Journal of Physics: Conference Series **331** (2011) 042010 doi:10.1088/1742-6596/331/4/042010

New data libraries and physics data management tools

M Han[1], M G Pia[2], M Augelli [3], S Hauf [4], C H Kim[1], M Kuster [4] L Moneta [5], L Quintieri [6], P Saracco [2] and H Seo[1]

[1] Department of Nuclear Engineering, Hanyang University, Seoul 133-791, Korea
[2] INFN Sezione di Genova, Genova 16146, Italy
[3] CNES, 31401 Toulouse, France
[4] Technische Universität Darmstadt, IKP, Germany
[5]CERN, CH 1211 Geneva 23, Switzerland
[6] INFN Laboratori Nazionali di Frascati, 00044 Frascati, Italy

E-mail: Maria.Grazia.Pia@cern.ch

Abstract. A number of physics data libraries for Monte Carlo simulation are reviewed. The development of a package for the management of physics data is described: its design, implementation and computational benchmarks. This package improves the data management tools originally developed for Geant4 electromagnetic physics models based on data libraries. The implementation exploits recent evolutions of the C++ libraries appearing in the C++0x draft, which are intended for inclusion in the next C++ ISO Standard. The new tools improve the computational performance of physics data management.

1. Introduction

Data libraries are extensively used in Monte Carlo simulation. They play an important role as a collaborative tool across Monte Carlo codes: they allow sharing physics modeling features in a variety of simulation environments and facilitate comparisons of simulations based on different codes. The paper reviews some recent developments in this domain.

Data libraries that are currently used by Monte Carlo codes have been evaluated with respect to experimental data and two new ones have been recently developed: a data library of proton and alpha ionization cross sections and one of electron ionization cross section.

The paper also reports the first results of the development of a new package for the management of physics data. This software tool extends and improves the data management tools originally developed for Geant4 [1, 2] physics models; it exploits generic programming techniques to achieve configuration flexibility and versatility at dealing with various features of the data, and recent evolutions of the C++ libraries, which are intended for inclusion in the next C++ ISO Standard. The design and implementation features of this package have been extensively benchmarked; test results are reported and discussed.

Due to the length limitations and copyright constraints imposed by the conference proceedings, this paper summarizes the main features and results of the research in progress; extensive details and the full set of results are meant to be included in dedicated publications in scholarly journals.

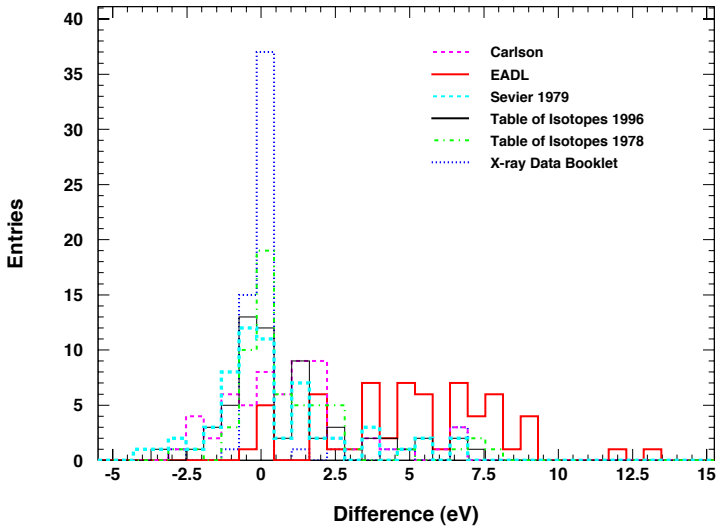

Figure 1. Difference between various compilations of electron binding energies and high precision experimental data of [12]: compilations of Carlson (dashed pink line) [13], Table of Isotopes 1996 (thin solid black line) [14], Table of Isotopes 1978 (dotted-dashed green line) [15], X-ray Data Booklet (dotted blue line) [16], Sevier (dashed turquoise line) [17] and EADL (thick solid red line).

2. Evaluation and development of physics data libraries

The Evaluated Atomic Data Library (EADL) [3] is used by Geant4 low energy electromagnetic package [4, 5] and by other Monte Carlo codes. It collects various atomic data, including electron binding energies and probabilities of radiative and non-radiative transitions in the atomic relaxation process. The current version has been in use for about two decades; nevertheless, despite the fundamental character of this data library, limited documentation is available in the literature regarding its accuracy.

