Hindawi Publishing Corporation
Advances in High Energy Physics
Volume 2015, Article ID 136093, 7 pages
http://dx.doi.org/10.1155/2015/136093

Research Article

HepSim: A Repository with Predictions for High-Energy Physics Experiments

S. V. Chekanov

HEP Division, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, USA

Correspondence should be addressed to S. V. Chekanov; chekanov@anl.gov

Received 8 December 2014; Accepted 3 February 2015

Academic Editor: Marco Battaglia

A file repository for calculations of cross sections and kinematic distributions using Monte Carlo generators for high-energy collisions is discussed. The repository is used to facilitate effective preservation and archiving of data from theoretical calculations and for comparisons with experimental data. The HepSim data library is publicly accessible and includes a number of Monte Carlo event samples with Standard Model predictions for current and future experiments. The HepSim project includes a software package to automate the process of downloading and viewing online Monte Carlo event samples. Data streaming over a network for end-user analysis is discussed.

1. Introduction

Modern theoretical predictions quickly become CPU intensive. A possible solution to facilitate comparisons between theory and data from high-energy physics (HEP) experiments is to develop a public library that stores theoretical predictions in a form that is suited for calculation of arbitrary experimental distribution on commodity computers. The need for such library is driven by the following modern developments.

(i) The Standard Model (SM) predictions should be substantially improved in order to find new physics that can potentially exhibit itself within theoretical uncertainties, which are currently at the level of 5%–10% for quantum chromodynamics (QCD) theory. Currently, such uncertainties are the main limiting factor for precision measurements, as well as searches for new physics beyond the SM. An increase in theoretical precision leads to highly complex, CPU intensive computations. Such calculations are difficult to achieve on commodity computers. In many cases, it is easier to read events with predictions generated after a proper validation, rather than generating them for every measurement or experiment.

(ii) Searches for new physics often include event scans in different kinematic domains. This means that the outputs from theoretical predictions should be sufficiently flexible to accommodate large variations in event selection requirements and to narrow down search results. A theory "frozen" in the form of histograms is often difficult to deal with since histograms need to be computed for each experimental cut.

(iii) The current method to generate predictions for experimental papers lacks transparency. Usually, such calculations are done by experiments using computational resources that are often unavailable for theorists. Theoretical calculations are typically done through a "private" communication between data analyzers and theorists, without public access to the original code or data that are the result of the computations performed for publications. For example, common samples with SM predictions can be useful for a comparison between different experiments that often use different selection cuts.

Let us give an example illustrating the first point. A single calculation of γ + jet cross section at a next-to-leading order (NLO) QCD typically requires several hours on a commodity computer. Sufficient statistical precision for a falling transverse momentum spectrum ($p_T(\gamma)$), typical for HEP, requires several independent calculations with different minimum $p_T(\gamma)$ cuts. Next, the calculations of theoretical uncertainties, such as those with renormalisation scale variations or with different sets of the input parton-density functions (PDFs), require several additional runs. Thus, a single high-quality prediction for a publication may require up to 10000 CPU hours. Finding a method to store Monte Carlo (MC) events with full systematic variations in highly compressed archive files that can be processed by experimentalists and theorists becomes essential. We will come back to this example in the next sections.

A creation of the library with common data from theoretical models for HEP experiments can be an important step to simplify data analysis, to ensure proper validation, accessibility, and preservation over the long term for new uses. The idea of storing MC predictions (including NLO calculations) in a form of "*n*-tuples," that is, an ordered list of records with detailed information on separate (weighted or unweighted) events, is not new; one way or the other, many Monte Carlo (MC) and NLO programs can write data on event-by-event bases into files that can be subsequently read by analysis programs. The missing part of this approach is a common standard layout for such files, a transparent public access, and an easy-to-use software toolkit to process such data for an arbitrary experimental observable. The HepSim project aims to achieve this goal.

A number of community projects exist that simplify theoretical computations and comparisons with experimental data, such as MCDB (a Monte Carlo Database) [1], Professor [2] (a tuning tool for MC event generators), Rivet [3] (a toolkit for validation of MC event generators), and APPLgrid [4] (a method to reproduce the results of full NLO calculations with any input parton distribution set). In the past, JetWeb [5] (a WWW interface and database for Monte Carlo tuning) addressed similar questions of comparing data with theory. Among these tools, the closest repository that focuses on storing data with theoretical predictions is the MCDB Monte Carlo database developed within the CMS Collaboration. This publicly available repository mainly includes the CompHEP MC events [6] in the HepML format [7].

