Electromagnetic physics vectorization in the GeantV transport framework

Guilherme Amadio[1], Ananya[1], *John* Apostolakis[1], *Marilena* Bandieramonte[1,2,*], *S. P.* Behera[3], *Abhijit* Bhattacharyya[3], *Rene* Brun[1], *Philippe* Canal[4], *Federico* Carminati[1], *Gabriele* Cosmo[1], *Vitalji* Drohan[1], *Victor Daniel* Elvira[4], *Krzysztof* Genser[4], *Andrei* Gheata[1], *Mihaela* Gheata[1,5], *Ilias* Goulas[1], *Farah* Hariri[1], *Vladimir* Ivanchenko[1,6], *Gul Ruk* Khattak[1], *Dmitri* Konstantinov[1], *Harpool* Kumawat[3], *Jose Guilherme* Lima[4], *Jesus* Martinez Castro[8], *Pere* Mato[1], *Patricia* Mendez[1], *Aldo* Miranda Aguillar[8], *Katalin* Nikolics[1], *Mihaly* Novak[1], *Elena* Orlova[1], *Kevin* Pedro[3], *Witold* Pokorski[1], *Alberto* Ribon[1], *Dmitry* Savin[1], *Ryan* Schmitz[1], *Raman* Sehgal[3], *Oksana* Shadura[1,7], *Shruti* Sharan[1], *Sofia* Vallecorsa[1], *Sandro Christian* Wenzel[1], and *Soon* Yung Jun[4]

[1]CERN, Meyrin 1211, Switzerland

[2]University of Pittsburgh, PA 15260, USA

[3]Bhabha Atomic Research Centre, Mumbai 400085, IN

[4]Fermi National Accelerator Lab, IL 60510, US

[5]Institute of Space Science, Magurele 077125, RO

[6]Tomsk State University, Tomsk 634050, Russia

[7]University of Nebraska, NE 68588, US

[8]Centro de Investigación en Computación, 07738 Gustavo A. Madero, Mexico

Abstract. The development of the GeantV Electromagnetic (EM) physics package has evolved following two necessary paths towards code modernization. A first phase required the revision of the main electromagnetic physics models and their implementation. The main objectives were to improve their accuracy, extend them to the new high-energy frontier posed by the Future Circular Collider (FCC) programme and allow a better adaptation to a multi-particle flow. Most of the EM physics models in GeantV have been reviewed from theoretical perspective and rewritten with vector-friendly implementations, being now available in scalar mode in the alpha release. The second phase consists of a thorough investigation on the possibility to vectorise the most CPU-intensive physics code parts, such as final state sampling. We have shown the feasibility of implementing electromagnetic physics models that take advantage of SIMD/SIMT architectures, thus obtaining gains in performance. After this phase, the time has come for the GeantV project to take a step forward towards the final proof of concept. This takes shape through the testing of the full simulation chain (transport + physics + geometry) running in vectorized mode. In this paper we will present the first benchmark results obtained after vectorizing a full set of electromagnetic physics models.

[*] Corresponding author: marilena.bandieramonte@cern.ch

1 Introduction

Large Hadron Collider (LHC) experiments rely heavily on Monte Carlo simulations of particle transport and interaction with detector material. These simulations are among the most time consuming parts of the HEP software work-flow, since the simulated data-sets have to be accurate and extensive, and each simulation is computationally expensive. The increase in the integrated luminosity, expected in the next decade of the future LHC runs (HL-LHC), and the consequent increase in the amount of experimental data to be analyzed, implies the need to simulate larger and more accurate samples to avoid that the systematic errors due to the simulations will become dominant. The GeantV R&D activity [1, 2] is underway with the goal of speeding up the detector simulation by exploiting modern computing architecture platforms, in order to bring the HEP community a step forward towards accomplishing the goals posed by the compelling future physics programmes. The project investigates potential computational benefits of using a multiple track transportation approach instead of the classical single particle transportation flow. This is done in order to improve code and data locality in the process, and artificially enhance the data-level parallelism (DLP) of the simulation software enabling SIMD/SIMT execution models to combine the benefits of vectorization and multithreaded approaches. In a typical HEP simulation, geometry and transportation take a significant portion of the total execution time. However, the execution time is usually dominated by the physics components, and mainly by the electromagnetic physics. An important fraction of the simulations, in fact, is normally spent on the simulation of electromagnetic showers inside calorimeters. Jets simulations, having electromagnetic components deriving from the decay of neutral pions, have also a significant weight in the total simulation time. While working on the performance, the GeantV project aims also at revising the physics models, in order to improve where possible their accuracy and extend their range of validity to the new high-energy frontier, that will be reached with the FCC programme. Most of the EM physics models in GeantV have been reviewed from a theoretical perspective and rewritten with vector-friendly implementations. The next step towards the final demonstration of the performances obtainable from the GeantV prototype consists in the vectorization of the electromagnetic physics models.

