SHERPA mini documentation

(version 1.0.9)

Timo Fischer[1], Tanju Gleisberg[1], Stefan Höche[1],
Frank Krauss[1], Thomas Laubrich[1], Andreas Schälicke[2],
Steffen Schumann[1], Frank Siegert[1], Jan Winter[1]

[1] Institute for Theoretical Physics, TU Dresden, D-01062 Dresden, Germany
[2] DESY-Zeuthen, Platanenallee 6, D-15738 Zeuthen, Germany
email: info@sherpa-mc.de

December 8, 2006

Abstract

This is a short primer on how to get started and how to use the SHERPA framework for the *Simulation for High-Energy Reactions of PArticles*. It is not meant to fully reflect the physics implemented in this program nor does it give an account of the program structure or the coding principles underlying it. For different physics aspects and especially for external code that has been linked, corresponding references are given and users are encouraged to cite them accordingly. Also, users of this framework are kindly asked to refer to its original publication [1].

Contents

Introduction

This mini-guide is meant as a help for users to get started with SHERPA. It does not replace a careful study of the manual(s) or the many excellent publications on different aspects of event generation or physics at collider experiments. Users of the code are encouraged to read and to cite original work of the authors or others whenever justified.
This short documentation is written for those mainly interested in getting started as quickly as possible. Reading it should be sufficient to understand, which switches and options have been provided, how to use them, and how the output may be interpreted.
For those, who are interested in playing with this tool and, maybe, in adding some new features to the program, a part II explaining the possibilities of easily extending the code will be made available soon.

It should be stressed that in the construction of such a code a number of implicit assumptions, unjustified simplifications of complicated situations, potential bugs and other shortcomings of the authors are included. Therefore, the results of event generators, independent of their quality, should always be verified and cross-checked with results obtained by the programs of other authors.
It should also be stressed that there is a number of physics processes SHERPA cannot simulate (yet). These include, among others,

- DIS-like situations. SHERPA may calculate cross sections for DIS and similar processes; however, the parton shower of APACIC is not ready to deal with the situation of having only one space-like evolution to be done.
- Soft hadron collisions. SHERPA has no means to simulate reggeon exchange processes etc.. Therefore, soft events are not accounted for. This is also true for pile-up events, which are still out of SHERPAs current scope.

In addition, the hadronization and most of the hadron decays are still completely handled by Pythia; an interface is provided.

The authors welcome suggestions, bug reports, etc. and endeavour to answer any question as quickly as possible. Furthermore they try to provide user support as far as possible. In order to improve SHERPA, please do not hesitate to send any form of comment to their common mail account

info@sherpa-mc.de .

For bug fixings, updates, etc., please refer to the homepage

http://www.sherpa-mc.de .

Part I
Basics: Getting started with SHERPA

1 Structure of the SHERPA framework

The SHERPA package is constructed in a modular fashion, where each module encapsulates different aspects of event generation for high-energy particle reactions. Each module resides within its own namespace. In the names of the modules a version number is included. For instance, there are the modules ATOOLS-2.0 or AMEGIC++-2.0. The label "++" thereby signals that the corresponding module has been developed independently and, therefore, is equipped with a separate documentation[1]. After the SHERPA package has been installed in a directory, each module is located in its own subdirectory of the same name. The main steering module defining how the events are actually generated is called – in its current version – SHERPA-1.0. Altogether, the following modules are currently distributed with the SHERPA framework:

- ATOOLS-2.0

  This is the toolbox for all other modules. Since the SHERPA framework does not rely on CLHEP etc., the ATOOLS contain classes with mathematical tools like vectors and matrices, organization tools such as read-in or write-out devices, and physics tools like particle data or classes for the event record.

- BEAM-1.0

  This module manages the treatment of the initial beam spectra for different colliders. At the moment two options are implemented for the beams: they can either be monochromatic, and therefore require no extra treatment, or, for the case of an electron collider, laser backscattering off the electrons is supported leading to photonic initial states[2].

- PDF-1.0

  In this module the handling of initial state radiation (ISR) is located. It provides interfaces to various proton [4, 5] and photon [6] parton density functions (PDFs), and to the LHAPDF interface (version 3) [7], which makes a full wealth of proton PDFs available. In addition, an (analytical) electron structure function is supplied there as well.

- MODEL-1.0,

  This module comprises the basic physics parameters (like masses, mixing angles, etc.) of the simulation run. Thus it specifies the corresponding physics model. At the moment three different physics models are supported: the Standard Model (SM), its Minimal Supersymmetric extension (MSSM) [8] and the ADD model of large extra dimensions [9]. For the SM and the ADD model most of the required parameters are directly read in. Within SHERPA an interface to the program HDecay for the calculation of Higgs boson width(s) in the SM and MSSM [10] is also provided but not linked to the main code. The same holds true for the MSSM, where an interface to the program Isasusy 7.67 [11] is provided but not linked by default. Rather than using ISAJET, the new version of SHERPA obtains its MSSM spectrum through SLHA conform files [12].

[1] However, two modules have no documentation yet, namely AMISIC++ and HADRONS++. Also, the documentation for the version 1.0 of AMEGIC [2], is not up to date anymore. Documentation for these modules should be made available in near future.

[2] It should be stressed, however, that the parametrization [3], which has been implemented, is valid for the proposed TESLA photon collider only. This is because various assumptions concerning the laser parameters and especially the initial energy of the electrons have been made.

- EXTRA_XS-1.0

  In this module a (limited) collection of analytic expressions for simple $2 \rightarrow 2$ processes within the SM and the corresponding classes embedding them into the SHERPA framework are provided. This includes methods used for the definition of the parton shower evolution, such as colour connections and the hard scale of the process. However, the classes for phase space integration, which are common with AMEGIC, are located in a special module called PHASIC.

- AMEGIC++-2.0

  AMEGIC [2] is SHERPA's preferred matrix element generator, which employs the method of helicity amplitudes [13, 14]. It works as a generator, which generates generators:
  During the initialization run the matrix elements for a set of given processes within the SM, the MSSM or the ADD model, as well as their specific phase space mappings are created by AMEGIC and stored in library files. In the initialization of the production run, these libraries are linked to the program. They are used to calculate cross sections and to generate single weighted or unweighted events based on them. Also, a directory may be specified to store the cross sections, the status of the phase space integrators, and the maximal weights of the processes. This information may be used to accelerate the potentially lengthy calculation phase for further event generation runs.
  AMEGIC has been tested for multi-particle production in the Standard Model [15] and its MSSM implementation has been tested in [16].

- PHASIC++-1.0

  Here all classes dealing with the Monte Carlo phase space integration are located. By default the adaptive multi-channel method of [17, 18] together with a Vegas optimization [19] for the single channels are used for the evaluation of the initial state (laser backscattering, initial state radiation) and final state integrals. In addition, final state integration accomplished by Rambo [20] and Sarge [21] is supported.

- APACIC++-2.0

  APACIC [22, 23] contains classes for the simulation of both the initial and the final state parton shower [3]. The shower evolution is organized through an ordering in the parton virtual mass. Coherence effects are accounted for by explicit ordering of the opening angles in subsequent branchings[4]. All features for a consistent merging with matrix elements [28, 29] are included, however the main part of the merging procedure is situated in the SHERPA module itself.

- AMISIC++-1.0

  AMISIC contains classes for the simulation of multiple parton interactions according to [30]. SHERPA extends the treatment of multiple interactions by allowing for the simultaneous evolution of an independent parton shower in each of the subsequent (semi-)hard collisions. This shower evolution is done by the APACIC module. The beam-beam remnants are organized such that partons which are adjacent in colour space are also adjacent in momentum space. The corresponding classes for beam remnant handling reside in the SHERPA module.

- HADRONS++-0.0

[3] From version 1.0.9. final state QED radiation off quarks and leptons can optionally be included

[4] This is similar to the way Pythia [24, 25] handles the parton shower. Herwig [26, 27] in contrast ensures coherence by taking angles as ordering parameter.

HADRONS++-0.0 is a module that treats hadron and τ-lepton decays. The resulting decay products respect full spin correlation to their mothers (if desired). For τ decays several models have been implemented (such as the Kühn-Santamaría Model and form factor parameterisations from Resonance Chiral Theorie). For further details, see section 4.3. For the hadrons so far no decay matrix elements or form factors have been implemented, thus all hadron decays that are switched on are treated as isotropic decays. At the moment, the branching ratios of the lightest mesons are given in corresponding data files. Anyways, currently Pythia still is invoked by default to deal with hadron decays.

- SHERPA-1.0

  Finally, SHERPA is the steering module that initializes, controls and evaluates the different phases in the entire process of event generation. Furthermore, all necessary routines for the combination of parton showers and matrix elements, which are independent of the specific parton shower are found in this module. For details concerning the implementation of these routines, cf. [31]. In addition, this subpackage provides an interface to the Lund String Fragmentation of Pythia 6.214 including its hadron decay routines.

- In addition to the modules of SHERPA, some scripts for alleviated installation of the package and easier handling are distributed. They are located in the subdirectory TOOLS. In almost all cases, only the makeinstall and the makedist script will be necessary.

The actual executable of the SHERPA generator is located in the subdirectory SHERPA-1.0.9/Run and is called Sherpa. Using the default setting, input files have to be provided in the current working directory, all the output files are then located in this directory as well.

2 Installing SHERPA

SHERPA is distributed as a tarred and gzipped file named Sherpa-<version>.tar.gz. The file for the latest version can be unpacked in the current working directory with

```
tar -zxvf Sherpa-1.0.9.tar.gz .
```

All modules stated above are located as directories in the current working directory. To install the framework, the makeinstall script is used, which is located in the subdirectory TOOLS. To guarantee a successful installation, the following tools should be available on the system: make, autoconf, automake and libtool. Furthermore, a FORTRAN compiler must be available. The authors recommend to use the following command line

```
./TOOLS/makeinstall -t
```

in order to start the installation. While processing the installation a log file called sherpa_install.log is created, which records the entire installation procedure. This is useful in cases where the automatic compilation process fails. Then, users are invited to send the log file as a bug report to the authors.
The makeinstall script provides a number of combinable options listed and described in Table 1, including a help option available through the flag -h.
If SHERPA should translate its events to the HepMC format [32], the CLHEP-package has to be linked. For this purpose the configure script in the SHERPA-1.0.9 directory relies on the correct setting of the system variable CLHEPDIR, usually something along the lines of /usr/local/CLHEP or similar. To run the makeinstall script properly, the --copt-option has to be specified with --enable-clhep, such that the corresponding call reads like

`./TOOLS/makeinstall --copt --enable-clhep` .

If the user prefers to employ and customize his/her own install script, `makeinstall` is equipped with the option `-s`, constructing a template of such a script.
The `SHERPA` package has successfully been compiled, installed and tested on SuSE, RedHat / Scientific Linux and Debian / Ubuntu Linux systems using the GNU C++ compiler versions 2.95/3.2/3.3 and 4.0 as well as on Mac OS X 10 using the GNU C++ compiler version 4.0. In any case the GNU FORTRAN compiler g77 has been employed.
GCC version 2.96 is **not** supported.
For those daring users, who modified parts of `SHERPA` according to their own needs or who added own modules and who would like to distribute the code, the `makedist` script is available. Its options are listed in Table 2. Note that a successful generation requires the following preparation. Before creating a distribution, all automatically generated process libraries must be removed. These libraries can be found in a directory called `./Process`, which is either situated in the current working directory or in the one explicitly specified by having set a path before. Removing the libraries is then simply accomplished through `rm -rf Process/`.

3 First runs with SHERPA

3.1 Running SHERPA

Whenever a new process is initialized in `SHERPA` – including the case of a freshly installed program using the default setup – `SHERPA` needs to run twice in order to calculate cross sections or to generate events. This is because in most cases the matrix elements are generated by `AMEGIC` during the first run. In this initialization run started by changing into the `./SHERPA-1.0.9/Run` directory and executing

`./Sherpa PATH=specified_dir` ,

the Feynman diagrams for the contemplable process(es) are constructed and translated into helicity

`-c`	configure before compiling (subset of '-t')
`-t`	rebuild 'Makefile.in's' and 'configure's'; configure before compiling
`-f`	display the full information on the setup
`-s`	create a simple install script to be customized by the user
`--clean`	execute 'make clean' in every module before compiling
`--no-abort`	ignore exit status of 'make'
`--copt`	define option for 'configure'
`--mopt`	define option for 'make' (default is '-j2')
`--cxx`	define CXX flag for 'make'
`--f`	define Fortran flag for 'make'
`--force-tar`	force extraction of tar files (enables '-e')
`--clean-up`	execute 'make clean' in every module and exit
`--rm-libs`	remove shared libraries; does not remove '.lo' files
`--make-only`	do not install
`-h`	display this help and exit

Table 1: Options of the `makeinstall` script.

`-f`	include full time information in distribution name
`-n <name>`	write compressed distribution to `<name>.tar.gz`
`--clean`	execute 'make clean' before 'make dist'
`--configure`	configure before 'make dist' (enabled with '–total')
`--total`	rebuild 'Makefile.in's' and 'configure's' before 'make dist'
`--copt`	define option for 'configure'
`--no-abort`	ignore exit status of 'make'
`-h`	display this help and exit

Table 2: Options of the `makedist` script.

amplitudes. Furthermore suitable phase space mappings are produced. The amplitudes and the integration channels are stored in library files that need to be compiled and linked to the framework. The library files are found in a directory `Process`, which is, together with all the data files, either in the current working directory or in the directory specified by the `PATH`, which may be taken as an absolute path or relative to the `./SHERPA-1.0.9/Run` directory. In order to carry out the compilation and linking procedure, the `makelibs` script located in the same directory where the `Process` directory can be found has to be run according to

`./makelibs` .

In the second run then the cross sections are evaluated and events are generated. In most cases it is advantageous to store the cross sections obtained during the production run for later use. This can be achieved through explicitly creating a result directory and giving an additional option in the command line via

`./Sherpa PATH=specified_dir RESULT_DIRECTORY=specified_dir/Result` .

However, two remarks are in order here:
In the case when e.g. a coupling has been changed, the `Result` directory has to be cleaned, in order to correctly get the new cross section.
On top of this, whenever the physics setup is considerably changed, i.e. whenever particles are switched on or off or whenever the physics model is changed, it is absolutely mandatory to also erase the library files. For this purpose the `Process` (and `Result`) directory located in the current working directory or in the one given through the (respective) `PATH` has (have) to be erased by applying

`rm -rf Process/` (and `rm -rf Result/`) .

The urge to erase old library files is because in principle they are blind to the specific physics environment they are created in and meant for. So (ab)using them in another frame will in nearly all cases lead to drastically wrong results or to the crashing of `SHERPA`.

3.2 Default setup: Hadron production in $e\gamma$ collisions

3.2.1 The setup

The distributed version of `SHERPA` comes with a default setup[5] for the following type of process to be simulated within the SM:

$$e^-e^- \to e^-\gamma \to e^-\gamma \to e^-d\bar{d} \to e^- + \text{hadrons}\,,$$

where in the first step the photon is produced through laser backscattering off one of the incoming electrons. In the second step the other electron experiences some initial state radiation according to a structure function (S.F.), before the electron and photon produce a $d\bar{d}$ pair through some kind of peripheral scattering. In calculating the cross section for this process and for the generation of events, some cuts are applied on the initial and final state, namely

$$\begin{array}{lllllllll} E_\perp(e^-) & > & 1\text{GeV}, & E_\perp(d) & > & 1\text{GeV}, & E_\perp(\bar{d}) & > & 1\text{GeV}, \\ \cos\theta_{e^-} & \in & [-0.9,\,0.9], & \cos\theta_d & \in & [-0.9,\,0.9], & \cos\theta_{\bar{d}} & \in & [-0.9,\,0.9]\,, \end{array}$$

and

$$y_{\text{cut}} \;=\; 0.004 \;\; (\text{Durham algorithm})\,,$$

where the angles are w.r.t. the positive beam axis. Then, the parton shower starts before the resulting final state partons are afterwards translated into hadrons. To modify this default process or to set up `SHERPA` for a completely different collider with a different process in a different model, various data files (of the form `*.dat`) may be changed. Details and instructions for doing this can be found in Sec. 4.
The default setup comes in the output mode "info+events" (`OUTPUT=3`). The output generated in the "info" mode contains the very basic information about the status of the generator. In the "events" mode, the events are printed on screen. If also the "tracking" mode is enabled (see Table 3) `SHERPA` lets the user know what it is currently working on. In particular, this includes

- messages concerning the modules that have been initialized for the run with some basic information about the setup;
- a frequent update of the result of eventual integration of processes that have been initialized.

For more options to output events in different formats and for information on how to interpret the generic `SHERPA` output, cf. Sec. 6.

3.2.2 First run

For the first – the initialization run – the output would look like

```
$ ./Sherpa PATH=EGamma/
Welcome to Sherpa, User Name. Initialization of framework underway.
-----------------------------------------------------------------------------
-----------    Event generation run with SHERPA started .......   -----------
-----------------------------------------------------------------------------
............................................... |       +
............................................... ||  |       +  +
.................................       ....  | |        /   +
................ ................   _,_ |  ....  ||        +|  +  +
..............................  __.'  ,\|  ...  ||    /    +|         +
............................. (  \    \   ...  | |  |   + + \         +
```

[5] The corresponding data files are located in the `SHERPA-1.0.9/Run` directory.

```
.............................  (    \   -/  .... ||       +    |          +
........ .................... <S   /()))))~~~~~~~~##     +     /\    +
............................ (!H   (~~)))))~~~~~~#/     +  +    |  +
................ ........... (!E   (~~~)))))     /|/    +         +
............................ (!R   (~~~)))))   |||   + +           +
..... ...................... (!P    (~~~~)))   /|  + +          +
............................ (!A>    (~~~~~~~~~##        + +        +
............................. ~~(!    '~~~~~~~~ \       +     + +      +
...............................  `~~~QQQQQDb //   |         + + +        +
...................... ..........   IDDDDP||     \  + + + + +            +
...................................  IDDDI||       \                    +
................................... IHD HD||        \ + +  + + + + +      +
..................................  IHD ##|            :-) + +\          +
......... .............. ......... IHI ## /      /   +  + + + +\       +
.................................. IHI/ /       /      + + + +        +
.................................. ## | | /    / + +      + + /      +
...................... /TT\ .....  ##/ ///  / + + + + + + +/       +
......................./TTT/T\ ... /TT\/\\\ / + + + + + + +/   \         +
version 1.0.9 ......../TTT/TTTT\...|TT/T\\\/   +    ++  + /
-----------------------------------------------------------------------------

     SHERPA version 1.0.9

     Authors:        Tanju Gleisberg, Stefan Hoeche, Frank Krauss,
                     Steffen Schumann, Frank Siegert, Jan Winter
     Former Authors: Timo Fischer, Ralf Kuhn, Thomas Laubrich,
                     Andreas Schaelicke

     This program uses a lot of genuine and original research work
     by other people. Users are encouraged to refer to
          the various original publications.

          Users are kindly asked to refer to the documentation
          published under JHEP 0402 (2004) 056.

          Please visit also our homepage

       http://www.sherpa-mc.de

          for news, bugreports, updates and new releases.

-----------------------------------------------------------------------------

Initialize the Standard Model from EGamma/ / Model.dat
Initialized the beams Laser_Backscattering*Monochromatic
Initialized the ISR[1] : (None)*(SF)
Initialized the ISR[2] : (None)*(SF)
Initialized the Beam_Remnant_Handler.
Amegic::InitializeProcesses :
   Process initialization started; new libraries may be created.
ERROR in Single_Process::Tests()
   No compiled & linked library found for process P2_3_8_7_100_32_13_0
   but files already written out !
   Interrupt run and execute "makelibs" in '/path/to/SHERPA-1.0/Run/EGamma'.
Sherpa: Single_Process::Tests throws normal exit:
   Failed to load library.
Return_Value::PrintStatistics(): Statistics {
  Generated events: 0
}
Time: 0.43 s (clocks=43) on Mon Nov  6 12:49:33 2006
 (User: 0.05 s ,System: 0.02 s ,Children User: 0 s ,Children System: 0.01)
```

