Computational Methods in Astrophysics

Exercise 1 – Introduction to Fortran 90 and IDL

This exercise will help you to become acquainted with some basic features of Fortran 90 and IDL, in order to be able to solve the problems arising in the various parts of our computational course. Problem 2e) is optional and designed for advanced students already familiar with programming.

Preliminary remarks. To work most efficiently, always have at least two windows open, one to edit the F90-program or IDL-procedure (`emacs` *progname* `&`), and the other one for testing/running the program/procedure. Don't forget to save your modifications to disk (`CTRL x CTRL s` in emacs) and to recompile the code (`ifort ...` or `.run` *procedure*, respectively) before testing them.

Make extensive use of the various help-files (pdf-format) contained in the introductory section of our home page, as well as of the idl help system.

Don't be shy to test various possibilities. The most effective way to learn about a certain device is to play, play and play with it!!! (At least with respect to computers.)

Problem 1 *Capabilities of IDL*

a) To judge the capabilities of IDL, take a bit of time (roughly 15 minutes) and have a look at the IDL-demo tour. This can be started (from within IDL) with the command

`demo_tour`

b) After having got an impression, try to play around a bit. You can have a more decent look into the various possibilities, if you start

`demo`

alone. Try to find the demo showing the polynomial fit. In which section is it contained, and which maximum degree can be fitted within the demo?

c) Open the "About" entry of the demo-menu to find out where the corresponding source-codes are contained within the IDL-directory. Use then the

`locate` command

to find the absolute path to the relevant IDL-directory, which on our system is called *idl_5.5*. Switch to the source-code directory, and try to identify the procedure (with extension *.pro*) containig the polynomial fit demo. What is its name? With a bit of detective work, this should be easy to find out.

Open this procedure, and find the line where the polynomial fit is actually performed. List the corresponding statement.

Problem 2 *IDL and F90 programming*

On our homepage (intro-section), you will find the idl-procedure *gamow.pro*, which can be used to display the "Gamow-peak" as a function of various parameters.

Remember that only the actual presence of this peak allows for nuclear fusion in stellar cores (and shells), and results from the competition of the Maxwellian velocity distribution of the colliding particles

$$p(E_{\rm kin}) \propto \exp(-E_{\rm kin}/kT_{\rm plasma}),$$

and the probability to tunnel the Coulomb wall between two charged particles,

$$p(E_{\rm kin}) \propto \exp(-b/\sqrt{E_{\rm kin}}),$$

where b is defined, e.g., inside the procedure. In particular, *gamow.pro* displays the individual probabilities (without leading factors) and the combined probability, as a function of the particles' kinetic energy in units of $kT_{\rm plasma}$.

a) Study the procedure carefully and try to understand *each* statement, in particular the way how the x-grid is initialized (in a very effective way). Note also the way idl handles the arrays (without explicitly declaring them).
The units of the Boltzmann-constant are given in `cgs`. Which changes would have to be performed if this quantity was given in `SI` units?

b) Play around with the procedure and try to find the plasma temperature where the Gamow peak

1. for proton-proton collisions
2. for He-He collisions

is located at 10^{-10} (in the units used here, i.e., discarding leading factors), which should give you a fair guess about the required temperatures to start the corresponding fusion processes.

c) Write an analogous Fortran 90 program for the first lines of *gamow.pro* calculating the various probabilites (including the line which defines "ptot"), using only the default values for the colliding particles. Within the program, `write` the energy grid and all three probabilities into a file (one line per `kt, p1(kt),p2(kt),ptot(kt)`).

Use the programs given in the F90 manual (homepage!) as an example.

Modify the idl procedure in such a way that these quantities are printed out, and compare with the content of the F90 output file. Improve your program until the results are the same (to within the chosen precision).

d) Copy the idl-routine to another file *my_gamow.pro* and modify this procedure in the following way.[1] Delete all lines until "window, 0" and use the routine *readcol.pro* (in combination with the function *numlines.pro*, both on the homepage) to read the output-file created by your F90 program into the routine. Follow the description given in *readcol.pro* for the required syntax. (Hint: `readcol,'file',a,b`).

e) (optional, for advanced students)
Modify the original version of *gamow.pro* finally in such a way that the following information is printed out to the terminal

- the maximum value of the combined probability, i.e., the height of the peak, and its position in units $kT_{\rm plasma}$.

[1] Don't forget to change the procedure name accordingly!

- the FWHM (full width of half maximum, i.e., the width of the peak between the two positions where half of the maximum is reached), again in units of $kT_{\rm plasma}$.

Try to use the `where` command to locate the corresponding positions. Calculate the corresponding values for p-p collisions and $T_{\rm plasma} = 10^6, 10^7$ and 10^8 K.

Include all programs/procedures into your solution. Have a lot of fun.