GeantV alpha release

G Amadio[1], Ananya[2], J Apostolakis[1], M Bandieramonte[1], S Behera[3], A Bhattacharyya[3], R Brun[1], P Canal[4], F Carminati[1], G Cosmo[1], V Drogan[5], L Duhem[6], D Elvira[4], K Genser[4], A Gheata[1], M Gheata[1,7], I Goulas[1], F Hariri[1], V Ivantchenko[1], S Jun[4], P Karpinski[1], G Khattak[1], D Konstantinov[1], H Kumawat[3], G Lima[4], J Martínez-Castro[11], P Mendez Lorenzo[1], A Miranda-Aguilar[11], K Nikolics[1], M Novak[1], E Orlova[9], W Pokorski[1], A Ribon[1], R Sehgal[3], R Schmitz[8], S Sharan[1], O Shadura[1], S Vallecorsa[1,10], S Wenzel[1]

[1] CERN - European Organization for Nuclear Research, Geneva, Switzerland
[2] IIT- Indian Institute of Technology, India
[3] Bhabha Atomic Research Center, India
[4] Fermi National Accelerator Laboratory, USA
[5] National University of Kyiv, Ukraine
[6] Intel Corporation
[7] Institute of Space Sciences, Romania
[8]University of Minnesota, USA
[9]Higher School of Economics, Moscow, Russia
[10]Gangneung-Wonju National University, Gangneung, South Korea
[11]Centro de Investigación en Computación, Mexico City, Mexico

andrei.gheata@cern.ch

Abstract. In the fall 2016, *GeantV* went through a thorough community evaluation of the project status and of its strategy for sharing the R&D results with the LHC experiments and with the HEP simulation community in general. Following this discussion, *GeantV* has engaged onto an ambitious 2-year road-path aiming to deliver a beta version that has most of the final design and several performance features of the final product, partially integrated with some of the experiment's frameworks. The initial *GeantV* prototype has been updated to a vector-aware concurrent framework, which is able to deliver high-density floating-point computations for most of the performance-critical components such as propagation in field and physics models. Electromagnetic physics models were adapted for the specific *GeantV* requirements, aiming for the full demonstration of shower physics performance in the alpha release at the end of 2017. We have revisited and formalized *GeantV* user interfaces and helper protocols, allowing to: connect to user code, provide recipes to access efficiently MC truth and generate user data in a concurrent environment.

1. Introduction

Improving the CPU performance is a major objective of simulation R&D and the related program of work is connected to the LHC schedule for the high luminosity phase. Adapting the simulation workflow and algorithms to better profit from the FLOPS potential of modern architectures is very important. The R&D undertaken by the *GeantV* project to increase the data and instruction locality has shown [1] that

important performance improvements are within reach even in highly complex frameworks such as particle transport simulation.

The *GeantV* project aims at speeding-up particle transport simulation by a factor between 2 and 5 by recasting the traditional approach into a more cache and vector-friendly form, targeting instruction-level parallelism. Tracks are regrouped per-step into "baskets" collecting particles that go through a common processing phase, allowing to perform repetitive work and to use SIMD registers even for algorithms that are not internally vectorizable.

A first R&D phase revealed an important speed-up potential coming from geometry computation. This led to a vectorization-capable geometry package demonstrating excellent performance in both scalar and multi-particle modes. The scalar version of the *VecGeom* [2] package is now part of the official releases of *Geant4* [3], while the vectorization abstraction layer *VecCore* [4] developed in the context of *GeantV* was released as an independent package, adopted for vectorization by *ROOT* [5]. The major *GeantV* challenges after this first period of R&D were to extend the basket approach to other processing stages and to develop the first version of the electromagnetic physics package, preparing the field for a fully vectorized EM shower simulation.

A thorough review of the project goals revealed the need to expose *GeantV* to the community for early integration and testing with experiment frameworks, targeting high-luminosity LHC for the production phase. As a result, *GeantV* went through deep transformations touching the core scheduler, the interfaces and the physics package, aiming to deliver an alpha version at the end of 2017 and a beta version in 2018. The following sections briefly describe these transformations and the general features to be expected in the first release, and on-going R&D works expected to be delivered in 2018.