This paper discusses a public repository with Monte Carlo simulations (including NLO calculations) designed for fast calculation of cross sections or any kinematic distribution. This repository was created during the Snowmass Community Studies [8] in 2013 that had one of the goals of archiving MC simulation files for future experiments. In comparison with the MCDB repository, the proposed repository stores files in a highly compressed format that is better suited for archiving, has a simplified data access model with a possibility of data streaming from the web, and includes tools to perform calculations of kinematic distributions.

2. Technical Requirements

A number of software requirements must be met in order to achieve the goal of creating an archive of events from theory predictions for the HEP community.

(i) Data should be stored in compact files suitable for network communication. In particular, the data format should minimize the usage of fixed-length data types and utilize the "varint" approach which use fewer bytes for smaller numbers compared to larger, less common numbers. For example, such data serialization is implemented for integer values in the Google's Protocol Buffers library [9]. For typical HEP events, large numbers (such as energies, masses, and particle identification numbers) are usually less common, and this can be used for very effective compression. It is desirable if MC event samples have file sizes of the order of tens of GBs or less for effective exchange and wide usage.

(ii) An important requirement for the public access is to be able to read the data in a number of programming languages, on any computational platform, with a minimum overhead of installing and configuring the software needed for analysis. Therefore, the data format should be multiplatform from the ground, with the possibility to process such data on Linux, Windows, and Mac computers. Likewise, the files should be self-describing and well suited for structured data, similar to XML. The self-describing feature is needed to store data from different MC generators created by different authors; thus data attributes can be vastly different and should be accessed by name. The documentation of data layout should be the part of the file, without external documentation of position field. The programming language used to read the data should be well suited for concurrency (multithreading).

(iii) Public access via the HTTP protocol is one of the important requirements since this will allow streaming the simulated data to the Web browsers which, in future, can have a functionality of processing and analyzing the data. Although the samples can be located on the grid, our previous experience shows that sharing event samples using the grid access model is less suited for wide community due to security restrictions. A more effective data access, such as the GridFTP protocol, can be added in future.

(iv) When possible, theoretical uncertainties should be encapsulated inside the files. For example, events should include central weights plus all associated systematic variations. Such "all in one" approach will significantly simplify the calculations: a single pass over the data files can be sufficient to create final predictions with all uncertainties.

(v) In addition to the general availability of the data with simulations, the project should provide benchmark cross sections and most representative figures with distributions. All produced plots should be accompanied by analysis programs in order to illustrate the data access.

The above requirements represent a number of software challenges. For example, the usage of the ROOT [10] data-analysis program may be insufficient due to (a) ineffective fixed-length data representation leading to large file sizes. The usage of the variable-byte encoding leads to files that are 30–40% smaller compared to ROOT and other existing fixed-length data formats after compression and (b) a complexity in dealing with the C++ system programming language. From the other hand, the usage of ROOT should be well supported since this is the main analysis environment for HEP experiments.

The choice of the programming language may look obvious at first given that C++ is the preferred choice of HEP experiments. However, this can introduce certain limitations since C++ requires professional programming expertise that is typically available only for system programmers. Scripting languages, such as Python, should be an essential part of the project.

The scope of this project and its implementation substantially depend on the usage of high-performance computers.

3. Current Implementation

The HepSim repository with data samples from leading-order and NLO MC generators is currently available for validations and checks. The database is accessible using the URL link given in [11].

The HepSim project includes the following parts:

(i) a front-end of HepSim with user login;

(ii) a back-end server (or servers) that stores HepSim data;

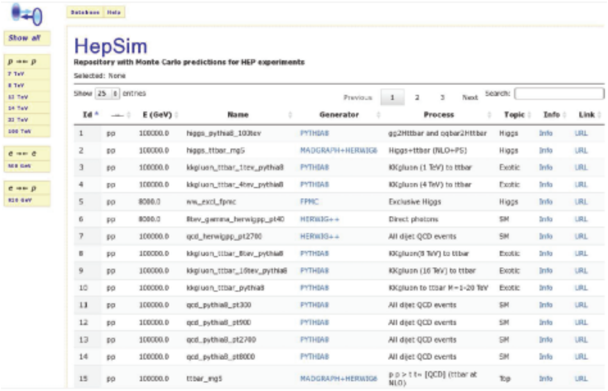
(iii) a software toolkit that allows accessing and processing data.

