Vol. 37 (2006) *ACTA PHYSICA POLONICA B* No 8

THE GENIE UNIVERSAL, OBJECT-ORIENTED NEUTRINO GENERATOR*

C. Andreopoulos

for the GENIE Collaboration

Particle Physics Department, CCLRC Rutherford Appleton Laboratory,
Harwell-Chilton Campus, Oxfordshire OX11 0QX, UK

(Received June 22, 2006)

A Universal Object-Oriented/C++ Neutrino Monte Carlo Generator (GENIE) is briefly described. The purpose of this large scale software system is to become the "canonical" Monte Carlo for Neutrino Interaction Physics whose validity will extend to all neutrino types and nuclear targets in the energy range from a few MeV to hundreds of TeV. GENIE attempts to unify the Monte Carlo generation approaches used by a host of different, smaller procedural systems in a modern object-oriented software design. It is already a mature software system that currently consists of $\sim 100\,000$ lines of C++ code (~ 350 classes organised in ~ 40 packages). The first official, extensively validated, release of the GENIE Monte Carlo (version 2.0.0) is now publicly available. This production version is primarily intended for the on-going analyses of the MINOS experiment, since it features a complete adaptation of NeuGEN [H. Gallagher, *Nucl. Phys. Proc. Suppl.* **112**, 188 (2002)], its presently used legacy Monte Carlo generator.

PACS numbers: 3.15.+g, 24.10.Lx

1. Introduction

Over the last few years, through out the field of high energy physics (HEP), we are witnessing an enormous effort committed in migrating many popular procedural-based Monte Carlo (MC) software systems into state-of-the-art object-oriented systems of significantly higher complexity. Some of the most distinctive cases are, for example, the widely used GEANT [1], HERWIG [2] and PYTHIA [3] MCs. This certainly reflects a radical change in our approach to scientific computing. Along with the time invariant

---

* Presented at the XX Max Born Symposium "Nuclear Effects in Neutrino Interactions", Wrocław, Poland, December 7–10, 2005.

requirements that all modelled physics is correct and extensively validated with experimental data in a variety of benchmark calculations, many new requirements arise. These requirements relate to the way large HEP software systems are developed and maintained, by wide geographically-spread collaborations over a typical time span of $\sim$ 20 years during which they will undergo many (initially unforeseen) extensions and modifications to accommodate new needs. This puts a stress on software qualities such as re-usability, maintainability, robustness, modularity and extensibility. Software engineering provides many well proved techniques that can improve the quality and lifetime of HEP software.

In neutrino physics, the requirements of a new event generator are more demanding. Whereas, object-oriented MCs in other HEP fields were evolved from a well established legacy system, in neutrino physics no "canonical" MC exists. The challenge for the authors of a next-generation neutrino MC generator is therefore two-fold. Not only they need to produce a state-of-the-art software system, but also to combine all different approaches to neutrino event generation into a single framework, and, through that, evaluate the validity of each approach and shape a "canonical" neutrino MC. The task obviously requires a wide joint collaboration between theorists and experimentalists including the authors of many previous generation neutrino MCs.

GENIE[1] is a ROOT-based [5] universal, object-oriented/C++ neutrino MC generator that encompasses and supersedes a host of successful Fortran neutrino MC generators, such as GENEVE [6], NEUT [7], NeuGEN [8] and NUX [9] that have been used extensively in the design and exploitation of many previous and current neutrino experiments. Whereas the previous generation of neutrino MC generators were built within specific experiments and were tuned to specific energy ranges and nuclear targets, GENIE is intended to describe the neutrino interaction physics for all neutrino types on all nuclear targets in the energy range from few MeV to hundreds of TeV. The project is supported by a wide collaboration of physicists from all major experiments establishing GENIE as a major HEP event generator collaboration.

GENIE can be used both as a stand-alone generator for fast 4-vector level simulations or it can be integrated, as the back-end primary kinematics generator, to the full simulation chain of a neutrino experiment. In addition, it can be used as a tool for event re-weighting, marginalisation and systematic error evaluation or as a tool for neutrino interaction model validation and tuning.