The quantitative estimate of the accuracy of EADL has been the object of recent investigations; these studies have shown that some parts of EADL do not reflect the state-of-the-art in the respective fields. Regarding radiative transition probabilities, Hartree-Fock [6, 7] calculations appear more accurate [8] than the Hartree-Slater [9, 10] ones tabulated in EADL. The inner shell binding energies and ionization energies tabulated in EADL have also been subject to validation with respect to experimental data [11]; a sample of results concerning ionization energies is shown in Figure 1. The binding energies in EADL appear in general less accurate than other compilations of electron binding energies available in the literature. The full set of results deriving from this validation process will be documented in a dedicated paper, whose publication in a scholarly journal is foreseen after the CHEP 2010 conference.

The Evaluated Electron Data Library [18] contains cross sections, secondary particle spectra and other physics data pertinent to electron transport. It covers the energy range from 10 eV to 100 Gev. A detailed validation study of its ionization cross sections has been performed; the main results are summarized in [19, 20, 21]. The accuracy of EEDL ionization cross sections appears satisfactory above 250 eV, while it is inadequate at lower energies.

Recent developments and validation with respect to experimental data [19, 20, 21] have identified two models of electron ionization cross section, the Binary-Encounter-Bethe [22] model and the Deutsch-Märk [23] one, as more accurate than EEDL at lower energies. The compilation of a data library, meant for public distribution, is in progress to make precalculated tabulations

of these cross sections available to the scientific community.

A data library of ionization cross sections by proton and α particle impact has been assembled, based on the extensive development and validation documented in [24]. It includes tabulations based on various theoretical and empirical models. The procedure for its public release by RSICC is in progress at the time of writing this paper.

The current Geant4 radioactive decay simulation uses datasets which are based on the Evaluated Nuclear Structure Data Files (ENSDF) [25] to obtain half lives, decay branches, energy levels and level intensities of the decaying nucleus. An assessment of the accuracy of Geant4 simulations based on these data is in progress [26, 27]; preliminary results have highlighted some discrepancies, which are currently object of investigation.

3. New tools for physics data handling

Various physics data structures and software design features were investigated to evaluate whether they would contribute to improve the computational performance of the data library management in Geant4. While computational performance was the main objective driving the study, the intent of simplifying the software design provided complementary motivation for the R&D described in the following sections. More agile software facilitates its maintenance and possible future evolution; it also supports the transparency of its semantics, thus facilitating its appropriate use.

3.1. Test configuration

The benchmarks were based on software released in Geant4 9.4.beta version and data released in G4EMLOW6.13.

The full set of benchmarks were executed on a Intel Core Duo CPU E8500 equipped with a 3.16 GHz processor and 4 GB of memory, running under Linux SLC5. GNU C++ compiler gcc 4.3.5 was used in this configuration. A subset of tests were executed on a Microsoft Windows system configured with Intel CPU U4100 with 1.30 GHz processor, with 1.96 GB of memory, running under Windows XP SP3. The MSVC++9 C++ compiler (with SP1) was used for these tests.

Two types of tests were performed to evaluate the computational performance of the code respectively at loading and retrieving data. The load test consisted of loading the data corresponding to a number of instantiated elements between 1 and 100; each experiment was repeated 100 times, and the whole series was repeated 10 times. The retrieve test consisted of finding the data associated with a randomly chosen atomic number; the finding procedure was repeated one million times, and the whole experiment was repeated 10 times.

3.2. Data structure

Some of the original data are structured as a single file, which encompasses data for one hundred elements. Loading the data needed for the elements required in a simulation, i.e. corresponding to the materials present in the experimental set-up, requires parsing the whole data file; this input-output (I/O) operation is expensive. To improve the agility of the loading process, such data files were split into individual files, each one associated with one element.

The gain in performance resulting from this modification depends on the number of data sets to be loaded in a simulation run; it is better than a factor 2 for simulation configurations where less than approximately 80 different elements are present.

The performance of data management is affected by the quantity of physics data to be handled. Large physics tabulations require large memory allocation for storing the data, time to load them into memory and to search through them.

A study was performed to evaluate on quantitative ground whether the size of the original data libraries could be reduced without affecting the precision of physics calculations. For this

purpose a test was developed to verify if the suppression of a given datum in the tabulations would allow the calculation of values of comparable accuracy through interpolation between adjacent data points, over the whole interval between them. The suppression of an original data point was considered tolerable if one could reproduce the same interpolated values as the original data tabulation within 0.01%.

The fraction of data that may be suppressed without significantly loosing precision in the simulation, and the consequent gain in performance, vary according to the physics data type. For instance, the gain in performance is approximately 60% for Compton scattering functions.

3.3. Use of forthcoming C++ features

The forthcoming edition of the C++ language [30] includes several new features. The R&D project evaluated whether the data management implementation could profit from a type of container, a so-called hash map, which is not available in the current Standard Template Library (STL), but is foreseen for inclusion in the new C++ standard.