Below we will discuss these three parts of the HepSim project in more detail.

3.1. HepSim Front-End. The HepSim has a front-end that stores metadata using a SQL database engine. The front-end is written in the PHP and JavaScript languages. The MC files are stored on a separate file storage with a URL access. There are no requirements to store data on the same web server; data can be scattered over multiple URL locations and it is up to the user to document data locations. The SQL front-end can be used to search the database using dataset description, MC generator name, production process, cross section, and other metadata that are included in the description of MC files.

In order to add an entry to the database, a user should be registered. Upon the registration, a dataset should be added by creating a metadata record with dataset name, physics process, the name of the MC generator, file sizes, a text short description, file format, and the URL location of the dataset. Figure 1 shows the HepSim database front-end that lists available samples.

The help menu of HepSim describes how to perform a bulk download of multiple files from the repository and how to read events using minimum requirements for software setup. A more advanced usage is explained on the wiki linked to the HepSim web page.



HepSim

Repository with Monte Carlo predictions for HEP experiments

Selected: None

Show 25 entries

Previous 1 2 3 Next Search:

Id	—	E (GeV)	Name	Generator	Process	Topic	Info	Link
1	pp	100000.0	higgs_pythia8_100tev	PYTHIA8	gg2Httbar and qqbar2Httbar	Higgs	Info	URL
2	pp	100000.0	higgs_ttbar_mg5	MADGRAPH+HERWIG6	Higgs+ttbar (NLO+PS)	Higgs	Info	URL
3	pp	100000.0	kkgluon_ttbar_1tev_pythia8	PYTHIA8	KKgluon (1 TeV) to ttbar	Exotic	Info	URL
4	pp	100000.0	kkgluon_ttbar_4tev_pythia8	PYTHIA8	KKgluon (4 TeV) to ttbar	Exotic	Info	URL
5	pp	8000.0	ww_excl_fpmc	FPMC	Exclusive Higgs	Higgs	Info	URL
6	pp	8000.0	8tev_gamma_herwigpp_pt40	HERWIG++	Direct photons	SM	Info	URL
7	pp	100000.0	qcd_herwigpp_pt2700	HERWIG++	All dijet QCD events	SM	Info	URL
8	pp	100000.0	kkgluon_ttbar_8tev_pythia8	PYTHIA8	KKgluon(8 TeV) to ttbar	Exotic	Info	URL
9	pp	100000.0	kkgluon_ttbar_16tev_pythia8	PYTHIA8	KKgluon (16 TeV) to ttbar	Exotic	Info	URL
10	pp	100000.0	kkgluon_ttbar_pythia8	PYTHIA8	KKgluon to ttbar M=1-20 TeV	Exotic	Info	URL
11	pp	100000.0	qcd_pythia8_pt300	PYTHIA8	All dijet QCD events	SM	Info	URL
12	pp	100000.0	qcd_pythia8_pt900	PYTHIA8	All dijet QCD events	SM	Info	URL
13	pp	100000.0	qcd_pythia8_pt2700	PYTHIA8	All dijet QCD events	SM	Info	URL
14	pp	100000.0	qcd_pythia8_pt8000	PYTHIA8	All dijet QCD events	SM	Info	URL
15	pp	100000.0	ttbar_mg5	MADGRAPH+HERWIG6	p p > t t~ [QCD] (ttbar at NLO)	Top	Info	URL

FIGURE 1: The HepSim front-end to access MC datasets.

3.2. HepSim Back-End. The front-end of HepSim includes URL links to the actual data that are located on a separate data server. The HepSim data servers can be distributed in several locations, since the front-end does not impose any particular requirements on the data servers.

As a basis for the HepSim public library, the ProMC [12, 13] file format has been chosen. This choice is motivated by the possibility to store data with arbitrary layout using variable-byte encoding, including log files from MC generators, and the possibility to stream data over the network. ProMC is implemented as a simple, self-containing library that can easily be deployed on a number of platforms including high-performance computers, such as IBM BlueGene/Q.