---

[1] GENIE stands for Generates Events for Neutrino Interaction Experiments.

2. Event generation framework and drivers

In GENIE[2] the generated events are stored at the GHEP event record a custom, STDHEP-like, record that is implemented as a ROOT [5] TClonesArray container. Each record entry (a GHepParticle), represents either an initial, intermediate or final state particle or a generator book-keeping action. Typically a GHepParticle holds all information relevant to a generated particle such as name, status code, PDG code, indices of mother and daughter particles, 4-momentum vector, 4-position relative to the interaction vertex, and polarisation angles. Furthermore, the GHEP record holds information with event, rather than particle, scope such as the cross sections for the selected event and the selected event kinematics, the event weight, a series of event flags and an interaction summary. Additionally, the GHEP record features a "spontaneous rearrangement" capability which maintains the compactness of the daughter lists at any given time, and a host of particle querying methods.

The event generation is seen as series of well-defined steps built around the GHEP event record. Each such step (an event generation module) which encapsulates a well defined event generator operation (*e.g.* the generation of event kinematics, the final state primary lepton generation, intra-nuclear rescattering and so on) can "visit" (*Visitor* design pattern [10]) the event record and modify it accordingly. An ordered list of such event generator modules is referred to as an event generation thread. This is the essence of the GENIE event generation abstraction! Although different event generation modules can perform very different tasks, they can be treated uniformly by focusing on their common operational aspect: Visiting and (potentially) modifying the event record. Treating the event generator modules uniformly and standardising an event generator module interface allows us to build a flexible and extensible system where modules can be dynamically plugged in and out of the event generation threads. It is worth noting that this strategy hardly restricts the way event generation can proceed which, if at all possible to be coded, can always be thought of as a series of steps.

Many of the GENIE event generator modules that build up the event generation threads are quite generic and do not contain model-dependent physics implementation themselves but rather contain, as part of their configuration, references to other algorithms. As an example, the module generating kinematics for QEL events does not contain the actual code for the QEL differential cross section but rather its configuration contains a reference to a pre-configured cross section algorithm that will be requested from the GENIE Algorithm Factory (see below). Again, as these algorithm refer-

---

[2] The GENIE internals will only be very briefly outlined here. Further information can be found in `http://www.genie-mc.org`.

ences are part of the external configuration, it gives enormous flexibility to the event generation framework.

As an event generation thread can generate a class of events only, there can be multiple event generation threads running concurrently. Each such thread declares a list of all interactions it can generate for any given initial state and can hand over the cross section algorithm to be used for these interactions. By iterating over all loaded event generation threads and querying for their interaction lists, the drivers assemble the grand list of interactions that can be generated at the current event generation job and build an association between each interaction and a cross section algorithm. This association is used for selecting interactions to generate and thus bootstrapping event generation cycles. When an interaction to be generated has been selected, the Chain of Responsibility pattern [10] is being used to locate, within the loaded list of threads, the one that can generate the selected interaction. The responsibility for event generation is then delegated to the thread which coordinates the successive operation of its event generation modules on the event record. A flexible system based on exceptions thrown by the modules and caught by the thread allows one to revoke the effect of any set of processing modules, once a dead-end of the generation process has been encountered, and retry without aborting the current generation cycle as long as the event is not unphysical.

GENIE provides the drivers needed to put its event generation framework in motion. Two event generation layers are supported to generate events starting from either (a) a given initial state, or (b) a given beam flux and detector geometry. Generating events for a given initial state corresponds to a basic operational layer that, already, includes all the relevant neutrino interaction physics. Generating events for a given flux and detector geometry, although significantly more complex, contains no extra input physics and is built on top of the basic layer. However, the latter defines the interface to beam Monte Carlos and detector geometries (*e.g.* ROOT, GEANT4). Rather than standardising the format for geometry and neutrino flux descriptions, which would be monolithic and inflexible, GENIE standardises the geometry and flux driver interfaces. Concrete implementations of these interfaces are loaded into the GENIE event generation drivers, extending GENIE event generation capabilities and allowing it to seamlessly integrate new geometry descriptions and beam fluxes. These drivers need not be part of the base GENIE distribution and they can be loaded into GENIE as external pluggins. Currently, GENIE contains some concrete driver implementations corresponding to popular detector geometry systems (such as ROOT/GEANT4) and neutrino fluxes (such as the FLUKA 3-D atmospheric neutrino flux [11]).

