Tracking Fast Simulation

David Brown, LBNL

SuperB meeting 2 June 2008
Elba, Italy

FastSim History

- Old (BaBar PravdaMC)
  - True particle 4-vectors are smeared to directly produce analysis objects (BtaCandidates)
  - Additional properties (PID) are simulated via adhoc functions
  - No energy loss, interaction, inefficiency, pat. rec. confusion, decay, no track smear @ dirc, ...
  - Difficult to understand, maintain or extend
- New (PACRAT + revised PravdaMC)
  - Algorithm described in the rest of this talk
  - First code stored in superB subversion repository

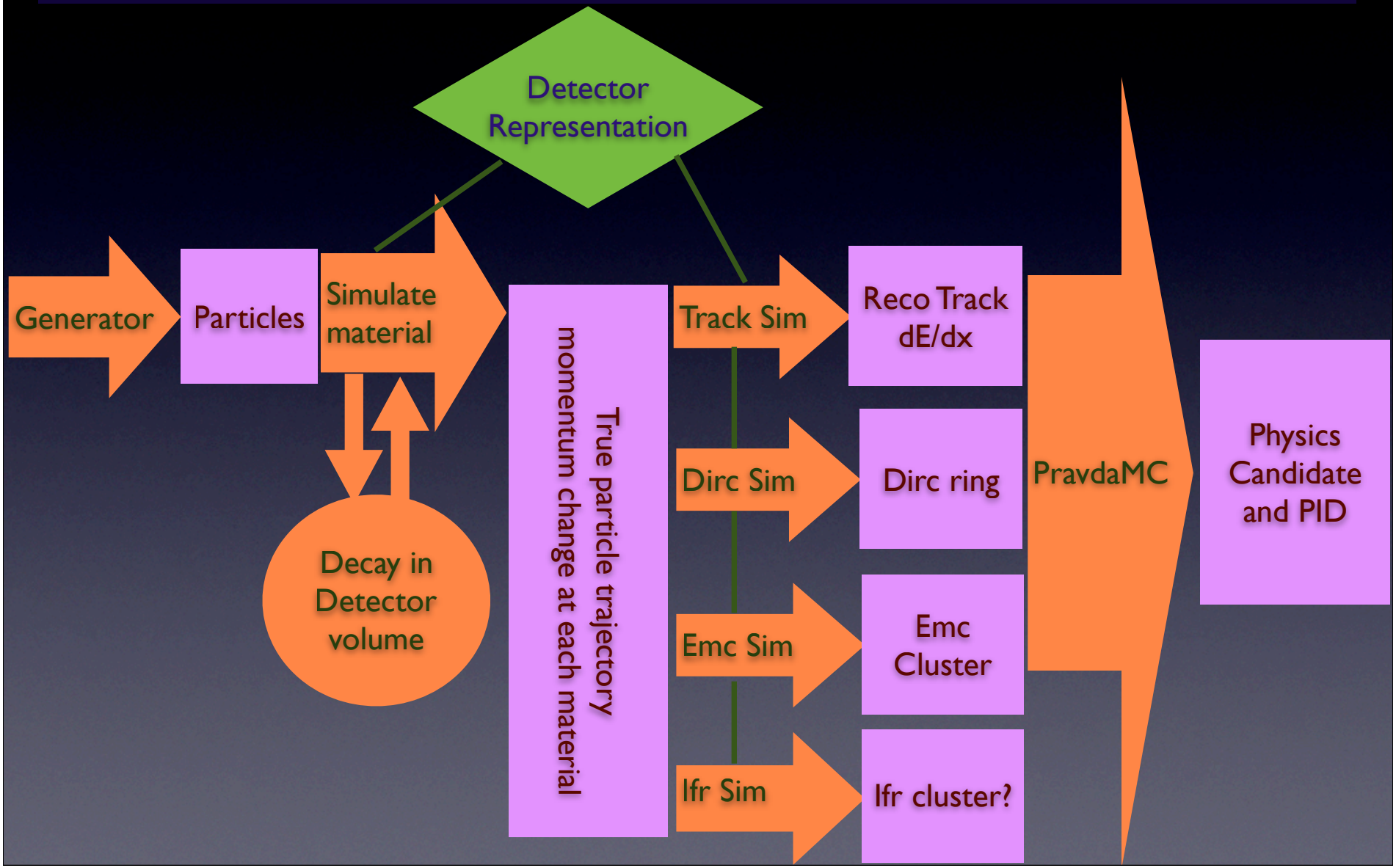
SuperB Fast Sim Goals

- Physics Reach studies
  - Processes dominated by core resolution
  - Processes with physics-dominated backgrounds
  - Change in reach for different detector options
- Detector Optimization
  - Tracking configuration
  - Resolution requirements from physics benchmarks
- Framework to mix in full sim results
  - Background sensitivity studies
  - Outer detector optimization?

Fast Simulation Process

- Generation of primary particles (EvtGen)
- Simulation of particle interaction with the detector material (active + passive)
  - scattering, energy loss, interaction, ...
  - equivalent to the job of G4
- Simulation of the detector response to the particle passage through active material
  - Turn energy deposition into ‘reconstructed’ quantity
  - equivalent to BBSim in BaBar
- Assemble analysis objects

Fast Sim Data Flow



Detector Representation

- Detector divided into simple ‘elements’
  - Cylinders for tracking layers, beampipe, ...
- Thickness modeled ‘locally’
  - not true volumes
- Details handled statistically
  - Si overlaps, Dch wires
- Material properties from G4 database
  - Elements and Composites
- Geometry and other params from ascii file