The ProMC format is based on a dynamic assignment of the needed number of bytes to store integer values, unlike the traditional approaches that use the fixed-length byte representations. The advantage of this "varint" feature has been discussed in [12, 13]. We will illustrate this using another example relevant to NLO QCD calculations. To store a single event together with theoretical uncertainties created by a NLO program, one needs to write the information on a few particles together with the event weights representing theoretical uncertainties. For the γ + jet example discussed previously, we need to store a few particles from the hard scattering (where one outgoing particle is photon), together with event weights from different sets of PDFs. Although the central weight can be stored as a floating point number without losing the numerical precision, other weights can be encoded as integer values representing deviations from the central weight. This approach can take the advantage of the compact varint "compression." For example, if a central weight, denoted as PDF(0), is estimated with MSTW2008 PDF [14], 40 associated eigenvector sets for PDF uncertainties can be represented as integer numbers:

$$w_n = \left[1000 \times \left(1 - \frac{\mathrm{PDF}\,(n)}{\mathrm{PDF}\,(0)}\right)\right], \quad n = 1, \ldots, 40, \qquad (1)$$

that is, in the units of 0.1% with respect to the central wight PDF(0). The factor 1000 is arbitrary and can be changed depending on the required precision. In many cases, integer values w_n are close to 0, leading to 1-2 bytes in the varint encoding. Therefore, a single event record with all associated eigenvector PDF sets will use less than 100 bytes.

The ProMC files can be read in a number of programming languages supported on the Linux and Windows platforms. The default language to read and process files for validation purposes was chosen to be Java, since it is well suited for web-application programming and is available on all major computational platforms. To process data from HepSim, the ROOT data-analysis program [10] developed at CERN can be used. In addition to the ProMC, the HepSim database can include datasets in other popular formats, such as HepML [7], HEPMC [15], or StdHEP [16].

3.3. Available Datasets. Currently, the HepSim repository contains events generated by PYTHIA [17], MadGraph [18], Jetphox [19], MCFM [20], NLOJet++ [21], FPMC [22], and HERWIG++ [23] generators. The repository includes event samples for pp colliders with the centre-of-mass energies of 8, 13, 14, and 100 TeV. In some cases, together with the detailed information on produced particles, full sets of theoretical uncertainties (scale, PDF, etc.) are embedded inside the files as discussed in Section 3. A number of simulated samples were created using the IBM BlueGene/Q (located at the Argonne Leadership Computing Facility) and the ATLAS Connect virtual cluster service, the descriptions of which are beyond the scope of this paper.

The total size of a typical HepSim dataset is less than 100 GB. The largest simulated sample stored in HepSim and used in physics studies [24] contains 400 million pp collision events at the center-of-mass energy of 100 TeV. Each event contains more than 5000 particles on average, totaling more than 2 trillion generated particles. The total size of this sample is 4.2 TB.

Each particle in a typical HepSim dataset is characterized by four-momentum, position, and several quantum numbers. In many cases, the event records are "slimmed" after removing unstable particles and final-state particles with transverse momentum less than 300–400 MeV. The most essential parton-level information on vector bosons, $b-$ and $t-$ quarks is kept.

When possible, ProMC files with NLO predictions include deviations from the central event weight in the form of integer values as discussed in the previous section. This typically leads to a very compact representation of events from NLO generators using the "varint" encoding since large systematic deviations are less common than small ones.

4. HepSim Software Toolkit

The HepSim toolkit is designed for download, validation, and viewing Monte Carlo event samples. On Linux/Mac, the HepSim software can be downloaded and installed as

```
curl http://atlaswww.hep.anl.gov/asc/
hepsim/hs-toolkit.tgz| tar -xz source
hs-toolkit/setup.sh.
```

To use this package, Java 7 (or 8) should be installed. There are no other requirements to use this package.

Let us consider several commands from the package "hs-toolkit" that can help to download and analyze HepSim Monte Carlo samples.

(i) The command to show all files associated with a given dataset is

```
hs-ls [name]
```

where "[name]" is the dataset name. Alternatively, the dataset name can be replaced with the URL location of the dataset on the web.

(ii) To search for a specific URL by name or dataset description, use this command

```
hs-find [word]
```

where "[word]" is a word that matches your criteria. This command returns a list of sites where the given word is present in the dataset names or description.

(iii) The files can be downloaded in a multiple threads as

```
hs-get [name] [OUTPUT DIR] [Nr of threads]
[Nr of files] [Pattern]
```

where "[OUTPUT DIR]" is the name of the output directory, "[Nr of threads]" is the number of threads for data download, "[Nr of files]" is the maximum number of files for download, and [Pattern] is a (optional) pattern that the regular expression engine attempts to match in the file names. Alternatively, the dataset name "[name]" can be replaced with the URL of the dataset.