2. A generic vector scheduling framework

Making use of the SIMD pipelines available today even in commodity PC's has been already investigated in *GeantV* [6]. An important lesson resulting from this R&D is that vectorizing only some of the computing intensive algorithms is not sufficient for enabling significant overall gains from the SIMD pipelines. The entire simulation workflow needs to be optimized to sustain a low-overhead continuous vector data flow. *GeantV* prototyped and put in production an approach that splits the stepping procedure for tracks into stages, to accumulate several particles before actually performing the actions involved by each stage.

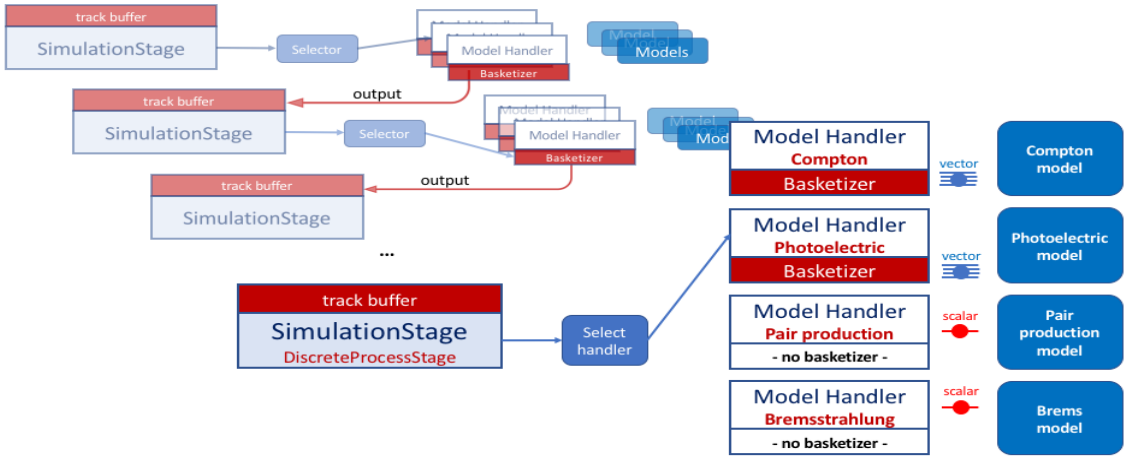

Figure 1 The GeantV stepping procedure decomposed in stages. Each stage loops over input tracks, selecting among a set of models corresponding to the stage. Tracks may be either "basketized" with others and sent to vectorized algorithms, or processed by the selected model in scalar mode. This approach improves the locality and cache coherency compared to the classical stack-based approach.

As illustrated in Figure 1, the generation of the final states from discrete physics processes is an example of a simulation stage, but the same mechanism is suitable for propagation in magnetic field or geometry queries. Particles entering a stage are directed to different physics model handlers that accumulate suitable tracks into baskets, executing the different model algorithms in multi-particle mode.

This kind of design changes the traditional stack-based execution into a multi-stream execution pattern. It creates the premises for increasing code and data locality while allowing to feed multiple data to vectorized code.

The new version of the *GeantV* scheduler (version 3) corrects many of the drawbacks of the previous implementations. The memory management has been fully re-written to prevent bloating due to event mixing in the most challenging conditions: high number of threads, geometry of large complexity or low production cuts producing many secondary particles. The implementation uses a special buffer that prioritizes the transport of particles of older generation, allowing to "consume" previous particle showers before generating and transporting new ones. Figure 2 shows the maximum resident memory as function of the number of threads in the case of a complex geometry setup and in the case of very large EM showers. Even if the scalability of the new scheduler is not improved much, the single-threaded *GeantV* applications are 40%-80% faster now due to the increased locality induced by the generalized vector flow, as shown in Figure 3. Vectorization benefits are not considered in this comparison and they are expected to increase the performance gain when available for physics and field propagation.

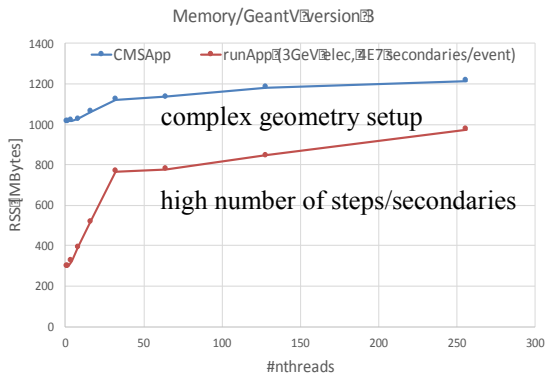

Figure 2 Resident Set Size (RSS) vs number of threads. Memory in GeantV scheduler (version 3 is controlled by applying a policy prioritizing the transport of older generation secondary particles, while re-using the containers released by completed tracks.