3.2.3 Second run

By performing `./makelibs` in the mentioned directory the freshly generated process libraries will be compiled and stored as shared libraries. Restarting the run produces the following output:

```
$ ./Sherpa PATH=EGamma/
Welcome to Sherpa, User Name. Initialization of framework underway.
----------------------------------------------------------------------------
-----------    Event generation run with SHERPA started .......   -----------
----------------------------------------------------------------------------
............................................... |       +
............................................... ||  |       +  +
.................................        ....  | |         /   +
................. ...............   _,_ |  ....  ||         +|  +  +
..............................  __.'  ,\|  ...  ||    /    +|          +
............................ (  \    \   ...  | |  |   + + \         +
...........................  (    \   -/  .... ||       +    |          +
........ ..................  <S   /()))))~~~~~~~~##     +     /\    +
............................ (!H   (~~)))))~~~~~~#/     +  +    |  +
................ ........... (!E   (~~~)))))     /|/    +         +
............................ (!R   (~~~)))))   |||   + +           +
..... ...................... (!P    (~~~~)))   /|  + +          +
............................ (!A>    (~~~~~~~~~##        + +        +
............................. ~~(!    '~~~~~~~~ \       +     + +      +
...............................  `~~~QQQQQDb //   |         + + +        +
....................... ..........   IDDDDP||     \  + + + + +            +
..................................   IDDDI||      \                      +
.................................. IHD HD||        \ + +  + + + + +      +
.................................  IHD ##|           :-) + +\          +
......... .............. ......... IHI ## /      /   +  + + + +\       +
.................................. IHI/ /       /      + + + +        +
.................................. ## | | /    / + +      + + /      +
....................... /TT\ .....  ##/ ///  / + + + + + + +/       +
......................./TTT/T\ ... /TT\/\\\ / + + + + + + +/   \         +
version 1.0.9 ......../TTT/TTTT\...|TT/T\\\/   +    ++  + /
----------------------------------------------------------------------------

     SHERPA version 1.0.9

     Authors:        Tanju Gleisberg, Stefan Hoeche, Frank Krauss,
                     Steffen Schumann, Frank Siegert, Jan Winter
     Former Authors: Timo Fischer, Ralf Kuhn, Thomas Laubrich,
                     Andreas Schaelicke

     This program uses a lot of genuine and original research work
     by other people. Users are encouraged to refer to
          the various original publications.

          Users are kindly asked to refer to the documentation
          published under JHEP 0402 (2004) 056.

          Please visit also our homepage

       http://www.sherpa-mc.de

          for news, bugreports, updates and new releases.

----------------------------------------------------------------------------

Initialize the Standard Model from EGamma/ / Model.dat
Initialized the beams Laser_Backscattering*Monochromatic
Initialized the ISR[1] : (None)*(SF)
```

```
Initialized the ISR[2] : (None)*(SF)
Initialized the Beam_Remnant_Handler.
Amegic::InitializeProcesses :
   Process initialization started; new libraries may be created.
VP2_3_8_7_100_32_13_0 loaded.
   No new libraries have been created.
Initialized the Matrix_Element_Handler for the hard processes : Amegic
Initialized the Shower_Handler.
Initialized the Fragmentation_Handler.
Lund_Interface::AdjustProperties Adjusted mass of f(0)(980) (10221) from 0.98 to 1 to allow Pythia decays.
Lund_Interface::AdjustProperties Adjusted mass of K(1)(1270) (10313) from 1.272 to 1.29 to allow Pythia decays.
Lund_Interface::AdjustProperties Adjusted mass of K(1)(1270)+ (10323) from 1.272 to 1.29 to allow Pythia decays.
Lund_Interface::AdjustProperties Adjusted mass of h(1)(1380) (10333) from 1.386 to 1.4 to allow Pythia decays.
Lund_Interface::AdjustProperties Adjusted mass of D_1(H) (20423) from 2.43267 to 2.372 to allow Pythia decays.
Initialized the Hadron_Decay_Handler, Decay model = Lund
=========================================================================
Start calculating the hard cross sections. This may take some time.
All_Processes::CalculateTotalXSec for 2_3_P_e-___e-_d_db
In Single_Process::CalculateTotalXSec(./Results) for 2_3_P_e-___e-_d_db
Starting the calculation. Lean back and enjoy ... .
0.0559047 pb +- ( 0.00201617 pb = 3.60644 % ) 5000
full optimization:  ( 1 s elapsed / 75 s left / 76 s total )
0.0556218 pb +- ( 0.00121102 pb = 2.17725 % ) 10000
full optimization:  ( 2 s elapsed / 71 s left / 73 s total )
0.0548822 pb +- ( 0.000690785 pb = 1.25867 % ) 15000
full optimization:  ( 3 s elapsed / 68 s left / 71 s total )
0.0556203 pb +- ( 0.000538071 pb = 0.967401 % ) 20000
full optimization:  ( 4 s elapsed / 66 s left / 70 s total )
0.0553835 pb +- ( 0.000448293 pb = 0.809434 % ) 25000
full optimization:  ( 5 s elapsed / 65 s left / 70 s total )
0.0551784 pb +- ( 0.000388052 pb = 0.703268 % ) 30000
full optimization:  ( 6 s elapsed / 64 s left / 70 s total )
0.0551378 pb +- ( 0.000296605 pb = 0.537935 % ) 40000
full optimization:  ( 9 s elapsed / 61 s left / 70 s total )
0.0551541 pb +- ( 0.000242959 pb = 0.440508 % ) 50000
full optimization:  ( 11 s elapsed / 59 s left / 70 s total )
0.0548977 pb +- ( 0.000207347 pb = 0.377696 % ) 60000
full optimization:  ( 13 s elapsed / 57 s left / 70 s total )
0.0549955 pb +- ( 0.000184597 pb = 0.335659 % ) 70000
full optimization:  ( 15 s elapsed / 55 s left / 70 s total )
0.0549855 pb +- ( 0.000167178 pb = 0.304041 % ) 80000
full optimization:  ( 18 s elapsed / 52 s left / 70 s total )
0.0549393 pb +- ( 0.00015351 pb = 0.279418 % ) 90000
full optimization:  ( 20 s elapsed / 50 s left / 70 s total )
0.0549762 pb +- ( 0.00014336 pb = 0.260767 % ) 100000
full optimization:  ( 22 s elapsed / 48 s left / 70 s total )
0.0549787 pb +- ( 0.00013448 pb = 0.244604 % ) 110000
full optimization:  ( 25 s elapsed / 45 s left / 70 s total )
0.0550293 pb +- ( 0.000127277 pb = 0.23129 % ) 120000
full optimization:  ( 27 s elapsed / 43 s left / 70 s total )
0.0549893 pb +- ( 0.000121199 pb = 0.220405 % ) 130000
full optimization:  ( 29 s elapsed / 41 s left / 70 s total )
0.0549795 pb +- ( 0.000115765 pb = 0.210559 % ) 140000
full optimization:  ( 31 s elapsed / 39 s left / 70 s total )
0.0550212 pb +- ( 0.000111065 pb = 0.201858 % ) 150000
full optimization:  ( 34 s elapsed / 36 s left / 70 s total )
0.0549912 pb +- ( 0.000106724 pb = 0.194075 % ) 160000
full optimization:  ( 36 s elapsed / 34 s left / 70 s total )
0.054969 pb +- ( 0.00010291 pb = 0.187214 % ) 170000
full optimization:  ( 38 s elapsed / 32 s left / 70 s total )
0.0549498 pb +- ( 9.95174e-05 pb = 0.181106 % ) 180000
full optimization:  ( 41 s elapsed / 29 s left / 70 s total )
```

```
0.0549529 pb +- ( 9.64683e-05 pb = 0.175547 % ) 190000
full optimization:  ( 43 s elapsed / 27 s left / 70 s total )
0.054967 pb +- ( 9.36718e-05 pb = 0.170414 % ) 200000
full optimization:  ( 45 s elapsed / 25 s left / 70 s total )
0.0549447 pb +- ( 9.10524e-05 pb = 0.165716 % ) 210000
full optimization:  ( 47 s elapsed / 23 s left / 70 s total )
0.0549531 pb +- ( 8.86975e-05 pb = 0.161406 % ) 220000
full optimization:  ( 50 s elapsed / 20 s left / 70 s total )
0.0549066 pb +- ( 8.63996e-05 pb = 0.157357 % ) 230000
full optimization:  ( 52 s elapsed / 18 s left / 70 s total )
0.0549186 pb +- ( 8.4333e-05 pb = 0.15356 % ) 240000
integration time:  ( 54 s elapsed / 16 s left / 70 s total )
0.0549066 pb +- ( 8.0704e-05 pb = 0.146984 % ) 260000
integration time:  ( 59 s elapsed / 11 s left / 70 s total )
0.0548865 pb +- ( 7.74025e-05 pb = 0.141023 % ) 280000
integration time:  ( 63 s elapsed / 7 s left / 70 s total )
0.054845 pb +- ( 7.44725e-05 pb = 0.135787 % ) 300000
integration time:  ( 68 s elapsed / 2 s left / 70 s total )
0.054853 pb +- ( 7.31674e-05 pb = 0.133388 % ) 310000
integration time:  ( 70 s elapsed / 0 s left / 70 s total )
2_3_P_e-___e-_d_db : 0.054853 pb +/- 0.133388 %, exp. eff: 5.67728 %.
Store result : xs for 2_3_P_e-___e-_d_db : 0.054853 pb +/- 0.133388%,
       max : 2.48134e-09
Calculating the hard cross sections has been successful.
==========================================================================
==========================================================================
Sherpa will start event generation now : 2 events
==========================================================================
----------------------------------------------------------
-- SHERPA generates events with the following structure --
----------------------------------------------------------
Perturbative       : Signal_Processes:Amegic
Perturbative       : Hard_Decays:
Perturbative       : Jet_Evolution:Apacic
Perturbative       : Multiple_Interactions:None
Hadronization      : Beam_Remnants
Hadronization      : Hadronization Lund
Hadronization      : Hadron_Decays
---------------------------------------------------------
```

This is followed by the output of the two events. Note that, by default the integration results have been stored in the file `./Results/MC_2_3_P_e-___e-_d_db`[6].

3.2.4 Third run

In a further run with the same setup, the integration results are read in; the output then looks like:

```
$ ./Sherpa PATH=EGamma/
Welcome to Sherpa, User Name. Initialization of framework underway.
-----------------------------------------------------------------------------
-----------    Event generation run with SHERPA started .......   -----------
-----------------------------------------------------------------------------
............................................... |       +
............................................... ||  |       +  +
...................................        ....  | |         /   +
................. ................   _,_ |  ....  ||         +|  +  +
```

[6]Storing the integration results implies that the user wants to keep these information. For this, the directory `Results` has to be created by the user himself before running `SHERPA` either in the current working directory or in the one specified by the `PATH`. Furthermore, this directory has to be emptied whenever a parameter, such as the c.m. energy, has been changed so that the cross section needs to be recalculated.

```
.............................  __.'  ,\|  ...  ||    /   +|          +
............................ (  \    \   ...  | |  |   + + \         +
...........................  (    \   -/  .... ||       +    |         +
........ .................  <S   /()))))~~~~~~~~##     +     /\    +
.......................... (!H   (~~)))))~~~~~~#/     +  +    |  +
............... .......... (!E   (~~~)))))     /|/    +         +
.......................... (!R   (~~~)))))   |||   + +           +
..... ..................... (!P    (~~~~)))   /|  + +           +
.......................... (!A>    (~~~~~~~~~##        + +        +
...........................  ~~(!    '~~~~~~~ \       +     + +      +
.............................  `~~~QQQQQDb //   |         + + +        +
...................... .........   IDDDDP||     \  + + + + +            +
................................   IDDDI||       \                     +
................................. IHD HD||        \ + +  + + + + +      +
................................  IHD ##|           :-) + +\          +
........ ............. ......... IHI ## /      /   +  + + + +\       +
................................ IHI/ /       /      + + + +        +
................................ ## | | /    / + +      + + /      +
..................... /TT\ .....  ##/ ///  / + + + + + + +/       +
...................../TTT/T\ ... /TT\/\\\ / + + + + + + +/   \         +
version 1.0.9 ......../TTT/TTTT\...|TT/T\\\/   +    ++  + /
-----------------------------------------------------------------------------

     SHERPA version 1.0.9

     Authors:        Tanju Gleisberg, Stefan Hoeche, Frank Krauss,
                     Steffen Schumann, Frank Siegert, Jan Winter
     Former Authors: Timo Fischer, Ralf Kuhn, Thomas Laubrich,
                     Andreas Schaelicke

     This program uses a lot of genuine and original research work
     by other people. Users are encouraged to refer to
          the various original publications.

          Users are kindly asked to refer to the documentation
          published under JHEP 0402 (2004) 056.

          Please visit also our homepage

       http://www.sherpa-mc.de

          for news, bugreports, updates and new releases.

-----------------------------------------------------------------------------

Initialize the Standard Model from EGamma/ / Model.dat
Initialized the beams Laser_Backscattering*Monochromatic
Initialized the ISR[1] : (None)*(SF)
Initialized the ISR[2] : (None)*(SF)
Initialized the Beam_Remnant_Handler.
Amegic::InitializeProcesses :
   Process initialization started; new libraries may be created.
VP2_3_8_7_100_32_13_0 loaded.
   No new libraries have been created.
Initialized the Matrix_Element_Handler for the hard processes : Amegic
Initialized the Shower_Handler.
Initialized the Fragmentation_Handler.
Lund_Interface::AdjustProperties Adjusted mass of f(0)(980) (10221) from 0.98 to 1 to allow Pythia decays.
Lund_Interface::AdjustProperties Adjusted mass of K(1)(1270) (10313) from 1.272 to 1.29 to allow Pythia decays.
Lund_Interface::AdjustProperties Adjusted mass of K(1)(1270)+ (10323) from 1.272 to 1.29 to allow Pythia decays.
Lund_Interface::AdjustProperties Adjusted mass of h(1)(1380) (10333) from 1.386 to 1.4 to allow Pythia decays.
Lund_Interface::AdjustProperties Adjusted mass of D_1(H) (20423) from 2.43267 to 2.372 to allow Pythia decays.
```

```
Initialized the Hadron_Decay_Handler, Decay model = Lund
=========================================================================
Start calculating the hard cross sections. This may take some time.
All_Processes::CalculateTotalXSec for 2_3_P_e-___e-_d_db
In Single_Process::CalculateTotalXSec(./Results) for 2_3_P_e-___e-_d_db
Read in channels from directory : ./Results/MC_2_3_P_e-___e-_d_db
ERROR in Single_Process::CalculateTotalXSec :
   Result of PS-Integrator and internal summation do not coincide for
2_3_P_e-___e-_d_db : 1.40873e-10 vs. 1.40873e-10
2_3_P_e-___e-_d_db : 0.054853 pb +/- 0.133388 %, exp. eff: 5.67728 %.
Calculating the hard cross sections has been successful.
=========================================================================
=========================================================================
Sherpa will start event generation now : 2 events
=========================================================================
---------------------------------------------------------
-- SHERPA generates events with the following structure --
---------------------------------------------------------
Perturbative       : Signal_Processes:Amegic
Perturbative       : Hard_Decays:
Perturbative       : Jet_Evolution:Apacic
Perturbative       : Multiple_Interactions:None
Hadronization      : Beam_Remnants
Hadronization      : Hadronization Lund
Hadronization      : Hadron_Decays
--------------------------------------------------------
```

This is again followed by the event output.

3.2.5 Event output

The event output is listed as:

```
Blob [0]( 11, Bunch            ,  from Beam 0, 1 -> 2 @ (0,0,0,0)
Incoming particles :
[a] 2 e-                   5 (     ->   11) [( 2.5000e+02, 0.0000e+00, 0.0000e+00, 2.5000e+02), p^2= 2.6113e-07, m= 5.1100e-04] (  0,  0)
Outgoing particles :
[a] 2 photon               2 (  11 ->    8) [( 2.0501e+02, 0.0000e+00, 0.0000e+00, 2.0501e+02), p^2= 0.0000e+00, m= 0.0000e+00] (  0,  0)
[a] 1 e-                   6 (  11 ->     ) [( 4.4987e+01, 0.0000e+00, 0.0000e+00, 4.4987e+01), p^2= 4.6989e-08, m= 5.1100e-04] (  0,  0)

Blob [0]( 10, Bunch            ,  from Beam 1, 1 -> 1 @ (0,0,0,0)
Incoming particles :
[a] 2 e-                   3 (     ->   10) [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), p^2= 2.6113e-07, m= 5.1100e-04] (  0,  0)
Outgoing particles :
[a] 2 e-                   3 (  10 ->    9) [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), p^2= 2.6113e-07, m= 5.1100e-04] (  0,  0)

Blob [16]( 9, Beam             ,  from Beam 1, 1 -> 2 @ (0,0,0,0)
Incoming particles :
[a] 2 e-                   3 (  10 ->    9) [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), p^2= 2.6113e-07, m= 5.1100e-04] (  0,  0)
Outgoing particles :
[G] 2 e-                   1 (   9 ->    7) [( 2.4999e+02, 0.0000e+00, 0.0000e+00,-2.4999e+02), p^2= 0.0000e+00, m= 0.0000e+00] (  0,  0)
[F] 1 photon               4 (   9 ->     ) [( 5.9660e-03, 0.0000e+00, 0.0000e+00,-5.9660e-03), p^2= 6.2315e-12, m= 0.0000e+00] (  0,  0)

Blob [16]( 8, Beam             ,  from Beam 0, 1 -> 1 @ (0,0,0,0)
Incoming particles :
[a] 2 photon               2 (  11 ->    8) [( 2.0501e+02, 0.0000e+00, 0.0000e+00, 2.0501e+02), p^2= 0.0000e+00, m= 0.0000e+00] (  0,  0)
Outgoing particles :
[G] 2 photon               1 (   8 ->    6) [( 2.0501e+02, 0.0000e+00, 0.0000e+00, 2.0501e+02), p^2= 1.4552e-11, m= 0.0000e+00] (  0,  0)

Blob [0]( 7, IS Shower        ,  from Beam 1, 1 -> 1 @ (0,0,0,0)
Incoming particles :
[G] 2 e-                   1 (   9 ->    7) [( 2.4999e+02, 0.0000e+00, 0.0000e+00,-2.4999e+02), p^2= 0.0000e+00, m= 0.0000e+00] (  0,  0)
Outgoing particles :
[G] 2 e-                   1 (   7 ->    1) [( 2.4999e+02, 0.0000e+00, 0.0000e+00,-2.4999e+02), p^2= 0.0000e+00, m= 0.0000e+00] (  0,  0)

Blob [0]( 6, IS Shower        ,  from Beam 0, 1 -> 1 @ (0,0,0,0)
Incoming particles :
```

```
[G] 2 photon               1 (   8 ->   6) [( 2.0501e+02, 0.0000e+00, 0.0000e+00, 2.0501e+02), p^2= 1.4552e-11, m= 0.0000e+00] (  0,  0)
Outgoing particles :
[G] 2 photon               1 (   6 ->   1) [( 2.0501e+02, 0.0000e+00, 0.0000e+00, 2.0501e+02), p^2= 1.4552e-11, m= 0.0000e+00] (  0,  0)

Blob [0]( 1, Signal Process    , 2 -> 3 @ (0,0,0,0)
Incoming particles :
[G] 2 photon               1 (   6 ->   1) [( 2.0501e+02, 0.0000e+00, 0.0000e+00, 2.0501e+02), p^2= 1.4552e-11, m= 0.0000e+00] (  0,  0)
[G] 2 e-                   1 (   7 ->   1) [( 2.4999e+02, 0.0000e+00, 0.0000e+00,-2.4999e+02), p^2= 0.0000e+00, m= 0.0000e+00] (  0,  0)
Outgoing particles :
[H] 2 e-                   2 (   1 ->   2) [( 2.1672e+02,-2.0139e+02,-7.9909e+01,-4.7731e+00), p^2= 7.2760e-12, m= 0.0000e+00] (  0,  0)
[H] 2 d                    3 (   1 ->   2) [( 1.9167e+02, 1.5859e+02, 9.4568e+01,-5.1435e+01), p^2= 0.0000e+00, m= 0.0000e+00] (601,  0)
[H] 2 anti-d               4 (   1 ->   2) [( 4.6615e+01, 4.2803e+01,-1.4659e+01, 1.1226e+01), p^2= 9.0949e-13, m= 0.0000e+00] (  0,601)
Data_Container:
   * Factorisation_Scale (2.05e+05)
   * ME_NumberOfTrials (1)
   * ME_Weight (1)
   * Process_Weight (1)
   * XS_NumberOfTrials (2)
   * XS_Weight (0.05485)

Blob [0]( 2, ME PS Interface FS, 3 -> 3 @ (0,0,0,0)
Incoming particles :
[H] 2 e-                   2 (   1 ->   2) [( 2.1672e+02,-2.0139e+02,-7.9909e+01,-4.7731e+00), p^2= 7.2760e-12, m= 0.0000e+00] (  0,  0)
[H] 2 d                    3 (   1 ->   2) [( 1.9167e+02, 1.5859e+02, 9.4568e+01,-5.1435e+01), p^2= 0.0000e+00, m= 0.0000e+00] (601,  0)
[H] 2 anti-d               4 (   1 ->   2) [( 4.6615e+01, 4.2803e+01,-1.4659e+01, 1.1226e+01), p^2= 9.0949e-13, m= 0.0000e+00] (  0,601)
Outgoing particles :
[H] 1 e-                   2 (   2 ->   3) [( 2.1672e+02,-2.0139e+02,-7.9909e+01,-4.7731e+00), p^2= 7.2760e-12, m= 3.1291e-06] (  0,  0)
[H] 2 d                    3 (   2 ->   4) [( 1.9273e+02, 1.5956e+02, 9.4236e+01,-5.1181e+01), p^2= 1.8616e+02, m= 1.3644e+01] (601,  0)
[H] 1 anti-d               4 (   2 ->   5) [( 4.5560e+01, 4.1833e+01,-1.4327e+01, 1.0972e+01), p^2= 1.0000e-04, m= 1.0000e-02] (  0,601)
Data_Container:
   * Core_Process (0)
   * OrderEWeak (2)
   * OrderStrong (0)

Blob [0]( 3, FS Shower         , 1 -> 1 @ (0,0,0,0)
Incoming particles :
[H] 1 e-                   2 (   2 ->   3) [( 2.1672e+02,-2.0139e+02,-7.9909e+01,-4.7731e+00), p^2= 7.2760e-12, m= 3.1291e-06] (  0,  0)
Outgoing particles :
[H] 1 e-                   2 (   3 ->    ) [( 2.1672e+02,-2.0139e+02,-7.9909e+01,-4.7731e+00), p^2= 7.2760e-12, m= 3.1291e-06] (  0,  0)
Data_Container:
   * FS_VS ((17,5,5,5,1))
   * IFS_VS ((<no entry>))

Blob [0]( 4, FS Shower         , 1 -> 4 @ (0,0,0,0)
Incoming particles :
[H] 2 d                    3 (   2 ->   4) [( 1.9273e+02, 1.5956e+02, 9.4236e+01,-5.1181e+01), p^2= 1.8616e+02, m= 1.3644e+01] (601,  0)
Outgoing particles :
[F] 4 d                   12 (   4 ->  12) [( 1.7151e+02, 1.4294e+02, 8.3203e+01,-4.5373e+01), p^2= 1.0000e-04, m= 1.0000e-02] (604,  0)
[F] 4 gluon               13 (   4 ->  12) [( 1.0723e+01, 9.0737e+00, 5.5458e+00,-1.3771e+00), p^2= 1.4211e-14, m= 1.6298e-07] (603,604)
[F] 4 gluon               11 (   4 ->  12) [( 7.6894e+00, 5.3269e+00, 4.7492e+00,-2.8629e+00), p^2= 2.1316e-14, m= 1.0261e-07] (602,603)
[F] 4 gluon                9 (   4 ->  12) [( 2.8107e+00, 2.2130e+00, 7.3803e-01,-1.5678e+00), p^2= 0.0000e+00, m= 1.4681e-08] (601,602)
Data_Container:
   * FS_VS ((17,5,5,5,1))
   * IFS_VS ((<no entry>))

Blob [0]( 5, FS Shower         , 1 -> 1 @ (0,0,0,0)
Incoming particles :
[H] 1 anti-d               4 (   2 ->   5) [( 4.5560e+01, 4.1833e+01,-1.4327e+01, 1.0972e+01), p^2= 1.0000e-04, m= 1.0000e-02] (  0,601)
Outgoing particles :
[H] 4 anti-d               4 (   5 ->  12) [( 4.5560e+01, 4.1833e+01,-1.4327e+01, 1.0972e+01), p^2= 1.0000e-04, m= 1.0000e-02] (  0,601)
Data_Container:
   * FS_VS ((17,5,5,5,1))
   * IFS_VS ((<no entry>))

Blob [0]( 12, Fragmentation     , 5 -> 14 @ (0,0,0,0)
Incoming particles :
[F] 4 d                   12 (   4 ->  12) [( 1.7151e+02, 1.4294e+02, 8.3203e+01,-4.5373e+01), p^2= 1.0000e-04, m= 1.0000e-02] (604,  0)
[F] 4 gluon               13 (   4 ->  12) [( 1.0723e+01, 9.0737e+00, 5.5458e+00,-1.3771e+00), p^2= 1.4211e-14, m= 1.6298e-07] (603,604)
[F] 4 gluon               11 (   4 ->  12) [( 7.6894e+00, 5.3269e+00, 4.7492e+00,-2.8629e+00), p^2= 2.1316e-14, m= 1.0261e-07] (602,603)
[F] 4 gluon                9 (   4 ->  12) [( 2.8107e+00, 2.2130e+00, 7.3803e-01,-1.5678e+00), p^2= 0.0000e+00, m= 1.4681e-08] (601,602)
[H] 4 anti-d               4 (   5 ->  12) [( 4.5560e+01, 4.1833e+01,-1.4327e+01, 1.0972e+01), p^2= 1.0000e-04, m= 1.0000e-02] (  0,601)
Outgoing particles :
[P] 1 pi-                  7 (  12 ->    ) [( 3.2969e+01, 2.7324e+01, 1.6099e+01,-9.0095e+00), p^2= 1.9480e-02, m= 1.3957e-01] (  0,  0)
[P] 2 rho(770)+            8 (  12 ->  20) [( 4.0172e+01, 3.3459e+01, 1.9589e+01,-1.0477e+01), p^2= 7.6402e-01, m= 8.7408e-01] (  0,  0)
```

```
[P] 2 eta                 9 (  12 ->   16) [( 6.0606e+01, 5.0576e+01, 2.9490e+01,-1.5659e+01), p^2= 2.9970e-01, m= 5.4745e-01] (  0,  0)
[P] 2 K                  10 (  12 ->   13) [( 3.7314e+01, 3.1275e+01, 1.8171e+01,-9.1550e+00), p^2= 2.4768e-01, m= 4.9767e-01] (  0,  0)
[P] 2 K*(892)-           11 (  12 ->   19) [( 5.9050e+00, 4.7438e+00, 2.8203e+00,-1.9097e+00), p^2= 7.6468e-01, m= 8.7446e-01] (  0,  0)
[P] 2 rho(770)+          12 (  12 ->   21) [( 9.3191e+00, 7.5084e+00, 5.2011e+00,-1.6834e+00), p^2= 5.8235e-01, m= 7.6312e-01] (  0,  0)
[P] 2 pi                 13 (  12 ->   14) [( 1.5660e+00, 1.0654e+00, 7.7691e-01,-8.3400e-01), p^2= 1.8220e-02, m= 1.3498e-01] (  0,  0)
[P] 1 pi-                14 (  12 ->     ) [( 2.0171e+00, 1.3131e+00, 8.7171e-01,-1.2510e+00), p^2= 1.9480e-02, m= 1.3957e-01] (  0,  0)
[P] 2 pi                 15 (  12 ->   15) [( 1.8248e+00, 1.4168e+00, 1.1394e+00, 7.8949e-02), p^2= 1.8220e-02, m= 1.3498e-01] (  0,  0)
[P] 2 rho(770)+          16 (  12 ->   22) [( 6.0980e+00, 5.8017e+00,-1.4346e+00, 9.5078e-01), p^2= 5.6376e-01, m= 7.5084e-01] (  0,  0)
[P] 2 eta                17 (  12 ->   17) [( 3.0112e+00, 2.7013e+00,-1.2047e+00,-1.3798e-01), p^2= 2.9970e-01, m= 5.4745e-01] (  0,  0)
[P] 2 eta                18 (  12 ->   18) [( 1.3618e+00, 1.1528e+00,-3.5741e-01,-3.1352e-01), p^2= 2.9970e-01, m= 5.4745e-01] (  0,  0)
[P] 2 rho(770)-          19 (  12 ->   23) [( 2.8601e+01, 2.6097e+01,-8.9021e+00, 7.5592e+00), p^2= 5.5399e-01, m= 7.4430e-01] (  0,  0)
[P] 1 pi+                20 (  12 ->     ) [( 7.5244e+00, 6.9572e+00,-2.3510e+00, 1.6331e+00), p^2= 1.9480e-02, m= 1.3957e-01] (  0,  0)

Blob [0]( 13, Hadron Decay      , 1 -> 1 @ (0,0,0,0)
Incoming particles :
[P] 2 K                  10 (  12 ->   13) [( 3.7314e+01, 3.1275e+01, 1.8171e+01,-9.1550e+00), p^2= 2.4768e-01, m= 4.9767e-01] (  0,  0)
Outgoing particles :
[P] 2 K(S)               21 (  13 ->   24) [( 3.7314e+01, 3.1275e+01, 1.8171e+01,-9.1550e+00), p^2= 2.4768e-01, m= 4.9767e-01] (  0,  0)

Blob [0]( 14, Hadron Decay      , 1 -> 2 @ (0.0002694,0.0001833,0.0001336,-0.0001435)
Incoming particles :
[P] 2 pi                 13 (  12 ->   14) [( 1.5660e+00, 1.0654e+00, 7.7691e-01,-8.3400e-01), p^2= 1.8220e-02, m= 1.3498e-01] (  0,  0)
Outgoing particles :
[P] 1 photon             22 (  14 ->     ) [( 1.0045e+00, 6.6073e-01, 5.5523e-01,-5.1398e-01), p^2= 2.2204e-16, m= 0.0000e+00] (  0,  0)
[P] 1 photon             23 (  14 ->     ) [( 5.6154e-01, 4.0468e-01, 2.2168e-01,-3.2002e-01), p^2=-5.5511e-17, m= 0.0000e+00] (  0,  0)

Blob [0]( 15, Hadron Decay      , 1 -> 3 @ (0.0002264,0.0001758,0.0001413,9.794e-06)
Incoming particles :
[P] 2 pi                 15 (  12 ->   15) [( 1.8248e+00, 1.4168e+00, 1.1394e+00, 7.8949e-02), p^2= 1.8220e-02, m= 1.3498e-01] (  0,  0)
Outgoing particles :
[P] 1 photon             24 (  15 ->     ) [( 6.4126e-01, 4.9327e-01, 3.9929e-01, 9.1992e-02), p^2=-5.5511e-17, m= 0.0000e+00] (  0,  0)
[P] 1 e+                 25 (  15 ->     ) [( 1.0676e+00, 8.3329e-01, 6.6730e-01,-1.1303e-02), p^2= 2.6112e-07, m= 5.1100e-04] (  0,  0)
[P] 1 e-                 26 (  15 ->     ) [( 1.1594e-01, 9.0246e-02, 7.2764e-02,-1.7403e-03), p^2= 2.6112e-07, m= 5.1100e-04] (  0,  0)
```