In this paper we present several benchmarks obtained from the vectorization of the main electromagnetic physics models. They have been tested by running the full simulation chain, i.e. transport, physics and geometry, and verifying the results in comparison with the corresponding Geant4 [3] simulations. This paper is organized as follows: Sect. 2 gives a generic introduction to vectorization with some additional consideration on what speedup factors can be reasonably expected when using SIMD registers. Sect. 3 gives an overview of the status of implementation of the electromagnetic physics library in GeantV. Sect. 4 describes what solutions have been designed and implemented to speed up the simulation physics code. Benchmark results are presented in Sect. 5. Lastly, Sect. 6 presents final conclusions.

2 Vectorization is good, when it happens

In recent years, the stall in the CPU frequency with the consequent advent of multi-core architectures, has led to an increasing need for performance and energy efficiency exploitation in modern processors. This promoted a resurrected interest in the Single Instruction Multiple Data (SIMD) vector units. SIMD instruction set extensions are quite popular today and commonly available in most instruction sets of both high-performance and embedded microprocessors. The major vendors support vector architectures and the trend goes towards increasingly wide and powerful vector units [4]. However, the real challenge is to use these instructions effectively, writing code that can truly take advantage of the ever increasing size of the underlying vector registers [5]. Vectorization requires often not only to change the algorithmic structure but also to modify the data layout. Two solutions to exploit vectorization are commonly used: the first is to implicitly achieve it with compiler-based automatic vectorization. The second is to explicitly enforce vectorization by using intrinsics: compiler-known functions that are built-in and can be directly expanded to machine instructions. Auto-vectorization can typically be applied in two main categories of algorithms: one are loop-based algorithms [6] where the multiple iterations of a loop can be converted in a single vector iteration. The second one, the super-word level parallelism (SLP) [7], can be applied in more general cases as it is directed to straight-line code and operates on repeated sequences of scalar instructions outside a loop. However, compiler-based vectorization is often not powerful enough and has a limited scope of applicability. On the other hand, the use of intrinsics is complex and significantly decreases the readability and maintainability of the code, leading to platform-specific implementations, to the detriment of code portability. GeantV project implements an innovative approach to vectorization, based on the combined use of VecCore [8] and the concept of templated backends [9, 10]. VecCore is a wrapper library that provides an abstraction layer on top of existing SIMD libraries, such as Vc [11] and UME::SIMD [12], that facilitate users to write generic vectorized code. The use of templated backends allows to write code that is generic and transparent to the platform-specific implementation details. Only at compilation time the code is specialized for a specific type of backend (scalar, SSE, AVX, AVX2...), allowing to enable vectorization while maintaining readability, maintainability and portability. This vectorization model is at the base of the success of the vectorized geometry library developed in the framework of the GeantV project, VecGeom [10], that is also integrated in the production version of Geant4 since release 10.2 and is being progressively adopted by the major LHC experiments. The same vectorization model is being applied to the EM physics library, focusing on computationally expensive physics calculations.

However, before discussing performance, it is fundamental to analyze and understand what speedup factors can be expected from vectorization, beyond the ideal value given by the vector width. In fact, benefits of vectorization are not always easy to exploit and the gain obtainable depends heavily on the algorithmic structure. Several are the factors that have to be taken into account. Functions with many math computations are likely to produce high speed-up factors, while functions bounded by memory accesses are not the best use-case to vectorize. Branching plays also an important role in vectorization: functions with minimal branching are more suitable for vectorization, because conditional statements usually reduce the vector registers population and may require, in most cases, to evaluate both branches for vectorized code. Even in presence of math-bounded algorithms without any branching, the maximum speed-up achievable is generally less than the vector register width. Some vector operations are in fact slower than the corresponding scalar ones in some CPUs. Consider, as an example, the reciprocal throughput for double precision division in an Intel® SandyBridge. This operation takes $10-20$ cycles to be executed in scalar mode while it requires $20-44$ cycles with vector registers. This means that the maximum speedup obtainable for a double precision division on this CPU would be of ~ 2 when the ideal one is 4. Another factor that has to be taken into consideration is the overhead that has to be paid to gather data into SIMD vectors, in order to be ready for vector operations. This overhead in some cases can exceed the vectorization gain if the data layout is not properly designed or the algorithmic structure is very complex. Furthermore the comparison with scalar executions is not always "fair" because the total execution time when running in scalar mode depends also on the number of execution units of the CPU available for specific instructions, i.e. the number of instructions that can be executed simultaneously.

