The new ATLAS Fast Calorimeter Simulation

J Schaarschmidt, on behalf of the ATLAS collaboration
The University of Washington in Seattle, Washington 98105, USA
E-mail: jana.schaarschmidt@cern.ch

Abstract. Current and future need for large scale simulated samples motivate the development of reliable fast simulation techniques. The new Fast Calorimeter Simulation is an improved parameterized response of single particles in the ATLAS calorimeter that aims to accurately emulate the key features of the detailed calorimeter response as simulated with Geant4, yet approximately ten times faster. Principal component analysis and machine learning techniques are used to improve the performance and decrease the memory need compared to the current version of the ATLAS Fast Calorimeter Simulation. A prototype of this new Fast Calorimeter Simulation is in development and its integration into the ATLAS simulation infrastructure is ongoing.

1. Introduction
The successful physics program of the ATLAS experiment [1] at the Large Hadron Collider (LHC) [2] strongly depends on simulated Monte Carlo (MC) samples that accurately predict the detector response. Increasing integrated and instantaneous luminosity resulting in larger number of multiple-interactions (pile-up), as well as limited computing resources, encourage the development of fast simulation techniques. It is likely that in the future the bulk production of simulated MC samples will be done using fast simulation. In ATLAS, about 90% of the simulation time with the GEANT4 [3] program is spent in the calorimeters. This is due to the complex shower development of an incoming primary particle, during which many additional particles are created. The ATLAS Fast Calorimeter Simulation (FastCaloSim) [4] was developed to provide a parameterized calorimeter response while decreasing the simulation time in the calorimeters by about one order of magnitude. An improved version of FastCaloSim is in development, and its main components and principles are described in this proceedings. A short outline of the ATLAS simulation infrastructure is given in Section 2. The current FastCaloSim is described in Section 3. The improvements of the new FastCaloSim are discussed in Section 4, and Section 5 is a summary.

2. The ATLAS Simulation Infrastructure
The ATLAS simulation infrastructure [5] is used to produce and validate the samples to be used in physics and performance studies. The production chain consists of the following steps:

- Event generation: Various MC generators are available to produce hard scatter or pile-up events and decay most of the emerging unstable particles. The event record is based on the common HepMC format.
• Detector simulation: The Geant4 toolkit is used to step through the various detector volumes and simulate the interactions of particles with the material according to the numerical models provided by the physics lists. It is also referred to as full simulation. Fast simulation can replace Geant4 in defined detector parts. The output is stored in the HIT\(^1\) format. Pile-up events typically undergo the full simulation and are then overlayed with the hard scatter event at the hit level.

• Digitization: In this step, the hits are converted into digits, which is the same format as that of the real detector. Typically, a digit is produced when a voltage or current on a readout-channel exceeds a given threshold during a time window. Some detector parts also include the signal shape as a function of time. Detector and electronics noise and cross talk effects are modelled as well. The output is raw data objects (RDO).

• Reconstruction: At the final step the digits are converted to real physics objects such as jets, muons, electrons, photons, etc., using various algorithms, such as jet finders, tracking or clustering software.

All the steps of the production chain undergo validation to ensure their correct functionality, performance and robustness.

The ATLAS integrated simulation framework (ISF)\(^6\) was developed to provide a flexible configuration of detailed or fast tools at every step of the chain. It also allows to use different simulators for different particles of the same event. This approach can be tailored to the specific needs of the various physics analyses. If all parts of the chain are substituted by fast techniques, then this is referred to as the Fast Chain\(^7\). With the ISF and the Fast Chain it is possible for example to process the hard-scatter with Geant4 and pass it to full digitization and reconstruction, while the pile-up events undergo fast simulation and fast digitization methods and finally a truth-assisted fast reconstruction. This approach is currently under validation.

3. Fast Simulation

While various algorithms for fast simulation exists (see\(^5\)), only the most relevant ones are described here. In general, fast simulation is faster, less precise and easier to tune to data than the full simulation.

Frozen-showers\(^8\) is a technique that replaces low-energetic particles from electromagnetic (elmag) shower produced by Geant4 by pre-simulated showers stored in a library. For the MC production for Run-2, it is used by default in the forward calorimeters (FCAL) in the full simulation.