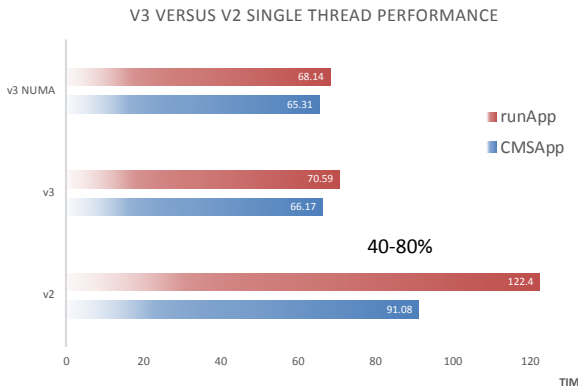

Figure 3 Single thread performance improvements compared to the previous version, induced by increased locality. Tests are shown for both a simple sampling calorimeter (upper) and a realistic CMS setup (middle), compared to previous version (v2)

The coming *GeantV* prototype release will be "hardware topology"-aware. The increase of the number of cores on modern hardware leads to more complex cache and memory hierarchies (Non-Uniform Memory Architecture or NUMA) with un-avoidable extra costs for non-local data access. Programs having frequent cache misses are more penalized by non-uniform memory accesses than the others. *GeantV* can run NUMA-aware on demand, activating a topology discovery service based on the *hwloc* library [7]. The application can configure its concurrency settings to deploy separate propagators on each locality node, which uses threads and data pinned to the node and minimizes the need of data transfers among different propagators.

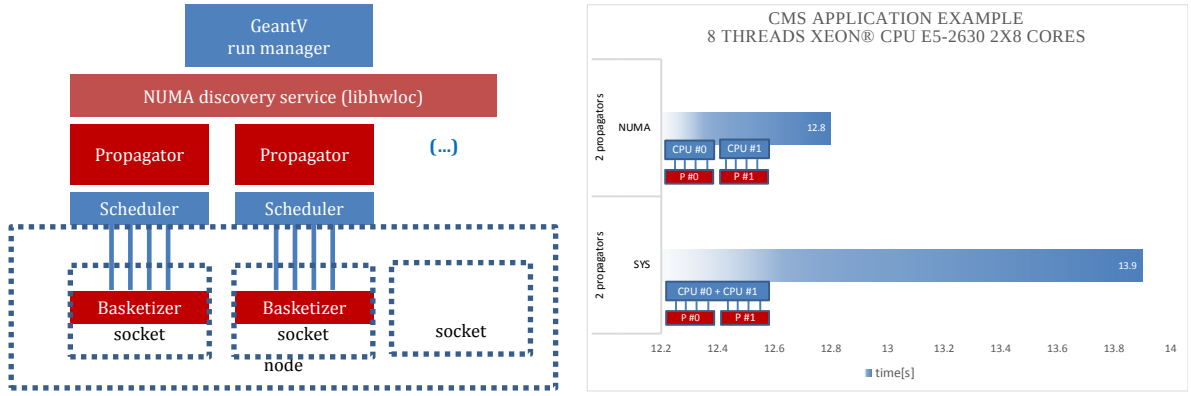

Figure 4 Left: schematic mapping of GeantV track transport services to machine topology. Right: Benefits of deployment of a number of propagators matching the number of locality nodes (sockets) of the machine. The alpha release of *GeantV* can be configured NUMA-aware, detecting the hardware topology and confining the data and processing flow to local resources as much as possible.

As shown in Figure 4, each propagator is pinned to a node, and it behaves almost like a separate process on its own locality, while different propagators are work-balanced by a single event server service. Compared to the scheduling done by the OS, this mode can bring benefits of up to 10% or more depending on the hardware.

3. GeantV physics in the alpha release

The development of electromagnetic showers in calorimeters plays a major role for the CPU performance of simulations. This is also true for jets because hadronic showers, induced by the hadron components of jets, have electromagnetic components coming from the decays of neutral pions. This implies that the development of vectorized EM physics models is one of the main areas where the effort should be focused in the short and medium term for developing a high-performance detailed simulation.