3 Electromagnetic physics modelling

As mentioned in the introduction, the ultimate goal of the GeantV R&D activity is to exploit the possible computational benefits of applying vectorisation techniques on high energy physics (HEP) detector simulation codes. From physics modelling point of view, the most intensively used and computationally most demanding part of these simulations is the description of electromagnetic (EM) interactions of e^-, e^+ and γ particles with matter. This is what motivated the choice of selecting EM shower simulation in HEP applications to demonstrate the possible computational benefits of applying track level vectorisation on HEP detector simulation codes.

Geant4 provides a unique variety of EM physics models to describe particle interactions with matter from the eV to PeV energy range with different level of physics accuracy. Each application areas can find a suitable set of models with the appropriate balance between the required accuracy of the physics description and the corresponding computational complexity. Moreover, Geant4 provides a pre-defined collection of EM physics models and processes for different application area in the form of EM physics constructors [13]. Among these the so-called EM *standard* physics constructor (i.e. EM Opt0) is the one that is recommended by the developers for HEP detector simulations. The corresponding Geant4 EM physics processes and models are summarised in Table 1. (right) [13].

According to the goal of the project, this EM *standard* physics constructor served as a guide line for the GeantV EM physics development. The physics processes and the corresponding models implemented in the GeantV framework are also listed in Table 1. (left). As it can be seen, all the EM processes are available in the GeantV physics framework apart from coherent scattering of photons and energy loss fluctuation. The latter was decided not to be implemented since the development of an alternative model is under investigation, while coherent scattering of photons does not play a significant role in case of general HEP applications. Moreover, all the physics processes are described by EM models based on exactly the same theoretical framework both in GeantV and Geant4 except the Coulomb scattering of e^-/e^+ for which the Goudsmit-Saunderson Geant4 model [14] was selected to be implemented in GeantV. Nevertheless, all the GeantV models are available in Geant4 that makes possible the one-to-one comparison of the results. Furthermore, these differences do not affect at all the final goal of the GeantV R&D activity and the GeantV physics list can provide reliable EM physics shower simulations in case of HEP applications.

The accuracy of each GeantV model implementation was carefully tested through individual model level tests by comparing both the computed final states as well as the integrated quantities (e.g. cross sections, stopping power) to those produced by the corresponding Geant4 version of the given model. Moreover, several simulation applications have been developed to test and verify the GeantV EM shower simulation accuracy including both a general, simplified sampling calorimeter and a complete CMS detector setups. In all cases, the GeantV simulation results (regarding quantities such as energy deposit distributions in a given part of the detector, number of charged and neutral particle steps, secondary particles, etc.) agreed to the corresponding Geant4 simulation results within the impressive 0.1 % difference. This level of accuracy provided a solid and reliable starting point for the *standard* EM physics vectorisation. More about the implementation details of the EM models and their vectorisation will be given in the following sections.

4 Vectorization of the electromagnetic physics code

Particle transport simulation codes solve the Boltzmann transport equation numerically by using the Monte Carlo technique which determines the stochastic nature of the detector simula-