Material Simulation

- Model particle through each Detector element
- Charged particles Scattering and energy-loss
  - Gaussian or Moliere scattering
  - Landau energy loss
  - Cumulative effects for outer detector
- Interaction probability given X_0, $N_{interaction}$
  - EM and/or Hadronic showers as appropriate
- Gamma Conversion
- Bremsstralung
- Decay

Tracking Response Simulation

- Hits are generated at each active layer
  - Can model Si strips, pixels, axial and stereo wires
  - Only active region generates hits (acceptance)
- Hit efficiency modeled as random loss
- True position is smeared by resolution function
  - Gaussian or non-Gaussian function
- Hit confusion can be modeled by proximity
  - Randomly pick true hit or other hit nearby
  - Proximity defined by 2-hit resolution, track resolution
  - Model confusion from background hits too

Track Reconstruction

- Use BaBar track hit class to store hits
  - Describes a line trajectory in space
  - Describes RMS resolution value, provided by hit
- Use BaBar DetectorModel to describe geometry and material
  - Describes elements as 2-D shapes
  - Same objects used in Detector simulation!
- Use BaBar Kalman fit to reconstruct tracks
  - Hits assigned by simulation (no Pat. Rec.)
  - Provides track parameters, covariance at origin

Detector Configuration

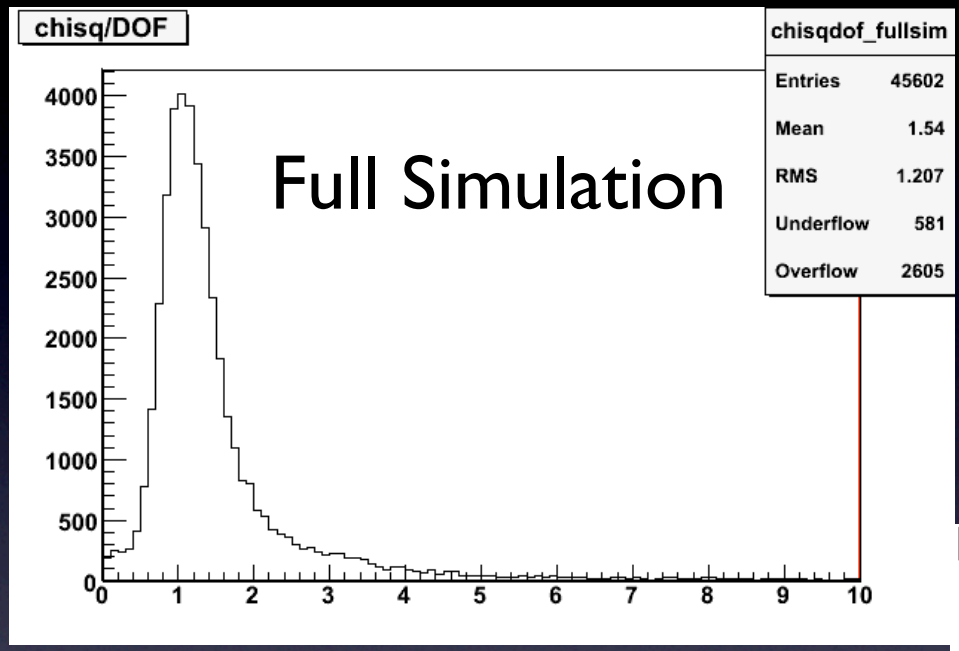
- Controlled by an ascii file
  - ‘arrays’ describing Detector Elements material, geometry, and measurement properties
  - Read by Detector Model constructor
  - Configurable but clumsy
- Current BaBar Detector configuration
  - 5 layers Si
    - ϕ: 10μm resolution, 96% efficiency, z: 20μm resolution, 94% efficiency
  - 40 layers Dch with 6° stereo, 180μm resolution
  - Passive components (beampipe, ...)
- Will provide SuperB prototype description