The proposed C++0x TR1 name for a hash table is *unordered_map* ; it will replace the various incompatible implementations of the hash table (called *hash_map* by the gcc and MSVC compilers). As its name implies, unlike the *map* class, the elements of an *unordered_map* are not ordered; this is due to the use of hashing to store objects. The main advantage of hash tables over other types of associative containers is speed.

This container is currently accessible as *std::tr1::unordered_map*. Until TR1 is officially accepted into the upcoming C++0x standard, *unordered_map* is available from the $< tr1/unordered\_map >$ header file and from $< unordered\_map >$ in MSVC. *unordered_map* can be used in a similar way to the *map* class in C++ STL.

The *map* containers currently used in Geant4 data management system were replaced by *unordered_map* ones. This modification has negligible effects on data loading performance, while it improves significantly the performance of retrieving the data. For instance, the time needed for the retrieval of pair production cross section data can be reduced by approximately up to 40%; the improvement in performance is more visible when a large number of elements are present in the simulation set-up.

3.4. Caching pre-calculated data

An improvement of the performance of the data management has been announced as part of the Geant4 9.3 release [31]; it consists of caching pre-calculated logarithms in base 10 of the data for use in logarithmic interpolation. The authors of this paper are not responsible for these modifications and do not claim any credit for them.

Some tests were performed to evaluate quantitatively the computational performance effects of caching pre-calculated logarithms in base 10 of the data. A test including 100 million calls to a logarithm in base 10 showed that the overhead due to a single call is on average of the order of 10%. The performance improvement at retrieving data due to caching pre-calculated logarithm in base 10 of the data is approximately 25%; however, caching the data adds some penalty to the data load procedure.

3.5. Software design

The original software design of the data library management exploited a Composite design pattern [32].

An alternative design approach was investigated. The handling of data concerning atoms, shells and materials was unified under the responsibility of one class. The problem domain analysis recognized that polymorphic behavior of data sets and interpolation algorithms were not necessary at runtime through dynamic binding, rather could be realized by exploiting template programming techniques.

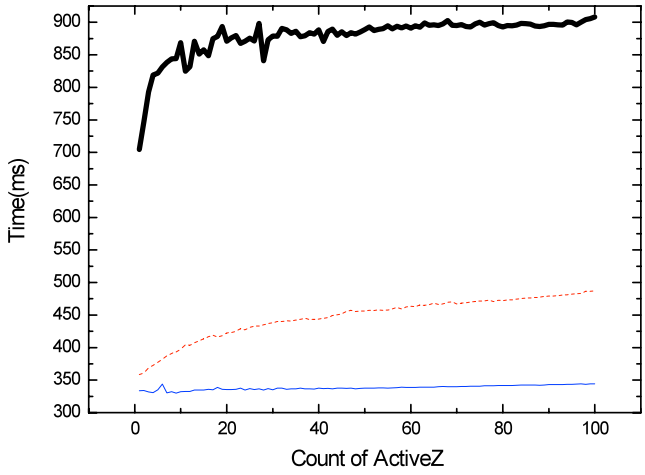

Figure 2. Improvement at retrieving pair production cross section data due to the design based on templates (dashed red curve) and on the use of unordered_map along with the new design (thin blue curve), with respect to the original classes (thick black curve).

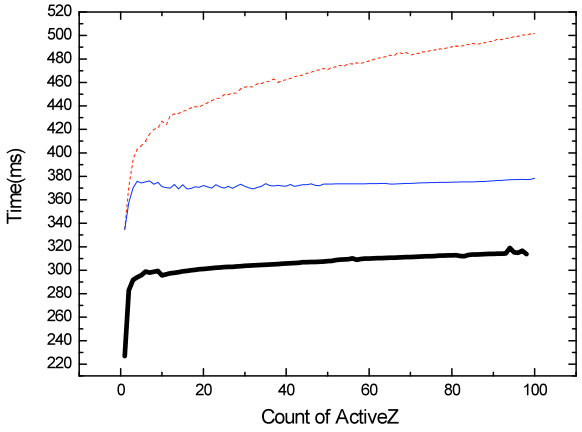

Figure 3. Improvement at retrieving Bremsstrahlung spectrum data due to the design based on templates (dashed red curve) and on the use of unordered_map along with the new design (thin blue curve), with respect to the original classes (thick black curve).