(iv) To check a single file from the dataset and to print its metadata, one can use the following command:

```
hs-info [URL].
```

Note that "[URL]" can be either a file URL or an absolute path of the file on a disk. This command is slower in the case of URL. In order to print an event on a Linux/Mac console, use

```
hs-info [URL] [Event number]
```

where the last argument is an event (integer) number.

(v) In order to look at all events using a GUI mode, use this command

```
hs-view [URL]
```

where, again, "[URL]" can be either a URL or the location of a file on the local disk. There is no limitation on file sizes for this command.

The above examples illustrate the fact that data can be streamed over a network, without storing data on the disk. The only limitation for this approach is the computer memory.

4.1. Data Validation. For validations of HepSim simulated samples, Jython, an implementation of the Python programming language in Java, is used. The Jython language has similar semantics to Python, but uses the Java virtual machine (JVM) which ensures platform independence of the analysis environment. The Jython scripts can straightforwardly be rewritten in Groovy, JRuby, and other scripting dynamic languages supported by JVM. In many cases, the validation code examples accompany the data sets and are publicly available for the users. The validation codes show how to read the ProMC files with simulation data and how to reconstruct cross sections when the event weights are required (i.e., for NLO programs). The Jython snippets were written using the SCaVis [25, 26] data-analysis framework for the Java platform, but any Java-based IDE (Eclipse, NetBeans or IntelliJ) should be sufficient to develop the codes as long as the needed jar libraries are included in the Java classpath. All basic analysis packages for HEP physics, such as a four-vector with the Lorentz transformations and different types of jet reconstruction algorithms, such as the popular k_T, anti-k_T and Cambridge/Aachen inclusive jet clustering algorithms ([27, 28] and references therein) for *pp* collisions, are supported by the SCaVis Java libraries. Jet algorithms for e^+e^- collisions are supported via the FreeHEP Java library [29].

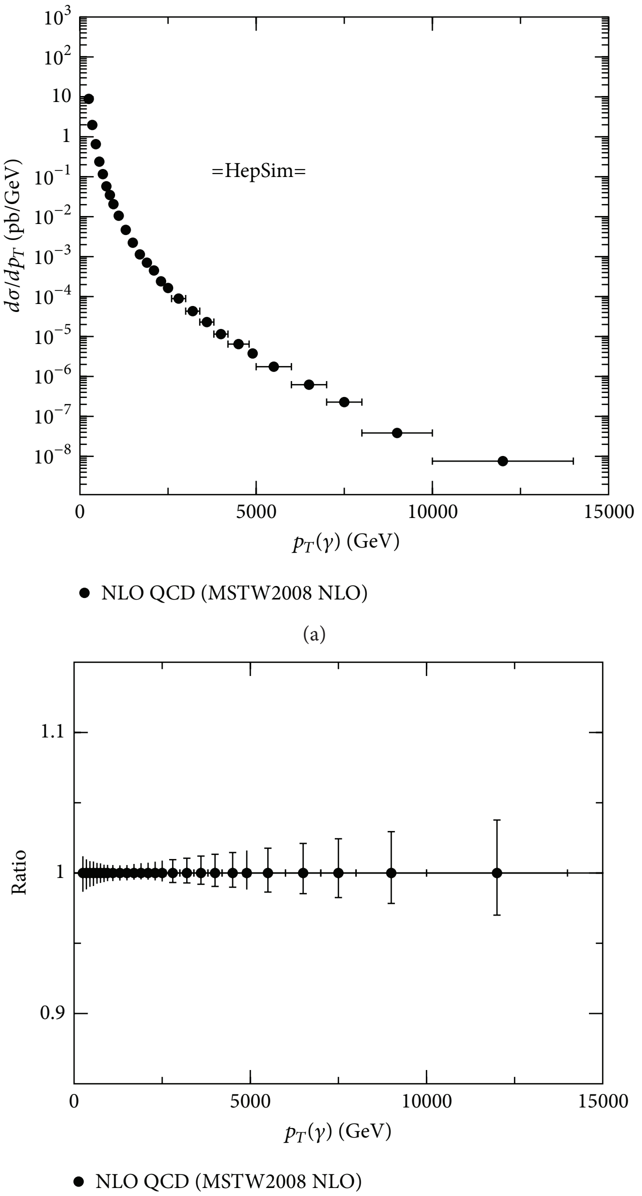
The validation scripts can read data either through the HTTP protocol or using files stored on local file systems. The analysis of files using streaming over a network is typically slower and thus is only recommended for a nonrepeatable analysis.