Electromagnetic physics description

particle	**processes**	**model(s)**	
		GeantV	**Geant4**
e^-	ionisation	Møller [100eV-100TeV]	Møller [100eV-100TeV]
	bremsstrahlung	Seltzer-Berger [1keV-1GeV]	Seltzer-Berger [1keV-1GeV]
		Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]	Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]
	Coulomb sc.	GS MSC model [100eV-100TeV]	Urban MSC model [100eV-100MeV]
			WentzelVI mixed model [100MeV-100TeV]
e^+	ionisation	Bhabha [100eV-100TeV]	Bhabha [100eV-100TeV]
	bremsstrahlung	Seltzer-Berger [1keV-1GeV]	Seltzer-Berger [1keV-1GeV]
		Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]	Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]
	Coulomb sc.	GS MSC model [100eV-100TeV]	Urban MSC model [100eV-100MeV]
			WentzelVI mixed model [100MeV-100TeV]
	annihilation	Heitler (2γ) [0-100TeV]	Heitler (2γ) [0-100TeV]
γ	photoelectric	Sauter-Gavrila + EPICS2014 [1eV-100TeV]	Sauter-Gavrila + EPICS2014 [1eV-100TeV]
	incoherent sc.	Klein-Nishina$^+$ [100eV-100TeV]	Klein-Nishina$^+$ [100eV-100TeV]
	e$^-$e$^+$pair production	Bethe-Heitler$^+$ [100eV-80GeV]	Bethe-Heitler$^+$ [100eV-80GeV]
		Bethe-Heitler$^+$ w. LPM [80GeV-100TeV]	Bethe-Heitler$^+$ w. LPM [80GeV-100TeV]
	coherent sc.	-	Livermore
+	energy loss fluct.	-	Urban

Table 1: Electromagnetic physics modelling in the GeantV prototype (left) compared to the Geant4 *standard* (Opt0) EM physics constructor [13] (right) .

tion algorithms. It implies generating samples of stochastic variables according to their probability distributions determined by the underlying physics, conditional branches depending on the actual outcome of such stochastic variables and further nondeterministic components in each simulation step. As mentioned in Section 2, the stochastic nature and conditional branches do not make these algorithms straightforwardly vectorizable.

A simulation step, limited by a discrete physics interaction, can be divided into two distinct parts from the physics modelling point of view: selecting the physics interaction with the corresponding interaction point and invoking the interaction itself. The former is driven by the integrated cross section values of the physics processes assigned to the given particle. In the EM physics modelling part, it practically means using cross section table lookups and interpolations, which are memory bounded operations with very little mathematical computations that, according to Section 2, do not make this part suitable for vectorization. However, the second part, that includes the computation of the post-interaction kinematical state of the primary particle if any as well as the generation of possible secondary particles, usually contains significantly more mathematical operations which makes it more appropriate for vectorization. Therefore, the final state computation part of the EM physics models is the primary target for exploring vectorization benefits in the context of EM physics simulation.

The final state computation usually includes generation of stochastic variables such as energy transfer, scattering or ejection angles from their probability distributions determined by the corresponding differential (in energy, angle) cross sections (DCS) of the underlying physics interaction. Since these probability density functions (PDF) are proportional to the DCS, which is usually a complex function of the primary particle, target material properties and very often available only in numerical form, the analytical inversion of corresponding cumulative distribution function (CDF) is unknown. Since analytical inverse transform method cannot be used, different numerical techniques need to be utilized to generate samples of these stochastic variables according to their probability distributions.

Geant4, as other particle transport simulation codes, makes extensive use of the composition-rejection method to generate random samples according to a given PDF. This algorithm contains an unpredictable number of loop executions since the exit condition depends on the outcome of the stochastic variable itself. Moreover, when the algorithm is vectorized over primary particle tracks, the exit condition is reached unpredictably for the different tracks filled into the vector register, resulting in undesired divergence and eventually loss of potential computational gain. Therefore, composition-rejection algorithm is not very suitable for vectorization.

Alternative techniques, with perfectly deterministic behaviour, have been investigated and will be discussed shortly in Section 4.1. As it will be shown, using these table-based sampling methods can accompany with significant memory overhead, that can restrict their general applicability. Therefore, a solution to overcome the vectorization barrier posed by the nondeterministic exit condition of the rejection-based sampling algorithm will be discussed in Section 4.2.

4.1 Deterministic sampling algorithms: *sampling tables*

A frequently used deterministic algorithm to efficiently generate samples of discrete stochastic variables is the alias method [15]. Since the stochastic variables used in the final state computations are all continuous, this method cannot be applied directly on the corresponding PDFs. Instead, an intermediate discrete random variable is introduced by partitioning the range of the original continuous variable into distinct intervals and defining the new discrete variable as the probability of having the original continuous variable lying in a given interval. The alias method can then be applied for fast, deterministic sampling of this new discrete variable. The outcome of this first step will determine one of the distinct intervals of the original continuous variable. A sample value of the original continuous random variable can be obtained by sampling from the corresponding continuous PDF over the selected interval. This can be done in a very efficient way by assuming linear behaviour of the PDF over each discrete interval. This approximation can be always fulfilled by choosing an appropriate partition of the original variable range and/or applying clever variable transformations to transform the original PDF to a smooth function. The smoother the transformed PDF the better the linear approximation, which implies less intervals that directly translates to smaller memory requirement.

