GeVSim Event Generator

Sylwester Radomski
GSI, Darmstadt

S.Radomski@gsi.de
Sylwester.Radomski@cern.ch

Content

- Motivation
- Features
- Event-by-Event capabilities of GeVSim
- Implementation
- Quality Assurance
- Web page survey

Motivation

- GeVSim is a simple Monte-Carlo event generator
  - 2 (two) classes, 400 lines of the (good) code
- Applications of GeVSim
  - testing algorythms
  - detector performance mesurement
- GeVSim is based on MevSim definition but do not inherit the code
- Revriten to Root/C++ to avoid
  - unnecesary interfaces (TMevSim)
  - problems in stack size (in big multiplicities)
  - 'private' random number generation
- Improved flexibility (event-by-event)

Features

- Partices in GeVSim are generated from a thermal distribution
  - no dynamics
  - parameters of a distribution are set explicite by the user
- Momentum distribution in (pT,Y)
  - user defined plugged by TF2 named "gevsimPtY"
  - four generic distribution (the same as in MevSim)
- Azimuthal distribution
  - deconvoluted form (Pt,Y) distribution
  - Direct and Eliptical flow (more to come)
- Carefuly tested, documentation in development

Momentum distribution

- In first generic model transverse momentum and rapidity is deconvoluted
- Formula

$$\frac{d^2N}{dp_T dy} = Ap_T \mathrm{e}^{-m_T/T} \mathrm{e}^{-y^2/2\sigma_y^2}$$

- Parameters
  - multiplicity
  - invers slope parameter (sometimes called temperature)
  - rapidity width distribution

Momentum distribution

- Second and Third model – termalised ball
- Formulae

$$\frac{d^2N}{dp_T dy} = Ap_T E \mathrm{e}^{-E/T}$$

$$\frac{d^2N}{dp_T dy} = \frac{Ap_T E}{\mathrm{e}^{E/T} - 1}.$$

- Parameters
  - multiplicity
  - invers slope parameter

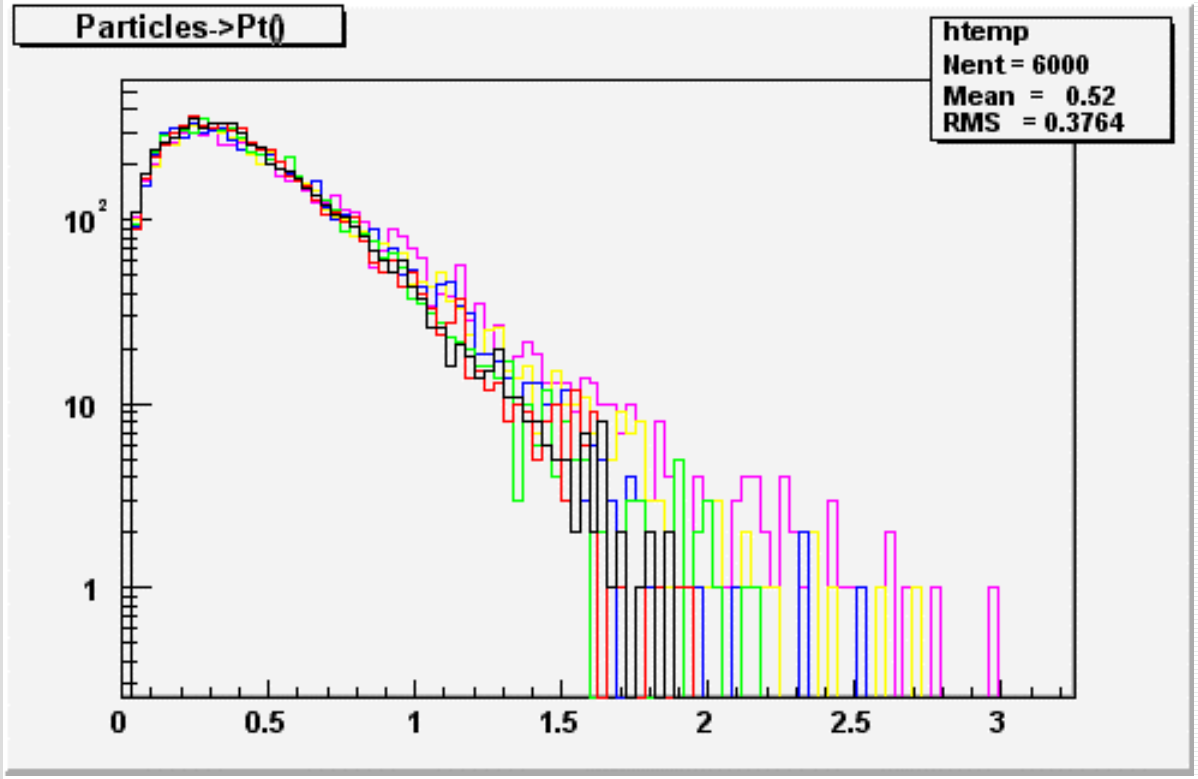
Expansion

- In first generic model transverse momentum and rapidity is deconvoluted
- Formula

$$\frac{d^2N}{dp_T dy} = Ap_T E \mathrm{e}^{-\gamma E/T} \times \left[\frac{\sinh(y_p)}{y_p} + \frac{T}{\gamma E}\left(\frac{\sinh(y_p)}{y_p} - \cosh(y_p)\right)\right]$$