Central to the overall design is the concept of the abstract "Algorithm", which can be found at the root of the inheritance tree of each GENIE algorithm and which, amongst other things, allows us to build a flexible XML [12]-based configuration system. Many different configurations can exist per algorithm, all uniquely identified by a name and the name of the algorithm they are intended for. At the GENIE initialisation phase, all external XML configuration files are parsed and each configuration set is held at an algorithm configuration pool as a Registry, a type-safe "value" → "type" associative container. At run-time all algorithms can look-up their configuration registry from the configuration pool using the name of the configuration set that was specified for them. Typically, pre-configured instances of GENIE algorithms are accessed through an Algorithm Factory [10]. Each algorithm is typically run through one of the numerous standardised interfaces which, in the algorithm inheritance tree, are to be found in the next level up from the abstract "Algorithm" root. Invoking all concrete GENIE algorithms through a standardised interface guarantees GENIE scalability and ensures the seamless integration of new concrete implementations of those interfaces.

3. Neutrino interaction physics

The neutrino interaction physics included in the default configuration of GENIE 2.0.0 is *identical with the physics included in* NeuGEN 3 which, given the underlying uncertainties, does a reasonably good job in describing the MINOS data. A comprehensive generator tuning to external data [13] has been performed in the context of the MINOS experiment.

In its default configuration, GENIE computes the CC quasi-elastic vN and NC elastic vN scattering cross sections using, respectively, implementations of the Llewellyn–Smith [14] and Hendrick and Li [15] models with

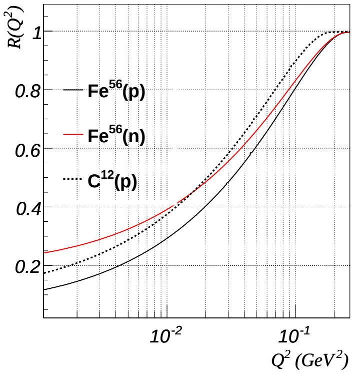

Fig. 1. Nuclear suppression factor from an analytical calculation of the Pauli-blocking effect.

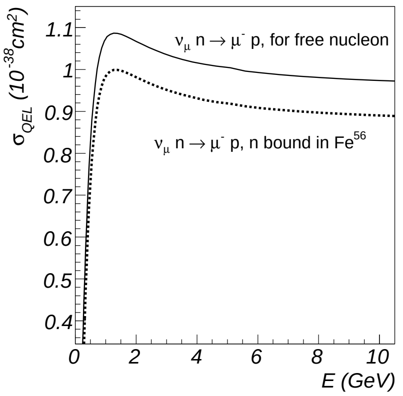

Fig. 2. QEL CC ν_μ cross section for free and bound nucleons.

