ASTR4004/ASTR8004
Astronomical Computing
Assignment 3

Christoph Federrath

due Friday, September 20, 2019

1 Reading data in IDL and making map plots

1. Download 3 HDF5 data files of the world's largest simulation of supersonic turbulence:
   EXTREME_proj_xy_000100
   EXTREME_proj_xy_000200
   EXTREME_proj_xy_000300
   These data are column density (2D) projections of the 3D gas density in the simulations. The 3 files are at 3 different times of the evolution. If you are interested in more details on these data and the simulations, please have a look at Federrath et al. (2016) and references therein.

2. Use the IDL program readhdf5data.pro to read in one of the simulation data files. Example code:
   `IDL> readhdf5data, filename, 'dens_proj_xy', coldens`
   Now the content of the dataset '`dens_proj_xy`' is read into the array called '`coldens`'. Use `reform` to strip off the superfluous 3rd dimension and convert the array to `double`.

3. First, make a plot of the 2D map. This can be done with the customised procedure write_eps_wcb.pro provided. Search for the "Main" part of the procedure and inspect the calling sequence based on the source code to work out the minimum amount of data/keywords that needs to be provided to `write_eps_wcb`. Produce a map with a resolution of 512^2 pixels. Add a colour bar (keyword `colorbar`). Use this procedure to make images of the 3 data files, scaled nicely (keyword `range`) and with colour table #5 (keyword `ctab`) and colour bar scaling, such that the contrast in the images comes out reasonably (e.g., use logarithmic scaling of the colour bar; use keyword `imagelog`). Use the additional keywords (e.g., `fontsize`) to control the font size, the tick format and title on the colourbar (keywords `tickformatmap` and `title`) to produce a high-quality plot. Let your script write the images out with file names:
   `EXTREME_proj_xy_000100.eps`,
   `EXTREME_proj_xy_000200.eps`, and
   `EXTREME_proj_xy_000300.eps`.

4. Code this all up such that you use procedures and/or functions in order to be able to call these for the 3 different files (avoid duplication of code!). The best is to define a `pro read_data, filename,...` procedure that reads the data for a given filename, then write another `pro` to make the plot based on which map you provide. Again, do not duplicate code; use functions, procedures, to make the steps in your code modular and applicable to the 3 different input files.

(6 points)

2 Re-binning and beam convolution

1. Let's add some beam smearing in order to simulate the effect of looking at these data through a telescope with a finite beam resolution.

2. Implement a Gaussian beam convolution based on the IDL function `gauss_smooth`. The function that does the convolution should take the original map scaled to a resolution of 512^2 (use the function `congrid` to rescale the resolution from the original to the target 512^2 pixel resolution) as an input. The 2nd input to `gauss_smooth` is the Gaussian standard deviation. Work out the correct standard deviation from the beam full-width-half-maximum (FWHM) parameter, such that the map is smoothed on physical scales of 3% of the physical size (side length) of the original map (in terms of the FWHM).

3. Write out the beam-smeared image of snapshot file number 300 from Section 1 above with `write_eps_wcb.pro` as before (using the exact same colour bar scaling, etc., as above), and visually compare the non-smoothed with the beam-smoothed image. The output file name should be
`EXTREME_proj_xy_000300_Gauss_smoothed.eps`

(3 points)

3 Statistical functions and probability density functions (PDFs)

1. Compute the mean, standard deviation, skewness and kurtosis of the log-normalised column density (intensity) of snapshot number 300 from Section 1 above,

$$\mathcal{I} = \ln\left(I/\langle I\rangle\right), \tag{1}$$

where $\langle I\rangle$ is the mean column density (intensity) and ln() is the natural logarithm (`alog` in IDL). Use the built-in functions `mean, stddev, skewness, kurtosis` and let your script write the results of these 4 statistical moments to the screen.