The adoption of template programming contributes to improving execution speed, since it eliminates the overhead due to the virtual table associated with inheritance. The performance gain at loading depends on the characteristics of the physics data: in some cases it is significant, while for some other kinds of physics data it is negligible. The performance is in general significantly improved at data retrieving, as illustrated in Figure 2 and Figure 3. These plots also show the combined effect of the new software design and of using the unordered_map foreseen in the new C++ standard.

The complete set of results will be documented and discussed in depth in a dedicated paper.

Conclusions

A significant effort has been invested into the evaluation of data libraries used by Monte Carlo codes and the development of new ones.

An R&D project evaluated the possibility of improving the design and computational performance of Geant4 physics data management system. Various issues were addressed: the improvement of the structure of the data themselves, the use of new features in the forthcoming C++ standard, the use of template programming to replace the current design based on the Composite design pattern. A recent improvement implemented in Geant4, consisting of caching some pre-calculated data, was quantitatively evaluated.

The effects on computational performance due to all the previously mentioned topics were independently measured. The overall improvement in computational performance is significant, both at loading and retrieving data; depending on the type of data, gains in data retrieving of approximately 30% up to almost a factor 3 can be achieved.

Acknowledgements

We thank CERN Directorate for support to the research described in this paper.

References

[1] Agostinelli S et al. 2003 *Nucl. Instrum. Meth. A* **506** 250
[2] Allison J et al. 2006 *IEEE Trans. Nucl. Sci.* **53** 270
[3] Perkins S T et al 1991 *Tables and Graphs of Atomic Subshell and Relaxation Data Derived from the LLNL Evaluated Atomic Data Library (EADL), Z=1-100* UCRL-50400 Vol. 30
[4] Chauvie S et al. 2001 *Proc. Computing in High Energy and Nuclear Physics* 337
[5] Chauvie S et al. 2004 *Conf. Rec. 2004 IEEE Nucl. Sci. Symp.* N33-165
[6] Scofield J H 1974 *Phys. Rev. A*, **9** 1041
[7] Scofield J H 1974 *Phys. Rev. A* **10** 1507
[8] Pia M G et al. 2009 *IEEE Trans. Nucl. Sci.* **56** 3650
[9] Scofield J H 1969 *Phys. Rev. A* **179** 9
[10] Scofield J H 1974 *Atom. Data Nucl. Data Tables* **14** 121
[11] Seo H et al. 2010 *Conf. Rec. 2010 IEEE Nucl. Sci. Symp.*
[12] Powell C J 1995 *Appl. Surf. Sci.* **89** 141
[13] Carlson T A 1975 *Photoelectron and Auger spectroscopy* Plenum, New York
[14] Firestone R B et al. 1996 *Table of Isotopes 8th ed.* John Wiley & Sons, New York
[15] Lederer M and Shirley V C 1978 {textitTable of Isotopes 7th ed. John Wiley & Sons, New York
[16] Thompson A C et al. 2009 *X-ray Data Booklet* Berkeley, CA, USA
[17] Sevier K 1979 *Atom. Data Nucl. Data Tables* **24** 323
[18] Perkins S T et al. 1997 *Tables and Graphs of Electron-Interaction Cross Sections from 10 eV to 100 GeV Derived from the LLNL Evaluated Electron Data Library (EEDL)* UCRL-50400 Vol. 31
[19] Seo H et al. 2010 *SNA+MC 2010 Conference* Full Paper no. 10255
[20] Seo H et al. 2010 *IEEE Nucl. Sci. Symp. Conf. Rec.*
[21] Seo H et al. 2011 *CHEP 2011 Proc.*
[22] Kim Y K and M. E. Rudd M E 1994 *Phys. Rev. A* **50** 3954
[23] Deutsch H and Märk D T1987 *Int. J. Mass Spectrom. Ion Processes* **79** R1
[24] Pia M G et al 2009 *IEEE Trans. Nucl. Sci.* **56** 3614
[25] Tuli J K 2001 *E*valuated Nuclear Structure Data File: A Manual for Preparation of Datasets Brookhaven National Laboratory
[26] Hauf S et al 2009 *IEEE Nucl. Sci. Symp. Conf. Rec.* 2060
[27] Hauf S et al. 2010 *SNA+MC Proc.* Full Paper no. 10275
[28] Augelli M et al. 2009 *IEEE Nucl. Sci. Symp. Conf. Rec.* 177
[29] Pia M G et al. 2010 *J. Phys. Conf. Ser.* **219** 042019
[30] ISO/IEC 2010 C++ JTC1 SC22 WG21 N3092
[31] Geant4 Release notes http://cern.ch/geant4 /support/ReleaseNotes4.9.3.html
[32] Gamma E, Helm R, Johnson R, Vlissides J 1995 *Design Patterns* Addison-Wesley, New York