The processing validation time for a typical simulated sample is less than 30 min on a desktop computer while, in some cases, the CPU time to generate such event samples is more than 8000 CPU hours (512 nodes times 16 cores) on IBM BlueGene/Q of the Argonne Leadership Computing Facility.

As an example, Figure 2 shows the γ + jet differential cross section for a *pp* collider at a center-of-mass energy of 100 TeV created using a validation script. The JETPHOX 1.3 program [19, 30], which implements a full NLO QCD calculation of both the direct and fragmentation contributions to the total cross section, was used to generate the prediction. The bottom plot shows the PDF uncertainty calculated from $N = 41$ PDF weights provided by the MSTW2008 NLO PDF [14]:

$$\frac{\sqrt{\sum_{i=1}^{N}\left(\sigma_i-\sigma_0\right)^2}}{\sigma_0}, \tag{2}$$

where σ_i is the cross section for the *i*th eigenvector of the MSTW2008 NLO set, and σ_0 is the cross section for the central MSTW2008 NLO set. Negative and positive values of the difference $(\sigma_i - \sigma_0)$ are treated separately. The output data sample is about 5 GB and includes 7 calculations generated with different minimum $p_T(\gamma)$ cuts. All PDF weights are included in the file record using the variable-byte encoding discussed before. The processing time is 30 min on a commodity computer using a Jython script which reads 4-momenta of particles and event weights. Thus, any distribution with arbitrary experimental cuts and histogram bin sizes can be repeated within this time.



(a)

(b)

Figure 2: A typical example of the HepSim library output showing the Jetphox NLO [19, 30] calculation for γ + jet cross section for a 100 TeV *pp* collider. The shaded band on the bottom plot shows the relative theoretical uncertainty estimated from 41 PDF sets of the MSTW2008 NLO PDF [14]. This validation plot was created using a Jython code on the Java platform.

4.2. Data Analysis. As discussed before, the HepSim repository is useful for a fast reconstruction of theoretical cross sections and distributions from four-momenta of particles using experiment-specific selection, reconstruction, and histogram bins. The analysis code in C++, Java and Python can be generated from the ProMC files as described in [12, 13]

For a full-scale analysis of HepSim data samples, the ROOT data-analysis program [10] developed at CERN can be used. How to compile data-analysis programs with ROOT is given in a number of examples that come with the ProMC package (inside the directory "examples"). The analysis can also be done using ROOT I/O, after converting data to the ROOT file format. However, this might be redundant since data in the ProMC format can be read by C++ programs directly.

The HepSim files can also be used as inputs for the DELPHES fast detector simulation program [31] which has a built-in reader for ProMC files.

5. Summary

The online HepSim manual [11] contains a description of how to search for simulated samples, how to download them in multiple threads, how to read data using Java, C++/ROOT, and CPython, and how to run a fast detector simulation. Currently, the database includes more than 70 event samples that cover a wide range of physics processes for pp collision energies from 7 TeV to 100 TeV. A number of publications based on the HepSim database are listed on the web page. The current focus of HepSim is to publish simulated events for the High Luminosity LHC and for studies of the physics potential of a future 100 TeV pp collider.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author would like to thank J. Proudfoot and E. May for discussion and validation. The submitted paper has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract no. DE-AC02-06CH11357. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357. A fraction of the simulated event samples presented in this paper were generated using the ATLAS Connect virtual cluster service.

References

[1] S. Belov, L. Dudko, E. Galkin, A. Gusev, W. Pokorski, and A. Sherstnev, “LCG MCDB—a knowledgebase of Monte-Carlo simulated events,” *Computer Physics Communications*, vol. 178, no. 3, pp. 222–229, 2008.

[2] P. Abreu, W. Adam, T. Adye et al., “Tuning and test of fragmentation models based on identified particles and precision event shape data DELPHI,” *Zeitschrift für Physik C: Particles and Fields*, vol. 73, no. 1, pp. 11–59, 1997.

[3] A. Buckley, J. Butterworth, D. Grellscheid et al., “Rivet user manual,” *Computer Physics Communications*, vol. 184, no. 12, pp. 2803–2819, 2013.

