PHANTOM 1.1

A Monte Carlo generator for six fermion physics at colliders. [1]

User Guide

Alessandro Ballestrero, Aissa Belhouari, Giuseppe Bevilacqua, Diogo Buarque Franzosi, Vladimir Kashkan and Ezio Maina

INFN, Sezione di Torino, Italy
and
Dipartimento di Fisica Teorica, Università di Torino, Italy

e-mail: ballestrero@to.infn.it,maina@to.infn.it,bevilacq@to.infn.it

URL: http://www.to.infn.it/∼ballestr/phantom

[1]Work supported by the European Union under contract MRTN-CT-2006-035505 and by Ministero dell' Università e della Ricerca Scientifica.

1 Introduction

The program has two modes of operation which are selected by the input value `ionesh`. If `ionesh=0` the program computes the cross section for one single process, specified by `iproc`. If the input variable `iflat` is set to `1` the program produces one or more integration grids depending on the number of channels required for a good mapping of phase space for the selected process. If `ionesh=1` the program generates unweighted events for a set of processes, which is specified by the user, using the previously produced grids. All the integration grids for each selected process must be included for a meaningful generation.

For integration and grid determination we use `VEGAS`.

The syntax of the input is almost identical to the one required by the CERN library routine `FFREAD`. Routines internal to `PHANTOM` are however used (`iread, rread`), so that real variables can (and must) be given in double precision.

All lines in the file of input must not exceed 80 characters, with the exception of filenames which can be 200 characters long. A `*` or `C` character at the beginning of a line identifies it as a comment line. Comment lines can be freely interspersed within the input file, with the only obvious exception that they must not interrupt a list of input values for a single array variable. The name of the variable to be read must be specified as the first word of a line (needs not to begin in column 1). Its value (values) must follow it. The list of values can span several lines. Variables which are not needed for the process under study will be ignored. They can be left in the file without harm. All variables actually read from the input file will be reproduced in the output.

An example of the input file `r.in` is included with the distribution,

All masses and energies are in GeV. Energies refer to the collider frame. All angles are in degrees.

All entries in the following sections, which describe input variables in more detail, are in the form:

`variable-name`, (`example-of-value(s)`):

or

`variable-name`, (`full-list-of-possible-values: val1/val2/...`):

If the list reduces to `0/1` we adopt the convention that `0` corresponds to `no` and `1` to `yes`.

2 Common inputs

`ionesh`, (`0/1`): this flags selects the basic operation mode of `PHANTOM` as explained in the introduction.

`idum`, (`-123456789`): random number generation seed. Must be a large negative integer.

`i_coll`, (`1/2/3`): type of collider: 1 corresponds to a pp accelerator, 2 corresponds to a $p\bar{p}$ accelerator, 3 corresponds to a e^+e^- accelerator. In this last case also the two following flags must be set:

`i_isr`, (`0/1`): no/yes initial state radiation for e^+e^- ILC.

`i_beamstrahlung`, (`0/1`): no/yes beamstrahlung for e^+e^- ILC.

`ecoll`, (`14000.d0`): total collider center of mass energy.

`perturbativeorder`, (`1/2/3`): if `perturbativeorder=1` the ampliude is evaluated at $O(\alpha_{em}^6)$ including all purely electroweak contributions. If `perturbativeorder=2` the ampliude is evaluated at $O(\alpha_{em}^4\alpha_s^2)$, including all one gluon exchange contributions and all processes with two external gluons. Finally, if `perturbativeorder=3` both contributions are evaluated including possible interference terms.

`i_massive`, (`0/1`): with 0 faster massless amplitudes are used unless there is at least a b quark line. with 1 massive amplitudes are used.

`PDFname`, (`/home/user/LHAPDF/cteq5l.LHgrid`): Full pathname of the PDF set in the Les Houches Accord distribution, which can be downloaded from http://projects.hepforge.org/lhapdf/ .

`i_PDFscale`, (`1/2`): If `i_PDFscale=1` the PDF scale is

$$Q^2 = M_W^2 + \frac{1}{6}\sum_{i=1}^{6} p_{Ti}^2. \tag{1}$$

where p_{Ti} denotes the transverse momentum of the i–th final state particle.
If `i_PDFscale=2` the PDF scale is set to

$$Q^2 = M_t^2 + p_T^2(t), \tag{2}$$

where $p_T(t)$ is the transverse momentum of the reconstructed top, for all processes in which a t and or $\bar{t}$ can be produced. For all other processes the scale is evaluated as in Eq.(1).

`rmh`, (`250.d0`): Higgs mass. If set equal to a negative number all diagrams involving a Higgs propagator are omitted. In the Unitary Gauge, which we use, this is equivalent to taking the infinite Higgs mass limit .