An important objective of the *GeantV* physics development is therefore to review the physics algorithms and their implementation in order to improve their accuracy and better adapt them to a multi-particle flow. Most of the EM physics models in *GeantV* were revisited with respect to the theory and rewritten with state-of-the-art implementations. The resulting improvements were adapted also to *Geant4*, which is of utmost importance for both validating and making available the new *GeantV* developments to the community.

The *GeantV* strategy is to first provide scalar versions having vectorizable data structures and general helpers (such as sampling methods), then to proceed with the vectorization of these models one-by-one, offering a testbed where the scalar and vector versions can be switched on demand. This will also facilitate the validation of the new developments and the benchmarking of their performance in a basket flow environment.

The alpha release will deliver vectorized geometry, but mainly scalar physics fully describing EM showers. The different *GeantV* models were tested and validated by test applications implemented in both *GeantV* and *Geant4*. Tests are showing results matching at per mil level for sensitive observables, such as visible energy per primary, number of steps or number of produced secondary particles of different types. Table 1 presents such a comparison for electron-induced showers in the ATLAS simplified calorimeter, while Figure 5 presents the validation of different options of the multiple scattering model in two different setups. The available physics models in the alpha release are: Bremsstrahlung, Ionisation and Coulomb scattering (with multiple scattering) for electrons/positrons and Compton, Conversion and Photoelectric for gammas, in the range [100eV-100TeV].

Table 1 Total energy deposited and cumulative track length for electron-induced showers in simplified ATLAS sampling calorimeter. The corresponding GeantV and Geant4 simulation results are matching within statistical errors.

10^5 1 [GeV] e- in ATLAS bar. simpl. cal. : 50 layers of [2.3 mm Pb + 5.7 mm lAr]; p.cut = 0.7 [mm]

	e^-/e^+: ionisation, bremsstrahlung, msc; γ: Compton, conversion							
	GeantV				Geant4			
material	E_d[GeV]	rms [MeV]	tr.l. [m]	rms [cm]	E_d[GeV]	rms [MeV]	tr.l. [m]	rms [cm]
Pb	0.69450	15.198	51.015	1.189	0.69448	15.234	51.016	1.192
lAr	0.22792	14.675	106.11	7.592	0.22796	14.656	106.13	7.582

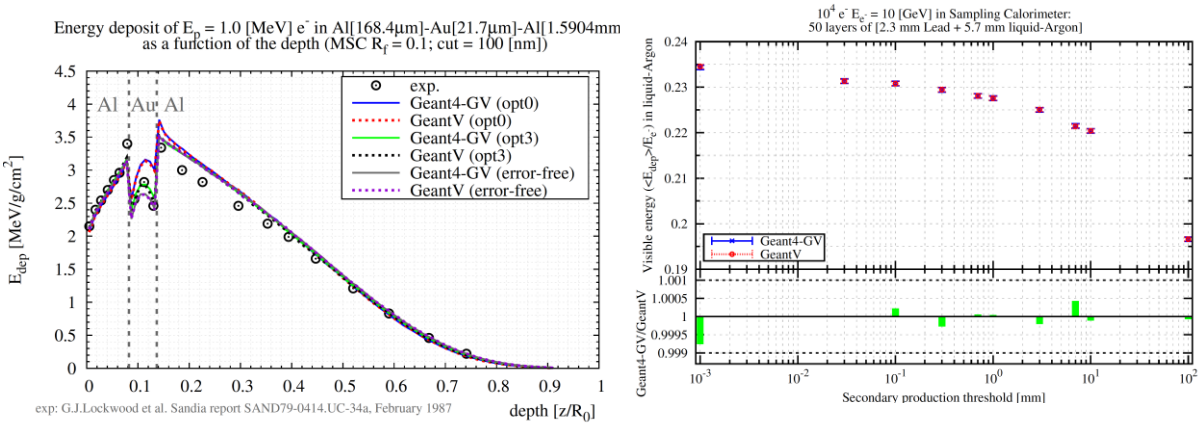

Figure 5 GeantV EM physics validation against Geant4 10.4 beta. Left: Energy deposit depending on depth for 1 MeV electrons in multi-layered Al-Au-Al target. Different options of the multiple scattering model have been compared for GeantV, Geant4 and data. Right: visible energy per primary for 10 GeV electrons in liquid-Argon for multi-layered sampling calorimeter. The Geant4-GV physics list matching the physics settings used in GeantV was consistently used in Geant4-based simulations.