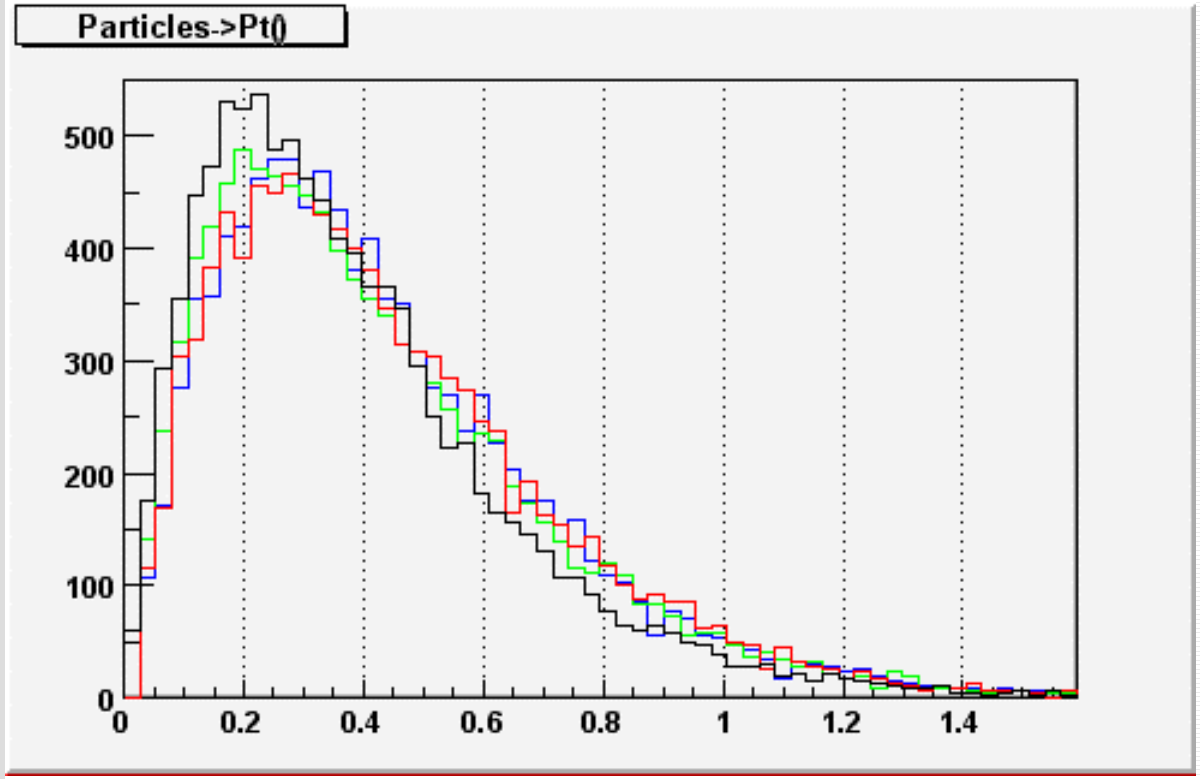
- Parameters
  - multiplicity
  - invers slope parameter (sometimes called temperature)
  - expansion velocity