`i_ccfam`, (`0/1`): if `i_ccfam=0` only the process(es) explicitely required by the user is(are) computed. If `i_ccfam=1` the required process(es) is(are) computed together with the reactions obtained interchanging the quarks or antiquarks of the first family with those of the second, and with the reactions obtained by Charge Conjugation. For instance if the user requires the process

$$u\bar{u} \to b\bar{b}c\bar{s}\mu^-\bar{\nu}_\mu$$

with `i_ccfam=1`, all the following processes are computed or generated:

$$\begin{aligned} u\bar{u} &\rightarrow b\bar{b}c\bar{s}\mu^-\bar{\nu}_\mu \\ c\bar{c} &\rightarrow b\bar{b}u\bar{d}\mu^-\bar{\nu}_\mu \\ \bar{u}u &\rightarrow \bar{b}b\bar{c}s\mu^+\nu_\mu \\ \bar{c}c &\rightarrow \bar{b}b\bar{u}d\mu^+\nu_\mu \end{aligned}$$

With the obvious exception of `ionesh` and `idum` all common inputs must have the same value while producing integration grids and while generating events.

2.1 Cuts

All cuts mentioned in this section are applied at hard parton level, before showering and hadronization. In the following `lep` will refer to *any charged* lepton, while `j` will refer to *any* final state quark/antiquark/gluon. When quarks are mentioned, antiquarks and gluons will always be implicitely included.

All cuts are specified by an integer of the type `i_flag` which specifies whether the corresponding cut is activated (`i_flag=1`) or not (`i_flag=0`) and by one or more values which define the extrema of the accepted region. The name of the flag in most cases is the name of the corresponding variable with `i_` prepended. Exceptions to this rules will be pointed out. In all other cases we will give only the variable name and the corresponding flag will be understood.

`e_min_lep`, (`20.d0`): minimum energy of any charged lepton.

`pt_min_lep`, (`10.d0`): minimum transverse momentum of any charged lepton.

`eta_max_onelep`, (`3.d0`): this cut requires at least one charged lepton to have $|\eta| <$`eta_max_onelep`. If other charged leptons are present in the final state they can be outside the selected range in η. Ignored if no charged leptons are present in the final state.

`eta_max_lep`, (`3.d0`): maximum absolute value of the pseudo–rapidity η of any charged lepton.

`ptmiss_min`, (`50.d0`): minimum missing transverse momentum. It is applied on the total neutrino transverse momentum.

`e_min_j`, (`20.d0`): minimum energy of any quark.

`pt_min_j`, (`10.d0`): minimum transverse momentum of any quark.

`eta_max_j`, (`6.5d0`): maximum absolute value of the pseudo–rapidity of any quark.

`i_eta_jf_jb_jc`, (`0/1`): requiring hard partons at large rapidities is often useful to disentangle vector boson fusion events from the QCD background. To this end, the `i_eta_jf_jb_jc` flag specifies whether the following triplet of related cuts are activated.

> `eta_def_jf_min`, (`2.d0`): minimum value of the pseudo–rapidity of the most forward quark.
>
> `eta_def_jb_max`, (`-2.d0`): maximum value of the pseudo–rapidity of the most backward quark.
>
> `eta_def_jc_max`, (`2.d0`): maximum absolute value of the pseudo–rapidity of the remaining two (*central*) quarks.

`pt_min_jcjc`, (`50.d0`): minimum transverse momenta of the two central quarks.

`rm_min_jj`, (`20.d0`): minimum invariant mass of any pair of quarks.

`rm_min_jlep`, (`30.d0`): minimum invariant mass of any charged–lepton–quark pair.

`rm_min_ll`, (`30.d0`): minimum invariant mass of any pair of opposite–sign same–flavour charged leptons.

`rm_min_jcjc`, (`60.d0`): minimum invariant mass of the two central quarks.

`rm_max_jcjc`, (`110.d0`): maximum invariant mass of the two central quarks.

`rm_min_jfjb`, (`100.d0`): minimum invariant mass of the most forward and most backward quark.

`eta_min_jfjb`, (`3.d0`): minimum absolute value of the difference in pseudo–rapidity between most forward and most backward quark.

`d_ar_jj`, (`0.7d0`): minimum separation in $\Delta R = \sqrt{\Delta\phi + \Delta\eta}$ between any two quarks.

`d_ar_jlep`, (`0.7d0`): minimum separation in ΔR between any quark and any charged lepton.

`thetamin_jj`, (`15.d0`): minimum angular separation between any two quarks.

`thetamin_jlep`, (`15.d0`): minimum angular separation between any quark and any charged lepton.