Some more hadron decays are following. The event is finished with the statement:

```
------------------------------------------------
Sherpa : Passed 1 events.
------------------------------------------------
```

At the same time this also indicates the start of a new event. Finally, the run ends up with a little statistics output:

```
In Event_Handler::Finish : Summarizing the run may take some time.
=========================================================================
Sherpa finished its simulation run with 0 errors.
=========================================================================
Return_Value::PrintStatistics(): Statistics {
  Generated events: 2
}
Time: 0.34 s (clocks=34) on Mon Nov  6 13:00:12 2006
 (User: 0.11 s ,System: 0.02 s ,Children User: 0 s ,Children System: 0)
```

3.3 Default setup: W production at the Tevatron Run I

Since SHERPA is mainly designed for the simulation of hadron collisions, a default setup[7] for the W boson production at the Tevatron Run I is also part of the distribution. This example should illustrate the usage of SHERPA in the framework of combining matrix elements and the parton shower [28, 29]. The matrix element processes included in this setup are the following:

$$
\begin{aligned}
p\bar{p} &\to jet\, jet \to e^-\bar{\nu}_e\,,\\
p\bar{p} &\to jet\, jet \to e^-\bar{\nu}_e + jet\,,\\
p\bar{p} &\to jet\, jet \to e^-\bar{\nu}_e + jet\, jet\,.
\end{aligned}
$$

[7]The corresponding data files are located in the SHERPA-1.0.9/Run/Tevatron1800 directory (Note that for running SHERPA in this setup the PATH variable has to be set to PATH=Tevatron1800).

Employing this setup SHERPA should be able to describe, for instance, the $p_\perp$ distribution of the W boson as measured by the D0 and CDF collaborations at Tevatron Run I, see also [33, 34, 35]. For more information about the merging of matrix elements and parton showers in the SHERPA framework, cf. Sec. 5.1. In any case please make sure that

- a jet finder with an appropriate $y_{\rm cut}$ has been specified (here 20 GeV),
- the Sudakov weights have been enabled (SUDAKOV_WEIGHT=1) and
- both showers, initial **and** final state shower, have been turned on.

In the output mode "errors" (OUTPUT=0) the output is very sparse. Then, the following output should be obtained for the initialization run:

```
$ ./Sherpa PATH=Tevatron1800 OUTPUT=0
Amegic::InitializeProcesses :
   Some new libraries were created and have to be compiled and linked.
   Type "./makelibs" in '/path/to/SHERPA-1.0/Run/Tevatron1800' and rerun.
Sherpa: Matrix_Element_Handler::Matrix_Element_Handler throws normal exit:
   Failed to initialize Amegic for hard interactions.
```

For the second run, i.e. the cross section integration and the event generation there should not be any output, if output mode "errors" was specified.

4 Specification of the parameters

There are two general ways of modifying the parameters of SHERPA:

1. Through data files:
   By default they are located in the ./SHERPA-1.0.9/Run directory and have the form Filename.dat.
2. Through command line input:
   The syntax[8] is ./Sherpa KEYWORD1=value1 KEYWORD2=value2 ... . To change, e.g., the path to the data files the corresponding command line is ./Sherpa PATH=pathname, where pathname is the path relative to the ./SHERPA-1.0.9/Run directory[9].

All over SHERPA, particles are defined by a particle code, which is in most cases identical to the one proposed by the PDG. These codes are displayed for the elementary particles in Particle.dat and for the hadrons in Hadron.dat, respectively. In both cases, antiparticles are characterized by a minus sign in front of their code. (So a final state μ^- has code 13, while a μ^+ has -13).

4.1 Particle data

4.1.1 Particle.dat

In Particle.dat all the elementary particles available throughout SHERPA are defined. The head of this file and a typical entry, here the d quark, looks like:

```
kf  Mass  Width  3*e  Y   SU(3)  2*Spin  maj  on  stbl  m_on  Name
1   .01   .0     -1   -1  1      1       0    1   1     0     d
```

[8] This is valid if one works in the ./SHERPA-1.0.9/Run directory.
[9] Specifying an absolute path should work as well.

Here kf – the particle code – labels the internal number characterizing the flavour within SHERPA, Mass is the (current) mass of the particle in GeV. The width of the particle is given by the parameter Width. The 3*e column determines three times the electric charge of the particle, the sign convention is such that the electron has charge $3 * e = -3$. The weak hypercharge is given by Y. The switch SU(3) determines if the particle is strongly interacting (1) or not (0); similarly, the flag maj accounts for the majorana character of a particle. Through the switch on, the particle can be excluded in the processes to be considered. Note that this switch should be used with extreme caution. The stbl flag is not used at the moment. In the near future it will steer the way of how SHERPA defines the width of particles and their decays. By m_on the kinematical mass of the particles can be set to zero within the matrix element generation and evaluation. This kinematical mass is completely decoupled from possible Yukawa couplings. Note that the masses of SUSY particles will, by default, be overwritten by those obtained from the spectrum generator, either Isasusy 7.67 or a SLHA conform file. Only for the first the widths of the sparticles are overwritten, the input of particle widths throught the Les Houches interface is not yet supported. The last entry of the row defines the name of the particle.

4.1.2 Hadron.dat

This file specifying the hadrons of SHERPA is in analogy to the file Particle.dat. All baryons and mesons that may appear as intermediate or final state particles within events produced by SHERPA are listed here. Using the Lund fragmentation and the associated hadron decay package, the only entries of importance are the kf codes (according to the PDG numbering scheme) and the names of the hadrons. All other parameters have no influence at present.

4.2 Data files

Switches and parameters for the general steering of the program are – by default – found in the file Run.dat. In particular, it can include the names (and the paths) of other input data files. In most cases the syntax to specify a parameter in these data files is

```
KEYWORD = value ,
```

where spaces are of no consequences. Exclamation marks indicate the begin of a comment line. However, it should be stressed that the read-in devices of the different SHERPA modules are case sensitive. For any input parameter of SHERPA which is a number, they allow the specification through an algebraic expression in C-style like

```
ISR_SMIN = sqr(1.3/900) .
```

If a parameter or switch has not been specified, a default parameter is given.
In the following, the data files and the most important parameters and switches are explained.

4.2.1 Run.dat

As mentioned before, general switches and parameters as well as names for additional input files are defined in Run.dat. The corresponding file of the default setup is displayed in Fig. 3, see the appendix A. The following parameters can be specified:

1. The number of events is set through EVENTS; if this is never specified, no events are generated.

2. The numerical accuracy can be given, e.g. for comparing two numbers.

3. The output level can be specified through `OUTPUT`. If not specified the output level is `0`. Different levels can be overlayed by simply adding the level number. For their meaning please cf. Table 3. By default, `SHERPA` produces coloured output. If however stdout does not correspond to a terminal, this option is automatically switched off. It may also be set manually through the switch `PRETTY_PRINT = {On,Off}`.

`0`	errors	all error messages, most general run information
`1`	events	event output
`2`	info	more detailed information about what is going on during the run
`4`	tracking	useful information in order to track problems a user might encounter, i.e. to approximately identify the place/cause of problems
`16`	logfile	write a log file containing information about the runtime environment and the setup

Table 3: The output levels in `SHERPA`.

4. Different data files can be read in, their names and purposes should be self-explanatory. In the example file shown in Fig. 3 data files for the beam and ISR initialization as well as for the handling of the analysis are listed together with their default value.

5. The internal analysis can be switched on or off by setting the `ANALYSIS` switch to `1` or `0`, respectively. An `ANALYSIS_OUTPUT` main directory can be specified through setting this keyword with a meaningful value. It will be created in the working directory. For detailed information, cf. Sec. 4.2.12.

6. Writing a log file can be initiated through a `LOG_FILE` setting, however it is only created when using the output mode "logfile" (cf. Table 3). It then contains all information about the setup. In case that a `SHERPA` run crashes unexpectedly, you can do the following to report the bug to us:

   - First check the Sherpa output for hints on the reason of program abort. If present, use them to figure out what has gone wrong, also employing manuals.
   - If this does not help, contact the Sherpa team, providing all information on your setup. That is
     - the status recovery directory "Status__¡date of crash¿" produced before the program abort.
     - The command line.
     - The installation log file.

7. `SHERPA` uses a random number generator as described in [36]. The two independent integer-valued seeds for this random generator are specified by the option `RANDOM_SEED=A B`. The seeds `A` and `B` may range from 0 to 31328 and from 0 to 30081, respectively. Only giving a single integer number or not specifying a `RANDOM_SEED` at all results in the old random number generator (`SHERPA 1.0.6` and older) to be used.

8. Some more options, which are not shown in the example `Run.dat` files that are given within the `SHERPA` distribution, can be specified in the `Run.dat` file. Some of them are explained below.

   (a) There are two ways to specify the event record. First of all, the switch `EVENT_MODE` can be set to either `Sherpa` (default), `HepMC`, or `HepEvt`, resulting in the corresponding structure to be filled internally (and printed to screen, if the output-level is set accordingly).

Second, if the events are to be written to file, the keywords SHERPA_OUTPUT, HEPMC_OUTPUT and HEPEVT_OUTPUT are available, see below. With these keywords you specify the filename's root, i. e. HEPEVT_OUTPUT=filename will create files named filename.#.hepevt, where the hash mark stands for an increasing number.

By default, each file contains 1000 events, but the number can be set by specifying the keyword FILESIZE. Use EVT_FILE_PATH to change the directory, where the files will be stored. To steer the precision of all double numbers written to file, the keyword OUTPUT_PRECISION is available, with a default value of 6.

To write events directly to gzipped files instead of plain text, you have to specify the option --enable-gzip during configure. (see section 2).

Note: You can use multiple output formats in the same run.

- SHERPA_OUTPUT: Full events in the Sherpa format.
- HEPEVT_OUTPUT: Full events in the HepEvt format.
- D0_HEPEVT_OUTPUT: Specifically for D0 analysis needs, there is this modified HepEvt output format available. It will output events in the HepEvt format, with one extra line added for each event, containing information about the signal process:
  enhance-factor Q^2 x_1 x_2 parton1 parton2
- HEPMC_OUTPUT: HepMC output according to the HepMC::writeGenEvent method.
- OLD_HEPMC_OUTPUT: HepMC output according to the GenEvent::print method. **Beware**: This is buggy even in recent CLHEP packages and might crash especially when using gcc4.

It should be stressed that in order to enable the HepMC event record, the user must enable the link to the CLHEP package. This is possible through a corresponding configure flag, see the section above dealing with the installation of SHERPA and how to apply the makeinstall script in particular.

It is of course also possible to link a user program to SHERPA, so that the events can be read and/or analyzed "on the flight". Two configurations are supported. The user can of course access SHERPA's internal Blob_List, which contains all necessary information about an event. Or you can access the HepMC::GenEvent, if EVENT_MODE is set to HepMC

(b) The running time of the code is given in user CPU seconds through TIMEOUT. This option is of some relevance when running SHERPA on a batch system. Since in many cases jobs are just terminated, this allows to interrupt a run, to store all relevant information and to restart it without any loss. This is particularly interesting when carrying out long integrations. Contrary, setting the TIMEOUT variable to -1, which is the default setting, translates into having no run time limitation at all.

(c) The spin correlation algorithm used by the HADRONS++ module is switched off per default. It can be switched on through SPIN_CORRELATIONS with the values 0 (off) or 1 (on).

4.2.2 Beam.dat

In Beam.dat, cf. Fig. 4 of appendix A for the corresponding file of the default setup, the following beam parameters are specified:

1. The two incoming particles are specified according to their particle code by BEAM_{1,2}. For (anti)protons and (positrons) electrons the relevant codes are given by $(-)2212$ or $(-)11$, respectively. For photons the code is 22. The energies of the incoming beams are defined through BEAM_ENERGY_{1,2}. The specification of the beam particles and energies is mandatory.

2. Also, in SHERPA polarized beams can be accounted for through specifying their polarization degree by BEAM_POL_{1,2} in terms of the helicity leading to values in the range of -1 to 1.

3. Having specified the "collider particles" they can be translated into the "incoming particles" (that is before ISR has been applied) through the application of a beam spectrum. The corresponding keyword is BEAM_SPECTRUM_{1,2}. Currently the following options are provided:
   - Monochromatic: The beam energy is not changed.
   - Simple_Compton: This is a simple Laser backscattering spectrum, which is often used in theoretical analyses.
   - Laser_Backscattering: This is a realistic spectrum according to the CompAZ parametrization.
   - Spectrum_Reader: Then a user defined spectrum can be specified by SPECTRUM_FILE_{1,2}.
4. The following parameters for the beam remnant handling can be varied: the mean value of the intrinsic $k_\perp$ and its variance through the keywords K_PERP_MEAN_{1,2} and K_PERP_SIGMA_{1,2}, respectively. Both values are given in GeV.
5. The minimal and maximal fraction of the total energy s – the sum of the two beam energies squared – carried by the particles after the beam treatment are determined by the parameters BEAM_SMIN and BEAM_SMAX. Possibly, the latter can be subject to further reduction due to the form of the chosen spectrum. The laser backscattering mode, for instance, has a limited maximal energy fraction that can be carried by the photon.
6. The remaining parameters deal with the setup of the laser used for backscattering. The energy and circular polarization of the lasers is given by E_LASER_{1,2} and P_LASER_{1,2}, respectively. For the latter, the range of allowed values is again -1 to $+1$. The keyword LASER_MODE denotes the different components of the laser backscattering photon spectrum. The default value is 0 (all components). The full set of options is summarized in Tab. 4.
   The spectra and polarization degrees for photons generated in laser backscattering via Laser_Backscattering and Simple_Compton are shown in Fig. 1. By changing the switch LASER_NON-LINEARITY non-linear Laser effects can be taken into account or not. Note that this option is valid only for the full laser backscattering spectrum.

4.2.3 ISR.dat

In ISR.dat, cf. Fig. 5 of appendix A for the corresponding file of the default setup, the following parameters for the initial state radiation and the PDFs are specified:

1. After the beam spectra have been applied, the two incoming particles are specified by BUNCH_{1,2} according to their particle code. For (positrons) electrons and (anti)protons the relevant codes are $(-)11$ and $(-)2212$, respectively. For photons the code can be given by 22. It is important to note that the beam spectra and the bunch particles have to fit, otherwise SHERPA will terminate its run with a corresponding error message during the initialization phase.

option	comment
0	default, the full spectrum
1	the pure Compton component of the spectrum
2	the pure two photon component of the spectrum
3	the pure re-scattering component of the spectrum

Table 4: Different options available for the variable LASER_MODE in the file Beam.dat and their associated choices for the used components of the entire photon spectrum produced via laser backscattering.

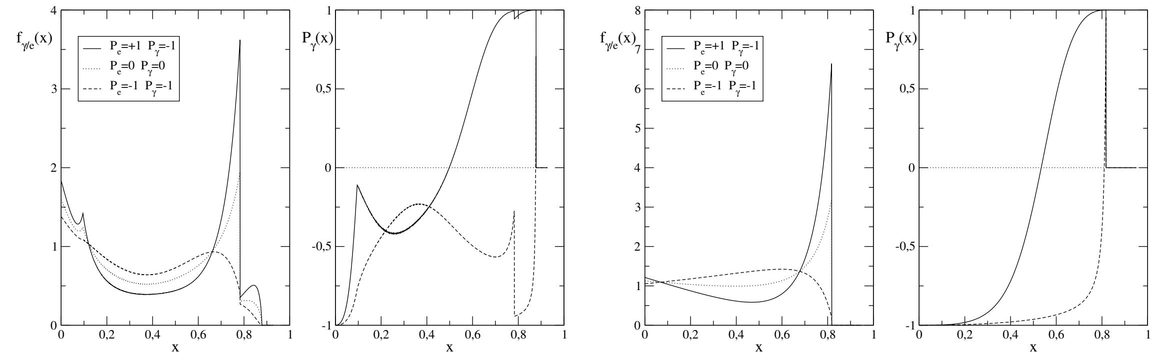

Figure 1: Laser backscattering spectra and degree of circular polarization for backscattered photons according to the CompAZ parametrization (left) and according to the simple Compton spectrum (right). Different combinations of electron and laser photon polarization are shown. The energy of the electron beam and the laser are $E_e = 250$ GeV and $\omega_\gamma = 1.17 \cdot 10^{-9}$ GeV, respectively ($x_e = 3.61$).

2. With the `ISR_`{1,2} switch one may allow for the consideration of an internal parton structure. If the incoming particle is a proton, the corresponding switch has to be set to `On`. For photons, ISR and hence an internal parton structure can be switched on or off. For leptons this switch steers the inclusion of a structure function.

3. The minimal and maximal fraction of the total energy s carried by the particles after the ISR treatment are constrained by the parameters `ISR_SMIN` and `ISR_SMAX`. The default values are 0 and 1, respectively.

4. In case the ISR handling is set up for leptons the inclusion of a structure function is further specified through the switches `ISR_E_ORDER` and `ISR_E_SCHEME`. The former (default value is 1) fixes the perturbative order in the fine structure constant α, whereas the latter (default value is `2`) determines the way of respecting non-leading terms in the structure function. Possible options for the last switch are `0` ("mixed" choice), `1` ("eta" choice), or `2` ("beta" choice).

5. For the case of protons a number of PDFs and different options are supported:
   - an interface to the Fortran CTEQ6 set of PDFs with their original grid,
   - an interface to the C++ version of MRST99,
   - and the LHAPDF version 3 interface, which provides access to a multitude of PDFs.[10]

   The variable `PDF_SET` determines which PDF is actually used, while `PDF_GRID_PATH` decides where to look for the required grid files. `PDF_SET_VERSION` finally selects the specific tuning out of the different possibilities provided by the authors of the respective set. The three available options for `PDF_GRID_PATH` and the corresponding PDFs are listed in Tab. 5.

4.2.4 `Model.dat`

The general model settings are displayed in Fig. 6 of appendix A. They have the following meaning:

1. The physics model used in the `SHERPA` run is defined through the variable `MODEL`, which can have one of the three self-explanatory values `SM`, `MSSM` or `ADD`, its default value is `SM`. In case the MSSM

[10]To link the LHAPDF libraries the environment variable `LHAPDFDIR` must be set to the corresponding path.

PDF_GRID_PATH	PDF_SET
CTEQ6Grid	cteq6m, cteq6d, cteq6l, cteq6l1
MRST99Grid	MRST99
PDFsets	Alekhin_100, Alekhin_1000, Botje_100, Botje_1000, Fermi2002_100, Fermi2002_1000, MRST2001, MRST98, cteq6, cteq6l, cetq6ll

Table 5: The proton PDFs available within SHERPA and their associated settings of the corresponding parameters PDF_GRID_PATH and PDF_SET. By choosing one of the PDFs out of the available PDF_SET options the value of the PDF_GRID_PATH has to be set accordingly.

is selected, the variable GENERATOR_ON must have the value 1. There are two possible ways to input low-scale MSSM spectra to SHERPA, by the parameter SUSY_GENERATOR either Isasusy 7.67 can be called (Isajet) or the Les Houches interface (LesHouches), that reads in SLHA compatible files, can be chosen.

2. SHERPA also provides different schemes to define the parameters of the electroweak sector. The scheme can be set by EW_SCHEME with its default value given by 0. The different schemes correspond to different choices of which physics parameters are considered to be fixed and which ones are considered to be derived from given magnitudes. Within SHERPA the following schemes have been provided:

   - Case 0:
     All EW parameters are explicitly given.
     Here M_W, M_Z and M_H are taken from Particle.dat, and the parameters 1/ALPHAQED(0), SIN2THETAW, VEV and LAMBDA have to be specified as well.
   - Case 1:
     All EW parameters are calculated out of M_W, M_Z, M_H and 1/ALPHAQED(0). The first three values are fixed in Particle.dat.
   - Case 2:
     All EW parameters are calculated out of 1/ALPHAQED(0), SIN2THETAW, VEV and M_H, whereas M_H again is given in Particle.dat.
   - Case 3:
     This choice correponds to the G_μ-scheme. The EW parameters are calculated out of the weak gauge boson masses M_W, M_Z, the Higgs boson mass M_H (all specified in Particle.dat) and the Fermi constant GF.

The SM parameters are defined through the settings, which are exhibited in Fig. 7. The parameters for physics beyond the SM are defined by the settings exhibited in the next two figures. Most of the keywords are self-explanatory, however, some comments are in order here.

1. The new default choice to input MSSM spectra to SHERPA is via the SUSY-Les-Houches-Accord format [12]. It is selected with the choice MSSM for the MODEL variable, GENERATOR_ON = 1 and SUSY_GENERATOR = LesHouches. The actual SLHA file has to be specified by LESHOUCHESINPUT. From this file the full low-scale MSSM spectrum is read, including sparticle masses, mixing angles etc. However, it has to be noted that the SHERPA Les Houches interface does not set the widths of the sparticles. For the those the value specified in the corresponding Particle.dat file are retained.