Expansion spectra



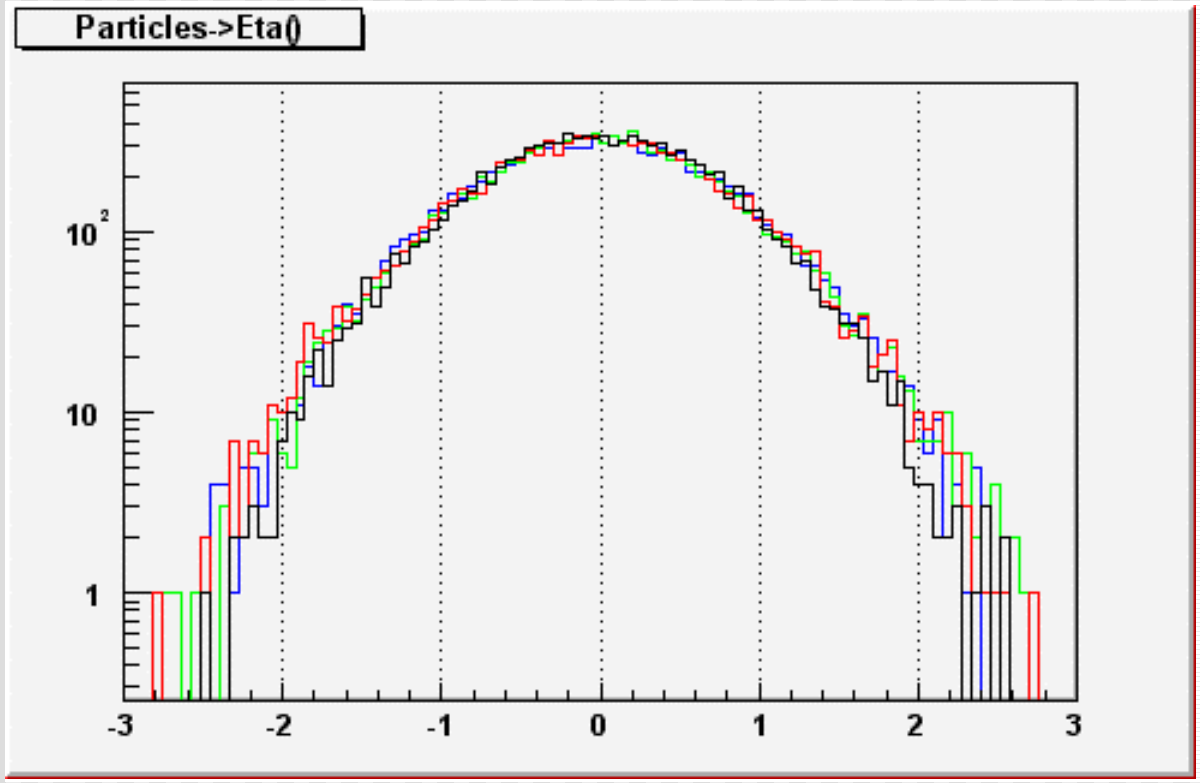
Sylwester Radomski
offline week, CERN

Spectrum comparison



Sylwester Radomski
offline week, CERN

Spectrum comparison



Azimuthal distribution

- Azimuthal is described with the use of Fourier transformation
  - first coeficient – direct flow
  - secodn coeficient – elliptical flow

- Formula:

$$\frac{1}{2\pi p_T}\frac{d^2N}{dp_T dy}\left[1+\sum_{n=1}^{6}2V_n(p_T,y)\cos[n(\phi-\Psi_R)]\right]$$

- Fourier coeficients defined on the particle type level
  - independent of the momentum
  - this will evolve

Parameter determination

- Parameters of distributions are determined on the event-by-event basis
  - determination with the use of named functions (TF1)
  - discovery of formula by gRoot->GetFunction("gevsimMult")
- Product of five numbers gives actual parameter
  - base value -from AliGeVSimParticle object
  - particle type level scaller
  - particle type level random number
  - event level scaller
  - event level random number
- Applications
  - systematic study of multiplicity dependance
  - the same for any other parameter (yelds, T, v2, expansion)
  - meaurements of the fluctuations

Parameter determination

- Parameters name patterns
  - global level – "gevsimParam" – eg. gevsimMult
  - global level random – "gevsimParam" – eg. gevsimMultRndm
  - local level – "gevsimPdgParem" – eg. gevsim211Temp
  - local level random – "gevsimPdgParamRndm" – eg. gevsim211V2Rndm

- Names of parameters
  - Mult, Temp, SigmaY (1), ExpVel (4)
  - V1, V2 – flow parameters
  - Psi – Reaction plane angle on global level only

Implementation

- Two classes – AliGeVSimParticle, AliGenGeVSim
- AliGeVSimParticles coresponds to one particle type
  - multiplicity
  - slope parameter, rapidity width, expansion velocity
  - flow coeficients
  - parameters in "AliGeVSimParticle" are subject to scalling
- AliGenGeVSim inherits AliGenerator and do actual job
  - Contain definitions of used formulae
  - "gevsimPt", "gevsimRapidity" (TF1)
  - "gevsimPtY_1" (TF2)
  - "gevsimPhi" (TF1)

Code Quality Assurance

- GeVSim was tested before submited
  - bug in TFormula
- Results on Web page
  - http://home.cern.ch/radomski

Conclusion

- GeVSim is working and tested
- Extensive documentation with examples in preparation
- MevSim is depricated and will disapear in future release
- Please use it – do not reinvent it
- If You need more information – ask
- If You need something special – GeVSim is extensible