Id: dispatch.tex,v 1.13 2001/12/12 16:49:05 gsalam Exp

Instructions for use of the dispatch package†

Mrinal Dasgupta
Gavin P. Salam

1 Introduction

The `dispatch` package is an optimised driver for DISENT [1] and DISASTER++ [2], designed for the calculation of event-shape distributions. Its main characteristic is that it reuses a single event to calculate contributions at several x and Q values. This translates to a speed improvement of up to an order of magnitude when studying observables at several x and Q values.

The algorithm, as currently implemented, is suitable for any observable whose definition, when expressed in the Breit frame, contains no dependence on x or on any dimensionful quantity other than through its ratio with Q (so for example there can be no dependence on the incoming proton momentum, or on the proton remnant, nor any cuts on absolute momenta, nor cuts in the lab frame). Examples of suitable observables are Breit-frame event shapes.

We shall frame our discussion in terms of weighted cross sections $\sigma^{(w)}$, where the weighting function will for example be the value observable itself if one wants to calculate its mean value, or it may be a product of Θ functions so as to obtain the cross section for the observable's value to be in a certain 'bin'. For such weighted cross sections we can write

$$\frac{xQ^4 d\sigma^{(w)}}{dxdQ^2} = 2\pi\alpha_{\mathrm{em}}^2 \left((1+(1-y)^2)F_2^{(w)} - y^2 F_L^{(w)}\right) \equiv \mathcal{D}^{(w)}(x,y,Q^2)\,, \qquad y = \frac{Q^2}{xs}\,, \quad (1)$$

where we have introduced a shorthand $\mathcal{D}^{(w)}(x,y,Q^2)$ for the weighted differential cross section. As is standard for the total cross section we have decomposed the result into transverse+longitudinal (2) and pure longitudinal components (L) weighted structure functions, $F_{2,L}^{(w)}$. These two weighted structure functions are themselves convolutions of weighted coefficient functions with parton distributions (both of which are vectors in flavour space):

$$F_{2,L}^{(w)}(x,Q^2) = \int_x^1 \frac{dz}{z}\, \boldsymbol{C}_{2,L}^{(w)}(z,\alpha_{\mathrm{s}}) \cdot x\boldsymbol{q}\left(\frac{x}{z}, Q^2\right)\,. \quad (2)$$

A Monte Carlo program such as DISENT or DISASTER++ returns event contributions which, after taking into account the weighting function and averaging over many events with the same value of $\xi = x/z$, correspond to the following function (again a vector in flavour space)

$$\boldsymbol{\mathcal{D}}_n^{(w)}(z,y) = 2\pi\alpha_{\mathrm{em}}^2 \left((1+(1-y)^2)\boldsymbol{C}_{2n}^{(w)}(z) - y^2 \boldsymbol{C}_{Ln}^{(w)}(z)\right)\,. \quad (3)$$

† Available from `http://www.lpthe.jussieu.fr/~salam/dispatch/`

where the n refers to the order of perturbation theory. There are two points of interest. Firstly, order by order, $\boldsymbol{\mathcal{D}}_n^{(w)}$ is a function only of z and y, but not of x or Q^2. Therefore, for a fixed y value, the event contributions provided by the Monte Carlo at a single x and Q^2 value are sufficient to allow us to calculate $\mathcal{D}_n(x, y, Q^2)$ for any x and Q^2 value, through the following convolution

$$\mathcal{D}_n^{(w)}(x, y, Q^2) = \int_x^1 \frac{dz}{z} \boldsymbol{\mathcal{D}}_n^{(w)}(z, y) \cdot x\boldsymbol{q}\left(\frac{x}{z}, Q^2\right) . \qquad (4)$$

Accordingly the `dispatch` program uses Monte Carlo events at a single value of x and Q^2 (chosen automatically) to calculate the $\mathcal{D}_n^{(w)}$ for a set of x and Q^2 values, all with the same value of y.