2. As already stated above, SHERPA also provides an interface to Isasusy 7.67 for the calculation of MSSM particle spectra and decay widths. Once Isasusy 7.67 is linked to the program it can be selected with the setting SUSY_GENERATOR = Isajet. The only inputs for the MSSM are a SUSY breaking mechanism via SUSY_SCENARIO and the corresponding high scale input parameters.
The available options for SUSY_SCENARIO and the parameters that have to be accordingly set read:
   - SUSY_SCENARIO = mSUGRA or SUGRA_with_enforced_unification
     - M0, M12, TAN(BETA), A0, SIGN(MU)
   - SUSY_SCENARIO = AMSB
     - M0, M32, TAN(BETA), A0, SIGN(MU)
   - SUSY_SCENARIO = mGMSB
     - MESSENGER_SCALE, LAMBDA_M, TAN(BETA), GRAVITINO_RATIO, SIGN(MU), NUMBER_MESSENGERS
   - SUSY_SCENARIO = non-minimal_GMSB
     - MESSENGER_SCALE, LAMBDA_M, TAN(BETA), GRAVITINO_RATIO, GAUGINO_RATIO, DELTA_M(Hu), DELTA_M(Hd), DELTA_M(Y), N5(U1), N5(SU2), N5(SU3), SIGN(MU), NUMBER_MESSENGERS
   - SUSY_SCENARIO = non-minimal_SUGRA
     - M0, M12, TAN(BETA), A0, M(U1)_GUT, M(SU2)_GUT, M(SU3)_GUT, M(e1R)_GUT, M(l1)_GUT, M(u1R)_GUT, M(d1R)_GUT, M(q1)_GUT, M(e2R)_GUT, M(l2)_GUT, M(u2R)_GUT, M(d2R)_GUT, M(q2)_GUT, M(e3R)_GUT, M(l3)_GUT, M(u3R)_GUT, M(d3R)_GUT, M(q3)_GUT, A(tau)_GUT, A(b)_GUT, A(t)_GUT, H(d)_GUT, H(u)_GUT, SCALE(SUSY), SIGN(MU)

   For the specific meaning of these parameters the reader is referred to the corresponding publication of Isajet. It should be noted that for all scenarios the value of the t quark mass is rather fixed to the value of the parameter Yukawa_t than to the value provided in Particle.dat.

3. The number of parameters necessary to specify an ADD model is rather small: the variable N_ED fixes the number of extra dimension, which should be present. Although its nature is rather to be constant than variable, the used Newtonian constant can be changed. Per default its value is set to the PDG value. The most important parameter of any ADD model is the size of the string scale M_S denoted by the parameter M_S. With the switch M_CUT one can restrict the c.m. energy of the hard process to be below this specified scale. Setting the value of KK_CONVENTION allows to change between three widely used conventions for the definition of M_S and the way of summing internal Kaluza-Klein propagators. For details of the implementation, the reader is referred to [37].

4.2.5 ME.dat

The ME.dat file (see Fig. 10 of the appendix A) steers the general settings for matrix element generation.

1. First of all, by setting ME_SIGNAL_GENERATOR, the user can choose between the AMEGIC++ ME generator or the library of simple $2 \to 2$ cross sections made available through the EXTRA_XS package.

2. SHERPA provides the option to generate either weighted or unweighted events after the integration has taken place. For unweighted events it seems a priori not so important to have the total cross sections. Nevertheless, the events of different processes are mixed according to the total cross sections after the cuts have been applied. Running the integration for some time guarantees that the multichannel integrator has been trained such that events with large weight are produced more often.

setting	Q^2	
0 = default	$\hat{s}$	
1	s_0 = fixed scale	
2	$4\hat{s}\hat{t}\hat{u}/(\hat{s}^2+\hat{t}^2+\hat{u}^2)$	for $2 \to 2$ processes
	$\sum_i p_{\perp,i}^2$	else
3	$\mathrm{Min}\{p_{\perp,i}^2\}$	

Table 6: Available scale schemes within SHERPA. Note that for the transverse momenta, i runs over all outgoing particles.

3. Through the variable SIGNAL_MODEL a subset of interactions out of a physical model can be chosen. However, in order to not exclude diagrams unintentionally due to missing interactions, this variable should be equal to the variable MODEL in Model.dat.

Additionally, different scale choices relevant at different stages of process integration and event generation are made available.

1. Gauge couplings in the matrix element can be set through COUPLING_SCHEME. It should be stressed that this setting occurs on the level of vertices. It is permanent and independent of what is suggested by the keyword "Running". In particular, the COUPLING_SCHEME = Fixed refers to using the ALPHAS(default) and 1/ALPHAQED(default) values given in Model.dat for the respective coupling. In contrast, when using the COUPLING_SCHEME = Running the couplings are evaluated at the c.m. scale s of the incoming beams starting the evolution at ALPHAS(MZ) and 1/ALPHAQED(0).

2. For the Yukawa couplings, the couplings in the matrix element can be set through YUKAWA_-MASSES. The options "Running" and "Fixed" correspond to the c.m. energy and the pole mass of the considered particle, respectively. Another keyword of interest is the YUKAWA_MASSES_FACTOR, which works as a prefactor for the couplings.

3. During the calculation of the matrix elements, the renormalization scale for the strong coupling constant and the factorization scale can be modified. SCALE_SCHEME selects the scale definition for both – a prefactor for the setting is available through SCALE_FACTOR. The KFACTOR_SCHEME then switches on or off the influence of the (possibly modified) renormalization scale on α_s, provided that COUPLING_SCHEME is set to "Running". The available scale schemes are listed in Table 6.

4. A fixed, user-defined scale squared can be defined through the keyword FIXED_SCALE.

5. The variable SUDAKOV_WEIGHT is the master switch for preparing the matrix elements in order to combine them with the parton shower, cf. Sec. 5.1.

4.2.6 Processes.dat

In Processes.dat the process(es) to be evaluated by the matrix element generator are defined. They also serve as seeds for the event generation.

1. The admixture of different hard signal processes in the events is determined according to the respective total cross section after the cuts have been applied, times, possibly, Sudakov weights from the merging procedure with the parton showers. The particles are distributed according to the respective matrix elements, again, this can be reweighted with some Sudakov form factors.

2. In setting up the processes it is necessary that the requested initial states are consistent with the settings in the file `ISR.dat`, otherwise `SHERPA` will terminate the run immediately. Therefore, elementary bunches like electrons have to be the initial state of the hard processes as well. For the case of proton bunches, the initial states can be either quarks or gluons.

3. To characterize the initial and final states of the processes, the particle codes displayed in the data file `Particle.dat` are needed. Besides elementary particles, `SHERPA` allows for containers that collect different particles with similar properties, namely

   - lepton (carrying number `90`),
   - neutrino (carrying number `91`),
   - fermion (carrying number `92`),
   - jet (carrying number `93`),
   - quark (carrying number `94`).

   These containers contain all massless particles of the denoted type and allow for a more efficient definition of initial and final states to be considered. The jet container consists of massless quarks (as defined by the mass value and/or the `mass_on` switch residing in the `Particle.dat` file) and the gluon.

4. The syntax of calling a specific process is:

```
Process : 93 93 -> -11 12 93
Order electroweak : 2
Order strong : 1
End process
```

   Here all processes of the kind $jet\ jet \to e^+\nu_e\ jet$ are initialized, where contributions originating from the production of heavy quarks have been left out.

   In version 1.0.9 processes that differ only in the multiplicity of jets (or any other particle) can be specified using the following short notation:

```
Process : 93 93 -> -11 12 93{3}
Order electroweak : 2
End process
```

   The number in curly brackets following a particle code specifies it's maximal multiplicity. The given example would generate all $jet\ jet \to e^+\nu_e$ processes with up to 3 extra jets in the final state.

   The parameters `Order electroweak` and `Order strong` restrict the number of electroweak or strong couplings occurring in single Feynman diagrams. The default values are `99`, meaning that all electroweak and strong contributions are taken into account.

5. Processes can be also specified by defining a production and subsequent decay processes. In this case `SHERPA` would generate matrix elements which feature full spin correlation to the final state particles. The syntax is:

```
Process : 93 93 -> 6[a] -6[b]
Decay : 6[a] -> 5 -23[c]
Decay : -6[b] -> -5 23[d]
```

```
Decay : -23[c] -> -13 14
Decay : 23[d] -> 1 -2
Order electroweak : 0
Order strong : 2
End process
```

In the example a *jet jet* initial state produces a top-pair which then enter into a decay chain. The full process is equivalent to *jet jet* $\rightarrow b\bar{b}\mu\bar{\nu}_\mu d\bar{u}$. However in this setup only amplitudes that exactly contain the given intermediate particles are taken into account.

The qualifiers in square brackets must be given to allow for a unique assignment of decaying particles. Any character can be used, but it has to be unique for the specific process. Particle container are only allowed for the initial or final state of the full process, not for particles that are subject to further decay. The given electroweak and strong order only relates to the core process (*jet jet* $\rightarrow t\bar{t}$).

This treatment is applicable for processes which are dominated by the resonance of a given intermediate particle. The resulting decay products respect full spin correlation to their mothers and are distributed according their widths. Since the number of amplitudes to be evaluated is usually reduced drastically compared to the full process, the computational effort is significantly smaller and thus much higher final state multiplicities can be considered.

Note that the treatment is an approximation to the full matrix element and only as good as the narrow width assumption for the intermediate particle. For some processes this might lead to gauge violations up to this order. The gauge violation displayed during the initialization of the process, however, can be much larger. This is no cause for concern and due to the fact that then the matrix element is evaluated only at a single phase space point, in general far away from the mass-shells of the intermediate particles.

The short notation for different jet multiplicities using curly brackets can be applied here as well. If specified, all possible combinations of production and decay processes will be generated.

6. Additionally, scale choices and prefactors can be defined on a process-by-process base different from the global settings. The keywords look similar to the ones used in `ME.dat` and correspondingly have the same meaning:

   - "`Scale scheme : `",
   - "`Scale factor : `",
   - "`KFactor scheme : `", and
   - "`Fixed scale : `".

7. To enable the multi-cut treatment described in [31] the user may set a process dependent value for the jet resolution cut y_{cut} via

   - "`YCUT : `".

   Alternatively an individual selector data file can be specified:

   - "`Selector file : `".

8. In case unweighted events are to be generated, the unweighting efficency may be increased through setting the keyword

   - "`Max_Epsilon : `".

The given value specifies the fraction ϵ, for which weights are allowed to be above a "ϵ-dependent maximum weight" $\omega^{\epsilon}_{\mathrm{max}}$. Meaningful values for `Max_Epsilon` are in the range of 10^{-3} to 10^{-4}. In any case it should be carefully checked that the considered distributions are not altered. To appropriately use this feature, it has to be ensured that the optimization of the phase space integration has been completed by switching on the `FINISH_OPTIMIZATION` option in the file `Integration.dat`, see Sec. 4.2.8.
Corresponding weight histograms, which are used to determine $\omega^{\epsilon}_{\mathrm{max}}$ are stored – if it has been provided by the user before – in the `Results` directory, see subdirectory `WD_<process_id>`. The first row of the histogram contains the decade logarithm of the weight, the second row the number of events in the corresponding bin.

9. The specification of an

   - "`Enhance_Factor :` "

   allows the enhancement of specific processes. E.g. a factor of 10 causes the process to produce 10 times more events than its fraction of the total number of events given by the corresponding cross section. This factor is stored in the output of the affected event. To regard for the enhancement a weight of 1/`Enhance_Factor` has to be applied in the analysis.

10. The keyword

    - "`Integration_Error :` ".

    allows to specify the relative integration error on a process-by-process base. If stated it has priority over the `ERROR` setting in `Integration.dat` and disables the `FINISH_OPTIMIZATION` option for the process.

Examples for process files are given in Figs. 11 and 12.

4.2.7 `Selector.dat`

`SHERPA` provides a number of predefined selectors on the matrix-element level to define kinematical cuts on external particles. These can be found and specified in the file `Selector.dat`.

1. Any selector is included through its keyword, followed by the particles it acts on, and by the minimal and maximal value of the variable to cut on.

2. Particles are called by their code, which is defined in `Particle.dat`.

3. Available predefined selectors, which can be used within `SHERPA`, are listed together with possible applications in Tab. 7.

4. There are selectors related to only one particle – then, of course, no second particle has to be specified.

5. All angle selectors are defined in terms of the cosine of the angle between two particles. For the case of the `BeamAngle` selector, the notation is specific: here, with the second particle specifier, one can choose that the angle is taken between the particle given by the first specifier and the first or second incoming (beam) particle by using the option `0` or `1`, respectively.

6. The thresholds for all energy and mass selectors have to be given in GeV.

A `Selector.dat` example file for the default setup of the `SHERPA` distribution is given in Fig. 13 of appendix A.

Keyword/Variable	Particle 1	Particle 2	Minimal value	Maximal value
Energy	22		10.	10000.
ET	22		10.	10000.
PT	22		10.	10000.
Rapidity	22		-2.5	2.5
PseudoRapidity	22		-2.5	2.5
Mass	1	-2	75.	85.
PT2	1	-2	75.	85.
Angle	11	22	-0.995	0.995
BeamAngle	11	0	-0.98	0.98
JetFinder			0.01	1.
ConeFinder			0.4	

Table 7: All possible types of cuts exemplified.

option	comment
0	pure Rambo
1	pure Sarge
2	Rambo + Sarge
3	Rambo + Multichannel
4 ... 6	pure Multichannel

Table 8: Available integration algorithms within `SHERPA` and the corresponding values of the `INTEGRATOR` variable, which can be set in `Integration.dat`.

4.2.8 `Integration.dat`

The setup of the phase space integration can be adjusted through the settings in the file `Integration.dat`. An example (the default setup of the `SHERPA` distribution) is given in Fig. 14, see appendix A. The possible settings are:

1. With the variable `ERROR` the maximal relative integration error (statistical error of the MC integration) is determined to which a total cross section is evaluated. The parameter is also important for the generation of unweighted events, since before the actual event generation can start the phase space has to be integrated as accurately as requested by the variable `ERROR`. Additionally the performance of the MC integrator is optimized at this stage. However, the appropriate choice of the integration accuracy is clearly process and purpose dependent. For processes with many final state particles, the usage of a very small integration error can cause very time-consuming calculations. In most cases an adequate choice seems to be of the order of 0.01.

2. The switch `FINISH_OPTIMIZATION` (if not declared its default value is set to `Off`) steers whether the phase space integration is performed at least until the integrator is fully optimized or not. In the case of unweighted events this ensures the best unweighting efficiency, however, might become very time-consuming for processes with many particles in the final state. Typically a statistical accuracy of 0.002 ... 0.005 is achieved after the full optimization.

3. With the variable `INTEGRATOR` the integration algorithm responsible for the final state phase space integration is selected; for the various available choices see Tab. 8.
   The default is the pure Rambo integrator, however, the authors recommend to employ one of the options 4 ... 6, namely the multichannel integrators. The three given multichannel methods differ in the way the set of single channels is generated during the initialization run.

- Option 4:
  Building up the channels is achieved through respecting the peak structure given by the propagators. The algorithm works recursively starting from the initial state.
- Option 5:
  This is an extension of option 4. In the case of competing peaks (e.g. a higgs boson decaying into W^+W^-, which further decay), additional channels are produced to account for all kinematical configurations where one of the propagating particles is produced on its mass shell.
- Option 6:
  In contrast to option 4 the algorithm now starts from the final state. The extra channels described in option 5 are produced as well. This has been optimized for configurations similar to Sec. 3.3.

  Note that the choice 4 ... 6 only affects the channel generation. Once a channel set has been produced, say by using option 4, switching to option 5 or 6 will not be sufficient to obtain the new channel set. For this, the old channel set has to be removed in the `Process` directory. However, after a completed Vegas optimization (see the `VEGAS` switch description below) the performance of the three channel sets is quite similar in most cases.
  Considering processes with real graviton production in the framework of the ADD model automatically appropriate channels will be created when choosing one of the options 4 ... 6, such that they correctly account for the required sum over the tower of external real gravitons.

4. The `VEGAS` switch assigns whether an additional Vegas optimization is performed for the channels of the multichannel integrator. The authors strongly recommend to use this option, since the integration efficiency improves significantly – typically by a factor of 2 ... 10. So, the default value is of course `On`.

In general, the best choice for the integration algorithm might depend on the considered process and the applied phase space cuts.

4.2.9 `Shower.dat`

An example for the parton shower settings provided through the file `Shower.dat` is exhibited in Fig. 15 of appendix A. The various keywords relate to the following features:

1. The variable `SHOWER_GENERATOR` specifies the shower module, which is responsible for the jet evolution. Currently, the only valid value is `Apacic`.

2. The main switches to steer the shower generation are `FSR_SHOWER` and `ISR_SHOWER`, which turn on and off the final and the initial state shower, respectively. Note that it is not necessary to turn on the showers to calculate cross sections with ISR, for this purpose it is sufficient to turn on `ISR_1` and/or `ISR_2` in the `ISR.dat` file. Contrary, in case of event generation for QCD processes at hadron colliders, it should be stressed that it is essential to have turned on both showers in order to obtain physical meaningful results.

3. The parton showers in the initial and final state terminate when no more parton can be emitted with a transverse momentum squared larger than the values given by `IS_PT2MIN` and `FS_PT2MIN`. The default choices are

$$p_\perp^2(\mathrm{IS}) \,=\, 4\,\mathrm{GeV}^2 \qquad \text{and} \qquad p_\perp^2(\mathrm{FS}) \,=\, 1\,\mathrm{GeV}^2\,.$$

With version 1.0.9 Sherpa provides the option to include QED final state radiation off charged leptons and quarks. To enable this option the parameter `FS_QED_SCHEME = 1` has to be specified. The cut-off for emissions off quarks is equal to the final state QCD shower cut-off. For emissions off leptons a separate cut-off has to be specified via `FS_PT2MIN_QED`. The current default is $0.0025\mathrm{GeV}^2$.

4.2.10 `MI.dat`

The multiple interaction scenario is set up according to the entries in the file `MI.dat`. Examples can be viewed in Fig. 16 of appendix A. So far multiple interactions can **only** be generated for proton–proton collisions[11]. The parameters are the following:

1. The variable `MI_HANDLER` specifies the multiple interaction handler. If multiple interactions shall be switched on, the setting is `Amisic`.
   In future there will be two more switches, selecting the hard and the soft model used in the underlying event. Since there is no soft model so far, the specification is `SOFT_MODEL_NAME = None`. Multiple interactions are finally switched on via `HARD_MODEL_NAME = Simple_Chain`[12].

2. The processes, which are to be generated in multiple interactions are given by `CREATE_GRID` followed by the specification of the process employing a similar nomenclature as in `Processes.dat`. The tag `MI_SCALE_SCHEME` sets the scale scheme, while `MI_K_FACTOR_SCHEME` sets the K factor scheme to be used for these processes, cf. the description of `ME.dat`, Sec. 4.2.5. `PS_ERROR` gives the error with which the phase space integrator calculates the cross section for the processes.

3. If `REGULATE_XS` is turned on, trivial regularization of the cross section is applied. `XS_REGULATOR` gives the type of the regulator to use. Currently there is only one option, `QCD_Trivial`, which leads to the multiplication of the differential cross section by $p_\perp^4/(p_\perp^2 + p_{\perp 0}^2)$ with $p_{\perp 0}$ given by `XS_REGULATION`.

4. `SCALE_MIN` sets the minimum scale $Q_{\rm min}$ for which events are to be generated. This scale may be rescaled with the use of an exponent α, which can be set through `RESCALE_EXPONENT`, and a reference energy scale $E_{\rm ref}$ made available through `REFERENCE_SCALE`. The rule that determines the cutoff scale is then $Q_{\rm min} \cdot (E_{\rm cms}/E_{\rm ref})^\alpha$.

5. The tag `PROFILE_FUNCTION` is used to specify the name of the profile function for the hadron profile in impact parameter space. The available options are `Exponential`, `Gaussian` and `Double_Gaussian`. The parameters of the double Gaussian are set through `PROFILE_PARAMETERS`, see Fig. 16.

4.2.11 `Fragmentation.dat`

The hadronization phase of `SHERPA` is arranged according to the settings in `Fragmentation.dat`. An example is given in Fig. 17 of appendix A. The meaning of the keywords is as follows:

1. The keyword `FRAGMENTATION` (default value `Lund`), steers whether the fragmentation is on (right now only handled by an interface to `Pythia` [25]), indicated by `Lund`), or not (`Off`). In the next release, a modified version of the cluster fragmentation model [38] will be made available as well.

2. The parameters of the Lund string fragmentation are set in the file specified by `LUND_FILE` (default is `Lund.dat`). The coarse features of the string breakup in the Lund model are characterized by three parameters, Lund-a through `PARJ(41)` (data file value: 0.431), Lund-b through `PARJ(42)`

[11] The multiple interaction treatment will be extended to include more classes of collisions in the future.
[12] The latter two switches are not shown in the example `MI.dat` file, since at present there is only one meaningful choice per switch.

(data file value: 0.878), and Lund-σ through PARJ(21) (data file value: 0.336). More Pythia parameters can be added to the Lund.dat file using the Pythia nomenclature in the common-blocks, e.g. MSTJ(11) for changing the fragmentation function for heavy quarks. In particular, particle decays may be enabled or disabled using the switch MDCYKF(<kfcode>,1), which is actually resolved to MDCY(PYCOMP(<kfcode>),1) during the initalisation of SHERPA.

3. The treatment of hadron and τ decays is specified by DECAYMODEL, allowed values are either Lund (the default), in which case Pythia performs all hadron decays, or Hadrons, which makes the HADRONS++ module performing the decays. Specifics for this setting can be found in Sec. 4.3. There are three options in Fragmentation.dat that can be specified to initialise the HADRONS++ module. These are:

   - DECAYPATH
     The path to the parameter files for the hadron and τ decays, with default Decaydata/. It is important to note that the path is supposed to start in the SHERPA Run directory. Moreover, it has to end with a slash, "/".
   - DECAYFILE
     The filename of the file that contains all particles that HADRONS++ will decay. For future references this file will be called "Hadron Decays File". This file must be stored in the path given under DECAYPATH. The default setting is HadronDecays.dat.
   - DECAYCONSTFILE
     This file contains several constants that are frequently used for hadron decays (see Sec. 4.3 for details). The default name is HadronConstants.dat.

4. There are two more switches which relate to fragmentation and hadron decays:

   - MASS_SMEARING = [0,1] (default: 1)
     Determines whether particles entering the hadron decay event phase should be put off-shell according to their mass distribution. It is taken care that no decay mode is suppressed by a potentially too low mass. For Pythia as hadron decay handler this means we are using its w-cut parameter, while for HADRONS we determine it dynamically from the chosen decay channel.
   - MAX_PROPER_LIFETIME = [mm]
     Switch for maximum proper lifetime $c\tau$ up to which particles are considered unstable. This will make long-living particles stable, even if they are set unstable in Hadron.dat. By default, this switch is turned off, i.e. everything is treated as specified in Hadron.dat.

4.2.12 Analysis.dat

SHERPA has a built-in framework, which is capable of carrying out simple analyses of the generated events. It is possible to restrict the analysis on single event phases: the level of the hard matrix element, the parton-shower evolution level, the hadronization level. On the other hand one can do some or all of these subanalyses in one single run.
The main switch to enable this analysis framework is ANALYSIS, which can be set in the file Run.dat or from the command line. ANALYSIS=1 and ANALYSIS=0 correspond to the analysis framework being switched on and off, respectively.

The results of the analysis are histograms, which are written to disk as text files. These files have the following form:

```
10 102 -0.221849 1.77815 0.004  0  100000
-0.221849  0
-0.201849  0.001
     .        .
     .        .
     .        .
-0.181849  0.0015
-0.161849  0.001
-0.141849  0.001
```

where the first line gives (in order of appearance) the type of the histogram, the number of bins, the lower and the upper range of the histogram, the underflow and the overflow weight and the number of entries. The type of the histogram is `10` in case the histogram is binned logarithmically; if it is binned linearly the histogram type is `0`. Each of all the other lines gives the lower edge of the corresponding bin and its entry. If error storage is enabled an additional item specifies the Monte Carlo error of the entry.

By default, the analysis setup will be read from the file `Analysis.dat`. The example files for the default and the `Tevatron1800` setup of the `SHERPA` distribution can be found in Fig. 18 of appendix A. It is however possible to specify another `filename` by setting the variable `ANALYSIS_DATA_FILE` in `Run.dat` with this `filename`.
The syntax of the setup file is as follows

1. Each analysis specification must be enclosed between the statements `BEGIN_ANALYSIS` and `END_ANALYSIS`. An event phase for the analysis must be specified through the keyword `LEVEL`. The possible phases are `ME`, `Shower` and `Hadron`. To have these or some of these analyses in one run, the corresponding values are simply concatenated by a space or a plus sign.

2. An output subdirectory for the analysis results may be given via `PATH_PIECE <directory>`. If a directory with the corresponding name does not exist yet, `SHERPA` will create it, if there are sufficient permissions on the selected medium.