dipole form factors. For nuclear targets a nuclear suppression factor is taken into account for both interaction modes using an analytical calculation of the Pauli-blocking effect (see Figs. 1 and 2). For computing exclusive NC and CC resonance neutrinoproduction cross sections, GENIE implements the calculation of Rein–Seghal [16] neglecting interference between resonances while the deep inelastic scattering cross section is modelled according to the Bodek–Yang model [17] including parameterisations for nuclear effects (such as shadowing and anti-shadowing, see Fig. 5) and longitudinal structure functions (Whitlow R parameterisation, Fig. 6). The DIS/RES joining scheme couples in the hadronization model: At $W > W_{\rm cut}$ resonance neutrinoproduction is included in the inclusive cross section computed from the Bodek–Yang model so exclusive contributions from the Rein–Seghal model are turned off. For $W < W_{\rm cut}$, exclusive resonance production cross section is computed using the Rein–Seghal model and is added on top of a non-resonance background computed from the Bodek–Yang cross section scaled by an appropriate W-dependent reduction factor. This factor corresponds to the reduction in the integral of the computed hadronic multiplicity distribution when 2-hadron and 3-hadron final states are suppressed by a tunable amount (NeuGEN's Rijk factors) to yield agreement between the computed total cross section and the world's multi-pion cross section data. Inclusive charm production cross section is computed using an implementation of the LO model of Aivazis et al [19]. The NC and CC coherent scattering cross section is described by the Rein–Seghal model [18]. All other processes that are not important in the few GeV energy range (such as diffractive scattering, QEL charm production or νe^- scattering) are currently neglected with the exception of inverse muon decay which is treated according to the Bardin–Dokuchaeva model [20]. The missing processes would be added at near future revisions.

Low invariant mass hadronization is handled by an empirical KNO-based model (see Figs. 3 and 4): Average hadronic multiplicities, $\langle n \rangle$, are computed using the empirical parameterisations of the form $\langle n \rangle = a + b \ln W^2$ [22] and their dispersion is taken into account using the KNO scaling law. Hadron

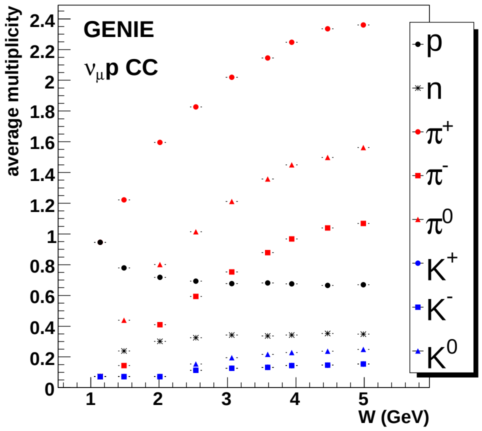

Fig. 3. Generated average hadronic multiplicities as a function of the hadronic invariant mass W, using GENIE's KNO-based hadronization model for no 2-hadron and 3-hadron suppressions taken into account.

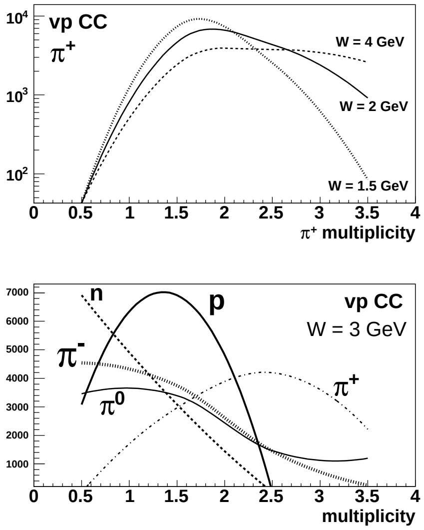

Fig. 4. Top: π^+ multiplicity at $W = 1.5$, 2 and 4 GeV. Bottom: Nucleon and pion multiplicity distributions at $W = 3$ GeV. All plots were generated using GENIE's KNO-based hadronization model for νp CC DIS interactions with no 2-hadron and 3-hadron suppressions taken into account.

IDs are decided based on general isospin and charge conservation arguments and their 4-momenta are obtained by sampling experimentally measured x_F and p_{T} distributions for the generated baryon (target fragment) and then performing a p_{T}-weighted phase space decay [23] for the generated mesons (mainly current fragments). At higher invariant masses GENIE switches to PYTHIA6 [3]. For charm production, a custom hadronization scheme is employed relying on experimental data for charm fractions and empirical fragmentation functions and p_{T} distributions.