For comparison to data, a fixed value of y is usually not suitable. However the second point of interest of eq. (3) is that all the y dependence comes from the linear weighting of the two y-independent functions, $\boldsymbol{C}_{2n}$ and $\boldsymbol{C}_{Ln}$. So if we have calculated $\mathcal{D}_n^{(w)}$ for two values of y, say y_1 and y_2, then we can obtain it for any third value, y_3, as follows:

$$\begin{aligned}\mathcal{D}_n^{(w)}(x, y_3, Q^2) = & \frac{(y_2 y_3 - y_2 - y_3)(y_3 - y_2)}{(y_1 y_2 - y_1 - y_2)(y_1 - y_2)} \mathcal{D}_n^{(w)}(x, y_1, Q^2) \\ & - \frac{(y_1 y_3 - y_1 - y_3)(y_3 - y_1)}{(y_1 y_2 - y_1 - y_2)(y_1 - y_2)} \mathcal{D}_n^{(w)}(x, y_2, Q^2) . \qquad (5)\end{aligned}$$

So one simply carries out two `dispatch` runs at different y values and then uses the `join-ybj2s` program to combine them with the correct weights so as to correspond to a fixed value of *ep* centre of mass energy, $\sqrt{s}$.

This that we have to do two full runs of `dispatch`. However in many practical situation, for example when comparing to the H1 event-shape distributions in [3] the relevant y values all satisfy $y \lesssim 0.4$ — choosing a low value for y_1 and a high value for y_2 it turns out that the weight for the high-y run is usually somewhat smaller than that for the low-y run, allowing one to generate fewer events at the high-y point. As an example, in [4] we used $y_1 = 10^{-3}$ and $y_2 = 0.9$, and generated only one tenth of the total number of Monte Carlo events at the higher y value. So there is relatively little overhead from generating the second high-y point.

1.1 Other features

Other features of `dispatch` include:

- Optimised histogramming routines for the error treatment (when considering many variables at many x-values, the histogramming — in particular the operations required to obtain a proper error — becomes a major overhead).
- A modified version of DISENT which eliminates many of the crashes which occur with lower cutoffs and higher `NPOW` values.

- Replacement of the default DISENT and DISASTER++ random number generators with the CERNLIB RM48 generator, allowing use of a large number of different random sequences (useful for runs in parallel).
- Output of results at regular intervals in case of crashes. If restarted after a crash the program automatically reads the last seed and continues from there.
- A tool, `join-renorm`, for combining runs carried out with different seeds.

2 Using dispatch

There are two separate executables for DISENT and DISASTER++: `dispatch-disent` and `dispatch-disaster`, to be found respectively in the `disent/` and `disaster/` directories of the distribution (after compilation, described in the `INSTALL` file).

A typical command to set off a Monte Carlo run (from the `disent/` directory) might be

```
> dipatch-disent -rundef ../examples/H1.rdf -ybj 1e-3 -nev 1e6
```

This will generate a DISENT run with 10^6 events at a y value of 10^{-3}. The run definition file `../examples/H1.rdf` specifies which parton distribution function(s) to use, which x and Q points to consider, and gives details of the event shapes to be studied.

By default the output will be sent to a file `gtopdraw.top`, though there is a command-line option (`-tdfl`) to send the output to a file of the user's choosing.

Below we describe in more detail both the run definition file and the command-line options.

2.1 The run-definition file

Various example run-definition files are given in the `examples` directory. Here we describe their structure. An example file is given below:

```
# For comparison with the H1 1997(?) data
( new run
( pdf list #------------------------------------------
pdf_MRS99_1  # MRS99 with imode=1
( xQ list #-------------------------------------------
      0.0039        7.5
      0.0060        8.7
      0.014         15.0
      0.020         17.8
      0.031         23.6
      0.056         36.7
```

```
      0.11          57.9
      0.18          81.3
( Elim    #--- consider events with (E_current / Q) > 0.1
       0.10
( event-shape list #---------------------------------
# evshp   low    high  nbin  log/lin  eslim
brd_tE   1e-3    1.0    100   lin     0.5
tau_zE   1e-3    1.0    100   lin     0.5
tau_tE   1e-3    0.5    100   lin     0.13
Cpr_E    1e-3    0.5    100   lin     0.5
rho_E    1e-3    0.25    50   lin     0.5
brd_zE   1e-3    1.0    100   lin     0.5
```

Anything following a '`#`' is a comment. Lines starting with a '`(`' are command lines. So '`( new run`' indicates that this is a new run definition (the file can contain several complete run definitions, one after the other).