3. The next step is to specify cuts to be applied on the final state particle list. If any of these cuts is not passed the event enters the analysis as a nil event. There are several possibilities to restrict the final state. To obtain a complete list of selectors including the syntax for their setup `SHERPA` provides the command line option `SHOW_ANALYSIS_SYNTAX=1`[13]. Running `SHERPA` with this command will stop the program after the printout of the syntax for the analysis setup. The interested user is kindly referred to inform about the various selectors by scanning the generated output. Each list entry containing the tag `Sel` in its name is in fact a selector. A special selector is `Trigger`.

4. Finally the observables have to be specified. For the list of observables available in `SHERPA`, the user is kindly asked to inform about them using the `SHOW_ANALYSIS_SYNTAX=1` option described above. To get used to the syntax specifying the analysis setup, the analysis example file for the default setup of the `SHERPA` distribution (see Fig. 18 of appendix A) may serve as a helpful guide.

5. At the end of the event generation run, there will be created a histogram output file for each observable. It resides in the directory with the name of the corresponding analysis phase within the subdirectory given by `PATH_PIECE`, which itself resides in the directory given by `ANALYSIS_OUTPUT`. The filename depends on the observable itself. For the one- and two-particle observables, it is a concatenation of the observable name and the name of the particle (or the names of both particles in the case of a two-particle observable) for which the analysis has been carried out. This

[13]To get these information, please, make sure that the `ANALYSIS` switch in `Run.dat` is turned on.

is preceeded by the default filename extension dat. In any case the filename is suggestive for the data contained in the file.

6. Within the analysis setup file C-style comments such as // <comment> are supported. Also the characters # and % can be used to indicate comments throughout the file.

It is possible to specify multiple main analyses in the setup file. These may, e.g., contain different restrictions on the final state. To avoid confusions, it is however necessary to specify different output directories for different analyses, in order to avoid that similar histograms will be overwritten.

4.3 Hadron Decay Data

HADRONS++ is a module within the SHERPA framework that is able to treat hadron and τ decays. The τ decays are already fully implemented in different models with various matrix elements and form factor parameterisations, which need many parameters.
Note that only those particles decay that are switched on and that are **not** set as stable in the Particle.dat and Hadron.dat. HADRONS++ employs three levels of data files. All names given here are the default names. They can be found in Decaydata/ which is a subdirectory of the SHERPA Run directory.

HadronDecays.dat ↓ decaying particles path to and name of the decay table	→	<particle>Decays/ <particle>Decays.dat ↓ decay channels (outgoing particles, BR) name of decay channel file	→	<mother>_<daughters>.dat ↓ parameters for decay channel
see (A) for details		see (B) for details		see (C) for details

(A) HadronDecays.dat consists of a table of particles that are to be decayed by HADRONS++. It has the following structure:

<kf-code>	->	<subdirectory>/	<filename>.dat
↓		↓	↓
decaying particle		path to decay table (note: entry must end with /)	name of decay table file
default names:		<particle>Decays/	<particle>Decays.dat

(B) The decay table contains information about outgoing particles for each channel, its branching ratio and eventually the name of the file that stores necessary parameters for a specific non-isotropic channel. If such a file is not specified HADRONS++ will produce it and modify the decay table file accordingly. Every hadron is supposed to have their own decay table in their own subdirectory. The structure of a decay table is

{<kf1,kf2,kf3,...>}		 /		<filename>.dat
↓	↓	↓
outgoing particles	branching ratio	name of decay channel file
default names:		<mother>_<daughters>.dat

It should be stressed here that the branching ratio which is explicitly given for any individual channel in this file is **always used**, independent of any evaluation. There is one keyword that can be set for each hadron individually:

1. NO_ANTI
   By default this option is also switched off. When switched on, HADRONS++ does not perform decays of the anti-particle of the corresponding hadron, no matter of their origin.

(C) A decay channel file contains information about parameters that are given to the class that evaluates the decay matrix element of that specific decay channel. The parameters given in a decay channel file are needed for the parameterisation of the matrix element. Some parameters can be global factors such as CKM matrix elements or kinematical parameters such as resonance masses, widths, etc.. However, the setting here do not have any influence on any other part of SHERPA.

If a necessary parameter is not given in the file, HADRONS++ takes its value from HadronConstants.dat. If it is not given their either, HADRONS++ takes the corresponding value from Model.dat or takes an initial default value. All decay channel files should have the following structure:

```
#<mother> --> <daughter1> <daughter2> ...    →  decay channel
#0        --> 1           2           ...    →  particle numbers for this channel
Channels {
    AlwaysIntegrate = <0,1>                  →  0... read result (default)
                                                1... integrate always
    <integrator 1> <weight>                  →  name of integrator and weight
    <integrator 2> <weight>
         ⋮
}
Parameters {
    <parameter> = <value>                    →  set parameter to specific value
    <parameter> = <value>                       (value can be a numbe or
    <parameter> = <value>                       a C-like syntax formula)
         ⋮
    <complex parameter> = <abs> <phase>      →  a complex parameter is specified by
                                                its absolute value and its phase
}
Result {
    <partial width> <error> <maximum>
}
```

The last three lines (Result) have quite an important meaning. If these lines are missing, HADRONS++ integrates this channel during the initialisation of SHERPA tuning the integration procedures. (In fact, this means that the completely ignored partial width of this decays is evaluated according to the implemented model.) After this integration, these last three lines are added in by the HADRONS++ module. If a result is written and AlwaysIntegrate is off (the default value) then HADRONS++ reads in the maximum for unweighting kinematics and the integration setup. Consequently, if some parameters are changed (also masses of incoming and outgoing particles) the maximum might change such that a new integration is needed in order to obtain correct kinematical distributions. There are two ways to enforce the integration: either by deleting the last three lines or by setting `AlwaysIntegrate` to 1. When a channel is re-integrated, `HADRONS++` copies the old decay channel file into `<mother>_<daughters>.dat.old`.

There are two scripts within the decay data directory:

1. `makeclean`:
   It goes through all decay channel files and deletes the result lines, usually the last three lines. It also deletes phase space functions and tabulated form factor functions that are used for τ decays. This way, `HADRONS++` is forced to re-integrate all its channels as well as it then must re-create the phase space functions for τ decays.

2. `makeold`:
   It renames all `<name>.dat.old` files back into `<name>.dat`.

Spin correlations The spin correlation algrorithm is fully implemented. It can be switched on through the keyword `SPIN_CORRELATIONS` in `Run.dat`, see Sec. 4.2.1 for more details.

Adding new channels If new channels are added to `HADRONS++` (choosing isotropic decay kinematics) a new decay table must be defined and the corresponding hadron must be added to `HadronDecays.dat`. The decay table merely needs to consist of the outgoing particles and branching ratios, i.e. the last column (the one with the decay channel file name) can safely be dropped. By running `SHERPA` it will automatically produce the decay channel files and write their names in the decay table.

Some details on τ decays τ decays are treated within the `HADRONS++` framework, even though the τ is not a hadron. In contrast to the implemented hadron decays, the τ's decay kinematics is fully implemented. Most parameters in the corresponding decay channel files are self-explaining or explained through comments. Here are some references for the implemented models for the semileptonic decays:

mode	reference (also references herein were used)
2 pseudoscalars	MPI-PAE/PTh 17/90 for Kühn-Santamaría
	hep-ph/9707347 for Resonance Chiral Theorie
3 pseudoscalars	hep-ph/9310270 for Kühn-Santamaría
	hep-ph/0312183 for Resonance Chiral Theorie
4 pions	CERN-TH.6793/93
4 pseudoscalars (with kaons)	isotropic
5, 6 pseudoscalars	isotropic
decays inolving η	isotropic

The choice of model, where applicable, is made through the keyword `FORM_FACTOR`. `RUNNING_WIDTH` is a switch which tells `HADRONS++` either so use constant resonance widths or running resonance widths. The default setup might change between different decay channels.

4.4 Command lines

For the command lines option of specifying parameters, all keywords of all data files are available, which refer to parameters that can be specified through a command structure like

`KEYWORD=value` .

In fact, for the command lines option it is crucial that no spaces separate the "=" mark and the keyword or value. In contrast, an arbitrary number of spaces is needed to separate the settings of different keywords/parameters/switches.
If events written out by SHERPA in a previous run should be read in, e.g. for another analysis, the useful keyword is EVTDATA and the command looks like

`./Sherpa EVTDATA=filename` .

5 Selecting processes – merging the matrix elements with the parton showers

5.1 How the merging works

SHERPA's most prominent feature is the consistent combination of matrix elements at the tree level with the parton shower. According to the algorithms proposed in [28, 29], this merging procedure leads exact results to (next-to) leading logarithmic order. The key idea is to separate the phase space for parton emission into a hard region of jet production – accounted for by suitable matrix elements – and the softer region of jet evolution – covered by the parton shower. Then, extra weights and vetoes are applied on the former and on the latter region, respectively, such that the overall dependence on the intersection cut is minimal. In SHERPA this merging proceeds in a fully automated way: for the user it is, in principle, sufficient to select some processes in `Processes.dat` and to switch on the `SUDAKOV_WEIGHT` in `ME.dat`, and the parton shower(s) in the initial and final states.

The merging proceeds along the following steps:

1. The total cross sections for all processes are calculated or read in by calling the method `CalculateTheHardProcesses()` of the `Initialization_Handler`. This is done during the initialization and preparation phase of SHERPA. It is important that the admixture of different processes is chosen with care; for inclusive processes, for instance, the number of extra jets in the processes must be continuous with "no gap". Also, no extra one-particle cuts (such as cuts on $E_\perp$, $p_\perp$, E, etc.) should be applied on particles through the file `Selector.dat` apart from the ones provided by the jet algorithm. The reason for this is that the parton shower automatically induces boosts not only along the beam axis – they tend to be comparably mild – but also in the transverse plane[14]. In contrast, invariant masses of pairs of particles tend to be almost stable under the parton shower treatment, their application therefore is uncritical.

2. It is of crucial importance that, if results are to be written out and recycled in later runs, already during the corresponding integration the parton showers have to be switched on in `Shower.dat`. Although the parton showers are of no direct consequence for the cross sections, they influence the information written out in the resulting data files.

3. Before the actual event generation starts, look-up tables for the Sudakov form factors are prepared. The Sudakov form factors depend on two scales, one of which is the jet resolution scale, which characterizes the softest scale for the emission of QCD particles accessible by the matrix elements.

[14] Remember that the parton shower is an expansion around the collinear limit; the initial-state parton shower, therefore, will respect to a good approximation the beam direction; in contrast, since critical transverse momenta are usually much smaller than the beam energy, any mild change in this direction really matters. This is why cuts on the pseudo rapidity of individual particles do not have such a tremendous impact.

This scale is ultimately defined by the jet algorithm and it is of utmost importance that the phase space for jet production is not further constrained. This guarantees that the two regions – jet production and jet evolution – match. Note that the Sudakov form factors for the highest jet multiplicity treatment are calculated "on the flight".

4. The hard matrix element of each event is selected according to the total cross section of the corresponding process. The particles are distributed according to the differential cross section. These events are generated through the method `GenerateOneEvent()` of the `Matrix_Element_Handler`, the information concerning momenta etc. is made available through corresponding simple access methods. The `Signal_Processes` handler will fill a tentative `Blob`[15].

5. Before the parton shower(s) may start, the parton configuration of the hard matrix element has to be analysed. `SHERPA` provides an interface structure, the most involved one being the `Amegic_Apacic_Interface`, employed by the `Event_Phase_Handler` for the parton shower, dubbed `Jet_Evolution`. In this interface, the partons are clustered backwards according to the same jet algorithm used for the cuts on the final-state phase space of the matrix elements. The clustering automatically yields the nodal values of each parton emission; in fact, by this "clustering-backwards" procedure, a pseudo classical limit of the radiation pattern has been constructed, which reproduces the parton shower history in its logarithmic approximation. Equipped with the nodal values, the Sudakov rejection weight (including the α_s reweighting) can be calculated. The parton configuration and, thus, the (tentative) hard event is accepted or rejected according to this weight. In case it is rejected, the event generation continues with the previous step, possibly leading to a completely new process with different flavours and different multiplicity.

6. In case the hard event is accepted, the parton shower is invoked, starting from the pseudo history constructed before. Each parton starts its evolution at the scale where it was produced, subject to the condition that no extra jet is produced. For the virtuality-ordered parton shower of `APACIC`, the scale is the corresponding invariant mass. The exception to this veto is for configurations with the maximal number of jets produced through the matrix elements. There, the parton shower has to cover the phase space for more jets than produced by the matrix elements, the veto therefore is on partons at scales harder than the softest scale of the hard event.

5.2 How the merging does not work

There are a number of ways, in which the merging procedure can fail. Some of them will be briefly discussed below.

5.2.1 Missing jets

When mixing processes with different numbers of jets, it is not a good idea to leave out some of them. For instance, for inclusive W production the user may want to initialize processes with a W final state plus a number of additional jets: $jet\,jet \to e\bar{\nu}_e, e\bar{\nu}_e + jet, e\bar{\nu}_e + 2\,jets, e\bar{\nu}_e + 3\,jets, \ldots$. The treatment for the highest jet multiplicity is such that the interplay of Sudakov suppression on the matrix elements and jet veto on extra jets in the parton shower allows for more than this highest number of jets in a fashion that reproduces the soft/collinear limit of the according matrix element(s). Therefore, omitting the highest jet multiplicity has a significant unpleasant effect only, if the user is interested in correlation on this number of jets that are sensitive to a full quantum-mechanically correct treatment.
However, missing the subprocesses with some of the lowest numbers of jets (like, for instance, the subprocess $jet\,jet \to e\bar{\nu}_e$) may lead to significant holes in the soft region of many distributions.

[15] For details concerning the event record, cf. Sec. 6.

Similarly, leaving out some intermediate numbers of jets will lead to unphysical bumps or holes in the distributions.

5.2.2 Extra cuts on the phase space of jets

As explained above, cuts on jets, especially on their $p_\perp$, energy, etc., have two effects: First of all they define the measure of the Sudakov suppression, second they define, which region is populated by the parton shower and which one by the matrix element. Both, the Sudakov suppression and the parton-shower phase space are defined solely by the jet definition. Applying cuts, e.g. on the $p_\perp$ of jets, which are harder than the jet cuts, will immediately result in a gap. This gap is the deeper the worse the mismatch of scales is. Therefore, applying extra cuts, especially on the jets, will almost always lead to wrong predictions of physics.

5.2.3 Wrong result directories

To store results of the matrix element calculations in a directory[16] and to recycle them in future runs seems to be a good idea. Especially when going to high final-state multiplicities, it saves quite a lot of CPU time. However, there are a number of potential mismatches, for example:

- different c.m. energy of the beams or different cuts on the final state;
- different values of α_s;
- the merging has been switched off during the calculation;
- different numbers of maximal jet multiplicities.

The latter two matter, since the results of the matrix element calculation are evaluated with the α_s at the merging scale, provided that the merging is switched on. This scale is either given by the jet resolution scale or by the scale of the softest jet in those matrix elements with the maximal jet multiplicity.

6 How to interprete the event record

6.1 `SHERPA`'s output format

`SHERPA`'s event record is based on a list of linked `Blobs`. They reflect some very general structure that transforms particles into other particles. Consequently they have a number $n_{\rm in}$ and a number $n_{\rm out}$ of incoming and outgoing particles, respectively[17]. Some care is taken that four-momentum conservation is locally fulfilled at each of the `Blobs`. Apart from the `Particles` attached to them, `Blobs` are located in space time, giving rise to production and decay positions of their particles. They are further specified by an identity number, a status flag, and a type – coded as an `enum` structure – such as `Signal_Process`, `ME_PS_Interface`, `IS_Shower` or `FS_Shower`.
Physical particles in `SHERPA` are implemented in their own class. As mentioned before, apart from those entering the event as beams and those leaving it as final-state particles, particles have a production and a decay blob, providing them with information of where they have been produced and where they decayed. This structure also incorporates in a natural way mother-daughter relations (which is therefore obsolete to encode in a different way). When the parton shower is switched on, partons, i.e. quarks and gluons, also carry two colour indices in a kind of $\mathbf{3}\otimes\bar{\mathbf{3}}$ representation, but with N_c going to infinity[18]. Of

[16] In Sec. 3.1 the user may find some hints how this can be achieved.
[17] In terms of programming, the `Blobs` own pointers to the respective `Particles` and these `Blobs` are responsible of erasing them after each event.
[18] By the way, this is one of the assumptions implicit in one way or the other in all event generators.

course, particles carry a four momentum, given in units of GeV and an id-number. Finally, particles also have a status flag, basically `1` or `2` for particles, which are still active, or particles, which are decayed or fragmented, respectively. Finally, more information can be obtained from an info flag with the meanings listed in Tab. 9.

info	meaning
G	initial-state parton for hard interaction
H	final-state parton from hard interaction
I	initial parton (shower initiator)
i	parton from ISR
F	final parton (active)
f	parton from FSR (decayed)
D	particle from hadron decays (active)
d	particle from hadron decays (decayed)
P	primary active hadrons, directly from fragmentation (active)
p	primary active hadrons, directly from fragmentation (decayed)

Table 9: Particle information codes and their meaning in the event record.

The full output of a typical event of the $e\gamma$ default setup is listed in appendix B, its structure is represented pictorially in Fig. 2. To exemplify it more in detail, consider the last `Blob`, which reads:

```
Blob [1]( 58, Hadron Decay      , 1 -> 3 @ (8.56e-05,-3.032e-05,-5.306e-05,-5.98e-05)
Incoming particles :
[d] 1 pi             147 ( 56 -> 58)  [( 2.9158e+00,-1.0320e+00,-1.8082e+00,-2.0370e+00), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon         148 ( 58 ->   )  [( 1.7688e+00,-6.0006e-01,-1.1475e+00,-1.2048e+00), 1.3323e-15] (0,0)
[D] 1 e+             149 ( 58 ->   )  [( 6.9901e-01,-2.6415e-01,-3.9638e-01,-5.1159e-01), 2.6010e-07] (0,0)
[D] 1 e-             150 ( 58 ->   )  [( 4.4805e-01,-1.6783e-01,-2.6423e-01,-3.2058e-01), 2.6010e-07] (0,0)
```

The first line is for the blob, its status ([1]), its number (58) and type (`Hadron Decay`), the number of incoming and outgoing particles ($1 \to 3$), and its four position in the notation $(t, \vec{x})$. Then the particles are listed with its particle info code, status, type[19], its number, information about the production and decay blob[20], its four momentum in the notation $(E, \vec{p})$, and finally its colour charges[21]. The colour flow is represented in the $N_c \to \infty$ limit by two numbers standing for the $\mathbf{3} \otimes \bar{\mathbf{3}}$ components.

6.2 Other formats to output events

`SHERPA` provides the possibility to output events – either to file or to screen – in two other output formats: the user may output events in the `HepEVT` common block structure or in the `HepMC` format [32]. The authors of `SHERPA` assume that the user is sufficiently acquainted with these formats when selecting them.

In order to write events into files, one of the keywords `SHERPA_OUTPUT`, `HEPEVT_OUTPUT`, or `HEPMC_OUTPUT` has to be provided with a name. See section 4.2.1 for information about these switches.

Acknowledgements

The authors of `SHERPA` would like to thank everybody who helped us to improve this package. We are very grateful that many people encouraged us to continue with this project. Special thanks go to all

[19] In most cases, when the particle is electrically neutral, the charge subscript is missing.

[20] For instance, the incoming π^0 is produced in `Blob 56` and decays in `Blob 58`.

[21] Obviously, hadrons, photons, etc. have no colour charges.

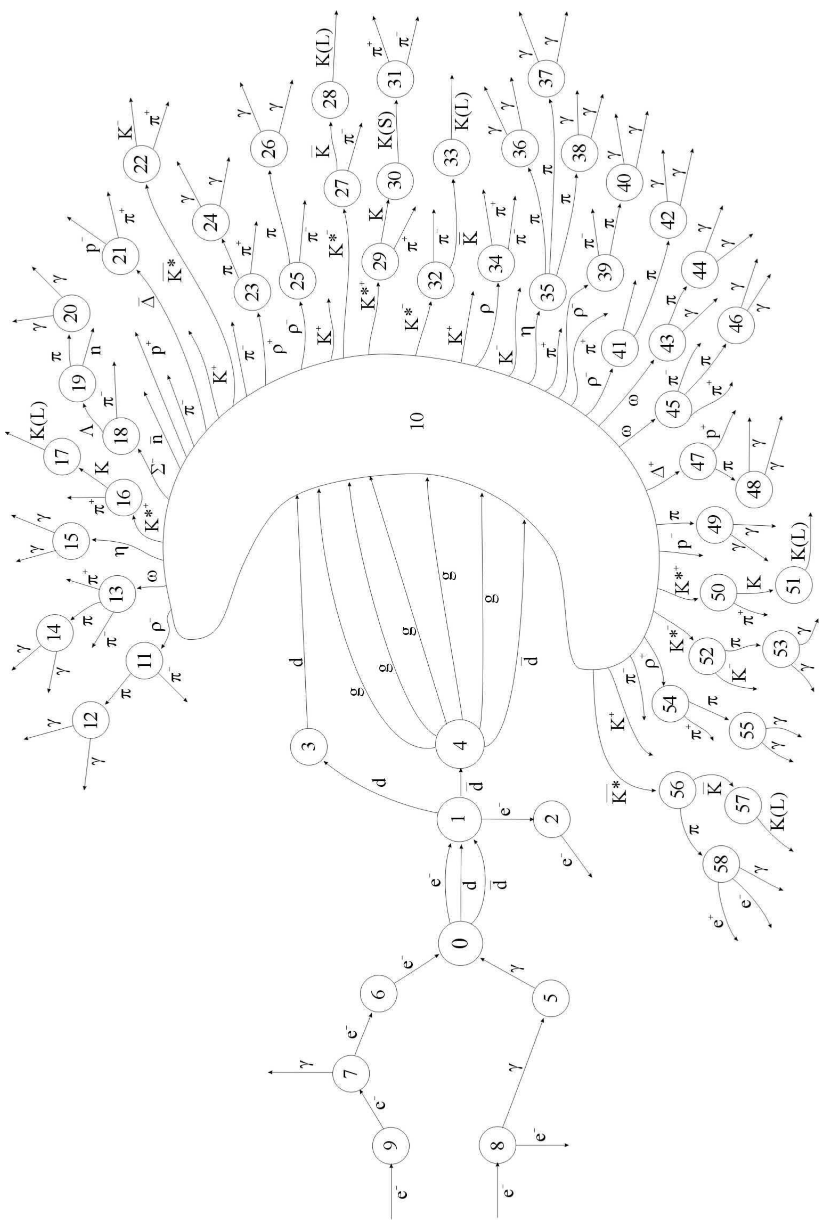

Figure 2: An example event, which is typical for the $e\gamma$ default setup.

users who actively supported the development by giving detailed comments and suggestions, and by providing bug reports and fixes to this new version and to all the previous releases. The authors are indebted to all people who promoted the usage of SHERPA for carrying out various physics analyses.

