

COMPUTING METHODS IN HIGH ENERGY PHYSICS

S. Lehti

Helsinki Institute of Physics

Spring 2025

Fast simulation

Introduction

Many simulation tasks, such as extracting rare signal events from the expected background, require a generation of millions of background events, of which only a handful survives after a proper selection. If detailed detector simulation is too time consuming to be applied, while still needing some reasonable estimates for the detector response, the solution can be found by parametrizing the detector response. The parametrized detector response simulation is called fast simulation.

Parametrizing and smearing provide an approximate but very fast method to simulate the detector response. It is a useful tool for quick first

inspection of properties of a given physics channel with large statistics.

Single particle performance

The key parameters for tracker resolution for single particles are

- ▶ transverse momentum p_T
- ▶ azimuthal angle ϕ_0
- ▶ polar angle θ
- ▶ transverse impact parameter d_0
- ▶ longitudinal impact parameter Z_{ip}

The transverse impact parameter is defined as the closest point of approach to the impact reference point (primary vertex).

Fast simulation

The resolution is studied with full simulation as a function of track p_T and η . The resolutions are fitted as a function of η for each value of p_T using polynomial functions up to power five, which is needed to describe the most irregularly shaped η -dependencies. These functions are used to give the resolution for fixed values of η in order to have points of resolution as a function of p_T .

An approximate parametrization of the p_T dependence of the impact parameter and angular resolution is expressed as

$$\sigma_{ip,ang}(p_T) = (a_1 \oplus a_2/p_T) + a_3$$

Likewise, a first approximation of the p_T resolution dependence on p_T

is of the form

$$\sigma_{p_T}(p_T) = b_1 \oplus b_2 p_T$$

where the first term corresponds to the resolution due to the finite detector precision and the second term the uncertainty due to multiple scattering.

In order to improve the fits, one may use a "covariance" or cross term, e.g. for impact parameter

$$\sigma_{d_0}(p_T) = \sqrt{a_1^2 + a_4/p_T + a_2^2/p_T^2} + a_3$$

In the functions the coefficients are calculated for discrete values of η_j . The resolution for arbitrary η is found by linear interpolation (extrapolation) from two points of

Fast simulation

77

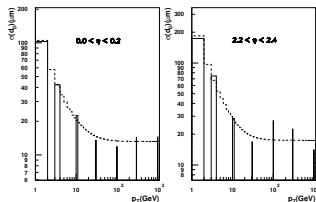


Figure B.1: An example of the transverse impact parameter fitted by the least squares method as a function of track p_T for two different η regions (central barrel and the edge of the end-cap).

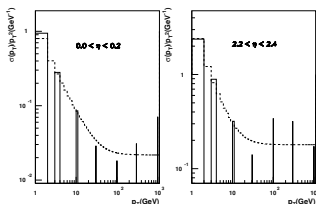


Figure B.2: The p_T resolution fitted as a function track p_T for two different η -regions.

$\sigma(\eta_j)$ with η_j 's closest to η , and used in smearing the ip.

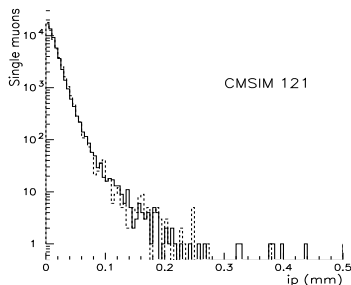


Figure: CMSIM (solid), and parameterized and smeared (dashed) ip.

Fast simulation

Particle showers

Calorimeters are described in terms of characteristic parameters such as energy resolution, position resolution, and shower separation, dynamical range and spatial coverage. A calorimeter can be homogeneous (like crystal calorimeter) or sampling calorimeter (sandwich structure with active and inactive layers).

The energy resolution of calorimeters is usually expressed in terms of $\Delta E/E$, ΔE being the overall statistical error including all error sources. At high energies, statistical contributions dominate. The relative resolution improves

with increasing E . The main tendency for $\Delta E/E$ is to follow the form k/\sqrt{E} where k depends on the calorimeter and readout characteristics. A general formula to get an approximate resolution is

$$\Delta E/E = \sqrt{A^2 + B^2} \quad (1)$$

where A is related to photon statistics in the photomultipliers, and B deals with the sampling fluctuations.

Containment describes the fraction of energy measured in the calorimeter. Energy not contained in the calorimeter may result in severe biases or simply be detrimental to the energy resolution.

Fast simulation

A detailed simulation of electromagnetic showers is computationally intensive. A parameterization of the spatial energy distribution of an electromagnetic shower, based on probability density functions allows speeding the simulation without compromising the simulation accuracy. The parameterized simulation is tuned to give the same results as the full detector response simulation.

CMS Fast Simulation

CMS Fast Simulation is a CMSSW-integrated tool to simulate and reconstruct events with the CMS detector, in view of doing physics analysis without being

penalized by either CPU time or disk space considerations, while still benefiting from an accurate simulation of the detector effects. The input of Fast Simulation is a list of particles from an event generator, propagated in the magnetic field to different layers of the various subdetectors, with which the particles may interact. The quasi-stable particles are allowed to decay.

The interactions simulated with Fast Simulation are

- ▶ electron bremsstrahlung
- ▶ photon conversion
- ▶ charged particle energy loss by ionization

Fast simulation

- ▶ multiple scattering
- ▶ electron, photon and hadron showering

The first 4 are applied to particles traversing the thin layers of the tracker, while the latter is parameterized in the electromagnetic and hadron calorimeters.

The muon simulation is mostly based on parameterization of resolutions and efficiencies.

As output, Fast Simulation delivers a series of "high-level objects", such as reconstructed hits for charged particles in the tracker layers and energy deposits in calorimeter cells, which can be used as inputs for the reconstruction algorithms.