[4] T. Carli, D. Clements, A. Cooper-Sarkar et al., “A posteriori inclusion of parton density functions in NLO QCD final-state calculations at hadron colliders: the APPLGRID project,” *European Physical Journal C*, vol. 66, no. 3, pp. 503–524, 2010.

[5] J. M. Butterworth and S. Butterworth, “JetWeb: a WWW interface and database for Monte Carlo tuning and validation,” *Computer Physics Communications*, vol. 153, no. 2, pp. 164–178, 2003.

[6] A. Pukhov, E. Boos, M. Dubinin et al., “CompHEP: a package for evaluation of Feynman diagrams and integration over multiparticle phase space,” User’s Manual, Preprint INP MSU 98-41/542, http://arxiv.org/abs/hep-ph/9908288.

[7] S. Belov, L. Dudko, D. Kekelidze, and A. Sherstnev, “HepML, an XML-based format for describing simulated data in high energy physics,” *Computer Physics Communications*, vol. 181, no. 10, pp. 1758–1768, 2010.

[8] HEP Community Summer Study (Snowmass), 2013, http://www.snowmass2013.org/.

[9] Google, Protocol buffers, google’s data interchange format, 2008, http://code.google.com/apis/protocolbuffers/.

[10] I. Antcheva, M. Ballintijn, B. Bellenot et al., “ROOT: a C++ framework for petabyte data storage, statistical analysis and visualization,” *Computer Physics Communications*, vol. 180, no. 12, pp. 2499–2512, 2009.

[11] S. Chekanov, The HepSim project, 2014, http://atlaswww.hep.anl.gov/hepsim/.

[12] S. V. Chekanov, “Next generation input-output data format for HEP using Google’s protocol buffers,” Tech. Rep. ANL-HEP-CP-13-32, 2013, Snowmass 2013 Electronic Proceedings, eConf SNOW13-00090.

[13] S. V. Chekanov, E. May, K. Strand, and P. van Gemmeren, “ProMC: input-output data format for HEP applications using varint encoding,” *Computer Physics Communications*, vol. 185, no. 10, pp. 2629–2635, 2014.

[14] A. Martin, W. Stirling, R. Thorne, and G. Watt, “Parton distributions for the LHC,” *The European Physical Journal C*, vol. 63, pp. 189–285, 2009.

[15] M. Dobbs, J. Hansen, L. Garren, and L. Sonnenschein, “HEPMC User Manual,” http://lcgapp.cern.ch/project/simu/HepMC/.

[16] L. Garren and P. Lebrun, “StdHep. A common output format for Monte Carlo events,” http://cepa.fnal.gov/psm/stdhep/.

[17] T. Sjöstrand, S. Mrenna, and P. Skands, “PYTHIA 6.4 physics and manual,” *Journal of High Energy Physics*, vol. 5, article 026, 2006.

[18] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, and T. Stelzer, “MadGraph 5: going beyond,” *Journal of High Energy Physics*, vol. 128, 2011.

[19] S. Catani, M. Fontannaz, J.-P. Guillet, and E. Pilon, “Cross section of isolated prompt photons in hadron-hadron collisions,” *Journal of High Energy Physics*, vol. 2002, no. 5, article 028, 2002.

[20] J. M. Campbell and R. Ellis, “MCFM for the tevatron and the LHC,” *Nuclear Physics B—Proceedings Supplements*, vol. 205-206, pp. 10–15, 2010.

[21] Z. Nagy, “Next-to-leading order calculation of three-jet observables in hadron-hadron collisions,” *Physical Review D*, vol. 68, no. 9, Article ID 094002, 2003.

[22] M. Boonekamp, A. Dechambre, V. Juranek et al., “FPMC: a generator for forward physics,” http://arxiv.org/abs/1102.2531.

[23] G. Corcella, I. G. Knowles, G. Marchesini et al., “HERWIG 6: an event generator for hadron emission reactions with interfering gluons (including supersymmetric processes),” *Journal of High Energy Physics*, vol. 2001, article 10, 2001.

[24] B. Auerbach, S. Chekanov, J. Love, J. Proudfoot, and A. Kotwal, “Sensitivity to new high-mass states decaying to $t\bar{t}$ at a 100 TeV collider,” *Physical Review D*, vol. 91, Article ID 034014, 2015.

[25] SCaVis, Scientific Computation and Visualization Environment, 2013, http://jwork.org/scavis/.