A Data files

```
!-------------------------------------------------------
!--- Run parameters ------------------------------------
!-------------------------------------------------------
EVENTS          = 2       ! Number of events
Num. Accuracy   = 1.e-10  ! Numerical accuracy
OUTPUT          = 7       ! Output level:
!                         ! 0=errors,1=events,2=info,
!   ! 4=tracking,6=info+tracking, etc.
!
PRETTY_PRINT = On ! Coloured output
!
BEAM_DATA_FILE     = Beam.dat
ISR_DATA_FILE      = ISR.dat
ANALYSIS_DATA_FILE = Analysis.dat
!
ANALYSIS        = 0  ! Analysis: 0=Off, 1=On
ANALYSIS_OUTPUT = analysis ! Analysis output directory
!
LOG_FILE = sherpa_run_ep.log  ! created if OUTPUT=4
```

Figure 3: The Run.dat file of the default setup. Most of the keywords should be self-explanatory, the output levels can be added and their individual definitions are explained in Tab. 3).

```
!---------------------------------------------------------
!-- Beam parameters --------------------------------------
!---------------------------------------------------------
BEAM_1          = 11                    ! possible beam particles:  P+, P-, e+, e-
BEAM_ENERGY_1   = 250.                  ! in GeV
BEAM_POL_1      = 0.                    ! Polarization degree -1 ... 1
BEAM_SPECTRUM_1 = Laser_Backscattering ! Monochromatic, Laser_Backscattering,
                                        ! Simple_Compton
K_PERP_MEAN_1 = 0.
K_PERP_SIGMA_1 = 0.
!
BEAM_2          = 11                    ! possible beam particles:  P+, P-, e+, e-
BEAM_ENERGY_2   = 250.                  ! in GeV
BEAM_POL_2      = 0.                    ! Polarization degree -1 ... 1
BEAM_SPECTRUM_2 = Monochromatic         ! Monochromatic, Laser_Backscattering,
                                        ! Simple_Compton
K_PERP_MEAN_2 = 0.
K_PERP_SIGMA_2 = 0.
!
BEAM_SMIN       = 0.3              ! Minimal fraction of nominal s after beam spectra
BEAM_SMAX       = 1.0              ! Maximal fraction of nominal s after beam spectra
!
E_LASER_1          = 1.17e-9            ! Laser energy in GeV
P_LASER_1          = 0.                 ! Laser polarization +-1
E_LASER_2          = 1.17e-9            ! Laser energy in GeV
P_LASER_2          = 0.                 ! Laser polarization +-1
LASER_MODE         = 0                  ! 0 = all, 1,2,3 = individual components
LASER_NONLINEARITY = On                 ! On/Off
```

Figure 4: The `Beam.dat` file of `SHERPA`'s default setup. A few comments are in order: all energies are in GeV, for particle codes cf. Figs. 19 and 20. Beam polarizations are given as (circular) polarization/helicity degree ranging from `-1` to `1`. `BEAM_SMIN` and `BEAM_SMAX` rule the s range ($E_{\rm cm}^2$) after the beam spectra have been applied, `Monochromatic` obviously results in $s = (E_1 + E_2)^2$. The laser specifications apply only, if `Laser_Backscattering` or `Simple_Compton` are selected. The `K_PERP` specifications steer the breakup of the beam remnants by specifying the $k_\perp$ distribution.

```
!-----------------------------------------------------
!-- ISR parameters -----------------------------------
!-----------------------------------------------------
BUNCH_1         = 22        ! possible beam particles:  P+, P-, e+, e-
ISR_1           = Off       ! On/Off
!
BUNCH_2         = 11        ! possible beam particles:  P+, P-, e+, e-
ISR_2           = On        ! On/Off
!
ISR_SMIN        = 0.01      ! Minimal fraction of nominal s for parton after ISR
ISR_SMAX        = 1.0       ! Maximal fraction of nominal s for parton after ISR
!
ISR_E_ORDER     = 1         ! Perturbative order of electron structure function
ISR_E_SCHEME    = 2         ! Beta-scheme : 0,1,2 , default = 2
!
PDF_SET         = cteq6l    ! For LHAPDF use the following PDFs in
                            ! in combination with PDFsets
                            ! Alekhin_100,Alekhin_1000,Botje_100,Botje_1000,
                            ! Fermi2002_100,Fermi2002_1000,
                            ! MRST2001,MRST98,cteq6,cteq6l,cetq6ll
                            !
                            ! alternatively use cteq6 via CTEQ6Grid
                            ! cteq6m,cteq6d,cteq6l,cteq6l1
                            !
                            ! Or use interface to MRST99 via MRST99Grid
                            ! MRST99
PDF_SET_VERSION = 1
PDF_GRID_PATH   = CTEQ6Grid ! PDFsets,MRST99Grid,CTEQ6Grid
```

Figure 5: The `ISR.dat` file of the default setup. `ISR_SMIN` and `ISR_SMAX` steer the s range after the bunch-particle parton transition has taken place. Note that some of the PDFs (cteq6-sets) are available either – in its "original form" – through the Fortran interface or through the LHAPDF interface. The one selected depends on the variable `PDF_GRID_PATH`. If the user chooses to simulate collisions involving a hadron or hadrons the corresponding ISR has to be switched on.

```
!-------------------------------------------------------
!-- Model parameters -----------------------------------
!-------------------------------------------------------
MODEL                 = SM       ! Model
!
GENERATOR_ON          = 0        ! use external spectrum generator
!
! SM parameters
!
EW_SCHEME             = 0        ! which parameters define the ew sector.
```

Figure 6: The general part of the file `Model.dat` of the default setup. The external spectra generators can be switched on or off (For SUSY they must be switched on at the moment!). The different schemes to set the electroweak parameters are listed in the text.

```
ORDER_ALPHAS       = 1  ! order of running coupling
ALPHAS(MZ)            = 0.1188   ! strong coupling at scale M_Z
ALPHAS(default)       = 0.0800   ! strong coupling
1/ALPHAQED(0)         = 137.036  ! inverse of alpha QED in the Thomson limit
1/ALPHAQED(default)   = 132.51   ! inverse of alpha QED
SIN2THETAW            = 0.2222   ! Weinberg angle at scale M_Z
VEV                   = 246.     ! Higgs vev
LAMBDA                = 0.47591  ! SM Higgs self coupling
YUKAWA_E              = 0.       ! Yukawa coupling of electron
YUKAWA_MU             = 0.105    ! Yukawa coupling of mu
YUKAWA_TAU            = 1.77     ! Yukawa coupling of tau
YUKAWA_U              = 0.       ! Yukawa coupling of u quark
YUKAWA_D              = 0.       ! Yukawa coupling of d quark
YUKAWA_S              = 0.       ! Yukawa coupling of s quark
YUKAWA_C              = 1.3      ! Yukawa coupling of c quark
YUKAWA_B              = 4.5      ! Yukawa coupling of b quark
YUKAWA_T              = 175.     ! Yukawa coupling of t quark
MSTRANGE_EFF          = 0.48     ! effective strange mass for higgs-loops.
CKMORDER              = 0        ! order of expansion of CKM matrix in Cabibbo angle
CABIBBO               = 0.22     ! Cabibbo angle (Wolfenstein parametrization)
A                     = 0.85     ! A (Wolfenstein parametrization)
RHO                   = 0.50     ! rho (Wolfenstein parametrization)
ETA                   = 0.50     ! eta (Wolfenstein parametrization)
```

Figure 7: The Standard Model part of the file `Model.dat` of the default setup. The settings are described in more detail in Sec. 4.2.4.

```
HIGGS_GENERATOR       = Hdecay   ! Generator for Higgs masses & widths
HIGGS_NNLO(M)         = 1        ! Use nnlo running of masses in higgs decays :
                                 ! allowed values 1,2
HIGGS_ON_SHELL_WZ     = 1        ! Only on-shell gauge bosons in higgs decays :
                                 ! 0 = double,1 = single
HIGGS_MASS_SCHEME     = 0        ! Scheme for higgs masses in higgs decays    :
                                 ! running/pole
HIGGS_NF_LIGHT_Q_IN_G = 5        ! Number of light flavours in higgs
                                 ! decays h->gg*->g qqbar
```

Figure 8: The part of the default setup file `Model.dat` steering the HDecay program. For more details the reader is referred to Sec. 4.2.4.

```
!
! SUSY parameters
!
SUSY_SCENARIO         = mSUGRA   ! also available : AMSB, mGMSB,non-minimal GMSB,
                                 !                  SUGRA with enforced unification,
                                 !                  non-universal SUGRA
M0                    = 200.     !
M12                   = 200.     !
A0                    = 100.     !
TAN(BETA)             = 3.       !
SIGN(MU)              = -1       !
!
! ADD parameters
!
N_ED                  = 2          ! Number of extra dimensions in ADD model
G_NEWTON              = 6.707e-39 ! Newton gravity constant
M_S                   = 2.5e3      ! string scale for ADD model,
                                   ! meaning depends on KK_CONVENTION
M_CUT                 = 2.5e3      ! cut-off scale for the c.m. energy
KK_CONVENTION         = 1          ! 0=const mass
                                   ! 1=simplified sum(HLZ)   2=exact sum(HLZ)
                                   ! 3=Hewett +1    4=Hewett -1
                                   ! 5=GRW
                                   !   (Lambda_T(GRW)=M_S
                                   !      for virtual Graviton exchange  or
                                   !    M_D=M_S
                                   !      for real Gravtion production)
                                   ! only KK_CONVENTION's 1,2 or 5 are applicable
                                   ! for real Graviton production!
```

Figure 9: The new physics parameters adjustable in the file `Model.dat` of the default setup. More details can be found in Sec. 4.2.4.

```
!-------------------------------------------------------
!-- ME generators --------------------------------------
!-------------------------------------------------------
ME_SIGNAL_GENERATOR   = Amegic      ! Internal or Amegic
EVENT_GENERATION_MODE = Unweighted
!
SIGNAL_MODEL          = SM          ! subset of a physical model
                                    ! or the whole SM, MSSM, ADD
!
COUPLING_SCHEME       = Fixed       ! Fixed   (default value) or
                                    ! Running (s for ME generators, pt for 2->2)
YUKAWA_MASSES         = Fixed       ! Fixed   (polemass) or
                                    ! Running (higgs mass)
YUKAWA_MASSES_FACTOR  = 1.          ! Additional prefactor for yukawas
!
SCALE_SCHEME          = 0           ! default is 0=shat,
KFACTOR_SCHEME        = 0           ! 1=as(pt2)/as(ecms)^nstrong,default=1
SUDAKOV_WEIGHT        = 0           ! apply sudakov weight on single events
!FIXED_SCALE          = 8315.25     ! fixed scale for the processes in GeV^2
SCALE_FACTOR          = 1.          ! factor the scale is multiplied with
```

Figure 10: This is an example for the ME.dat file, here the parameter settings for the default setup of SHERPA are shown.

```
%-------------------------------------------------------
%-- Processes to calculate -----------------------------
%-------------------------------------------------------
%
% photon electron -> electron d dbar
%
Process : 22 11 -> 11 1 -1
End process
%
```

Figure 11: The `Processes.dat` file is shown for the `SHERPA` default setup. Which particles belong to the process is specified by the particles' `kf` code.

```
%------------------------------------------------------
%-- Processes to calculate -----------------------------
%------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% jet jet -> electron anti-neutrino + n-jets
%
Process : 93 93 -> 11 -12
Order electroweak : 2
Order strong : 0
End process
%
Process : 93 93 -> 11 -12 93
Order electroweak : 2
Order strong : 1
End process
%
Process : 93 93 -> 11 -12 93 93
Order electroweak : 2
Order strong : 2
End process
%
%Process : 93 93 -> 11 -12 93 93 93
%Order electroweak : 2
%Order strong : 3
%End process
%
%Process : 93 93 -> 11 -12 93 93 93 93
%Order electroweak : 2
%Order strong : 4
%End process
```

Figure 12: The `Processes.dat` file of the alternative default setup – this is the one given in the directory `Run/Tevatron1800`.

```
!----------------------------------------------------------
!-- Parton level selectors -----------------------------
!----------------------------------------------------------
JetFinder   4.e-3  1.
ET         11       1.   10000.
ET          1       1.   10000.
ET         -1       1.   10000.
BeamAngle   1  0   -0.9      0.9
BeamAngle  -1  0   -0.9      0.9
BeamAngle  11  0   -0.9      0.9
```

Figure 13: The Selector.dat file, which belongs to SHERPA's default setup. A list of possible selectors available within SHERPA is given in Tab. 7.

```
!----------------------------------------------------------
!-- Phase space setup ------------------------------------
!----------------------------------------------------------
ERROR =  2.e-3   ! Error by calculating matrix-elements
!
FINISH_OPTIMIZATION = On ! Integrate until optimization is completed
!
INTEGRATOR = 4   ! Phasespace : Rambo=0, Sarge=1,
                 !              Rambo + Sarge=2,
         !              Rambo + Multichannel=3,
 !             Multichannel:
 !             =4 (Original channel generator)
 !             =5 (Original+extra channels for competing peaks
 !                 like in ->Higgs->ZZ->q qb q qb)
 !             =6 (New channel generator)
 !             RamboKK=7
 ! Choice 4-6 only affects channel generation!
VEGAS = On
```

Figure 14: The Integration.dat data file of the default setup. Note that RamboKK is the only choice when considering outgoing Kaluza-Klein resonances.

```
!-------------------------------------------------------
!-- Parton showers -------------------------------------
!-------------------------------------------------------
SHOWER_GENERATOR  = Apacic   !
FSR_SHOWER        = 1        !  1=On,0=Off
ISR_SHOWER        = 0        !  1=On,0=Off
!
IS_PT2MIN         = 4.       !
FS_PT2MIN         = 1.       !
```

Figure 15: The `Shower.dat` data file of the default setup. Showers should be consistently switched on or off. In case the user wishes to study the effect of the Sudakov form factors etc. from the merging procedure the showers must be switched on.

```
!==================================================!
!          Underlying Event Setup file            !
!==================================================!
!
! general parameters
!
MI_HANDLER = None  ! Amisic / None
!
```

```
!==================================================!
!          Underlying Event Setup file            !
!==================================================!
!
! general parameters
!
MI_HANDLER = None  ! Amisic / None
!
! hard underlying event parameters
!
CREATE_GRID 93 93 -> 93 93 ! processes to generate
MI_SCALE_SCHEME     = 2 ! scale scheme
MI_K_FACTOR_SCHEME  = 1 ! k factor scheme
PS_ERROR     = 3.0e-2  ! error for integration
VEGAS_MI     = 1 ! vegas channels 0 / 1
REGULATE_XS     = 0 ! regulate cross section
XS_REGULATOR     = QCD_Trivial ! regulation scheme
! QCD_Trivial
XS_REGULATION     = 2.1 ! regulation parameter
SCALE_MIN     = 2.1  ! minimum scale
RESCALE_EXPONENT    = 0.25 ! rescaling exponent
REFERENCE_SCALE     = 1960.0 ! reference energy scale
PROFILE_FUNCTION    = Double_Gaussian ! Gaussian / Double_Gaussian
PROFILE_PARAMETERS  = 1.0 0.2 0.5 ! size (must be 1), coresize,
! matter fraction
```

Figure 16: The MI.dat data file of the default setup is shown above. Below that the corresponding file for the Run/Tevatron1800 setup is exhibited. Note that, currently, multiple interactions may only be generated for proton–proton collisions.

```
!-------------------------------------------------------
!-- Fragmentation parameters----------------------------
!-------------------------------------------------------
FRAGMENTATION = Lund       ! Off, Lund (Pythia string fragmentation)
LUND_FILE     = Lund.dat   ! Pythia parameter file
```

Figure 17: This is an example for the `Fragmentation.dat` file, here the parameter settings for the default setup of `SHERPA` are shown.

```
// for details run 'Sherpa SHOW_ANALYSIS_SYNTAX=1'

// tag for the beginning of an analysis
BEGIN_ANALYSIS {

  // the analysis level
  LEVEL Hadron

  // the output subdirectory
  PATH_PIECE _run_ep

 // the observables
  PT  11  0.  250. 100 Lin
}
END_ANALYSIS
```

```
// for details run 'Sherpa SHOW_ANALYSIS_SYNTAX=1'

// tag for the beginning of an analysis
BEGIN_ANALYSIS {

  // the analysis level
  LEVEL Hadron

  // the output subdirectory
  PATH_PIECE _run_W_jets

 // the Triggers
 Trigger {
     OutList leptons_jets
     Finder  93  10.  -2.  2. 0.7
     Finder  11  15.  -1.  1.
     Finder -12  15. -20. 20.
     Counts -12   1  -1
     Counts  11   1  -1
 }

 Trigger {
     InList  leptons_jets
     OutList jets
     Keep 93
 }

 // the observables
 JetPT    0.   250. 100    1  1  6 Lin jets
 JetEta  -2.     2.  40    1  1  6 Lin jets
 HT       0.   600.  60            Lin leptons_jets

} // tag for the end of an analysis
END_ANALYSIS
```

Figure 18: The `Analysis.dat` file for the default (on top) and the `Run/Tevatron1800` (at bottom) setup.

kf	Mass	Width	3*e	Y	SU(3)	2*Spin	maj	on	stbl	m_on	Name
1	.01	.0	−1	−1	1	1	0	1	1	0	d
2	.005	.0	2	1	1	1	0	1	1	0	u
3	.2	.0	−1	−1	1	1	0	1	1	0	s
4	1.42	.0	2	1	1	1	0	1	1	0	c
5	4.7	.0	−1	−1	1	1	0	1	1	1	b
6	174.3	1.5	2	1	1	1	0	1	1	1	t
11	.000511	.0	−3	−1	0	1	0	1	1	0	e-
12	.0	.0	0	1	0	1	0	1	1	0	nu_e
13	.105	.0	−3	−1	0	1	0	1	1	0	mu-
14	.0	.0	0	1	0	1	0	1	1	0	nu_mu
15	1.777	2.36E-12	-3	−1	0	1	0	1	1	1	tau-
16	.0	.0	0	1	0	1	0	1	1	0	nu_tau
21	.0	.0	0	0	1	2	0	1	1	0	gluon
22	.0	.0	0	0	0	2	0	1	1	0	photon
23	80.419	2.048	-3	0	0	2	0	1	1	1	W-
24	91.188	2.446	0	0	0	2	0	1	1	1	Z
25	120.0	0.0037	0	0	0	0	0	1	1	1	h
27	100.0	.0	0	0	0	4	0	1	1	1	graviton
28	100.0	.0	0	0	0	0	0	1	1	1	gscalar
31	93.9932	.0074	0	0	0	0	0	1	1	1	h0
32	590.726	2.02	0	0	0	0	0	1	1	1	H0
33	585.68	2.25	0	0	0	0	0	1	1	1	A0
34	591.21	.827	-3	0	0	0	0	1	1	1	H-
41	212.293	0.01	3	0	0	1	0	1	1	1	chargino1
42	503.425	0.82	3	0	0	1	0	1	1	1	chargino2
43	115.459	.0	0	0	0	1	1	1	1	1	neutralino1
44	213.234	.0	0	0	0	1	1	1	1	1	neutralino2
45	482.791	.0	0	0	0	1	1	1	1	1	neutralino3
46	506.624	.0	0	0	0	1	1	1	1	1	neutralino4
47	706.822	.0	0	0	1	1	1	1	1	1	gluino
51	633.761	.0	2	1	1	0	0	1	1	1	sup_L
52	633.763	.0	2	1	1	0	0	1	1	1	scharm_L
53	659.643	.0	2	1	1	0	0	1	1	1	stop_2
54	612.413	.0	2	1	1	0	0	1	1	1	sup_R
55	612.415	.0	2	1	1	0	0	1	1	1	scharm_R
56	456.05	.0	2	1	1	0	0	1	1	1	stop_1
61	636.928	.0	−1	−1	1	0	0	1	1	1	sdown_L
62	636.928	.0	−1	−1	1	0	0	1	1	1	sstrange_L
63	611.299	.0	−1	−1	1	0	0	1	1	1	sbottom_2
64	610.76	.0	−1	−1	1	0	0	1	1	1	sdown_R
65	610.76	.0	−1	−1	1	0	0	1	1	1	sstrange_R
66	585.172	.0	−1	−1	1	0	0	1	1	1	sbottom_1
71	228.838	.0	−3	−1	0	0	0	1	1	1	selectron_L
72	228.838	.0	−3	−1	0	0	0	1	1	1	smuon_L
73	228.991	.0	−3	−1	0	0	0	1	1	1	stau_2
74	153.1918	.0	−3	−1	0	0	0	1	1	1	selectron_R
75	153.191	.0	−3	−1	0	0	0	1	1	1	smuon_R
76	152.782	.0	−3	−1	0	0	0	1	1	1	stau_1
81	219.858	.0	0	1	0	0	0	1	1	1	snu_1
82	219.858	.0	0	1	0	0	0	1	1	1	snu_2
83	219.812	.0	0	1	0	0	0	1	1	1	snu_3

Figure 19: Particle listings given `Particle.dat`. It is highly recommended not to change the following entries: `kf`, `3*e`, `Y`, `SU(3)`, `maj`, and `name`.

kf	Mass	Width	3*e	Y	2*Spin	on	stbl	Name
111	0.134976	7.8e-09	0	0	0	1	0	pi
113	0.77	0.1507	0	0	0	0	0	rho(770)
130	0.497672	1.273e-17	0	0	0	0	0	K(L)
211	0.13957	2.5284e-17	3	0	0	1	0	pi+
213	0.77	0.1507	3	0	0	0	0	rho(770)+
221	0.5473	1.18e-06	0	0	0	0	0	eta
223	0.78194	0.00841	0	0	0	0	0	omega(782)
310	0.497672	7.373e-15	0	0	0	0	0	K(S)
311	0.497672	0	0	0	0	1	0	K
313	0.8961	0.0505	0	0	0	0	0	K*(892)
321	0.493677	5.314e-17	3	0	0	1	0	K+
323	0.89166	0.0508	3	0	0	0	0	K*(892)+
331	0.95778	0.000203	0	0	0	0	0	eta'(958)
333	1.01941	0.00443	0	0	0	0	0	phi(1020)
411	1.8693	6.23e-13	3	0	0	0	0	D+
413	2.01	0	3	0	0	0	0	D*(2010)+
421	1.8646	1.586e-12	0	0	0	0	0	D
423	2.0067	0	0	0	0	0	0	D*(2007)
431	1.9685	1.41e-12	3	0	0	0	0	D(s)+
433	2.1124	0	3	0	0	0	0	D(s)*+
441	2.9798	0.0132	0	0	0	0	0	eta(c)(1S)
443	3.09688	8.7e-05	0	0	0	0	0	J/psi(1S)
445	3.55617	0.002	0	0	0	0	0	chi(c2)(1P)
511	5.2792	4.22e-13	0	0	0	0	0	B
513	5.3249	0	0	0	0	0	0	B*
521	5.2789	3.99e-13	3	0	0	0	0	B+
523	5.3249	0	3	0	0	0	0	B*+
531	5.3693	4.27e-13	0	0	0	0	0	B(s)
533	5.41630	0	0	0	0	0	0	B*(s)
541	6.594	0.	3	0	0	0	0	B(c)+
543	6.602	0.	3	0	0	0	0	B*(c)+
551	9.4	0.	0	0	0	0	0	eta_b
553	9.46037	5.25e-05	0	0	0	0	0	Upsilon(1S)
555	9.9132	0	0	0	0	0	0	chi(b2)(1P)
1112	1.62	0.15	-3	0	0	0	0	Delta(1620)-
1114	1.232	0.12	-3	0	0	0	0	Delta(1232)-
1212	1.62	0.15	0	0	0	0	0	Delta(1620)
1214	1.52	0.12	0	0	0	0	0	N(1520)
2112	0.939566	7.424e-28	0	0	0	0	0	n
2114	1.232	0.12	0	0	0	0	0	Delta(1232)
2116	1.675	0.15	0	0	0	0	0	N(1675)
2122	1.62	0.15	3	0	0	0	0	Delta(1620)+
2124	1.52	0.12	3	0	0	0	0	N(1520)+
2212	0.938272	0	3	1	1	1	1	P+
2214	1.232	0.12	3	0	0	0	0	Delta(1232)+
2216	1.675	0.15	3	0	0	0	0	N(1675)+
2222	1.62	0.15	6	0	0	0	0	Delta(1620)++
2224	1.232	0.12	6	0	0	0	0	Delta(1232)++
3112	1.19745	4.45e-15	-3	0	0	0	0	Sigma-
3114	1.3872	0.0394	-3	0	0	0	0	Sigma(1385)-
3116	1.775	0.12	-3	0	0	0	0	Sigma(1775)-

Figure 20: Some of the hadrons listed in `Hadron.dat`. It is highly recommended not to alter the entry given for `kf`.