4. User interfaces and examples

The *GeantV* recast of the simulation approach has several practical implications, several of which are affecting the user code. For instance, track mixing from different events is important for sustaining the basketized flow, and this will affect the user scoring methods. It is interesting to note, however, that several current or planned experiment frameworks will also support track-level parallelism. This will affect the data management and I/O on the user side, requiring concurrent bookkeeping and management of multiple events in flight. Extending the user API with vector signatures will not affect existing scalar user code, but it will allow new vectorized user code to exploit the particle-level parallelism offered by the framework.

The alpha release will introduce a set of interfaces defining the interaction with the user application, conceptually very similar to the *Geant4* ones. The user application will still be notified at the beginning/end of the run/event/primary tracks, and will be able to score in sensitive detectors, using similar methods as before. Some new paradigms derived from running concurrently a limited number of events will have impact on the data management. The signal produced in sensitive elements by different threads needs to be merged per event before being digitized. The user scoring data will need to have thread local state to prevent expensive locking required by thread safety.

GeantV provides on-demand services to deal with the new concurrency paradigm and ease the transition of user applications to the new environment. A series of simple to more complex examples demonstrating the complete workflow will be delivered with the first release. There will also be examples of the integration of *GeantV* with user applications providing their own event loop. We are targeting a first level of integration with task-based frameworks such as *CMSSW* [8] and *GAUDI* [9].

5. A ML approach for integrating fast simulation

The required increase in CPU performance for the LHC high-luminosity program cannot be achieved only with algorithmic improvements exploiting SIMD, structural changes and multi-particle flow. It is also necessary to develop fast sampling algorithms as well as fast simulation approaches that replace the simulation of full showers. The project aims to investigate further the ways in which full and fast simulation capabilities can be integrated in a single detector simulation framework.

The target is to integrate a simple example in the alpha release demonstrating fast simulation of a calorimeter using ML-based inference. This is the result of an ongoing R&D [10] testing a generic Generative Adversarial Networks (GAN) [11] based approach via computer vision techniques using as input 3D calorimeter data. The initial tests are able to reproduce the shower profiles and single cell response for mono-energetic particles coming from the same direction, in the 25x25x25 cells calorimeter (Figure 6).