Another possibility to obtain deterministic sampling algorithm is the numerical inverse-transform of the CDF by using a higher order interpolation scheme based on a discrete set of numerical values of the CDF. Since the inverse-transform method starts with a uniformly distributed random sample, this solution will include a search to identify the corresponding CDF interval to be used in the interpolation. Unfortunately, the search algorithm terminates after nondeterministic number of steps depending on the actual value of the initial uniform random variable. However, this issue can be resolved by combining the numerical inverse-transform with and appropriate alias method for the fast, deterministic sampling of this CDF interval.

In all above cases, the nondeterministic nature of the rejection-based sampling algorithm can be completely eliminated by employing these table-based alternative techniques making them more suitable for vectorization. Moreover, these sampling tables are prepared at initialisation incorporating all the complexity of the underlying distributions then used at run-time for generating samples with very simple computations. In case of computationally complex or numerical DCSs this can result in significant reduction of the run-time computation of sampling compared to rejection since the latter requires the evaluation of these complex functions at each time (possibly more than once) whenever a sample needs to be produced. Vectorization of the table based sampling will enhance this computational gain even further in these cases. On the other hand, the computation is usually more determined by memory accesses in case of using such tables compared to the composition-rejection algorithm because the values at different table indices need to be gathered into vector registers before making use of them. This makes these sampling table based algorithms less ideal for vectorization.

The biggest disadvantage of using sample tables however is their memory requirement. An appropriate set of discrete samples from the approximated function (PDF or inverse CDF) need to be stored in these tables together with some additional parameter values for the fast sampling that can satisfy the corresponding approximation. Furthermore, the fact that all the dependences of the underlying probability distribution need to be incorporated in the sampling tables at initialisation time in order to obtain fully deterministic sampling algorithm, indicates that multiple set of tables needs to be prepared when the corresponding DCS shows atomic number or material dependence. Moreover, the situation is even worse when alias method is involved in the sampling since the loss of monotonicity makes impossible to produce samples of the original continuous variable lying within an arbitrary restricted interval. This makes necessary to build separate set of sampling tables whenever a simple restricted interval sampling would be appropriate e.g. incorporating secondary production threshold dependence. As a final consequence, different sets of sampling tables need to be prepared very often as a function of the atomic number, material or material-secondary production threshold pairs available in the detector which can make these methods memory-consuming.

4.2 Making use of vectorization in case of composition-rejection

Rejection sampling is a widely used algorithm for generating random samples according to a given PDF. Its popularity lies in the fact that efficient algorithms i.e. with low rejection rate can be relatively easily composed even in case of complex PDFs while the algorithms usually have negligible memory requirements. However, as it has already been mentioned, the nondeterministic number of sampling loop executions results in unpredictably reached termination condition for the different tracks filled into the vector register: this causes undesired divergence and eventually loss in the efficiency of employing the full width of the vector register. A so-called *lane refilling* algorithm has been developed to overcome this issue. The algorithm makes sure that the full capacity of the available vector register is used in each rejection sampling loop iteration. It is achieved by replacing those track variables in the vector register for which the stochastic loop termination condition has been reached in the current iteration, with variables taken from untouched tracks while keeping for the next iteration those for which the sampling has not been completed yet. This algorithm ensures that the full width of vector register is exploited in each sampling loop iteration at the price of some data movements needed to refill the holes. It should be noted that a vectorized random number generator [16] is also necessary to be used in order to achieve the maximum performance gain offered by vectorization.

5 Performance results and discussion

Model level tests allow to execute specific components of the implemented EM physics models, such as the final state generation part, without the need of the whole simulation framework. Such test has been developed for each EM model for testing and verifying the GeantV implementation against the corresponding Geant4 one, as well as the *sampling table* based final state generation against the one using *rejection*. These model level tests also include the possibility to test and verify the *vectorized* implementations against the *scalar* versions. Speed-up factors, comparing the *scalar* implementations of the different sampling techniques (*table, rejection*) to their vectorized versions obtained with these micro-benchmarks, are shown in Fig.1.

