The Event Generator SHERPA

T. Gleisberg, S. Höche, F. Krauss, A. Schälicke, S. Schumann, J. Winter
Institut für theoretische Physik, TU Dresden, D-01062 Dresden, Germany
E-mail: steffen@theory.phy.tu-dresden.de

Abstract
In this contribution the multi-purpose event generation framework SHERPA is presented and the development status of its physics modules is reviewed. In its present version, SHERPA is capable of simulating lepton-lepton, lepton-photon, photon-photon and fully hadronic collisions, such as proton-proton reactions.

SHERPA [1] is an independent approach for a framework for event generation at high energy collider experiments. The program is entirely written in the object-oriented programming language C++. This is reflected in particular in the structure of the program. In SHERPA, the various tasks related to the generation of events are encapsulated in a number of specific modules. These physics modules are initialized and steered by the SHERPA framework. This structure facilitates a high modularity of the actual event generator and allows for the easy replacement/modification of entire physics models, e.g. the parton shower or the fragmentation model. The current version SHERPA-1.0.6 incorporates the following physics modules:

– ATOOLS-2.0.6: This is the toolbox for all other modules. 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-1.0.6: 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.
– PDF-1.0.6: In this module the handling of initial state radiation (ISR) is located. It provides interfaces to various proton and photon parton density functions, and to the LHAPDFv3 interface. In addition, an analytical electron structure function is supplied.
– MODEL-1.0.6: This module comprises the basic physics parameters (like masses, mixing angles, etc.) of the simulation run. Thus it specifies the corresponding physics model. Currently three different physics models are supported: the Standard Model (SM), its Minimal Supersymmetric extension (MSSM) and the ADD model of large extra dimensions. For the input of MSSM spectra a run-time interface to the program Isasusy 7.67 [2] is provided. The next release of SHERPA will in addition support the SLHA format of spectrum files [3].
– EXTRA_XS-1.0.6: In this module a 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 color connections and the hard scale of the process. The classes for phase space integration, which are common with AMEGIC, are located in a special module called PHASIC.
– AMEGIC++-2.0.6: AMEGIC [4] is SHERPA's own matrix element generator. It works as a generator-generator: 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 events based on them.

- PHASIC++-1.0.6: Here all classes dealing with the Monte Carlo phase space integration are located. As default the adaptive multi-channel method of [5] together with a Vegas optimization [6] for the single channels is used for the evaluation of the initial state and final state integrals.
- APACIC++-2.0.6: APACIC [7] contains classes for the simulation of both the initial and the final state parton shower. The sequence of parton emissions in the shower evolution is organized in terms of the parton's virtual mass as ordering parameter. Coherence effects are accounted for by explicit ordering of the opening angles in subsequent branchings. This treatment is similar to the Pythia [8] parton shower approach. All features for a consistent merging with matrix elements [9] are included.
- AMISIC++-1.0.6: AMISIC contains classes for the simulation of multiple parton interactions according to [10]. SHERPA extends this 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.
- SHERPA-1.0.6: 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 combining the parton showers and matrix elements, which are independent of the specific parton shower are found in this module. In addition, this subpackage provides an interface to the Lund String Fragmentation of Pythia 6.214 including its hadron decay routines.

SHERPA is publicly available from http://www.sherpa-mc.de. It has successfully been tested for various processes of great relevance at future colliders [11]. Present activities of developing SHERPA cover the modeling of the underlying event and an alternative fragmentation model [12].

References

[1] Gleisberg, T. and others, JHEP **02**, 056 (2004).
[2] Paige, F. E. and Protopescu, S. D. and Baer, H. and Tata, X., hep-ph/0312045.
[3] Skands, P. and others, JHEP **07**, 036 (2004).
[4] Krauss, F. and Kuhn, R. and Soff, G., JHEP **02**, 044 (2002).
[5] Kleiss, R. and Pittau, R., Comput. Phys. Commun. **83**, 141 (1994);
Berends, F. A. and Pittau, R. and Kleiss, R., Nucl. Phys. **B424**, 308 (1994).
[6] Lepage, G. P. CLNS-80/447.
[7] Kuhn, R. and Krauss, F. and Ivanyi, B. and Soff, G., Comput. Phys. Commun. **134**, 223 (2001);
Krauss, F. and Schälicke, A. and Soff, G., hep-ph/0503087.
[8] Sjöstrand, T., Comput. Phys. Commun. **82**, 74 (1994);
Sjöstrand, T. and Lönnblad, L. and Mrenna, S. and Skands, P., hep-ph/0308153.
[9] Catani, S. and Krauss, F. and Kuhn, R. and Webber, B. R., JHEP **11**, 063 (2001);
Krauss, F., JHEP **08**, 015 (2002).
[10] Sjöstrand, T. and van Zijl, M., Phys. Rev. **D36**, 2019 (1987).
[11] Gleisberg, T. and others, JHEP **09**, 001 (2003);
Gleisberg, T. and others, Eur. Phys. J. **C34**, 173 (2004);
Krauss, F. and Schälicke, A. and Schumann, S. and Soff, G., Phys. Rev. **D70**, 114009 (2004);
Krauss, F. and Schälicke, A. and Schumann, S. and Soff, G., hep-ph/0503280;
Gleisberg, T. and Krauss, F. and Schälicke, A. and Schumann, S. and Winter, J., hep-ph/0504032.
[12] Winter, J. and Krauss, F. and Soff, G., Eur. Phys. J. **C36**, 381 (2004).