PDFs: The line '`( pdf list`' indicates that a list consisting of one or more parton distribution function identifiers follow. For CTEQ [5] and GRV [6] (actually not yet supported!) the nomenclature is `pdf_` followed by the official name (entirely in capitals) of the parton distribution. For the MRS(T) distributions [7] rather than using the official name (which is not always very informative) we append a number to the end of the name to indicate the mode of the corresponding set. So `pdf_MRS99_2` designates the MRS99 set, with mode 2 (`cor02`, the higher-gluon set). The complete list of supported sets can be found by examining the constants at the beginning of `common/pdf_manager.f90`.

Several PDFs can be used in a single run but there are limitations: only one CTEQ and only one GRV distribution can be used at a time (because they load the whole table at each access of a different distribution). However several MRS(T) distributions can be used at once.

x and Q values: Following the command line '`( xQ list`' there should follow a list of pairs of x and Q values (note, not Q^2!), with Q given in units of GeV. Optionally a third value can be appended to each line, the ratio of the factorisation scale μ_F to Q. With DISENT the ratio μ_F/Q should always be the same, whereas with DISASTER++ in principle they can be different (but currently only $\mu_F/Q = 1$ is supported for DISASTER). A default common μ_F/Q value can also be set via the command-line option `-muF` (see below). The order of the x and Q points is unimportant.

E_{lim}/Q value: Following the command line '`( Elim`' there should follow a value for the ratio E_{lim}/Q. Events containing less than this fraction are not binned (this is a requirement following from infrared safety for many of the variables).

Currently the only way of obtaining results with several E_{lim}/Q values is to repeat whole run definition several times (*i.e.* starting from '`( new run`'), each with a different

`Elim` value. This leads to some overhead because of needless recalculation of parton distributions. If there is demand, a future version of `dispatch` may modify the way in which `Elim` is defined, so as to eliminate this restriction.

Event-shape list: The set of event shapes and the histogramming to be used for them is indicated by the lines following the instruction '`( event-shape list`'. Each line of the list should contain the following information (in order):

- the name of the event shapes (a complete list is to be found at the beginning of `common/dis-shapes.f90`).
- the lower limit of the histogram.
- the upper limit of the histogram.
- the number of bins in the histogram
- '`lin`' or '`log`' to indicate whether histogramming is for the variable itself or for its logarithm (the user-specified limits are then respectively for the variable or for its logarithm).
- A quantity referred to as `eslim`: the program calculates the total contribution for the event shape (never its log, even with log binning) to be above this limit. It is mainly a leftover from earlier versions of the program but should nevertheless be supplied (`eslim` can be put to an arbitrarily large value, but values ≤ 0 are probably better avoided)

For linear binning, with a view to use with the `disresum` package [8], it it advisable not to set the lower limit for linear binning to zero, since for the bin containing $V = 0$ to be well-defined one needs the NNLO virtual correction to $1 + 1$ jet production.

2.2 Command-line options

There are numerous command line options, some of which are common to both the DISENT and DISASTER++ version of `dispatch`. These are given immediately below, while those specific to one or the other MC generator are given afterwards. Default values for options are given in square brackets.

`-rundef filename.rdf`
: Required option; it sets the run definition file to be read in.

`-nev NEV` [10^6]
: Number of events.

`-ybj` y_{Bj} [0.001]
: The value to be chosen for y_{Bj} (recall that multiple x and Q values are done for fixed y_{Bj}).

`-nf` n_f [5]

The number of active flavours.

`-tdfl filename` [`gtopdraw`]

File to which output should be sent. Standard error and standard output are sent to `filename.err` and `filename.out`, while the main program output is sent to `filename.top`. If this option is *not* specified then standard error and output are not redirected (only the main program output is sent to a file). If the `filename.top` file already exists then (a) if `-iseq` corresponds (see below) the initial seed is set to be the last seed of the preexisting file; and (b) output is sent filenamexx.top where xx is smallest number for which filenamexx.top does not exist.

