Guide to Jet programs version 3.4

The algorithms in the jet program version 3.4 that is available from from these pages are described in detail in Z. Kunszt and D. E. Soper, *Phys. Rev. D* **46**,192 (1992). The algorithms behind the original version of the program are described in S. D. Ellis, Z. Kunszt and D. E Soper, *Phys. Rev. D* **40**, 2188 (1989).

Here we describe some aspects of the program that are not emphasized in these papers because they are more related to numerical error control than to fundamental physics. We refer specifically to the program *jet3_4.f*, which calculates the one jet inclusive cross section $d\sigma/dE_T$. The program *jjy3_4.f*, which calculates the two jet inclusive cross section $d\sigma/dM_{JJ}dy*$, is similar.

A Monte Carlo integration

The program is constructed to perform the integration that expresses the jet cross section, using the Monte Carlo method of numerical integration. There are two contributions, from $2\ partons \to 2\ partons$ and $2\ partons \to 3\ partons$. In the case of $2\ partons \to 2\ partons$, a single parton constitutes the jet. In the case of $2\ partons \to 3\ partons$, the jet can consist of a single parton, or a combination of two partons. The algorithm works by subtracting some pieces from $2\ partons \to 3\ partons$ and including them in $2\ partons \to 2\ partons$ in order to cancel infrared singularities. Descriptions of how this works are contained in the references given above.

If you want, you can think of the program as a Monte Carlo event generator working at the parton level and generating events with weights that represent the probabilities for those events. However, some of the weights are negative. Worse than that, the program will often generate a three parton event with a very negative weight along with an event with a very positive weight that is nearby in the three parton momentum space. This is part of the subtraction scheme, but it makes the interpretation of the program as an event generator a bit strained.

What is really calculated?

Let

$$F(E_T) \equiv \frac{d\sigma}{dE_T}. \tag{1}$$

Our goal is to calculate the cross section $F(E_T)$ for enough values of E_T to enable one to interpolate between them. One might think that the program would calculate the cross section $F(E_T)$ at each desired value of E_T by integrating over a bin centered at E_T. That is, one might calculate

$$F(E_T) \approx \frac{1}{\Delta}\int_{E_T-\Delta/2}^{E_T+\Delta/2} d\tilde{E}_T\ F(\tilde{E}_T). \tag{2}$$

However, this isn't very accurate unless $F(E_T)$ is a linear function of E_T. A more accurate method is to calculate

$$F(E_T) \approx \frac{F_A(E_T)}{\Delta} \int_{E_T-\Delta/2}^{E_T+\Delta/2} d\tilde{E}_T \; \frac{F(\tilde{E}_T)}{F_A(\tilde{E}_T)}. \tag{3}$$

where $F_A(E_T)$ is a simple function [called *TRIAL(E_T)* in the program] that is a good approximation to $F(E_T)$. Then the function that we are integrating is not only linear, it is nearly constant.

This is not quite what we do. There is a problem because with the subtraction method combined with Monte Carlo integration, we could get a large positive contribution to the integral just inside the bin combined with a large negative contribution that actually doesn't contribute because it is just outside of the bin. This would lead to large statistical fluctuations in the result. A simple modification fixes this. We calculate

$$F(E_T) \approx F_A(E_T) \int_0^\infty d\tilde{E}_T \; \rho(\tilde{E}_T) \; \frac{F(\tilde{E}_T)}{F_A(\tilde{E}_T)}. \tag{4}$$

where $\rho(\tilde{E}_T)$ is a smooth function with

$$\int_0^\infty d\tilde{E}_T \; \rho(\tilde{E}_T) = 1. \tag{5}$$

[We take $\rho(\tilde{E}_T)$ to be $1/\tilde{E}_T$ times a gaussian in $\log(\tilde{E}_T)$. The gaussian is centered on E_T and has a width $LNPJSMEAR$.]

The program lists the values of E_T it uses in the column labeled PJ [for $P_T^{\rm Jet}$]. It lists the results, call them $S(E_T)$, of the numerical integration in the column "Smeared XSECT." [Thus $S(E_T)$ is an approximation to the true cross section $F(E_T)$.] The associated integration error is given in the column labeled "Integration error."

Unsmearing

There is, of course, an error associated with smearing the actual cross section against the function ρ. The program does its best to estimate this error and correct for it. If $F(\tilde{E}_T)/F_A(\tilde{E}_T)$ were a polynomial $\tilde{P}(\tilde{x})$ in the variable $\tilde{x} = \ln(\tilde{E}_T)$, then smearing would produce another polynomial $P(x)$, where $x = \ln(E_T)$. The smearing is a linear operation on the coefficients of the polynomial. Thus, knowing the smeared polynomial $P(x)$, we could recover the unsmeared polynomial $\tilde{P}(\tilde{x})$. The program fits a polynomial $P(x)$ to $S(E_T)/F_A(E_T)$ as determined at the E_T points. Then it calculates the unsmeared polynomial $\tilde{P}(\tilde{x})$. Thus

$$\tilde{P}(\log(E_T)) \times F_A(E_T) \tag{6}$$

would be a good approximation to $F(E_T)$. The program lists these values in the column "Unsmeared Fit XSECT." A somewhat more reliable approximation is

$$S(E_T) + \left\{\tilde{P}(\log(E_T)) - P(\log(E_T))\right\} \times F_A(E_T) \tag{7}$$

which is given as the "Corrected XSECT."

The difference between "Corrected XSECT" and "Smeared XSECT" tells how big the correction due to smearing the cross section was. The difference between "Corrected XSECT" and "Unsmeared Fit XSECT" tells how accurate the polynomial fit was.

This procedure is useful mainly as a check. It is best to make sure that $LNPJSMEAR$ is small enough so that the smearing correction is smaller than the statistical errors of the integration. Then "Corrected XSECT" or "Smeared XSECT" are good approximations to the cross section (to within the statistical integration errors).

Davison E. Soper,
Institute of Theoretical Science, University of Oregon
Eugene OR 97403 USA
soper@bovine.uoregon.edu