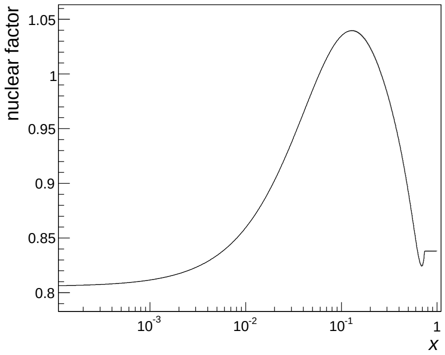

Fig. 5. Parameterisation of shadowing and anti-shadowing effects in DIS.

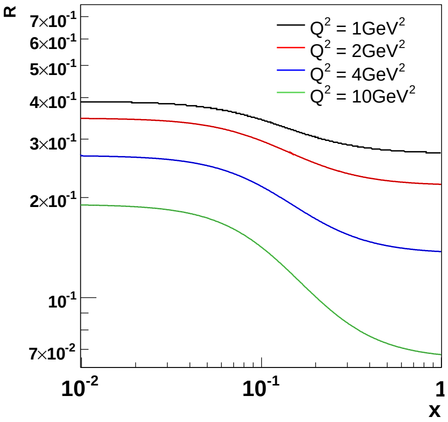

Fig. 6. Whitlow R parameterisation.

For intranuclear rescaterring GENIE is currently using a C++ adaptation of NeuGEN's updated INTRANUKE package [24] including the Ransome model for pion absorption and a re-tuned SKAT parameterisation for formation zone effects (see Fig. 7). In all event generation threads for nuclear targets, the target is treated as a relativistic Fermi gas [21] and off-shell kinematics are used consistently through-out the thread.

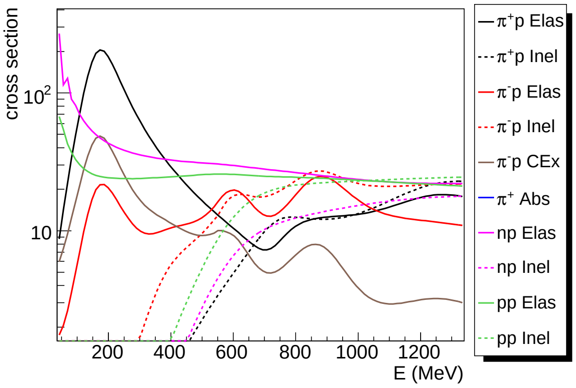

Fig. 7. Hadronic reaction cross sections from `http://gwdac.phys.gwu.edu` (2005 PWA solution) used as input to the INTRANUKE cascade MC.

A multitude of configuration parameters control the models mentioned above including axial and vector masses for QEL, RES and COH scattering, Rein–Seghal's FKR parameters, parameters controlling the average hadronic multiplicities as a function of W, parameters controlling the hadron production probabilities, the KNO distribution data, NeuGEN's Rijk parameters controlling the non-resonance background in $W < W_{\text{cut}}$, the parameter W_{cut}, CKM elements, the Bodek–Yang parameters controlling corrections to the scaling variable x and the GRVLO98 PDFs, the minimum Q^2 for PDF evaluation, the charm mass, which resonances to take into account in exclusive resonance production models, INTRANUKE parameters like the formation zone *etc.* All these parameters have been set to their best values. Using GENIE's external configuration system it is trivial to modify these configuration parameters or to plug in other non-default models into the event generation threads, such as, for example, the BBA form factors [25] instead of the dipole form factors *etc.*

4. Generator availability and license

GENIE is available from its CVS [27] repository hosted at CCLRC, Rutherford Laboratory from where one can access the development version and a series of recent "frozen" releases[3].

The repository is located on AFS [28] space and, in read-only mode, can be accessed anonymously. Read/write-mode access to the code repository is provided to the GENIE collaborators via SSH and key authentication.

---

[3] GENIE versions, following the common software versioning scheme, are named as i, j, k (and tagged as $R - i\_j\_k$) where i, j, k correspond to the major, minor and the revision index.