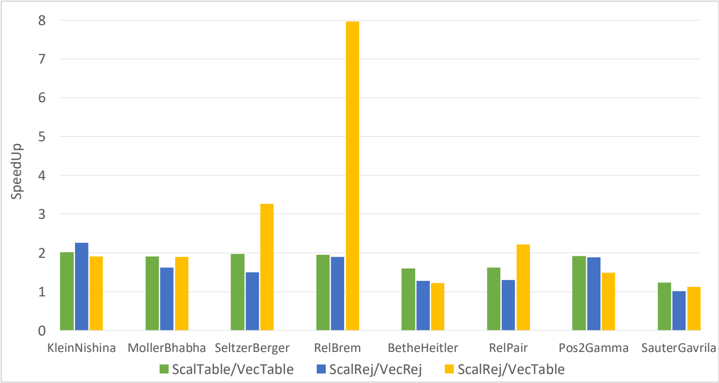

Figure 1: Speedup of the final state generation of different electromagnetic physics models obtained with SIMD vectorization in case of different sampling algorithms. The results were obtained by using Google Benchmarks [17] on an Intel® Haswell Core$^{\text{TM}}$ i7-6700HQ, 2.6 GHz, with Vc backend and AVX2 instruction set processing 256 tracks.

In particular, green bars refer to the *table*, the blue to the *rejection* based sampling algorithms and they show the speed-up values obtained by the *vectorization* of the corresponding codes in case of different EM physics models. Yellow bars show the combined, *table* based sampling plus *vectorization*, gain compared to the *scalar*, *rejection* based implementations. Since the values behind the green and yellow bars share the same denominator, the relation of their heights indicate the speed difference between the *scalar* implementations of the *table* and *rejection* based sampling algorithms. Similarly, since the blue and the yellow bars values share the same numerator, the relation of their heights shows the difference between the speed achievable vectorizing the *table* or the *rejection* based sampling algorithm.

As it was expected after the discussion given in Section 4, a wide range of performance variation of the algorithms as well as their vectorization gain can be observed in Fig.1. This is due to the fact that each of the investigated EM physics model translates to a finale state sampling algorithm with unique computational characteristics that will be more favourable for one sampling technique compare to the other.

Taking for example the Klein-Nishina model [18, 19] for Compton scattering of photons, one can see that the *rejection* based solution is faster than the sampling *table* based one since the green bar is higher than the yellow one. This is due to the fact that the computationally simple Klein-Nishina DCS makes possible the composition of a very efficient *rejection* based final state generation with a very low rejection rate and computational complexity that cannot be beaten by the sampling *table* based version. On the contrary, the DCS (in total energy transferred to one of the e^-/e^+ pair) in case of the high-energy model for e^-/e^+ pair production [18, 20–22] (RelPair) is significantly more complex compared to the Klein-Nishina DCS for Compton scattering. This results in a computationally more demanding *rejection* based final state generation compared to the *table* based one where all such complexities are handled at initialisation time as discussed in Section 4.1. Moreover, further advantage of using sampling *table* over the composition-*rejection* can be observed in case of the high-energy bremsstrahlung model (RelBrem), where the computationally expensive nature of the DCS comes together with a higher rejection rate due to the inclusion of the LPM suppression effect [18, 21, 22]. This makes the *rejection* based algorithm significantly ($\sim 4\times$) slower compared to the sampling *table* based one. This gain further enhanced up to 8.4 thanks to the efficient vectorization of the sampling *table* based algorithm. In general, the combined effects of choosing the appropriate sampling algorithm and its vectorization for a given EM physics model provides a speedup factor of 1.8-3.

The final state generation algorithm of some EM models follows different computational paths depending on some external conditions. For example, some corrections are computed only above a given primary particle energy but not below; or selecting a target atom for the interaction is necessary only in case of materials with multiple elements. These directly translate to particular computing paths with different computational complexities favouring one or the other sampling technique within the same model. This is demonstrated in Fig.2, where the benchmark results are shown for the high energy e^-/e^+ pair production model (RelPair) under different primary energy and target material (Pb or $PbWO_4$) conditions. One can see that the observable speed-up for the *rejection* based sampling depends on these conditions. At the same time, the *table* based sampling algorithms do not depend on these conditions, as it was discussed in Section 4.1., showing a constant speedup factor.