Fatras\(^9\) is a fast ATLAS track simulation, that employs the simplified reconstruction geometry rather than the simulation geometry, and uses an extrapolation engine to transport the track through the detector.

FastCaloSim\(^4\) is a parameterized calorimeter response, based on the Geant4 simulation of single particles, namely electrons and photons (representing elmag showers) and charged pions (representing hadronic showers). The single particles are generated in a fine grid of energies and eta\(^2\). Then, in the simulation step, a fitting parametrization is loaded into memory and the response (such as energy or energy density around the shower axis) is randomly sampled from that.

\(^1\) A hit object is a record of the deposited energy with time and position.

\(^2\) ATLAS uses a right-handed coordinate system with its origin at the nominal interaction point (IP) in the centre of the detector and the z-axis along the beam pipe. The x-axis points from the IP to the centre of the LHC ring, and the y-axis points upwards. Cylindrical coordinates \((r,\phi)\) are used in the transverse plane, \(\phi\) being the azimuthal angle around the z-axis. The pseudorapidity is defined in terms of the polar angle \(\theta\) as

\[
\eta = -\ln\tan(\theta/2).
\]
The FastCaloSim parametrization is split into the longitudinal and lateral shower developments. In the longitudinal direction, the energy deposited in each layer is of interest. The energy distribution is stored for each calorimeter layer in 2D histograms in ten bins of the shower depth (distance of the energy deposit from the calorimeter front face). The correlations of the energy deposit between the various layers are stored in correlation matrices. In the lateral direction, the energy density around the shower axis needs to be modelled. This is achieved by fitting the energy density with a radially symmetric, third order polynomial function, which is slightly modified to take into account asymmetries in case the particle is not perpendicular to the calorimeter surface. With this approach, FastCaloSim models well the average shower shape but has problems to describe the sub-structure of jets and is therefore not useable in analyses selecting boosted objects, as discussed in [10].

FastCaloSim is widely and successfully used in ATLAS. During the so-called MC15 campaign of the first year of Run-2, about three billion MC events were processed or re-processed with AtlFast2, while about 14 billion events were (re-)processed with the Geant4 simulation. AtlFast2 combines the full simulation in the inner detector and the muon spectrometer with the FastCaloSim. AtlFast2F utilises Fatras and FastCaloSim, but it uses the full simulation in the muon system. A comparison of averaged simulation time for full simulation and fast simulation programs is given in table 1. FastCaloSim reduces the simulation time by about a factor 10, and the additional use of Fatras reduces the simulation time by another factor 10, i.e., a factor 100 compared to full simulation.

Table 1: Averaged simulation time of physics processes in seconds, as obtained with various ATLAS software tools, taken from [5].

| Process          | Geant4 | AtlFast2 | AtlFast2F |
|------------------|--------|----------|-----------|
| Minimum Bias     | 551    | 31.2     | 2.13      |
| ttbar            | 1990   | 101      | 7.41      |
| Jets             | 2640   | 93.6     | 7.68      |
| Photons and jets | 2850   | 71.4     | 5.67      |
| W → eν           | 1150   | 57.0     | 4.09      |
| W → μν           | 1030   | 55.1     | 4.13      |

4. The new FastCaloSim
4.1. Energy parametrization
Along the longitudinal shower direction, energy is deposited in the various layers of the calorimeter that are hit by the shower, and the sum of those deposits will be referred to as the total deposited energy. The energy distributions are obtained from the Geant4 simulation, and they are converted into cumulative distributions in order to get a well-defined assignment of cumulative value vs. the deposited energy, expressed as the fraction of the total energy.