B An example event output

The full output of a typical event of the $e\gamma$ default setup looks like

```
Blob [0]( 9, Bunch            ,  from Beam 1, 1 -> 1 @ (0,0,0,0)
Incoming particles :
[a] 2 e-             19 (    ->  9)  [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), 2.6113e-07] (0,0)
Outgoing particles :
[a] 2 e-             15 (  9 ->  7)  [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), 2.6113e-07] (0,0)

Blob [0]( 8, Bunch            ,  from Beam 0, 1 -> 2 @ (0,0,0,0)
Incoming particles :
[a] 2 e-             17 (    ->  8)  [( 2.5000e+02, 0.0000e+00, 0.0000e+00, 2.5000e+02), 2.6113e-07] (0,0)
Outgoing particles :
[G] 2 photon          1 (  8 ->  5)  [( 1.4762e+02, 0.0000e+00, 0.0000e+00, 1.4762e+02), 0.0000e+00] (0,0)
[a] 1 e-             18 (  8 ->   )  [( 1.0238e+02, 0.0000e+00, 0.0000e+00, 1.0238e+02), 1.0693e-07] (0,0)

Blob [2]( 7, Beam             ,  from Beam 1, 1 -> 2 @ (0,0,0,0)
Incoming particles :
[a] 2 e-             15 (  9 ->  7)  [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), 2.6113e-07] (0,0)
Outgoing particles :
[G] 2 e-              2 (  7 ->  6)  [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), 1.4552e-11] (0,0)
[F] 1 photon         16 (  7 ->   )  [( 5.4499e-04, 0.0000e+00, 0.0000e+00,-5.4499e-04), 5.6921e-13] (0,0)

Blob [2]( 6, IS Shower        ,  from Beam 1, 1 -> 1 @ (0,0,0,0)
Incoming particles :
[G] 2 e-              2 (  7 ->  6)  [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), 1.4552e-11] (0,0)
Outgoing particles :
[G] 2 e-              2 (  6 ->  0)  [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), 1.4552e-11] (0,0)

Blob [2]( 5, IS Shower        ,  from Beam 0, 1 -> 1 @ (0,0,0,0)
Incoming particles :
[G] 2 photon          1 (  8 ->  5)  [( 1.4762e+02, 0.0000e+00, 0.0000e+00, 1.4762e+02), 0.0000e+00] (0,0)
Outgoing particles :
[G] 2 photon          1 (  5 ->  0)  [( 1.4762e+02, 0.0000e+00, 0.0000e+00, 1.4762e+02), 0.0000e+00] (0,0)

Blob [0]( 0, Signal Process    , 2 -> 3 @ (0,0,0,0)
Incoming particles :
[G] 2 photon          1 (  5 ->  0)  [( 1.4762e+02, 0.0000e+00, 0.0000e+00, 1.4762e+02), 0.0000e+00] (0,0)
[G] 2 e-              2 (  6 ->  0)  [( 2.5000e+02, 0.0000e+00, 0.0000e+00,-2.5000e+02), 1.4552e-11] (0,0)
Outgoing particles :
[H] 2 e-              2 (  0 ->  1)  [( 1.5910e+02, 9.2856e+01, 1.1283e+02, 6.2947e+01), 1.4552e-11] (0,0)
[H] 2 d               3 (  0 ->  1)  [( 1.5629e+01,-1.1111e+01, 1.0942e+01, 1.0488e+00),-9.9476e-13] (601,0)
[H] 2 anti-d          4 (  0 ->  1)  [( 2.2289e+02,-8.1746e+01,-1.2377e+02,-1.6637e+02), 1.4552e-11] (0,601)

Blob [0]( 1, ME PS Interface   , 3 -> 3 @ (0,0,0,0)
Incoming particles :
[H] 2 e-              2 (  0 ->  1)  [( 1.5910e+02, 9.2856e+01, 1.1283e+02, 6.2947e+01), 1.4552e-11] (0,0)
[H] 2 d               3 (  0 ->  1)  [( 1.5629e+01,-1.1111e+01, 1.0942e+01, 1.0488e+00),-9.9476e-13] (601,0)
[H] 2 anti-d          4 (  0 ->  1)  [( 2.2289e+02,-8.1746e+01,-1.2377e+02,-1.6637e+02), 1.4552e-11] (0,601)
Outgoing particles :
[H] 1 e-              2 (  1 ->  2)  [( 1.5910e+02, 9.2856e+01, 1.1283e+02, 6.2947e+01), 2.6112e-07] (0,0)
[H] 1 d               3 (  1 ->  3)  [( 5.5278e+00,-3.9296e+00, 3.8700e+00, 3.7093e-01), 1.0000e-04] (601,0)
[H] 2 anti-d          7 (  1 ->  4)  [( 2.3299e+02,-8.8927e+01,-1.1670e+02,-1.6569e+02), 5.3051e+03] (0,601)

Blob [1]( 2, FS Shower         , 1 -> 1 @ (4.256e-17,0.0001079,0.0001311,7.312e-05)
Incoming particles :
[H] 1 e-              2 (  1 ->  2)  [( 1.5910e+02, 9.2856e+01, 1.1283e+02, 6.2947e+01), 2.6112e-07] (0,0)
Outgoing particles :
[H] 1 e-              3 (  2 ->   )  [( 1.5910e+02, 9.2856e+01, 1.1283e+02, 6.2947e+01), 2.6112e-07] (0,0)

Blob [1]( 3, FS Shower         , 1 -> 1 @ (2.328e-20,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
```

```
[H] 1 d                  3 (  1 ->  3)  [( 5.5278e+00,-3.9296e+00, 3.8700e+00, 3.7093e-01), 1.0000e-04] (601,0)
Outgoing particles :
[H] 2 d                  5 (  3 -> 10)  [( 5.5278e+00,-3.9296e+00, 3.8700e+00, 3.7093e-01), 1.0000e-04] (601,0)

Blob [1]( 4, FS Shower         , 1 -> 6 @ (2.284e-26,-2.326e-15,-3.052e-15,-4.334e-15)
Incoming particles :
[H] 2 anti-d             7 (  1 ->  4)  [( 2.3299e+02,-8.8927e+01,-1.1670e+02,-1.6569e+02), 5.3051e+03] (0,601)
Outgoing particles :
[F] 2 anti-d             9 (  4 -> 10)  [( 7.3297e+01,-2.9305e+01,-4.3661e+01,-5.1062e+01), 1.0000e-04] (0,604)
[F] 2 gluon             10 (  4 -> 10)  [( 2.1309e+01,-9.4603e+00,-1.3720e+01,-1.3280e+01),-1.1369e-13] (604,603)
[F] 2 gluon             11 (  4 -> 10)  [( 4.6008e+01,-2.1263e+01,-2.3130e+01,-3.3609e+01),-4.5475e-13] (603,605)
[F] 2 gluon             12 (  4 -> 10)  [( 6.9869e+01,-3.7137e+01,-3.5404e+01,-4.7424e+01),-1.8190e-12] (605,602)
[F] 2 gluon             13 (  4 -> 10)  [( 1.3558e+01, 2.7436e+00,-2.0921e-01,-1.3276e+01),-2.8422e-14] (606,601)
[F] 2 gluon             14 (  4 -> 10)  [( 8.9509e+00, 5.4954e+00,-5.7053e-01,-7.0424e+00), 0.0000e+00] (602,606)

Blob [0]( 10, Fragmentation     , 7 -> 37 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[H] 2 d                  5 (  3 -> 10)  [( 5.5278e+00,-3.9296e+00, 3.8700e+00, 3.7093e-01), 1.0000e-04] (601,0)
[F] 2 gluon             13 (  4 -> 10)  [( 1.3558e+01, 2.7436e+00,-2.0921e-01,-1.3276e+01),-2.8422e-14] (606,601)
[F] 2 gluon             14 (  4 -> 10)  [( 8.9509e+00, 5.4954e+00,-5.7053e-01,-7.0424e+00), 0.0000e+00] (602,606)
[F] 2 gluon             12 (  4 -> 10)  [( 6.9869e+01,-3.7137e+01,-3.5404e+01,-4.7424e+01),-1.8190e-12] (605,602)
[F] 2 gluon             11 (  4 -> 10)  [( 4.6008e+01,-2.1263e+01,-2.3130e+01,-3.3609e+01),-4.5475e-13] (603,605)
[F] 2 gluon             10 (  4 -> 10)  [( 2.1309e+01,-9.4603e+00,-1.3720e+01,-1.3280e+01),-1.1369e-13] (604,603)
[F] 2 anti-d             9 (  4 -> 10)  [( 7.3297e+01,-2.9305e+01,-4.3661e+01,-5.1062e+01), 1.0000e-04] (0,604)
Outgoing particles :
[p] 1 rho(770)-         20 ( 10 -> 11)  [( 1.2471e+00,-6.4020e-01, 6.8008e-01,-1.9853e-01), 6.4341e-01] (0,0)
[p] 1 omega(782)        25 ( 10 -> 13)  [( 9.9838e-01,-2.8472e-01, 3.6596e-01,-4.1994e-01), 6.0542e-01] (0,0)
[p] 1 eta               31 ( 10 -> 15)  [( 9.2245e-01,-5.0842e-01, 5.3036e-01,-1.0690e-01), 2.9970e-01] (0,0)
[p] 1 K*(892)+          34 ( 10 -> 16)  [( 1.6360e+00,-9.3520e-01, 9.6687e-01,-1.5124e-01), 8.4426e-01] (0,0)
[p] 1 Sigma(1385)-      38 ( 10 -> 18)  [( 1.8401e+00,-3.0233e-01, 5.0023e-01,-1.0289e+00), 1.9859e+00] (0,0)
[P] 1 anti-n            45 ( 10 ->   )  [( 1.2169e+00,-2.2515e-01, 3.5057e-01,-6.5148e-01), 8.8279e-01] (0,0)
[P] 1 P+                46 ( 10 ->   )  [( 2.6677e+00, 4.0937e-01, 6.1054e-02,-2.4627e+00), 8.8035e-01] (0,0)
[P] 1 pi-               47 ( 10 ->   )  [( 2.0586e-01,-5.6533e-03, 3.3892e-02,-1.4738e-01), 1.9480e-02] (0,0)
[p] 1 anti-Delta(1232) 48 ( 10 -> 21)  [( 2.6303e+00, 9.3940e-01, 4.0424e-03,-2.1614e+00), 1.3643e+00] (0,0)
[P] 1 K+                51 ( 10 ->   )  [( 5.9648e-01, 1.0040e-01, 1.1523e-01,-2.9798e-01), 2.4364e-01] (0,0)
[p] 1 anti-K*(892)      52 ( 10 -> 22)  [( 5.8918e+00, 1.1157e+00,-4.0800e-02,-5.7054e+00), 9.1553e-01] (0,0)
[P] 1 pi-               55 ( 10 ->   )  [( 6.3338e-01, 2.9289e-01,-2.9108e-02,-5.4319e-01), 1.9480e-02] (0,0)
[p] 1 rho(770)+         56 ( 10 -> 23)  [( 1.9843e+00, 1.2018e+00,-2.6858e-01,-1.4084e+00), 4.3747e-01] (0,0)
[p] 1 rho(770)-         61 ( 10 -> 25)  [( 2.0000e+00, 3.4653e-01,-1.9033e-01,-1.8003e+00), 6.0281e-01] (0,0)
[P] 1 K+                66 ( 10 ->   )  [( 9.2083e-01,-2.2249e-01,-3.6337e-01,-6.5019e-01), 2.4364e-01] (0,0)
[p] 1 K*(892)-          67 ( 10 -> 27)  [( 1.5022e+00, 3.5843e-01,-3.1379e-01,-1.1287e+00), 7.5562e-01] (0,0)
[p] 1 K*(892)+          71 ( 10 -> 29)  [( 1.6741e+00,-3.6800e-01,-6.4651e-01,-1.1855e+00), 8.4383e-01] (0,0)
[p] 1 K*(892)-          77 ( 10 -> 32)  [( 1.3271e+00, 1.7693e-01,-3.3124e-01,-9.8395e-01), 6.5190e-01] (0,0)
[p] 1 rho(770)          81 ( 10 -> 34)  [( 1.6882e+00, 6.9078e-01,-2.4130e-01,-1.2957e+00), 6.3602e-01] (0,0)
[P] 1 K+                84 ( 10 ->   )  [( 2.4658e+00,-1.2341e+00,-1.2199e+00,-1.6809e+00), 2.4364e-01] (0,0)
[P] 1 K-                85 ( 10 ->   )  [( 2.2395e+01,-1.1895e+01,-1.1345e+01,-1.5202e+01), 2.4364e-01] (0,0)
[p] 1 eta               86 ( 10 -> 35)  [( 4.7620e+00,-2.4833e+00,-2.3945e+00,-3.2368e+00), 2.9970e-01] (0,0)
[P] 1 pi+               96 ( 10 ->   )  [( 6.5012e+00,-3.1226e+00,-3.2768e+00,-4.6645e+00), 1.9480e-02] (0,0)
[p] 1 rho(770)-         97 ( 10 -> 39)  [( 2.4726e+00,-1.0616e+00,-1.2099e+00,-1.8105e+00), 2.4485e-01] (0,0)
[P] 1 pi+              102 ( 10 ->   )  [( 2.7897e+00,-1.2837e+00,-1.4002e+00,-2.0382e+00), 1.9480e-02] (0,0)
[p] 1 rho(770)-        103 ( 10 -> 41)  [( 1.1895e+01,-5.4520e+00,-5.9614e+00,-8.6917e+00), 6.7624e-01] (0,0)
[p] 1 omega(782)       108 ( 10 -> 43)  [( 1.0234e+01,-5.2401e+00,-5.3440e+00,-6.9366e+00), 6.0872e-01] (0,0)
[p] 1 omega(782)       113 ( 10 -> 45)  [( 1.7194e+01,-9.0216e+00,-8.8963e+00,-1.1596e+01), 6.2817e-01] (0,0)
[p] 1 Delta(1232)+     119 ( 10 -> 47)  [( 1.4275e+01,-7.0942e+00,-8.0055e+00,-9.3760e+00), 1.4390e+00] (0,0)
[p] 1 pi               124 ( 10 -> 49)  [( 5.3411e+00,-2.6490e+00,-2.6627e+00,-3.7950e+00), 1.8220e-02] (0,0)
[P] 1 P-               127 ( 10 ->   )  [( 1.2549e+01,-5.7152e+00,-6.7410e+00,-8.8597e+00), 8.8035e-01] (0,0)
[p] 1 K*(892)+         128 ( 10 -> 50)  [( 1.5634e+01,-6.6310e+00,-8.7454e+00,-1.1098e+01), 7.9532e-01] (0,0)
[p] 1 K*(892)-         132 ( 10 -> 52)  [( 1.7685e+01,-7.3252e+00,-1.0155e+01,-1.2458e+01), 7.5939e-01] (0,0)
[p] 1 rho(770)+        137 ( 10 -> 54)  [( 1.3761e+01,-5.7607e+00,-8.4000e+00,-9.2239e+00), 5.4085e-01] (0,0)
[P] 1 pi-              142 ( 10 ->   )  [( 6.5900e+00,-2.6489e+00,-3.9410e+00,-4.5673e+00), 1.9480e-02] (0,0)
[P] 1 K+               143 ( 10 ->   )  [( 1.6158e+01,-6.5373e+00,-9.7092e+00,-1.1128e+01), 2.4364e-01] (0,0)
[p] 1 anti-K*(892)     144 ( 10 -> 56)  [( 2.4235e+01,-9.8402e+00,-1.4601e+01,-1.6632e+01), 6.8819e-01] (0,0)
```

```
Blob [1]( 11, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 rho(770)-        20 ( 10 -> 11)  [( 1.2471e+00,-6.4020e-01, 6.8008e-01,-1.9853e-01), 6.4341e-01] (0,0)
Outgoing particles :
[D] 1 pi-              21 ( 11 ->   )  [( 3.1329e-01, 1.3772e-02, 1.7689e-01, 2.1723e-01), 1.9480e-02] (0,0)
[d] 1 pi               22 ( 11 -> 12)  [( 9.3379e-01,-6.5398e-01, 5.0319e-01,-4.1576e-01), 1.8220e-02] (0,0)

Blob [1]( 12, Hadron Decay      , 1 -> 2 @ (0.0004512,-0.000316,0.0002431,-0.0002009)
Incoming particles :
[d] 1 pi               22 ( 11 -> 12)  [( 9.3379e-01,-6.5398e-01, 5.0319e-01,-4.1576e-01), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon           23 ( 12 ->   )  [( 7.3884e-01,-5.1113e-01, 4.4275e-01,-2.9767e-01), 6.6613e-16] (0,0)
[D] 1 photon           24 ( 12 ->   )  [( 1.9495e-01,-1.4285e-01, 6.0438e-02,-1.1809e-01), 7.6328e-17] (0,0)

Blob [1]( 13, Hadron Decay      , 1 -> 3 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 omega(782)       25 ( 10 -> 13)  [( 9.9838e-01,-2.8472e-01, 3.6596e-01,-4.1994e-01), 6.0542e-01] (0,0)
Outgoing particles :
[D] 1 pi-              26 ( 13 ->   )  [( 1.5365e-01, 4.9093e-02, 2.3910e-02,-3.3853e-02), 1.9480e-02] (0,0)
[D] 1 pi+              27 ( 13 ->   )  [( 3.7548e-01, 3.2588e-02, 3.2316e-01,-1.2652e-01), 1.9480e-02] (0,0)
[d] 1 pi               28 ( 13 -> 14)  [( 4.6925e-01,-3.6640e-01, 1.8889e-02,-2.5957e-01), 1.8220e-02] (0,0)

Blob [1]( 14, Hadron Decay      , 1 -> 2 @ (8.436e-05,-6.59e-05,3.422e-06,-4.666e-05)
Incoming particles :
[d] 1 pi               28 ( 13 -> 14)  [( 4.6925e-01,-3.6640e-01, 1.8889e-02,-2.5957e-01), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon           29 ( 14 ->   )  [( 3.2314e-01,-2.7506e-01, 6.6581e-02,-1.5597e-01), 1.2490e-16] (0,0)
[D] 1 photon           30 ( 14 ->   )  [( 1.4612e-01,-9.1335e-02,-4.7693e-02,-1.0360e-01), 2.0817e-17] (0,0)

Blob [1]( 15, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 eta              31 ( 10 -> 15)  [( 9.2245e-01,-5.0842e-01, 5.3036e-01,-1.0690e-01), 2.9970e-01] (0,0)
Outgoing particles :
[D] 1 photon           32 ( 15 ->   )  [( 2.2676e-01,-2.0934e-01,-8.6023e-02, 1.3944e-02),-5.5511e-17] (0,0)
[D] 1 photon           33 ( 15 ->   )  [( 6.9569e-01,-2.9908e-01, 6.1639e-01,-1.2084e-01),-2.7756e-16] (0,0)

Blob [1]( 16, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 K*(892)+         34 ( 10 -> 16)  [( 1.6360e+00,-9.3520e-01, 9.6687e-01,-1.5124e-01), 8.4426e-01] (0,0)
Outgoing particles :
[d] 1 K                35 ( 16 -> 17)  [( 1.0707e+00,-7.7307e-01, 4.6344e-01,-2.9388e-01), 2.4768e-01] (0,0)
[D] 1 pi+              37 ( 16 ->   )  [( 5.6530e-01,-1.6213e-01, 5.0344e-01, 1.4264e-01), 1.9480e-02] (0,0)

Blob [1]( 17, Hadron Decay      , 1 -> 1 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[d] 1 K                35 ( 16 -> 17)  [( 1.0707e+00,-7.7307e-01, 4.6344e-01,-2.9388e-01), 2.4768e-01] (0,0)
Outgoing particles :
[D] 1 K(L)             36 ( 17 ->   )  [( 1.0707e+00,-7.7307e-01, 4.6344e-01,-2.9388e-01), 2.4768e-01] (0,0)

Blob [1]( 18, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 Sigma(1385)-     38 ( 10 -> 18)  [( 1.8401e+00,-3.0233e-01, 5.0023e-01,-1.0289e+00), 1.9859e+00] (0,0)
Outgoing particles :
[d] 1 Lambda           39 ( 18 -> 19)  [( 1.4374e+00,-5.4837e-02, 5.1489e-01,-7.4377e-01), 1.2447e+00] (0,0)
[D] 1 pi-              44 ( 18 ->   )  [( 4.0277e-01,-2.4749e-01,-1.4655e-02,-2.8508e-01), 1.9480e-02] (0,0)

Blob [1]( 19, Hadron Decay      , 1 -> 2 @ (97.2,-3.708,34.82,-50.3)
Incoming particles :
[d] 1 Lambda           39 ( 18 -> 19)  [( 1.4374e+00,-5.4837e-02, 5.1489e-01,-7.4377e-01), 1.2447e+00] (0,0)
Outgoing particles :
[D] 1 n                40 ( 19 ->   )  [( 1.2864e+00,-1.1049e-01, 4.8005e-01,-7.2759e-01), 8.8279e-01] (0,0)
```

```
[d] 1 pi               41 ( 19 -> 20)  [( 1.5097e-01, 5.5655e-02, 3.4837e-02,-1.6179e-02), 1.8220e-02] (0,0)

Blob [1]( 20, Hadron Decay      , 1 -> 2 @ (97.2,-3.708,34.82,-50.3)
Incoming particles :
[d] 1 pi               41 ( 19 -> 20)  [( 1.5097e-01, 5.5655e-02, 3.4837e-02,-1.6179e-02), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon           42 ( 20 ->   )  [( 5.4302e-02,-3.6883e-02, 3.8133e-02, 1.1584e-02), 2.1684e-18] (0,0)
[D] 1 photon           43 ( 20 ->   )  [( 9.6670e-02, 9.2539e-02,-3.2962e-03,-2.7764e-02), 3.4694e-18] (0,0)

Blob [1]( 21, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 anti-Delta(1232) 48 ( 10 -> 21)  [( 2.6303e+00, 9.3940e-01, 4.0424e-03,-2.1614e+00), 1.3643e+00] (0,0)
Outgoing particles :
[D] 1 P-               49 ( 21 ->   )  [( 2.2895e+00, 9.3324e-01,-9.2661e-02,-1.8660e+00), 8.8035e-01] (0,0)
[D] 1 pi+              50 ( 21 ->   )  [( 3.4083e-01, 6.1571e-03, 9.6703e-02,-2.9546e-01), 1.9480e-02] (0,0)

Blob [1]( 22, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 anti-K*(892)     52 ( 10 -> 22)  [( 5.8918e+00, 1.1157e+00,-4.0800e-02,-5.7054e+00), 9.1553e-01] (0,0)
Outgoing particles :
[D] 1 K-               53 ( 22 ->   )  [( 3.7377e+00, 8.3931e-01,-3.3074e-01,-3.5935e+00), 2.4364e-01] (0,0)
[D] 1 pi+              54 ( 22 ->   )  [( 2.1541e+00, 2.7639e-01, 2.8994e-01,-2.1120e+00), 1.9480e-02] (0,0)

Blob [1]( 23, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 rho(770)+        56 ( 10 -> 23)  [( 1.9843e+00, 1.2018e+00,-2.6858e-01,-1.4084e+00), 4.3747e-01] (0,0)
Outgoing particles :
[D] 1 pi+              57 ( 23 ->   )  [( 9.3848e-01, 5.9193e-01,-4.1442e-01,-5.8235e-01), 1.9480e-02] (0,0)
[d] 1 pi               58 ( 23 -> 24)  [( 1.0458e+00, 6.0989e-01, 1.4585e-01,-8.2601e-01), 1.8220e-02] (0,0)

Blob [1]( 24, Hadron Decay      , 1 -> 2 @ (0.0001026,5.981e-05,1.434e-05,-8.104e-05)
Incoming particles :
[d] 1 pi               58 ( 23 -> 24)  [( 1.0458e+00, 6.0989e-01, 1.4585e-01,-8.2601e-01), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon           59 ( 24 ->   )  [( 7.8478e-01, 4.2349e-01, 7.8808e-02,-6.5598e-01), 4.4409e-16] (0,0)
[D] 1 photon           60 ( 24 ->   )  [( 2.6105e-01, 1.8639e-01, 6.7037e-02,-1.7002e-01), 4.1633e-17] (0,0)

Blob [1]( 25, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 rho(770)-        61 ( 10 -> 25)  [( 2.0000e+00, 3.4653e-01,-1.9033e-01,-1.8003e+00), 6.0281e-01] (0,0)
Outgoing particles :
[D] 1 pi-              62 ( 25 ->   )  [( 1.0349e+00, 1.8913e-01, 2.6187e-01,-9.7324e-01), 1.9480e-02] (0,0)
[d] 1 pi               63 ( 25 -> 26)  [( 9.6513e-01, 1.5740e-01,-4.5220e-01,-8.2704e-01), 1.8220e-02] (0,0)

Blob [1]( 26, Hadron Decay      , 1 -> 2 @ (0.0002391,3.897e-05,-0.000112,-0.0002049)
Incoming particles :
[d] 1 pi               63 ( 25 -> 26)  [( 9.6513e-01, 1.5740e-01,-4.5220e-01,-8.2704e-01), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon           64 ( 26 ->   )  [( 5.3089e-01, 1.5294e-01,-2.4229e-01,-4.4693e-01), 5.5511e-17] (0,0)
[D] 1 photon           65 ( 26 ->   )  [( 4.3424e-01, 4.4598e-03,-2.0991e-01,-3.8011e-01), 1.1102e-16] (0,0)

Blob [1]( 27, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 K*(892)-         67 ( 10 -> 27)  [( 1.5022e+00, 3.5843e-01,-3.1379e-01,-1.1287e+00), 7.5562e-01] (0,0)
Outgoing particles :
[d] 1 anti-K           68 ( 27 -> 28)  [( 6.2292e-01, 1.8368e-01,-1.3472e-01,-2.9744e-01), 2.4768e-01] (0,0)
[D] 1 pi-              70 ( 27 ->   )  [( 8.7925e-01, 1.7475e-01,-1.7907e-01,-8.3127e-01), 1.9480e-02] (0,0)

Blob [1]( 28, Hadron Decay      , 1 -> 1 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[d] 1 anti-K           68 ( 27 -> 28)  [( 6.2292e-01, 1.8368e-01,-1.3472e-01,-2.9744e-01), 2.4768e-01] (0,0)
Outgoing particles :
```

```
[D] 1 K(L)             69 ( 28 ->   )  [( 6.2292e-01, 1.8368e-01,-1.3472e-01,-2.9744e-01), 2.4768e-01] (0,0)

Blob [1]( 29, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 K*(892)+         71 ( 10 -> 29)  [( 1.6741e+00,-3.6800e-01,-6.4651e-01,-1.1855e+00), 8.4383e-01] (0,0)
Outgoing particles :
[d] 1 K                72 ( 29 -> 30)  [( 1.4783e+00,-2.5962e-01,-7.2658e-01,-1.1586e+00), 2.4768e-01] (0,0)
[D] 1 pi+              76 ( 29 ->   )  [( 1.9587e-01,-1.0838e-01, 8.0070e-02,-2.6970e-02), 1.9480e-02] (0,0)

Blob [1]( 30, Hadron Decay      , 1 -> 1 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[d] 1 K                72 ( 29 -> 30)  [( 1.4783e+00,-2.5962e-01,-7.2658e-01,-1.1586e+00), 2.4768e-01] (0,0)
Outgoing particles :
[d] 1 K(S)             73 ( 30 -> 31)  [( 1.4783e+00,-2.5962e-01,-7.2658e-01,-1.1586e+00), 2.4768e-01] (0,0)

Blob [1]( 31, Hadron Decay      , 1 -> 2 @ (228.5,-40.12,-112.3,-179.1)
Incoming particles :
[d] 1 K(S)             73 ( 30 -> 31)  [( 1.4783e+00,-2.5962e-01,-7.2658e-01,-1.1586e+00), 2.4768e-01] (0,0)
Outgoing particles :
[D] 1 pi-              74 ( 31 ->   )  [( 1.0101e+00,-2.0999e-02,-5.9385e-01,-8.0486e-01), 1.9480e-02] (0,0)
[D] 1 pi+              75 ( 31 ->   )  [( 4.6814e-01,-2.3862e-01,-1.3274e-01,-3.5371e-01), 1.9480e-02] (0,0)

Blob [1]( 32, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 K*(892)-         77 ( 10 -> 32)  [( 1.3271e+00, 1.7693e-01,-3.3124e-01,-9.8395e-01), 6.5190e-01] (0,0)
Outgoing particles :
[d] 1 anti-K           78 ( 32 -> 33)  [( 8.4955e-01,-1.0656e-01,-2.1956e-01,-6.4382e-01), 2.4768e-01] (0,0)
[D] 1 pi-              80 ( 32 ->   )  [( 4.7750e-01, 2.8349e-01,-1.1168e-01,-3.4013e-01), 1.9480e-02] (0,0)

Blob [1]( 33, Hadron Decay      , 1 -> 1 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[d] 1 anti-K           78 ( 32 -> 33)  [( 8.4955e-01,-1.0656e-01,-2.1956e-01,-6.4382e-01), 2.4768e-01] (0,0)
Outgoing particles :
[D] 1 K(L)             79 ( 33 ->   )  [( 8.4955e-01,-1.0656e-01,-2.1956e-01,-6.4382e-01), 2.4768e-01] (0,0)

Blob [1]( 34, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 rho(770)         81 ( 10 -> 34)  [( 1.6882e+00, 6.9078e-01,-2.4130e-01,-1.2957e+00), 6.3602e-01] (0,0)
Outgoing particles :
[D] 1 pi-              82 ( 34 ->   )  [( 8.6108e-01, 3.6195e-02,-2.5474e-01,-8.0980e-01), 1.9480e-02] (0,0)
[D] 1 pi+              83 ( 34 ->   )  [( 8.2716e-01, 6.5458e-01, 1.3435e-02,-4.8586e-01), 1.9480e-02] (0,0)

Blob [1]( 35, Hadron Decay      , 1 -> 3 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 eta              86 ( 10 -> 35)  [( 4.7620e+00,-2.4833e+00,-2.3945e+00,-3.2368e+00), 2.9970e-01] (0,0)
Outgoing particles :
[d] 1 pi               87 ( 35 -> 36)  [( 1.5249e+00,-7.7741e-01,-8.5896e-01,-9.8235e-01), 1.8220e-02] (0,0)
[d] 1 pi               90 ( 35 -> 37)  [( 2.1033e+00,-1.1072e+00,-9.1571e-01,-1.5301e+00), 1.8220e-02] (0,0)
[d] 1 pi               93 ( 35 -> 38)  [( 1.1338e+00,-5.9875e-01,-6.1987e-01,-7.2429e-01), 1.8220e-02] (0,0)

Blob [1]( 36, Hadron Decay      , 1 -> 2 @ (9.727e-05,-4.961e-05,-5.476e-05,-6.266e-05)
Incoming particles :
[d] 1 pi               87 ( 35 -> 36)  [( 1.5249e+00,-7.7741e-01,-8.5896e-01,-9.8235e-01), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon           88 ( 36 ->   )  [( 7.8089e-01,-4.2035e-01,-3.8431e-01,-5.3422e-01), 0.0000e+00] (0,0)
[D] 1 photon           89 ( 36 ->   )  [( 7.4404e-01,-3.5705e-01,-4.7464e-01,-4.4814e-01),-1.1102e-16] (0,0)

Blob [1]( 37, Hadron Decay      , 1 -> 2 @ (0.0001595,-8.398e-05,-6.941e-05,-0.000116)
Incoming particles :
[d] 1 pi               90 ( 35 -> 37)  [( 2.1033e+00,-1.1072e+00,-9.1571e-01,-1.5301e+00), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon           91 ( 37 ->   )  [( 1.3367e+00,-7.4046e-01,-5.2888e-01,-9.7922e-01),-2.2204e-16] (0,0)
```

```
[D] 1 photon           92 ( 37 ->   )  [( 7.6654e-01,-3.6670e-01,-3.8683e-01,-5.5089e-01),-1.1102e-16] (0,0)

Blob [1]( 38, Hadron Decay      , 1 -> 2 @ (0.000386,-0.0002039,-0.000211,-0.0002466)
Incoming particles :
[d] 1 pi               93 ( 35 -> 38)  [( 1.1338e+00,-5.9875e-01,-6.1987e-01,-7.2429e-01), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon           94 ( 38 ->   )  [( 7.3025e-02,-2.7355e-02,-1.5418e-02,-6.5929e-02), 0.0000e+00] (0,0)
[D] 1 photon           95 ( 38 ->   )  [( 1.0608e+00,-5.7139e-01,-6.0445e-01,-6.5836e-01), 0.0000e+00] (0,0)

Blob [1]( 39, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 rho(770)-        97 ( 10 -> 39)  [( 2.4726e+00,-1.0616e+00,-1.2099e+00,-1.8105e+00), 2.4485e-01] (0,0)
Outgoing particles :
[D] 1 pi-              98 ( 39 ->   )  [( 9.7410e-01,-4.3152e-01,-3.0219e-01,-8.0739e-01), 1.9480e-02] (0,0)
[d] 1 pi               99 ( 39 -> 40)  [( 1.4985e+00,-6.3005e-01,-9.0776e-01,-1.0031e+00), 1.8220e-02] (0,0)

Blob [1]( 40, Hadron Decay      , 1 -> 2 @ (7.704e-06,-3.265e-06,-4.639e-06,-5.153e-06)
Incoming particles :
[d] 1 pi               99 ( 39 -> 40)  [( 1.4985e+00,-6.3005e-01,-9.0776e-01,-1.0031e+00), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon          100 ( 40 ->   )  [( 1.0836e+00,-4.2604e-01,-7.0469e-01,-7.0434e-01),-2.2204e-16] (0,0)
[D] 1 photon          101 ( 40 ->   )  [( 4.1490e-01,-2.0401e-01,-2.0307e-01,-2.9881e-01),-5.5511e-17] (0,0)

Blob [1]( 41, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 rho(770)-       103 ( 10 -> 41)  [( 1.1895e+01,-5.4520e+00,-5.9614e+00,-8.6917e+00), 6.7624e-01] (0,0)
Outgoing particles :
[D] 1 pi-             104 ( 41 ->   )  [( 9.9065e+00,-4.5277e+00,-4.7553e+00,-7.4166e+00), 1.9480e-02] (0,0)
[d] 1 pi              105 ( 41 -> 42)  [( 1.9882e+00,-9.2428e-01,-1.2061e+00,-1.2750e+00), 1.8220e-02] (0,0)

Blob [1]( 42, Hadron Decay      , 1 -> 2 @ (0.0002054,-9.549e-05,-0.0001245,-0.0001317)
Incoming particles :
[d] 1 pi              105 ( 41 -> 42)  [( 1.9882e+00,-9.2428e-01,-1.2061e+00,-1.2750e+00), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon          106 ( 42 ->   )  [( 4.3851e-01,-2.3117e-01,-2.2094e-01,-3.0006e-01),-5.5511e-17] (0,0)
[D] 1 photon          107 ( 42 ->   )  [( 1.5497e+00,-6.9311e-01,-9.8513e-01,-9.7498e-01), 4.4409e-16] (0,0)

Blob [1]( 43, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 omega(782)      108 ( 10 -> 43)  [( 1.0234e+01,-5.2401e+00,-5.3440e+00,-6.9366e+00), 6.0872e-01] (0,0)
Outgoing particles :
[D] 1 photon          109 ( 43 ->   )  [( 4.0345e+00,-2.3630e+00,-1.8944e+00,-2.6654e+00), 7.1054e-15] (0,0)
[d] 1 pi              110 ( 43 -> 44)  [( 6.1999e+00,-2.8771e+00,-3.4496e+00,-4.2712e+00), 1.8220e-02] (0,0)

Blob [1]( 44, Hadron Decay      , 1 -> 2 @ (0.001608,-0.0007463,-0.0008947,-0.001108)
Incoming particles :
[d] 1 pi              110 ( 43 -> 44)  [( 6.1999e+00,-2.8771e+00,-3.4496e+00,-4.2712e+00), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon          111 ( 44 ->   )  [( 2.4598e-01,-9.8639e-02,-1.2526e-01,-1.8732e-01), 7.6328e-17] (0,0)
[D] 1 photon          112 ( 44 ->   )  [( 5.9539e+00,-2.7785e+00,-3.3243e+00,-4.0838e+00), 7.1054e-15] (0,0)

Blob [1]( 45, Hadron Decay      , 1 -> 3 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 omega(782)         113 ( 10 -> 45)  [( 1.7194e+01,-9.0216e+00,-8.8963e+00,-1.1596e+01), 6.2817e-01] (0,0)
Outgoing particles :
[D] 1 pi-             114 ( 45 ->   )  [( 5.9803e+00,-2.9108e+00,-3.1745e+00,-4.1465e+00), 1.9480e-02] (0,0)
[D] 1 pi+             115 ( 45 ->   )  [( 7.8790e+00,-4.2119e+00,-3.9890e+00,-5.3299e+00), 1.9480e-02] (0,0)
[d] 1 pi              116 ( 45 -> 46)  [( 3.3346e+00,-1.8989e+00,-1.7328e+00,-2.1195e+00), 1.8220e-02] (0,0)

Blob [1]( 46, Hadron Decay      , 1 -> 2 @ (0.0002711,-0.0001544,-0.0001409,-0.0001723)
Incoming particles :
[d] 1 pi              116 ( 45 -> 46)  [( 3.3346e+00,-1.8989e+00,-1.7328e+00,-2.1195e+00), 1.8220e-02] (0,0)
```

```
Outgoing particles :
[D] 1 photon          117 ( 46 ->   )  [( 3.1339e+00,-1.7839e+00,-1.6060e+00,-2.0148e+00),-3.5527e-15] (0,0)
[D] 1 photon          118 ( 46 ->   )  [( 2.0070e-01,-1.1504e-01,-1.2682e-01,-1.0470e-01), 0.0000e+00] (0,0)

Blob [1]( 47, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 Delta(1232)+       119 ( 10 -> 47)  [( 1.4275e+01,-7.0942e+00,-8.0055e+00,-9.3760e+00), 1.4390e+00] (0,0)
Outgoing particles :
[D] 1 P+              120 ( 47 ->   )  [( 1.2632e+01,-6.1639e+00,-7.2109e+00,-8.2878e+00), 8.8035e-01] (0,0)
[d] 1 pi              121 ( 47 -> 48)  [( 1.6430e+00,-9.3032e-01,-7.9458e-01,-1.0882e+00), 1.8220e-02] (0,0)

Blob [1]( 48, Hadron Decay      , 1 -> 2 @ (0.0008025,-0.0004544,-0.0003881,-0.0005316)
Incoming particles :
[d] 1 pi              121 ( 47 -> 48)  [( 1.6430e+00,-9.3032e-01,-7.9458e-01,-1.0882e+00), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon          122 ( 48 ->   )  [( 1.2099e+00,-7.3498e-01,-5.7364e-01,-7.7112e-01), 8.8818e-16] (0,0)
[D] 1 photon          123 ( 48 ->   )  [( 4.3306e-01,-1.9534e-01,-2.2093e-01,-3.1713e-01), 2.7756e-17] (0,0)

Blob [1]( 49, Hadron Decay      , 1 -> 2 @ (0.0009509,-0.0004716,-0.000474,-0.0006756)
Incoming particles :
[p] 1 pi              124 ( 10 -> 49)  [( 5.3411e+00,-2.6490e+00,-2.6627e+00,-3.7950e+00), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon          125 ( 49 ->   )  [( 7.5955e-02,-2.8922e-02,-5.0293e-02,-4.9024e-02),-6.0715e-18] (0,0)
[D] 1 photon          126 ( 49 ->   )  [( 5.2651e+00,-2.6201e+00,-2.6124e+00,-3.7459e+00), 1.4211e-14] (0,0)

Blob [1]( 50, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 K*(892)+        128 ( 10 -> 50)  [( 1.5634e+01,-6.6310e+00,-8.7454e+00,-1.1098e+01), 7.9532e-01] (0,0)
Outgoing particles :
[d] 1 K               129 ( 50 -> 51)  [( 1.2392e+01,-5.3835e+00,-6.7334e+00,-8.8874e+00), 2.4768e-01] (0,0)
[D] 1 pi+             131 ( 50 ->   )  [( 3.2419e+00,-1.2475e+00,-2.0120e+00,-2.2104e+00), 1.9480e-02] (0,0)

Blob [1]( 51, Hadron Decay      , 1 -> 1 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[d] 1 K               129 ( 50 -> 51)  [( 1.2392e+01,-5.3835e+00,-6.7334e+00,-8.8874e+00), 2.4768e-01] (0,0)
Outgoing particles :
[D] 1 K(L)            130 ( 51 ->   )  [( 1.2392e+01,-5.3835e+00,-6.7334e+00,-8.8874e+00), 2.4768e-01] (0,0)

Blob [1]( 52, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 K*(892)-        132 ( 10 -> 52)  [( 1.7685e+01,-7.3252e+00,-1.0155e+01,-1.2458e+01), 7.5939e-01] (0,0)
Outgoing particles :
[D] 1 K-              133 ( 52 ->   )  [( 1.4076e+01,-6.0204e+00,-8.1329e+00,-9.7730e+00), 2.4364e-01] (0,0)
[d] 1 pi              134 ( 52 -> 53)  [( 3.6085e+00,-1.3047e+00,-2.0221e+00,-2.6854e+00), 1.8220e-02] (0,0)

Blob [1]( 53, Hadron Decay      , 1 -> 2 @ (0.000483,-0.0001747,-0.0002706,-0.0003595)
Incoming particles :
[d] 1 pi              134 ( 52 -> 53)  [( 3.6085e+00,-1.3047e+00,-2.0221e+00,-2.6854e+00), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon          135 ( 53 ->   )  [( 2.9502e+00,-1.1145e+00,-1.6331e+00,-2.1897e+00),-3.5527e-15] (0,0)
[D] 1 photon          136 ( 53 ->   )  [( 6.5823e-01,-1.9027e-01,-3.8896e-01,-4.9576e-01),-2.2204e-16] (0,0)

Blob [1]( 54, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 rho(770)+       137 ( 10 -> 54)  [( 1.3761e+01,-5.7607e+00,-8.4000e+00,-9.2239e+00), 5.4085e-01] (0,0)
Outgoing particles :
[D] 1 pi+             138 ( 54 ->   )  [( 3.2455e+00,-1.2680e+00,-1.8071e+00,-2.3749e+00), 1.9480e-02] (0,0)
[d] 1 pi              139 ( 54 -> 55)  [( 1.0516e+01,-4.4927e+00,-6.5929e+00,-6.8490e+00), 1.8220e-02] (0,0)

Blob [1]( 55, Hadron Decay      , 1 -> 2 @ (0.001351,-0.0005772,-0.0008469,-0.0008799)
Incoming particles :
[d] 1 pi              139 ( 54 -> 55)  [( 1.0516e+01,-4.4927e+00,-6.5929e+00,-6.8490e+00), 1.8220e-02] (0,0)
```

```
Outgoing particles :
[D] 1 photon          140 ( 55 ->   )  [( 1.3676e+00,-6.2195e-01,-8.5715e-01,-8.6538e-01),-1.5543e-15] (0,0)
[D] 1 photon          141 ( 55 ->   )  [( 9.1479e+00,-3.8707e+00,-5.7357e+00,-5.9836e+00),-1.4211e-14] (0,0)

Blob [1]( 56, Hadron Decay      , 1 -> 2 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[p] 1 anti-K*(892)    144 ( 10 -> 56)  [( 2.4235e+01,-9.8402e+00,-1.4601e+01,-1.6632e+01), 6.8819e-01] (0,0)
Outgoing particles :
[d] 1 anti-K          145 ( 56 -> 57)  [( 2.1319e+01,-8.8081e+00,-1.2793e+01,-1.4595e+01), 2.4768e-01] (0,0)
[d] 1 pi              147 ( 56 -> 58)  [( 2.9158e+00,-1.0320e+00,-1.8082e+00,-2.0370e+00), 1.8220e-02] (0,0)

Blob [1]( 57, Hadron Decay      , 1 -> 1 @ (2.504e-09,-2.652e-08,2.612e-08,2.504e-09)
Incoming particles :
[d] 1 anti-K          145 ( 56 -> 57)  [( 2.1319e+01,-8.8081e+00,-1.2793e+01,-1.4595e+01), 2.4768e-01] (0,0)
Outgoing particles :
[D] 1 K(L)            146 ( 57 ->   )  [( 2.1319e+01,-8.8081e+00,-1.2793e+01,-1.4595e+01), 2.4768e-01] (0,0)

Blob [1]( 58, Hadron Decay      , 1 -> 3 @ (8.56e-05,-3.032e-05,-5.306e-05,-5.98e-05)
Incoming particles :
[d] 1 pi              147 ( 56 -> 58)  [( 2.9158e+00,-1.0320e+00,-1.8082e+00,-2.0370e+00), 1.8220e-02] (0,0)
Outgoing particles :
[D] 1 photon          148 ( 58 ->   )  [( 1.7688e+00,-6.0006e-01,-1.1475e+00,-1.2048e+00), 1.3323e-15] (0,0)
[D] 1 e+              149 ( 58 ->   )  [( 6.9901e-01,-2.6415e-01,-3.9638e-01,-5.1159e-01), 2.6010e-07] (0,0)
[D] 1 e-              150 ( 58 ->   )  [( 4.4805e-01,-1.6783e-01,-2.6423e-01,-3.2058e-01), 2.6010e-07] (0,0)
```