While the sampling *table* based final state generations have a constant runtime under any external conditions, the efficiency of a given *rejection* algorithm can change significantly with the primary particle energy. This is illustrated in Fig. 3 where the relative speed of applying these two techniques in case of the Bethe-Heitler e^-/e^+ pair production model [18, 23] is shown as a function of the primary γ particle energy. It can be seen that the two algorithms perform similarly at lower primary γ particle energies while the *rejection* algorithm becomes ~35 % faster at higher γ energies simply due to the smaller rejection rate.

These results indicate that one needs to investigate all the available final state generation algorithms in order to select the most efficient one, depending on many conditions such as the complexity of the underlying DCS, target material composition or primary particle energy. The GeantV physics framework have been designed by taking all these considerations into account, allowing to choose the most efficient algorithm for final state generation depending on the primary particle energy or detector region. This makes possible to obtain the maximum performance gain while keeping the memory consumption of the algorithms under control even in the case of the most complex HEP detector simulation applications.

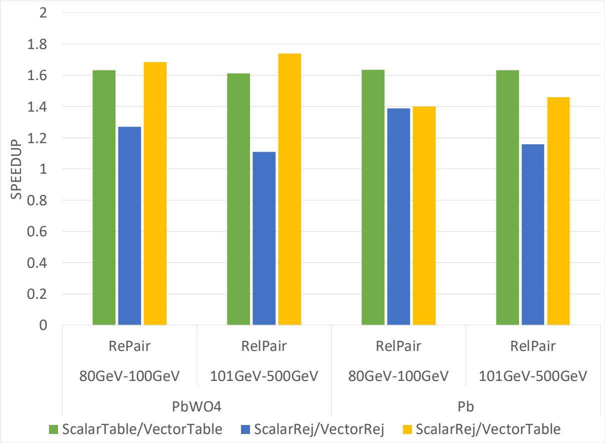

Figure 2: Microbenchmark results for final state generation in case of the high energy e^-/e^+ pair production model under different primary energy (80-100 [GeV] and 101-500 [GeV]) and target material conditions (left: $PbWO_4$, right: Pb).

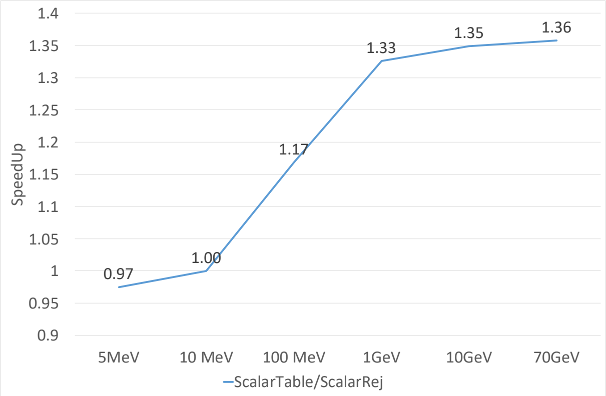

Figure 3: Speedup of the *rejection* based final state sampling compared to the sampling *table* based one in case of the Bethe-Heitler e^-/e^+ pair production model [18, 23] as a function of the primary γ particle energy.

6 Conclusions

The ultimate goal of the GeantV R&D activities is to show the potential benefits of applying track level vectorization on HEP detector simulation codes. Taking into account its computational significance, EM shower generation was selected as the target for providing the final proof-of-concept to demonstrate the computational advantage of vectorization under realistic HEP detector simulation conditions. To this end, a complete set of vectorized EM physics models, recommended for HEP applications, has been implemented for e^-, e^+ and γ particles within the GeantV multi-particle transportation environment. This provides the possibility of vectorized EM shower simulation in HEP applications with a difference in the relevant observables of less than per mille level compared to the corresponding Geant4 results. In order to achieve the maximum available performance, different algorithmic solutions for the final state generations have been investigated in case of all implemented EM models. As it has been shown, the computational diversity of these physics models directly translates to a variation of the optimal algorithmic solution as a function of the models. Moreover, the dependence of the underlying physics on some external conditions such as target material composition or primary particle energy can introduce further variations on the selection of the most suitable algorithm even in the case of a single model. The flexibility of the GeantV physics framework can fully cope with this diversity allowing to use the most optimal solutions as a function of primary particle energy or detector region. These bring the GeantV R&D activity to a stage of possessing all physics modelling related requirements to fulfil its final goal.