`-iseq n` [1]

Sets the random number sequence to be used. For details, see the documentation of the CERNLIB `rm48` random number, which replaces the default generators supplied with DISENT and DISASTER++.

`-wf nev` [10^6 for DISENT, 5.10^5 for DISASTER++]

The maximum interval (in events) after which intermediate results should be written to the output file (whose name stays the same at each write). For small total numbers of events the interval between writes may be smaller. Note that for DISASTER++ the normalisation is correct only for the final write. This is due to internal technical complications which cause the intermediate writes to have a normalisation which is off by a couple of percent — if one wishes to use the information from an incomplete run one should renormalise a posteriori (the `join-renorm` program offers this option).

`-muF` μ_F/Q [1.0]

Sets the default value for the ratio μ_F/Q (which can optionally be reset in the run definition file). Note that $\mu_F/Q \neq 1$ is not yet supported for DISASTER++.

`-cutoff cutoff` [10^{-12}]

Value for the internal generator cutoff.

Generator specific options. There are various generator specific options relating to the optimisation of phase space generation and other aspects of the calculation which can be controlled. For the DISENT generator, the following options are supported:

`-npow1 NPOW1` [2]

`-npow1 NPOW2` [4]

The values for the `NPOW1` and `NPOW2` internal DISENT parameters, which govern the generation of $1+2$ and $1+3$ parton event respectively. Higher values lead to a larger fraction of soft and collinear events (see the DISENT documentation for details).

`-cf` C_F [4/3]
: The value of the C_F colour factor.

`-ca` C_A [3]
: The value of the C_A colour factor.

`-tr` T_R [1/2]
: The value of the T_R colour factor.

`-order ORDER` [2]
: Whether to do a LO (`ORDER`$=1$) or a NLO (`ORDER`$=2$) calculation.

For the DISASTER++ generator the options that are supported are all related to the adaptive procedure for optimising phase space generation

`-ipdf ipdf` [1]
: Sets the PDF to be use for the adaptation (the number indicates the position in the list of PDFs in the first run definition).

`-ivar ivar` [1]
: Sets the event-shape to use for the adaptation (the number indicates the position in the list of event shapes in the first run definition)..

`-adapt adaptation_rule` [1]
: Indicates exactly what quantity is to be used for the adaptation. The quantity is a combination of some integral over the histogram, the mean value of the event shape and the probability of the current hemisphere containing an energy less than `Elim`. With `-adapt 1` the integral over the histogram is weighted uniformly, with `-adapt 2` it is weighted with $\sqrt{|V|}$ (or $\sqrt{|\log V|}$ for logarithmic binning), and with `-adapt 3` it is weighted with $4V$ (or $4\log V$ for logarithmic binning). Depending on the kind of binning being used some adaptation options may be unstable.

`-niter niter` [5]
: The number of VEGAS adaptation runs to be done before the main run. The fraction of events devoted to all adaptation runs taken together is 25%, and unlike the settings in the default DISASTER++ example program, the number of events requested (`-nev`) corresponds to the number of event generated.

2.3 Output format

Output can be examined with topdrawer (see Paolo Nason's page [9]). Various comment lines are included to describe the characteristics of the run. They are on lines starting with a '`(`', because this is the topdrawer comment character.

The output file should mostly be self-explanatory. However a few points are to be kept in mind:

- All quantities are given as coefficients of powers of $\alpha_s/2\pi$.

- All quantities are normalised to the leading order total cross section, `sigma0` (except the leading order total cross section itself, for which the result is defined in a MC-specific manner).

- The NLO contribution to the total cross section has large errors in DISASTER++. A separate run would be needed to generate accurate NLO total cross sections and since NLO cross sections are determined calculated by other methods it was decided not to waste time generating them with DISASTER++. Note also that the version of DISENT used (0.1) gives incorrect answers for the NLO cross section if `NPOW1` (see below) is different from the default value of 2.

- `sigElim1` and `sigElim2` are the $\mathcal{O}(\alpha_s)$ and $\mathcal{O}(\alpha_s^2)$ contributions for the energy in the current hemisphere to be less than `Elim`. Similarly `sigESlim1` and `sigESlim2` are the contributions for the event shape to be larger than `eslim`. The mean values that are given are always for the variable, never the logarithm.