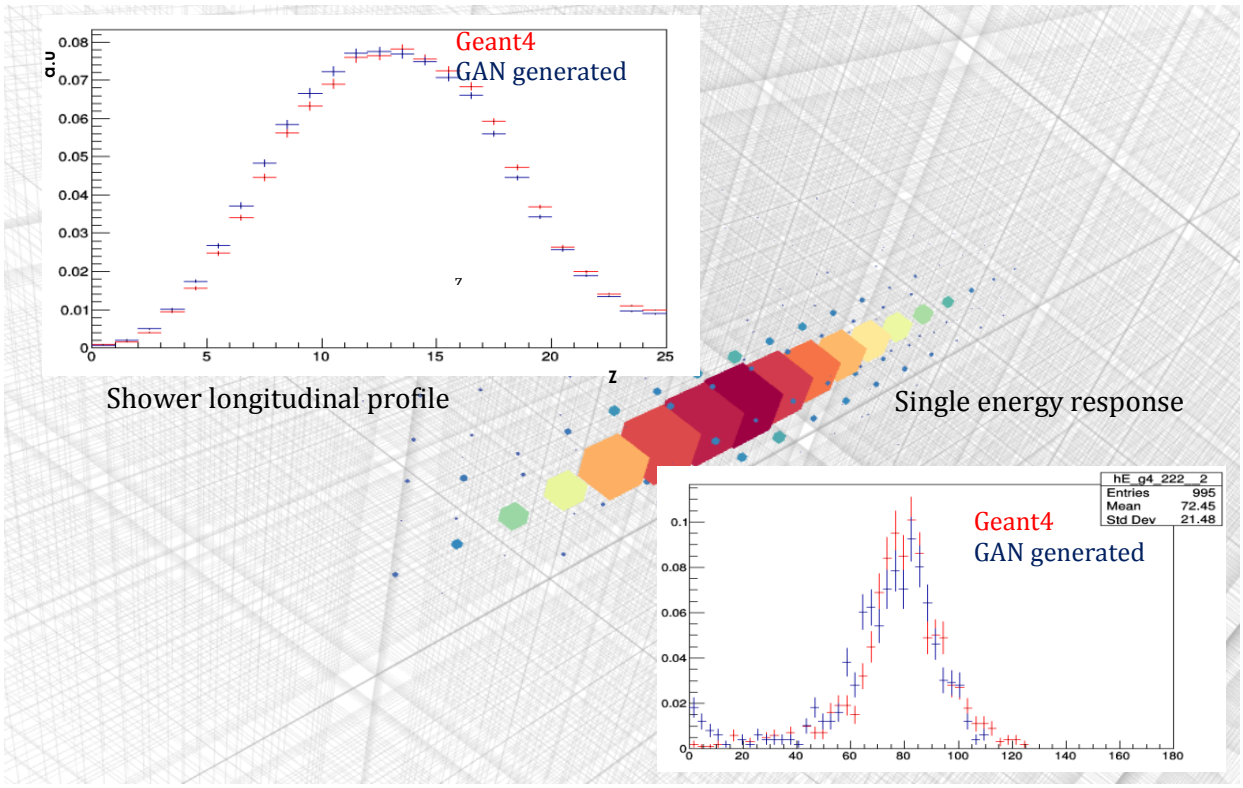

Figure 6 GAN approach for fast simulation of the 25x25x25 cells CLIC ECAL calorimeter. Shower profiles and single cell response for the trained network describe well the input data.

After a first phase trying to understand the limitations of such approach (such as training time versus re-usability of the trained network), we will try to generalize it, using multi-objective regression (momentum and angular distributions, particle type, …). The next phase aims at generalizing this approach to other types of detectors in an adaptive manner, optimizing for example the network topology for the problem to solve.

6. Conclusions

After a first phase of intensive R&D, the *GeantV* prototype has tagged an alpha release [12]. This version delivers only part of the design performance, but allows the community to have a first look at a new technology featuring track-level parallelism. This will benefit both *GeantV* and the applications using it. The alpha tag features the specific *GeantV* interfaces, and several examples of different degrees of complexity demonstrating the complete workflow, each one having a correspondent *Geant4* application.

The alpha release features full EM physics for electrons, positrons and gamma particles in scalar mode, demonstrated in different concurrency scenarios, using either internal event loop and static threads or external event loop in a task-based approach.

We expect a very close collaboration and feedback from experiments during 2018, including early tests of framework integration with *GeantV* for the beta release at the end of the year. The beta release is expected to feature full *GeantV* performance for vectorized EM physics, and a production-quality geometry.

Acknowledgements

The authors wish to acknowledge the contribution of Intel to the *GeantV* project through the Intel Parallel Computing Centre (IPCC) program. We also want to acknowledge the very useful technical contributions of CERN openlab, which allowed performing several studies and performance optimizations.

References

[1] J Apostolakis et al 2015 Adaptive track scheduling to optimize concurrency and vectorization in GeantV J Phys: Conf Ser **608** 012003

[2] J Apostolakis et al 2015 J. Phys.: Conf. Ser. **608** 012023

[3] S Agostinelli et al 2003 *Nuclear Instruments and Methods* **A 506** (53pp)

[4] G Amadio et al 2017 Speeding up experiments software with VecCore, a portable SIMD library *ACAT 2017*

[5] ROOT project, "ROOT" [software] version 6.13/02, 2018. Available from https://root.cern.ch/content/release-61302 [accessed 2018-04-09]

[6] Apostolakis J, Brun R, Carminati F and Gheata A 2012 *J.Phys: Conf. Ser.* **396** 022014 http://iopscience.iop.org/1742-6596/396/2/022014

[7] Portable Hardware Locality project, "hwloc" [software], version 2.0.1, 2018. Available from https://www.open-mpi.org/projects/hwloc [accessed 2018-04-09]

[8] CMSSW project, "CMSSW Application Framework" [software], 2018. Available from https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookCMSSWFramework [accessed 2018-04-09]

[9] Gaudi project, "Gaudi" [software] version v30r2, 2018. Available from http://gaudi.web.cern.ch/gaudi [accessed 2018-04-09]

[10] S Vallecorsa et al 2017 A Machine Learning tool for fast simulation *ACAT 2017*

[11] Goodfellow I J et al 2014 Generative Adversarial Networks. *ArXiv e-prints*: 1406.2661

[12] GeantV project, "GeantV" [software] version 0.3.0, 2018. Available from https://gitlab.cern.ch/GeantV/geant/tree/alpha [accessed 2018-04-09]