References

[1] G. Amadio, J. Apostolakis, M. Bandieramonte, A. Bhattacharyya, C. Bianchini, R. Brun, P. Canal, F. Carminati, L. Duhem, D. Elvira et al., Journal of Physics: Conference Series **664**, 072006 (2015)

[2] G. Amadio, J. Apostolakis, M. Bandieramonte, S. Behera, A. Bhattacharyya, R. Brun, P. Canal, F. Carminati, G. Cosmo, V. Drogan et al., Journal of Physics: Conference Series **1085**, 032037 (2018)

[3] J. Allison, K. Amako, J. Apostolakis, P. Arce, M. Asai, T. Aso, E. Bagli, A. Bagulya, S. Banerjee, G. Barrand et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **835**, 186 (2016)

[4] T. Sato, H. Mori, R. Yano, T. Hayashida, pp. 107–114 (2012)

[5] M. Hassaballah, S. Omran, Y.B. Mahdy, Comput. J. **51**, 630 (2008)

[6] D. Nuzman, I. Rosen, A. Zaks, SIGPLAN Not. **41**, 132 (2006)

[7] J. Shin, M. Hall, J. Chame, *Superword-Level Parallelism in the Presence of Control Flow*, in *Proceedings of the International Symposium on Code Generation and Optimization* (IEEE Computer Society, Washington, DC, USA, 2005), CGO '05, pp. 165–175, ISBN 0-7695-2298-X, `http://dx.doi.org/10.1109/CGO.2005.33`

[8] G. Amadio, P. Canal, D. Piparo, S. Wenzel, Journal of Physics: Conference Series **1085**, 032034 (2018)

[9] J. Apostolakis, M. Bandieramonte, G. Bitzes, G. Cosmo, J. de Fine Licht, L. Duhem, A. Gheata, G. Lima, T. Nikitina, S. Wenzel, *First experience with portable high-performance geometry code on GPU*, in *Proceedings, GPU Computing in High-Energy Physics (GPUHEP2014): Pisa, Italy, September 10-12, 2014* (2015), pp. 95–102

[10] J. Apostolakis, M. Bandieramonte, G. Bitzes, R. Brun, P. Canal, F. Carminati, G. Cosmo, J.C.D.F. Licht, L. Duhem, V.D. Elvira et al., Journal of Physics: Conference Series **608**, 012023 (2015)

[11] M. Kretz, V. Lindenstruth, Softw. Pract. Exper. **42**, 1409 (2012)

[12] P. Karpiński, J. McDonald, *A High-performance Portable Abstract Interface for Explicit SIMD Vectorization*, in *Proceedings of the 8th International Workshop on Programming Models and Applications for Multicores and Manycores* (ACM, New York, NY, USA, 2017), PMAM'17, pp. 21–28, ISBN 978-1-4503-4883-6, `http://doi.acm.org/10.1145/3026937.3026939`

[13] *Geant4 documentation: Guide for Physics Lists*, `http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/PhysicsListGuide/html/index.html`, accessed: 2018-12-09

[14] S. Incerti, V. Ivanchenko, M. Novak, Journal of Instrumentation **13**, C02054 (2018)

[15] A.J. Walker, ACM Trans. Math. Softw. **3**, 253 (1977)

[16] J. Apostolakis, S.Y. Jun, *Parallel random number generators: Vecrng* (2018), `https://indico.cern.ch/event/727112/contributions/3097294/attachments/1708255/2753368/VRNG-Geant42018.pdf`

[17] G. LLC, *Google benchmarks*, `https://github.com/google/benchmark` (2016-2018)

[18] *Geant4 documentation: Physics Reference Manual*, `http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/PhysicsReferenceManual/html/index.html`, accessed: 2018-12-09

[19] W. Heitler, *The quantum theory of radiation* (Courier Corporation, 1984)

[20] Y.S. Tsai, Reviews of Modern Physics **46**, 815 (1974)

[21] A.B. Migdal, Physical Review **103**, 1811 (1956)

[22] S. Klein, Reviews of Modern Physics **71**, 1501 (1999)

[23] H. Bethe, W. Heitler, **146**, 83 (1934)