- The histograms are given in a format where the first column is the centre of the bin, the second column is the bin contents (divided by bin size) and the third column is the error.

2.4 Restrictions and observations

Not everything is supported everywhere: some things not possible, others just not yet ready. For a summary, see table 1.

	DISENT	DISASTER++
Adaptation	✗	✓
Common $\mu_F \neq Q$	✓	✗
Different $\mu_F \neq Q$	✗	✗

Table 1: What is and is not yet supported for DISENT and DISASTER++; sometimes it is because of lack of support within the MC, other times it is because it has yet to be implemented within `dispatch`.

3 Associated tools

The tools described here, while an inherent part of the `dispatch` framework, are actually distributed with the `disresum` package [8]. The reason is that they require parsing (and in the case of `join-renorm` convolution) routines that are in any case to be found in `disresum`.

3.1 join-renorm

When many identical runs have been carried out with different seeds (or if runs have crashed and been restarted from the last previous seed) it is useful to combine them into a single run. The `join-renorm` utility performs this combination for as many files as are specified on the command line. The weights given to the different files are independent for each quantity and each histogram bin and are based on the errors quoted for each quantity in each file (except for quantities with zero error which are thrown out). This (rather than weighting in proportion to the number of events) allows to extract optimal benefit from runs carried out with different adaptation procedures or `NPOW` values.

The utility should be used as follows

```
> join-renorm file1.top file2.top ...
```

where up to 200 files can be specified. If among these files there are results from incomplete DISASTER++ runs (*e.g.* following a crash) then the option `-renorm` should be specified as the first argument: with incomplete DISASTER++ runs, `dispatch` is unable to reconstruct the correct normalisation. This is because in order to obtain the correct total weight DISASTER++ generates a number of events which differs slightly from that requested, and that number is known only after completion of the run. The option '`-renorm`' calculates the correct normalisation factor based on the exact and Monte-Carlo value of `sigElim1` (from the first PDF set).

The result from `join-renorm` is sent to standard output. Note that it correctly sets the value for the total number of events, however the information provided in the output concerning the random number sequence is unreliable.

3.2 join-ybj2s

As mentioned in the introduction, the basis of the method for generating many x and Q points is to do runs at two different values of y_{Bj}, and then combine them with appropriate weights. This is done by `join-ybj2s`, which should be used as follows

```
> join-ybj2s rts file-y1.top file-y2.top
```

The program calculates the appropriate y values for each x, Q point based on the value of $\sqrt{s}$ (`rts`) and then weights the results from the two files appropriately. The result is sent to standard output.

References

[1] S. Catani and M. H. Seymour, Phys. Lett. B **378** (1996) 287 [hep-ph/9602277]; S. Catani and M. H. Seymour, Nucl. Phys. B **485** (1997) 291 [hep-ph/9605323].

[2] D. Graudenz, hep-ph/9710244.

[3] C. Adloff *et al.* [H1 Collaboration], Eur. Phys. J. C **14** (2000) 255 [Erratum-ibid. C **18** (2000) 417] [hep-ex/9912052].

[4] M. Dasgupta and G. P. Salam, arXiv:hep-ph/0110213.

[5] H. L. Lai *et al.* [CTEQ Collaboration], Eur. Phys. J. C **12** (2000) 375 [hep-ph/9903282]; see also `http://www.phys.psu.edu/~cteq/CTEQ5Table/`.

[6] M. Gluck, E. Reya and A. Vogt, Eur. Phys. J. C **5** (1998) 461 [hep-ph/9806404]; see also `http://durpdg.dur.ac.uk/hepdata/grv.html` .

[7] A. D. Martin, R. G. Roberts, W. J. Stirling and R. S. Thorne, Eur. Phys. J. C **14** (2000) 133 [hep-ph/9907231];
A. D. Martin, R. G. Roberts, W. J. Stirling and R. S. Thorne, hep-ph/0110215.
see also `http://durpdg.dur.ac.uk/hepdata/mrs.html`

[8] The `disresum` package can be obtained from `http://www.lpthe.jussieu.fr/~salam/disresum/`.

[9] The program and documentation for `topdrawer` can be found at `http://cern.ch/nason/misc/`.