References

[1] T. Gleisberg et al., JHEP 0402 (2004) 056, arXiv:hep-ph/0311263.

[2] F. Krauss, R. Kuhn and G. Soff, JHEP 0202 (2002) 044, arXiv:hep-ph/0109036.

[3] A.F. Zarnecki, Acta Phys. Polon. B 34 (2003) 2741, arXiv:hep-ex/0207021.

[4] A.D. Martin et al., Eur. Phys. J. C 14 (2000) 133, arXiv:hep-ph/9907231.

[5] J. Pumplin et al., JHEP 0207 (2002) 012, arXiv:hep-ph/0201195.

[6] M. Gluck, E. Reya and A. Vogt, Phys. Rev. D 46 (1992) 1973.

[7] See the homepage `http://vircol.fnal.gov/`.

[8] H.E. Haber and G.L. Kane, Phys. Rept. 117 (1985) 75.

[9] N. Arkani-Hamed, S. Dimopoulos and G.R. Dvali, Phys. Lett. B 429 (1998) 263, arXiv:hep-ph/9803315.

[10] A. Djouadi, J. Kalinowski and M. Spira, Comput. Phys. Commun. 108 (1998) 56, arXiv:hep-ph/9704448.

[11] F.E. Paige et al., arXiv:hep-ph/0312045.

[12] P. Skands et al., JHEP 0407 (2004) 036, arXiv:hep-ph/0311123.

[13] R. Kleiss and W.J. Stirling, Nucl. Phys. B 262 (1985) 235.

[14] A. Ballestrero, E. Maina and S. Moretti, Nucl. Phys. B 415 (1994) 265, arXiv:hep-ph/9212246.

[15] T. Gleisberg et al., Eur. Phys. J. C 34 (2004) 173, arXiv:hep-ph/0311273.

[16] K. Hagiwara et al., arXiv:hep-ph/0512260.

[17] R. Kleiss and R. Pittau, Comput. Phys. Commun. 83 (1994) 141, arXiv:hep-ph/9405257.

[18] F.A. Berends, R. Pittau and R. Kleiss, Nucl. Phys. B 424 (1994) 308, arXiv:hep-ph/9404313.

[19] G.P. Lepage, (1980), CLNS-80/447.

[20] R. Kleiss, W.J. Stirling and S.D. Ellis, Comput. Phys. Commun. 40 (1986) 359.

[21] P.D. Draggiotis, A. van Hameren and R. Kleiss, Phys. Lett. B 483 (2000) 124, arXiv:hep-ph/0004047.

[22] R. Kuhn et al., Comput. Phys. Commun. 134 (2001) 223, arXiv:hep-ph/0004270.

[23] F. Krauss, A. Schalicke and G. Soff, (2005), arXiv:hep-ph/0503087.

[24] T. Sjöstrand, High-energy physics event generation with PYTHIA 5.7 and JETSET 7.4 82 (1994) 74.

[25] T. Sjöstrand et al., (2003), arXiv:hep-ph/0308153.

[26] G. Corcella et al., JHEP 0101 (2001) 010, arXiv:hep-ph/0011363.

[27] G. Corcella et al., (2002), arXiv:hep-ph/0210213.

[28] S. Catani et al., JHEP 0111 (2001) 063, arXiv:hep-ph/0109231.

[29] F. Krauss, JHEP 0208 (2002) 015, arXiv:hep-ph/0205283.

[30] T. Sjöstrand and M. van Zijl, Phys. Rev. D 36 (1987) 2019.

[31] A. Schalicke and F. Krauss, JHEP 0507 (2005) 018, arXiv:hep-ph/0503281.

[32] M. Dobbs and J.B. Hansen, Comput. Phys. Commun. 134 (2001) 41.

[33] F. Krauss et al., Phys. Rev. D 70 (2004) 114009, arXiv:hep-ph/0409106.

[34] F. Krauss et al., Phys. Rev. D 72 (2005) 054017, arXiv:hep-ph/0503280.

[35] T. Gleisberg et al., Phys. Rev. D 72 (2005) 034028, arXiv:hep-ph/0504032.

[36] G. Marsaglia, A. Zaman and W. Zang, (1987), Florida State University Report FSU-SCRI-87-50.

[37] T. Gleisberg et al., JHEP 0309 (2003) 001, arXiv:hep-ph/0306182.

[38] J. Winter, F. Krauss and G. Soff, Eur. Phys. J. C 36 (2004) 381, arXiv:hep-ph/0311085.