`thetamin_leplep`, (`15.d0`): minimum angular separation between any pair of charged leptons.

`i_usercuts`, (`0/1`): this flag determines whether additional user specified cuts are required. These additional requirements must be implemented in a routine called `IUSERFUNC` an example of which is provided with the distribution.

NOTE: explain meaning of relevant variables: pcoll(called p in cuts.f), then p0 and all other variable computed by cuts.

3 Inputs for `ionesh=0`

In this case the program proceeds in two steps. The first one is called thermalization. It determines the relative weight of each channel in the multichannel integration and it produces a first instance of `PHANTOM` space grids, one per channel. At least three thermalization iterations are performed. At the end of the third iteration all channel whose relative weight is smaller than 10^{-3} are eliminated. If any channel is discarded, two aditional thermalization iterations are performed, regardless of the flag `itmx_therm` mentioned below. The grids produced in the thermalization stage are then used as a starting point for the second step which consists of one integration per channel. Each integration will typically consist of several iterations and at each iteration the `PHANTOM` space grid will be refined in an effort to decrease the overall variance. A number of iterations between 3 and 5 is normally the best choice. If higher precision is requested it is usually more convenient to increase `ncall` rather than `itmx`. The user must be aware of the fact that if no point survives the cuts during an iteration, either during thermalization or at the integration stage, `VEGAS` will stop with an error.

At pp and e^+e^- colliders, when `ionesh=0` the process is computed exactly as specified by the user, assuming the first incoming particle to be moving in the $+z$ direction and the second one in the $-z$ direction. Even though at a pp collider this violates the symmetry between the two initial state protons, it can be useful for testing and for specialized studies. The full cross section can be easily obtained for unlike incoming partons by symmetrizing all distributions with respect to the beams and multiplying by a factor of two. At a $p\bar{p}$ collider the default for unlike incoming partons is to sum over the two possible assignement of the two partons to the beams. The behaviour at pp colliders can, and typically should, be modified when `ionesh=1` by the flag `i_exchincoming`. As a consequence cross sections computed with `ionesh=1` can be different from those computed with `ionesh=0`.

`iproc`, (`3 -4 2 -2 3 -3 13 -14`): specifies the desired process using the standard Monte Carlo particle numbering scheme:

d	u	s	c	b	e^-	ν_e	μ	ν_μ	τ	ν_τ	g
1	2	3	4	5	11	12	13	14	15	16	21

Antiparticle are coded with the opposite sign. The first two entries represent the initial state partons. Therefore the example corresponds to the reaction

$$s\bar{c} \to u\bar{u}s\bar{s}\mu\bar{\nu}_\mu. \tag{3}$$

`acc_therm`, (`0.01d0`): the maximum relative accuracy during thermalization. When accuracy `acc_therm` is reached thermalization stops.

`ncall_therm`, (`300000 2000000`): the maximum number of points for each iteration during thermalization. The first component refers to the number of calls for the first 3 iterations (alpha(i) determination), the second one to the calls for the remaining iterations (thermalization) `itmx_therm`, (`5`): the maximun number of times the grid is adapted and the multichannel weights are adjusted during thermalization. Ignored if any small channel is discarded.

`acc`, (`0.005d0`): the integration accuracy. When accuracy `acc` is reached the program stops.

`ncall`, (`5000000`): the maximum number of points for each iteration of the actual integration. `VEGAS` will in general use a number of `ncall_therm` and `ncall` lower than the input ones. The actual value is written in output, where also the number of points which survive all the cuts (`effective ncall`) is reported.

`itmx`, (`5`): the maximum number of iterations used to evaluate the integral and to refine the grid.

`iflat`, (`0/1`): this flag must be set to `1` in order to produce the `PHANTOM` space grids for later unweighted event generation. If `iflat=1` the program also returns the maximum weight $w0$ produced in the next–to–last iteration, the maximum weight $w1$ produced in the last iteration and the number of point with weight greater than $w0 * 1.1$ visited during the last iteration.

4 Inputs for `ionesh=1`

`nunwevts`, (`10000`): number of unweighted events to be produced. The program stops only when `nunwevts` events have been generated.

`iwrite_event`, (`0/1`): if `iwrite_event=1` the generated parton level events are written in a file named `phamom.dat` using the Les Houches AccordFile Format[**?**].

`iwrite_mothers`, (`0/1`): if `iwrite_mothers=1` information about intermediate particles (mothers) in .dat file are reported.