Instructions on how to obtain the GENIE source code can be found in [4] along with installation instructions and other documentation. GENIE is known to build on many platforms and has no OS proprietary or non-free dependency.

Note that GENIE is distributed under a *license agreement*. The license which is included with the source code, or can be found at the GENIE web page [4], allows anyone to use, copy and distribute the source code and the documentation provided that the copyright notice is included verbatim. Additionally, users may modify the source code and/or the documentation but the license *prohibits* distribution of modified source code, binaries or documentation without the explicit consent of the GENIE authors.

5. Future plans

GENIE version 2.0.0 is an important milestone in the GENIE development. It is the first production release with extensively validated physics models. It features a fully functional and very generic object-oriented event generation framework. In addition GENIE features a very rich toolkit including a tool for neutrino interaction model validation (NuValidator), access to the world's neutrino data, tools for event re-weighting and marginalisation, event generation drivers, geometry and flux drivers. We have now a stable code base as a starting point for further extensions and improvements. At the next stage we would give priority to extending GENIE validity to more nuclear targets and new energy regimes and in improving critical components, especially focusing on the intranuclear re-scattering models.

Future minor releases (the 2.* series) would include better algorithms for multi-dimensional integration, namely implementations of the VEGAS and MISER multi-dimensional adaptive MC integrators that make use of importance and/or stratified sampling. That is expected to improve both the cross section calculation speed and our control of numerical errors.

As far as additional physics inputs are concerned, future *minor* releases would feature new event generation threads for event classes such as diffractive scattering, QEL charm production and elastic νe^- scattering. These event classes are not important in the few GeV energy range that we are presently most concerned with so they have been neglected. Additionally, amongst other additions, we would include important new cross section models for resonance neutrinoproduction [29], migrate from PYTHIA6 [30] to its new C++ version PYTHIA8 [31], include an interface for LHAPDF [32] (and phase-out PDFLIB [33]), take into account the electron atomic orbital velocity in νe^- (currently neglected in inverse muon decay, the only modelled νe^- process), remove the $m/E \to 0$ assumption when computing the final state primary lepton polarisation angles to properly compute them for muons and taus and interface GENIE with TAUOLA [34] to properly take into account tau polarisation effects at its decay.

Over the next few months, and in preparation for a new *major* release (3.0.0), we would attempt to integrate into the GENIE framework models and corrections more relevant to Oxygen and Argon based on the NEUT [7] and GENEVE [6] experience in collaboration with their authors. We are also aiming in substantial improvements at the hadronization and intranuclear re-scattering models. There is currently an ongoing effort for improving the currently available INTRANUKE cascade MC (Dytman, Gallagher *et al.,*) that hopefully would be included in the next major release. In parallel and in high priority we would pursue incorporating the most rigorous available simulations of intranuclear re-scattering physics into our neutrino event simulations, by developing interfaces to DPMJET [36] and FLUKA [35] hadron transport MCs.

Additionally, we would attempt to extend the GENIE validity down to the sub-MeV and up to multi-TeV energy ranges so that this important tool can be also utilised at reactor neutrino experiments and neutrino telescopes, in addition to the LBL accelerator and atmospheric neutrino experiments for which it was originally intended.

6. Summary

We briefly discussed the rationale behind developing a Universal Object-Oriented Neutrino Monte Carlo Generator, and outlined the software design, the physics, the current status and the plans for further development.

A very detailed GENIE 2.0.0 release note describing the GENIE design, drivers and tools, the default physics and the physics configuration parameters is to be published at the hep-ex archives.

7. GENIE collaboration

The GENIE Collaboration consists of C. Andreopoulos (CCLRC, Rutherford), F. Cavanna (INFN, L'Aquila U.), J. Damet (LAPP), S. Dytman (Pittsbourgh U.), H. Gallagher (Tufts), Y. Hayato (ICRR, Tokyo), S. Kretzer (BNL), A. Meregaglia (ETHZ), D. Naples (Pittsbourgh U.), G. Pearce (CCLRC, Rutherford), A. Rubbia (ETHZ), M. Whalley (DURHAM).