2. Now write a procedure (again, a new `pro`) to make a histogram and finally a probability density function (PDF) of $\mathcal{I}$. Use the IDL function `histogram` with keywords `locations, min, max, nbins`. Produce a plot of the PDF, labelled with '$\mathcal{I}$' on the x-axis and 'PDF($\mathcal{I}$)' on the y-axis.

3. Write the PDF locations (x-axis values), the histogram (output of `histogram`, i.e., integers), the PDF (normalised histogram, i.e., such that $\int \mathrm{PDF}\, \mathrm{d}\mathcal{I} = 1$), and the cumulative distribution function (CDF) of $\mathcal{I}$ to an ASCII text file (four columns separated with white space and fixed column length of 20 characters per column) named:
`EXTREME_proj_xy_000300_pdf.txt`

4. Write another procedure to read this ASCII file and data columns back into memory. As always and before, use functions and/or procedures to avoid duplication of code.

5. Write a procedure to compute the mean and standard deviation of the PDF data (read from the columns in the file) by summation over the PDF. Let your script write out the first two moments (mean and standard deviation) computed that way, to the screen.

6. Now compare the statistical moments of the PDF computed from the intrinsic functions in point 1 above with those computed from the summation over the PDF. Is there a difference? Why? How can this be fixed? *Write a comment directly into the script to answer this, with no more than 3 sentences.*

(8 points)

4 Averaging data, making plots with error bars, and fitting

1. Produce PDF ASCII files (as in the previous section) of all 3 datasets from Section 1 above.

2. Read them back in, one-by-one, and average all 3 column density PDFs over time to produce a time-averaged PDF with error bars. Be careful to use the same `min` and `max` values and number of bins when you make the histograms, such that they all have the same `locations` and can be directly averaged without having to interpolate.

3. Produce a plot of the averaged PDF, including error bars, where the error bars should be the standard deviation of the variations over the 3 datasets above, computed for each PDF bin. Error bars can be over-plotted with the procedure `oploterror` from the IDL astronomy library. Write the plot out as a file called:
`EXTREME_proj_xy_000300_pdf_averaged.eps`

4. (Optional) Use the MPFIT library to write a procedure to fit the time-averaged PDF with a Gaussian function (making use of the error bars while fitting). Over-plot the fit and report the fitted mean and standard deviation.

(3 points)

5 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 dM as a function of stellar mass M by Chabrier (2005),

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

1. Write an IDL script that defines this function $\mathrm{IMF}(M)$ on a logarithmically-scaled grid of mass M from $M_{\mathrm{min}} = 10^{-2}\,\mathrm{M}_\odot$ to $M_{\mathrm{max}} = 10^{2}\,\mathrm{M}_\odot$. Use a reasonable number of bins (sampling points). In general, what is a reasonable number of bins (sampling points)?

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

3. Compute the mode (most frequent) mass of the distribution.

4. Compute the average mass of stars from this distribution. Note that the integral over Eq. (2) is not normalised to 1, so do the normalisation first and then sum over the normalised version of Eq. (2) to compute the average star mass from this IMF.

5. (Optional) How does the average mass change when M_{max} is increased, say to $1000\,\mathrm{M}_\odot$? Is the average mass finite for $M_{\mathrm{max}} \to \infty$?

(5 points)

(Total 25 points)

IMPORTANT: Please submit IDL scripts (`.pro`). Do not include files already provided (such as the input data maps or `write_eps_wcb.pro`; we already have them and they are provided via the links above). Use indentation in your scripts to separate procedures/functions, loops, conditional sections, etc. Comment the scripts sufficiently, in particular, note that this will also be used as the write-up (for this assignment, you do not have to produce a separate write-up; just the script with comments). Make sure that everything runs out-of-the-box; only tell us the name of the main function/pro call (if not obvious) to start the entire analysis pipeline. Test this on malice.anu.edu.au if you developed your code on a different computer.

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

Federrath, C., Klessen, R. S., Iapichino, L., & Hammer, N. J. 2016, High Performance Computing in Science und Engineering - Garching/Munich 2016, (arXiv:1607.00630)