The computer time needed to simulate an event in Fast Simulation is about 3 orders of magnitude smaller than that needed in the full simulation.

Tracker response

Fast Simulation uses a simplified tracker geometry, adequate for the required level of accuracy. The thickness of the active layers is tuned to reproduce the fully simulated number of bremsstrahlung photons above a certain energy threshold. After tuning, the total number of radiation lengths traversed in the tracker is in agreement with the full geometry.

While propagating in the magnetic

Fast simulation

field through tracker layers, charged particles experience multiple scattering and energy loss by ionization. The intersections between the modified trajectories and each tracker layer define the position of "simulated hits". Each simulated hit is turned with a certain efficiency to a "reconstructed hit". The position is obtained from a gaussian smearing of the simulated hit position. The gaussian resolution used in the smearing process is obtained from the full simulation for each silicon strip and pixel layer. The detailed procedure was developed in view of reproducing the b-tagging performance with requested level of accuracy.

Calorimeter response to e/γ

The showers of electrons and photons which impinge on the ECAL are simulated in 2 steps. In the first step, the shower is developed following a parameterization as if ECAL were a homogeneous medium. This approximation is realistic since the CMS ECAL is made of contiguous crystals. The deposited energy is integrated over longitudinal slides including uncertainties due to the limited photo-statistics and the longitudinal non-uniformity in the crystals.

In the second step, the energy spots in each slide are distributed in space and placed into the actual crystal

Fast simulation

geometry. The time used in this step is kept to a reasonable value by limiting the 2D spot-crystal assignment to a small 7x7 crystal grid in a plane perpendicular to the shower longitudinal component. The energy collection is then simulated taking into account energy leakage, losses due to gaps in the geometry, and shower enlargement due to the magnetic field.

When all electrons and photons are processed, the electronic noise is simulated and zero suppression applied, and a list of reconstructed hits is built. The fast and full simulations agree at the level of per mil in the barrel region, and at the level of per cent in the endcaps, for

energies ranging from 1 GeV to 1 TeV.

Calorimeter response to hadrons

Charged and neutral hadrons are propagated to the ECAL, HCAL and forward calorimeter entrance after their interactions with the tracker layers. Their energy response is derived from the full simulation. Single charged pions are fully simulated for p_T values of 2, 5, 10, 30, 50, 100 and 300 GeV, uniformly distributed in pseudorapidity between -5 and 5. The reconstructed energy is collected in 5x5 HCAL tower matrices and in the corresponding 25x25 ECAL crystals. The energy distributions are then sliced into η bins. These

Fast simulation

distributions are fitted to a gaussian, the mean value and the sigma of which are tabulated as a function of the energy and pseudorapidity, used in turn to smear the hadron energy response in the fast simulation.

Linear interpolation and extrapolation is used.

Other hadrons are simulated as if they were pions with the same transverse energy. The agreement between fast and full simulations is satisfactory.

Response to muons

The response of the muon chambers is simply parameterized to reproduce the efficiencies and resolutions of the full simulation. At the moment,

the detailed particle propagation stops at the entrance of the calorimeters. The muon interactions with calorimeters is simulated much as the pion interactions. The fast muon simulation results agree within 1.5% with the full simulation results.

Usage

Fast Simulation source code is available as part of the CMSSW code. The documentation can be found under the link

<https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBook>

-> *6.5 Simulating and Reconstructing events with Fast Simulation*

Unfortunately the above link needs a

Fast simulation

password. There is a copy of the examples in

http://www.mv.helsinki.fi/home/slehti/ComputingMethodsInHEP/CMSPublic_SWGuide

FastSimulationExamples.pdf

Start Fast Simulation simulation by creating the working area. First the environment needs to be set to get access to scram. We can take the latest available version on kale by

```
scram p CMSSW CMSSW_10_6_23
```

```
&& cd CMSSW_10_6_23/src
```

The GEN-fragments for official CMS production are located in the git directory

Configuration/Generator

In addition one needs to check-out any code which is called by the GEN-fragment, like the undelying

event parameters. The configuration file for CMSSW is generated from the GEN-fragment with `cmsDriver.py`:

```
cmsDriver.py GENfragment_cfi.py
-s GEN,SIM,RECOBEFMIX,DIGI,L1,L1Reco,
RECO,HLT:@frozen25ns
--fast --conditions auto:run2_mc
--eventcontent AODSIM
--beamspot NominalCollision2015
--era Run2_25ns
--magField 38T_PostLS1 --datatier AODSIM
-n 10 --no_exec
```

```
cmsRun
GENfragment_cfi.py_GEN_SIM...HLT.root
```

Even if the GEN-fragments available in the repository do not contain your favourite channel, they can be used as working examples.

In Fast Simulation the event generator input can be generated and fed to the analysis on the fly or

Fast simulation

one can store the events in ROOT files for later analysis with a separate analysis program.

There are example config files in the directory

FastSimulation/Configuration/test

which can be customized for user's needs by making some modifications. One can change for example the average number of pile-up events, simulate (or not) hits in the tracking system and calorimetry, etc: ...

```
process.maxEvents=  
    cms.untracked.PSet(input=cms.untracked.int32(5))  
...  
# If you want to turn on/off pile-up  
process.famosPileUp.PileUpSimulator.averageNumber=5.0  
# If you want to choose simulated detectors  
process.famosSimHits.SimulateCalorimetry =  
True  
process.famosSimHits.SimulateTracking = True  
# process.famosSimHits.SimulateMuons = False  
...
```

Remember that fast simulation is not a full simulation, it gives more approximate results, and it cannot always be used. Sometimes it is, however, the only practical option, and often a very good starting point for a “first look” in a new physics channel.