REFERENCES

[1] S. Agostineli *et al., Nucl. Instrum. Methods* **A506**, 250 (2003).

[2] M.H. Seymour, `hep-ph/0506202`.

[3] M. Bertini, L. Lonnblad, T. Sjostrand, *Comput. Phys. Commun.* **134**, 365 (2001).

[4] GENIE Collaboration, `http://www.genie-mc.org`.

[5] ROOT Collaboration, `http://root.cern.ch`.

[6] F. Cavanna, GENEVE, *Nucl. Phys. Proc. Suppl.* **112**, 183 (2002).

[7] Y. Hayato, NEUT, *Nucl. Phys. Proc. Suppl.* **112**, 171 (2002).

[8] H. Gallagher, NeuGEN, *Nucl. Phys. Proc. Suppl.* **112**, 188 (2002).

[9] A. Rubbia, NUX, talk presented in NuINT01.

[10] E. Gamma *et al.*, *Design Patterns*, ISBN 0-201-63361-2, ed. Addison-Wesley, 1995.

[11] G. Battistoni *et al.*, *Astropart. Phys.* **19**, 269 (2003).

[12] XML Specification, `http://w3.org/xml/`.

[13] C. Andreopoulos, D. Bhattacharya, H. Gallagher, R. Gran, W.A. Mann, M. Kordosky, J. Morfin, D. Naples, P. Vahle, "Event Generator Uncertainties and the 1E+20 POT ν_μ CC Analysis", MINOS internal document.

[14] C.H. Llewellyn Smith, *Phys. Rep.* **C3**, 261 (1972).

[15] R.E. Hendrick, L. Li, *Phys. Rev.* **D19**, 779,(1979).

[16] D. Rein, L.M. Seghal, *Ann. Phys.* **133**, 79 (1981).

[17] A. Bodek *et al.*, `hep-ph/0411202`.

[18] D. Rein, L.M. Seghal, *Nucl. Phys.* **B223**, 29 (1983).

[19] M.A.G. Aivazis, F.I. Olness, W.K. Tung, *Phys. Rev.* **D50**, 3085 (1994).

[20] D.Yu. Bardin, V.A. Dokuchaeva, *Nucl. Phys.* **B287**, 839 (1987).

[21] A. Bodek, L.J. Ritchie, *Phys. Rev.* **D23**, 1070 (1981).

[22] N. Schmitz, Proceedings of the 10th International Symposium, Lepton and Photon Interactions at High Energies, Bonn 1981, Germany, p. 527.

[23] F. James, "Monte Carlo Phase Space", CERN 68-15 (1968).

[24] R. Merenyi *et al.*, *Phys. Rev.* **D45**, 743 (1992); R.D. Ransome, *Nucl. Phys. Proc. Suppl.* **B139**, 2008 (2005).

[25] R. Bradford, A. Bodek, H. Budd, J. Arrington, `hep-ex/0602017`.

[26] E. Paschos *et al.*, `hep-ph/0107261`.

[27] CVS Collaboration, `http://www.nongnu.org/cvs/`.

[28] AFS Collaboration, `http://www.openafs.org/`.

[29] O. Lalakulich, E.A. Paschos, G. Piranishvili, `hep-ph/0602210`.

[30] PYTHIA-6 and JETSET Collaboration, `http://www.thep.lu.se/dtorbjorn/Pythia.html`.

[31] See "Status and Plans for PYTHIA8", CMS CPT week, CERN, 12 May 2006.

[32] `http://hepforge.cedar.ac.uk/lhapdf/`.

[33] PDFLIB Collaboration, `http://cernlib.web.cern.ch/cernlib/`.

[34] S. Jadach, Z. Was, R. Decker, J.H. Kuhn, *Comput. Phys. Commun.* **76**, 361 (1993).

[35] FLUKA Collaboration, `http://www.fluka.org/`.

[36] J. Ranft, `hep-ph/9911213`.