The energy deposits per layer are correlated with each other. If, for example a large amount of energy is lost in the first layers, less will be deposited in the layers behind. In order to decorrelate those energy deposits, a principal component analysis (PCA) is used. A PCA is a transformation of a set of variables into a set of orthogonal and uncorrelated, so-called principal components. The first component has the largest variance. In order to achieve a better decorrelation, the events are divided into bins of the first (and/or second) component. The bins have approximately the same number of events, and typically a number of bins between 5 and
The information that needs to be stored is the cumulative energy distribution in each PCA bin and the PCA matrix (coming from the second PCA transformation) in each PCA bin. Those have to be loaded into memory for the simulation later on; it is therefore desired to keep the storage as small as possible. For that, a neural-network based regression is used. The training and evaluation of the regression is done with a multi-layer perceptron (MLP), as implemented in the TMVA [11] package. An iterative MLP training is employed, starting from two neurons (i.e., hidden layers), up to ten neurons. After each training, the regression is evaluated and the agreement with the original distribution (from GEANT4) is tested. If the largest relative deviation between input and output is less than 5%, the iteration stops and the MLP weights are stored. An example of this iteration is displayed in figure 1. If the regression fails, the histogram holding the cumulative energy distribution is rebinned iteratively and the bin contents and bin borders are stored instead.

Finally, in the simulation step, for each PCA bin a set of uncorrelated random numbers is drawn (one per calorimeter layer), the inverse PCA transformation is applied, and the regression output is calculated from the MLP weights. The result is then a linear superposition of the deposited energy in each of the 5-10 PCA bins, and can be compared to the GEANT4 inputs. Two examples are displayed in figure 2. A good agreement of full and fast simulation is observed.

4.2. Shower Shape Parametrization

The shower shape is characterised by the energy density around the shower axis. Since it is a radially symmetric problem, the variables in which the shape is described are an angle $\alpha$ (around the axis) and a radius $r$ (distance from the axis). The energy density is then defined as the amount of energy deposited by the hits per area unit, in a given calorimeter layer, a given PCA bin, and for specific values of the particle energy and eta. The first step is to define a suitable binning of the energy density; this is achieved by requiring approximately the same number of hits per area unit, leading to eight bins in $\alpha$ direction, and a variable number of bins in the $r$ direction, whereas smaller bins are chosen for the radii closer to the shower axis, where

![Figure 1: Illustration of the iterative regression, here for the fraction of the total energy that is deposited by pions in the first layer of the Tile calorimeter. The upper panel shows the cumulative distribution of the energy fraction obtained from GEANT4 and the evaluation of various regressions, while the lower panel shows the difference between the regressions and the input distribution. With more neurons, the regression usually performs better. In this example, the iteration is successful with four neurons.](image-url)
Figure 2: The output of the fast energy simulation (in blue) compared to the GEANT4 inputs (black dots). Figure (a) displays the fraction of the total energy deposited by pions in the third layer of the hadronic end cap calorimeter (HEC). Figure (b) shows the total energy deposited by photons, generated with an energy of 50 GeV. The p-value of a Chi2 test ($\chi^2$) and the probability of a Kolmogorov-Smirnov test (KS) are also displayed.

the bulk of the energy is distributed.

This 2D histogram is then used to train a regression, with the MLP method from TMVA. A illustration of the input (the energy density from GEANT4) to the output (the evaluation of the 2D regression) is displayed in figure 3. The example chosen is pions with an energy of 50 GeV, in the first PCA bin and in the first layer of the emag barrel calorimeter. Both plots show good agreement, proving that the method works. In the next step, which is currently being worked on, the hit distribution is sampled from the MLP weights. The aim is to store only the MLP weights for each shower shape, as it is done for the energy parametrization.

Determining the optimal parameters of the neural network (especially the number of neurons used) can be time consuming, and sometimes the regression will not give good results with a sufficiently small number of weights to be stored. Therefore, additionally a simple lateral shape modelling method has been developed for cases where less precise shape information is needed, such as low energetic particles. In that simple approach, a 1D Gaussian function is fit to the core of the hit distribution around the shower axis, after projecting it to the x- and y-axis. In the simulation step, random numbers are sampled from the two 1D functions. The core of the hit distribution can be well approximated, but the tails are not well described.

4.3. Hit-to-Cell Assignment

Once the energy and the shower shape have been modelled, and a hit distribution is simulated, the hits need to be assigned to the calorimeter cells. The cell container is then eventually passed to the digitization step.