[26] S. Chekanov, *Scientific Data Analysis Using Jython Scripting and Java*, Springer, London, UK, 2010.

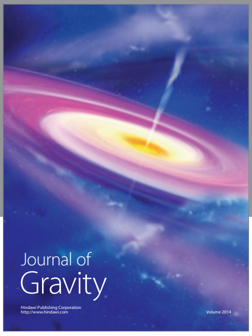
[27] S. D. Ellis and D. E. Soper, "Successive combination jet algorithm for hadron collisions," *Physical Review D*, vol. 48, no. 7, pp. 3160–3166, 1993.

[28] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-k_t jet clustering algorithm," *Journal of High Energy Physics*, vol. 2008, no. 4, article 063, 2008.

[29] FreeHEP Java Libraries, http://java.freehep.org/.

[30] P. Aurenche, J. P. Guillet, E. Pilon, M. Werlen, and M. Fontannaz, "A new critical study of photon production in hadronic collisions," *Physical Review D*, vol. 73, no. 9, Article ID 094007, 10 pages, 2006.

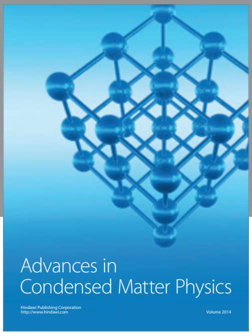
[31] J. de Favereau, C. Delaere, P. Demin et al., "DELPHES 3: a modular framework for fast simulation of a generic collider experiment," *Journal of High Energy Physics*, vol. 2014, article 57, 2014.

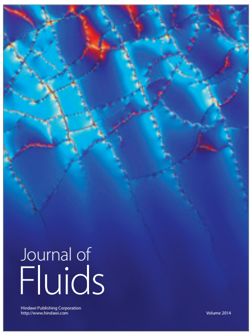

Journal of
Gravity

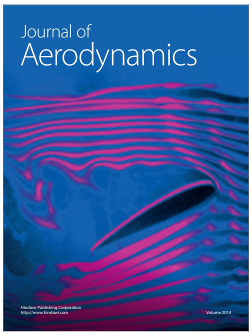
Journal of
Photonics

The Scientific
World Journal

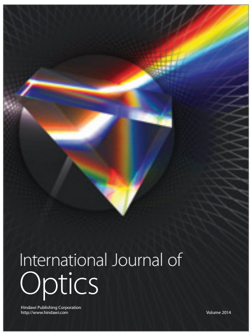
Journal of
Soft Matter

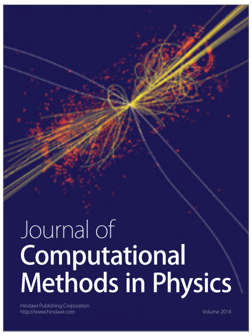

Advances in
Condensed Matter Physics

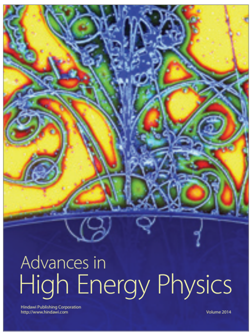

Journal of
Fluids


Journal of
Aerodynamics

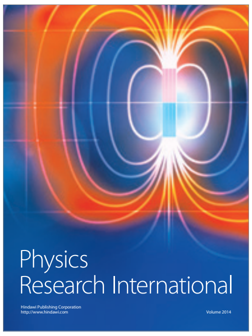
International Journal of
Statistical Mechanics

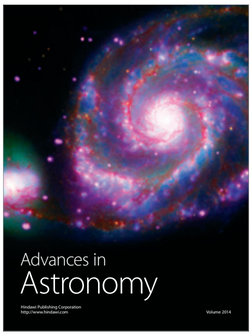

International Journal of
Optics


Journal of
Computational
Methods in Physics


Advances in
High Energy Physics

Journal of
Thermodynamics


Physics
Research International

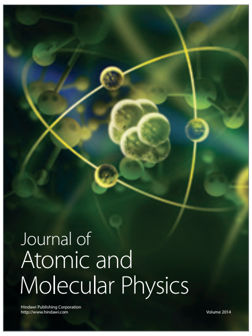

Advances in
Astronomy

Journal of
Solid State Physics

Journal of
Astrophysics

International Journal of
Superconductivity

Journal of
Biophysics


Journal of
Atomic and
Molecular Physics