relative) number of newborn stars $dN$ in stellar mass bins $d\log_{10} M$ as a function of stellar mass $M$ by Chabrier (2005),

$$dN/d\log_{10} M = \begin{cases} 0.093 \exp\left[-\frac{(\log_{10} M - \log_{10} 0.2)^2}{2\times(0.55)^2}\right] & \text{for} \quad M \leq 1 \\ 0.041\, M^{-1.35} & \text{for} \quad M > 1, \end{cases} \tag{2}$$

where $M$ is in units of $\mathrm{M}_\odot$ throughout.

1. From this equation, derive the form for $dN/dM$ and define $\mathrm{IMF}(M) = dN/dM$.

2. Write a python script that discretises $\mathrm{IMF}(M)$ on a logarithmically-scaled grid of mass $M$ from $M_{\mathrm{min}} = 10^{-2}$ to $M_{\mathrm{max}} = 10^2$. Logarithmically-scaled means that the grid of $M$ is uniformly spaced in bins of $\log_{10} M$, which will then be log-spaced in $M$ (i.e., the bin width is linearly increasing with increasing $M$). Use a reasonable number of bins (sampling points) that captures the functional form well. In fact, make the number of bins a parameter or keyword for your script, so you can easily test with different numbers of bins. Treat the bins defined this way as the bin edges throughout.

3. Plot your discretised function in a log-log plot and label the axes appropriately.

4. Compute the mode (most frequent/probable) mass of IMF($M$).

5. Compute the average mass of stars from IMF($M$) by numerical integration, using a sum over all bins. Implement a *staggered binning approach*, where the average in each bin is approximated as the *arithmetic mean of the $M$ and IMF values at the bin edges*. Note that the integral over IMF($M$) will not be automatically normalised to 1, so do the normalisation first (such that the function becomes a PDF) and then sum over this PDF to compute the average star mass.

6. Test how the resulting average star mass depends on the number of bins (sampling points). Produce a plot of the average $M$ as a function of the number of bins. How many bins are needed to converge on the average star mass to within 1% accuracy?

7. Compute the average mass for $M_{\max} \to \infty$. Approximate this limit numerically by choosing a sufficiently large number for $M_{\max}$, such that the mean mass is converged to at least 2 significant figures.

(50%)

(Total 100%)

**For Sections 1 and 2, please submit gnuplot scripts (not Bash scripts). In your writeup, please describe each of the steps you took for each of the assignment steps above, including the respective command lines. Include the figures as they are after each step. Send 2 gnuplot scripts with the state at the end of Sections 1 and 2 (check that your scripts do not produce any errors or warnings when run, and add comments to each code line/block of the script). For Section 3, submit only 1 python script; make sure that this runs without assuming anything in the python path; instead define all required functions (defs) inside the single python script; also comment, so it is easy to understand what each line of code does; make the output of the script such that it is straightforward to see which answer is for which respective subpart of Section 3.**

**Return of assignments is via Turnitin (see respective assignment link at course Wattle page). To upload the files, please make a tarball named <Uni-ID>.tar.gz that contains all the submission files (2 gnuplot scripts, 1 python script, and 1 pdf file as a writeup of Sections 1 and 2 with the respective images after each step of Sections 1 and 2).**

## References

Chabrier, G. 2005, in Astrophysics and Space Science Library, Vol. 327, The Initial Mass Function 50 Years Later, ed. E. Corbelli, F. Palla, & H. Zinnecker, 41