Physics 331 – Computer Programs for Parton Distributions

On the Physics 331 course web site, there is a file `partons.tar` that contains some computer programs for working with parton distributions. This note gives some basic documentation on these programs. If you have any difficulty following these instructions, please email the professor for help.

How do I unpack the distribution and run the programs?

Log in to a UNIX machine (*e.g.*, elaine at Stanford). Create a new directory by typing `mkdir partons`. Use your Web browser to download the file `partons.tar`, and store it in this directory. Type `cd partons` to enter the directory, and type

```
tar -xvf partons.tar
```

The distribution should then unpack, and you will find the files:

```
jets.cpp jets.h makefile plotxf.cpp ppcollider.cpp
```

together with two directories called `engine` and `mrst`. You can edit the files using your favorite text editor. For what is needed in Physics 331, you do not have to edit the material in the two directories. The directory `engine` contains basic reference material and a Monte Carlo integrator. The directory `mrst` contains an interface to one of the parton distribution fits by Martin, Roberts, Stirling, and Thorne (hep-ph/0110215).

To compile the program `plotxf.cpp`, type `make plotxf`. To run this program, type `plotxf`. To compile the program `ppcollider.cpp`, type `make ppcollider`. To run it, type `ppcollider`.

The output of the programs is in the form of gnuplot graphics files. To view a gnuplot file, type `gnuplot` to open the gnuplot command-line environment. Then, to view the file `A.gp`, type `load 'A.gp'`. To print the file, you can either take a screen shot or convert it to postscript. To convert the file to postscript, here are the steps. In the gnuplot environment, type:

```
> set terminal postscript color
> set output 'A.ps'
> load 'A.gp'
```

The plot `A.gp` will be transferred to a postscript file `A.ps`.

What is going on in `plotxf`, and how can I change it?

The core working part of `plotxf` is the `for` loop (line 26 or line 54) that steps over `x` from 0 to 1, evaluates the parton distribution, and sends the data to a gnuplot file. The parton distributions are computed by the C++ class `P` of the type `mrst`. The parton distribution for the up quark in the proton is accessed as

```
P.xfu(x,Q)
```

Note that this function returns $xf_u(x,Q)$, which is smoother than $f_u(x,Q)$. The other parton distributions in the proton are returned by

`P.xfd(x,Q) P.xfs(x,Q) P.xfc(x,Q) P.xfb(x,Q) P.xfg(x,Q)`

`P.xfubar(x,Q) P.xfdbar(x,Q) P.xfsbar(x,Q) P.xfcbar(x,Q) P.xfbbar(x,Q)`

The material in lines 9–12 and 36–40 define `gnuplot` objects, which create gnuplot graphics files. For example, line 9 creates a `gnuplot` object `Gu`, writing to a file `uquarks.gp`, with a plot with x limts 0 to 1 and y limits 0 to 1. Line 39 is the command that sets the `gnuplot` object `Gg` to have a plot with a log scale in the x direction. (Note that the corresponding `for` loop has exponential steps.) Please examine the lines 25, 30, 34. For each curve in the gnuplot, you must `open` and `close` the object; when you are done with the object, you must `finish` it.

What is going on in `ppcollider`, and how can I change it?

The key working part of `ppcollider` is a Monte Carlo integrator over a cross section formula:

$$\sigma = \int dx_1 \int dx_2 \int d\cos\theta_{cm} \frac{\hat{s}}{2} \cdot f_1(x_1, Q) f_2(x_2, Q) \frac{d\sigma}{d\hat{t}}(\hat{s}, \hat{t}, \hat{u}) \tag{1}$$

where f_1 and f_2 are parton distributions. The cross section—everything after the $\cdot$ in (1), is computed by the subroutine `crosssection()` in line 19 of the program. This subroutine can use the parton distributions described above. At the time the program is called, the following kinematic variables associated with the parton-parton reaction are defined:

`x1 x2 p pt cost sint shat that uhat Q alpha_s`

These are, respectively, the longitudinal fractions of the partons from hadrons 1 and 2, the CM momentum of the partons (assumed massless), the transverse momentum of the partons, the cosine and sine of the scattering angle in the CM frame, the 3 Mandelstam variables for the parton reaction, a sensibly chosen value of the momentum scale Q, and the value of $\alpha_s(Q)$. The values of the basic $2 \to 2$ parton cross sections presented in Section 10.4 of Peskin and Schroeder and available as functions such as `ggtogg(shat,that,uhat)`, for P&S eq. (17.78). The complete list of these functions is given in `jets.h`, lines 45–55.

By modifying the function `crosssection()`, you can make the program `ppcollider` display the results of various parton-parton reactions. To plot multiple curves in the same program, the mechanism is a little sophisticated. You have to notice that lines 6–29 are a C++ class definition. If you define a new class, with a different name and a different `crosssection()` function, you can use this class to compute a second parton-parton cross section in the same program.

I hope that whatever else in this code that you will need is self-explanatory. Please have fun modifying these programs and exploring the shapes of parton distributions and their physical effects in cross sections.