`i_exchincoming`, (`0/1`): This flag is relevant only for *pp* colliders and is ignored in all other cases. If `i_exchincoming=0` the process is generated exactly as required, assuming the first incoming particle to be moving in the $+z$ direction and the second one in the $-z$ direction. If `i_exchincoming=1` the two initial particle are assigned at random to the two protons, doubling the corresponding cross section if they are not identical. As a consequence cross sections computed with `ionesh=1` can be different from those computed with `ionesh=0`.

4.1 Cuts

`iextracuts`, (`0`): This flag determines whether additional cuts are required at the generation stage. Input files for `ionesh=1` must contain the set of cuts, defined in the common input section, used while generating integration grids. As a consequence additional cuts will be effective only if they are more stringent than those imposed in the `ionesh=0` step. The additional cuts must be defined setting `iextracuts=1` and then setting the full set of extra cuts. The corresponding names are equal to those in the common input section with `os` appended.

`i_usercutsos`, (`0`): this flag determines whether additional user specified cuts are required at the generation stage. These additional requirements must be implemented in a routine called `IUSERFUNCOS` an example of which is provided with the distribution. Obviously, the comments concerning the relationship between cuts in the grid–generation and event-generation stage also apply to the user specified cuts.

4.2 Processes

`nfiles`, (`3`): number of input `phavegas`-type files to be included in the generation. This input should be immediately followed, with no intervening blank line, by `nfiles` filenames, each on a separate line, with a maximum length of 200 characters, as in the following example:

```
nfiles   3
/home/user/dir1/phavegas01.dat
/home/user/dir1/phavegas02.dat
/home/user/dir2/phavegas01.dat
```

All the integration grids for each selected process must be included for a meaningful generation.

5 Makefile

In the distribution one can find an example of a makefile. It can be used for compiling with Intel ifort, Portland Group pgf77/pgf90 on linux or with a Tru64 unix compiler, or can be easily adapted for other compilers Notice however that PHANTOM cannot be compiled with g77 as this compiler does not allow the use of structures. Intel compilers work fine both on linux and on MacOS.

For the compilation one needs to have a pdf library LHAPDF which can be obtained by http://projects.hepforge.org/lhapdf . The actual location of the library must be indicated in the makefile, as in the given example.

In the present release the evolution of α_s is conputed with Pythia routines and for such a reason one needs to compile also the version of Pythia included in this distribution.

6 Scripts

The number of reactions which contribute to a given final state, say $4jl\nu$, can be very large, particularly at a hadron collider. By making use of symmetries, the task of integrating all relevant processes and producing the corresponding grids can be substantially simplified. Indeed all reaction which differ by charge conjugation and by the symmetry between first and second quark family can be described by the same matrix element. This extends to processes which are related by a permutation of the lepton families. At most they differ by the Pdf's convolution and possibly by a parity tranformation, which are accounted for automatically by `PHANTOM` if `i_ccfam=1` is set. Determining the minimal set of reactions for which grids need to be computed can be tedious and error prone. Therefore we have included in the distribution a Perl script `setupdir_LHC.pl` which handle the task for proton proton colliders.

The script can be executed with, for instance:

```
perl setupdir_LHC.pl [-options ...]
```

where options include:

```
-basedir      directory which contains the exe file
                (full pathname).                       ./
-dirtreeroot  root of new tree (full pathname).        mh0
-template     input template file.                     template.st0
-executable   executable file.                         phantom.exe
-inputstring  must contain only leptons and gluons     ""
              must be enclosed in double quotes: "e e_ mu vm_".
-quarks      number of quarks to be inserted
              (even integer > 0 and <= 10).
-Top          number of top/antitop quarks required in the
              final state. If the option is not set any number of
              tops is accepted. If the option is of the form
              n+ with n an integer any reaction with at least
              n tops are accepted.
-system       batch system. Allowed values are:
              SGE: Torino
              LSF: Cern
              Rio: Rio
-help         prints usage details.                      -
```

long options can be abbreviated up to one letter, eg. `-basedir X` can be passed as `-b X`. In square brackets the default values are reported. Foe each reaction the scripts create a directory which contains the corresponding `r.in` input file which is generated using the given template which must contain all input parameters with the exception of `iproc` which is supplied by the script. A runfile called `run` is also created. In the root directory of the new tree the script creates a file called `LSFfile` which can be used for submitting all integration jobs to the CERN LSF batch system. For different batch systems the user should appropriately edit the Perl scripts.

References

[1] J. Alwall *et al.*, A Standard format for Les Houches event files. Written within the framework of the MC4LHC-06 workshop: Monte Carlos for the LHC: A Workshop on the Tools for LHC Event Simulation (MC4LHC), Geneva, Switzerland, 17-16 Jul 2005, *Comp. Phys. Commun.* **176** (2007) 300, [hep-ph/0609017].