The simplified geometry tool is used to assign the hit positions to cells. This tool neglects the accordion shape of the electrodes in the Liquid Argon calorimeter (which assures a crack-
Figure 3: Energy density in bin of $\alpha$ and $r$, presented in the x-y plane. Figure (a) shows the distribution obtained from Geant4, used to train a regression. Figure (b) displays the evaluation of the regression, which approximates well the input.

less symmetry in phi), and assumes that cells are cuboids. In Geant4 this accordion structure is included. In FastcaloSim, the accordion shape is emulated by applying a hit replacement probability function, also called \textit{wiggle function}. This function slightly displaces the hit in phi direction before the geometry tool is called. Only if the hit is in the center of given cell, the probability that the hit is assigned to that cell is one. If the hit is positioned toward the edge of a cell, there is a non-zero probability that the hit is assigned to the neighbouring cell. The parameters of this hit replacement depend on the calorimeter layer, and they are tuned to Geant4.

In figure 4 the effect of the hit replacement is illustrated. The plots show the ratio of the cell energy from Geant4 to the cell from the simplified geometry, as a function of the distance of the cell to the shower center in the eta-phi plane, for a sample of simulated charged pions. After the hit replacement is applied, the energy ratio is flat and no structure in phi is observed.

4.4. Prototype Development
The completion of the new FastCaloSim is ongoing. Soon a prototype version will be available that is integrated into the ATLAS software framework, Athena [12], and will be used for the next MC campaign (MC16) during Run-2. Extensive validation is planned, and also tuning to data is foreseen.

5. Summary
FastCaloSim is a parameterized calorimeter response, based on the inputs from the detailed simulation with Geant4, but approximately ten times faster. The improved FastCaloSim employs machine learning techniques to reduce memory need and to improve the modelling of the shower shape. Nested PCA transformations are used to decorrelate the energy deposits per layer, and to define a binning that is used for both the energy and the shower shape parametrization. The accordion structure of some of the calorimeter parts is emulated with a dedicated hit replacement. The prototype version is in development.
Figure 4: The ratio of the cell energy from GEANT4 to the cell from the simplified geometry tool, (a) before, and (b) after the application of the hit replacement. The ratio is shown as a function of the distance of a cell to the shower center for pions in the second layer of the elmag barrel calorimeter. The hit replacement emulates the accordion structure of the real detector, that would otherwise lead to over- and under estimation for the cell energy in phi direction, which is visible in figure (a), but not in figure (b).

References
[1] ATLAS collaboration: The ATLAS experiment at the CERN Large Hadron Collider, JINST 3 S08003, 2008.
[2] Evans, Lyndon and Bryant, Philip: LHC Machine, JINST 3 S08001, 2008.
[3] Agostinelli, S. and others: GEANT4 - A Simulation Toolkit, Nucl. Instr. and Meth. A 506, 2003.
[4] ATLAS collaboration: The simulation principle and performance of the ATLAS fast calorimeter simulation FastCaloSim, ATL-PHYS-PUB-2010-013, 2010.
[5] ATLAS collaboration: The ATLAS Simulation Infrastructure, Eur. Phys. J. C 70 823-874, 2010.
[6] Ritsch, E. on behalf of the ATLAS collaboration: The ATLAS Integrated Simulation Framework, ATL-SOFT-SLIDE-2013-114, 2013.
[7] Basalaev, A. on behalf of the ATLAS collaboration: The Fast Simulation Chain for ATLAS, ATL-SOFT-SLIDE-2016-617, 2013.
[8] Gasnikova, K. on behalf of the ATLAS collaboration: Frozen-shower simulation of electromagnetic showers in the ATLAS forward calorimeter, ATL-SOFT-SLIDE-2016-735, 2016.
[9] Edmonds, K. and others: The Fast ATLAS Track Simulation (FATRAS), ATL-SOFT-PUB-2008-001, 2008.
[10] ATLAS collaboration: Performance of the Fast ATLAS Tracking Simulation (FATRAS) and the ATLAS Fast Calorimeter Simulation (FastCaloSim) with single particles, ATL-SOFT-PUB-2014-01, 2014.
[11] Hoecker, A. and others: TMVA - Toolkit for multivariate data analysis, Proc. of Science PoS ACAT 040, 2007.
[12] ATLAS collaboration: ATLAS Computing Technical Design Report, ATLAS-TDR-017, CERN-LHCC-2005-022, 2005.