Current Performance

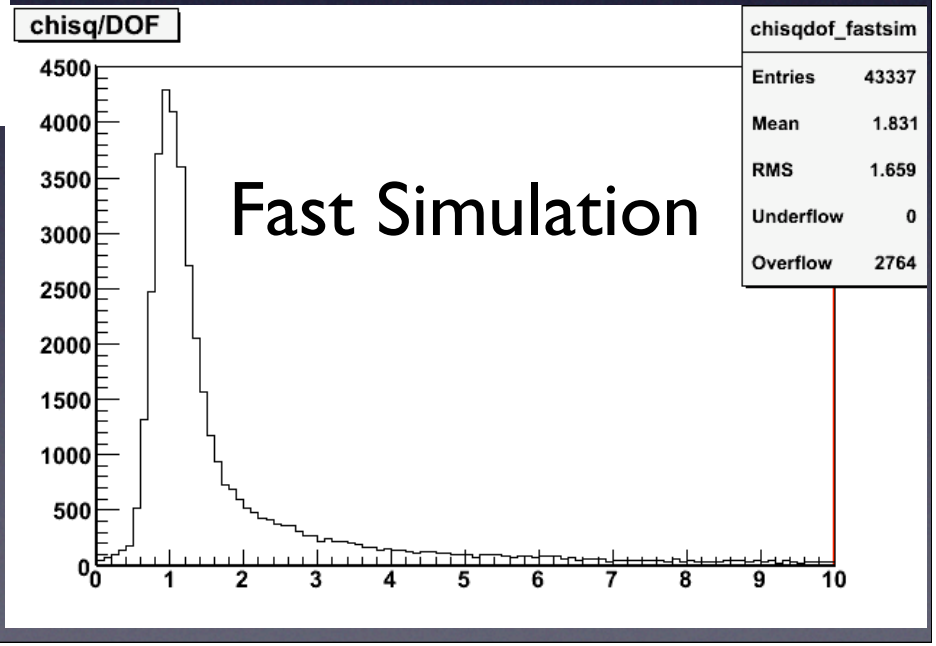
- BaBar detector configuration (including dirc)
- $\Upsilon \rightarrow B\bar{B}$, $B \rightarrow \pi^+\pi^-$
- 2GHz Opteron dual core SL4
- Sim ~1 ms/event, track fit ~5 ms/event
- EvtGen ~11 ms/event, total ~25 ms/event

AppAST:	0.00	10000	15.16	1.51600	0.00 BtaMicroPidKilling
AppAST:	0.00	10000	21.72	2.17200	0.00 BtuTupleMaker
AppAST:	0.00	10000	15.48	1.54800	0.00 RacTestInput
AppAST:	0.00	10000	51.88	5.18800	0.00 PmcReconstruct
AppAST:	0.00	10000	9.81	0.98100	0.00 PmcSimulate
AppAST:	0.00	10000	107.89	10.78900	0.00 GfiEvtGen

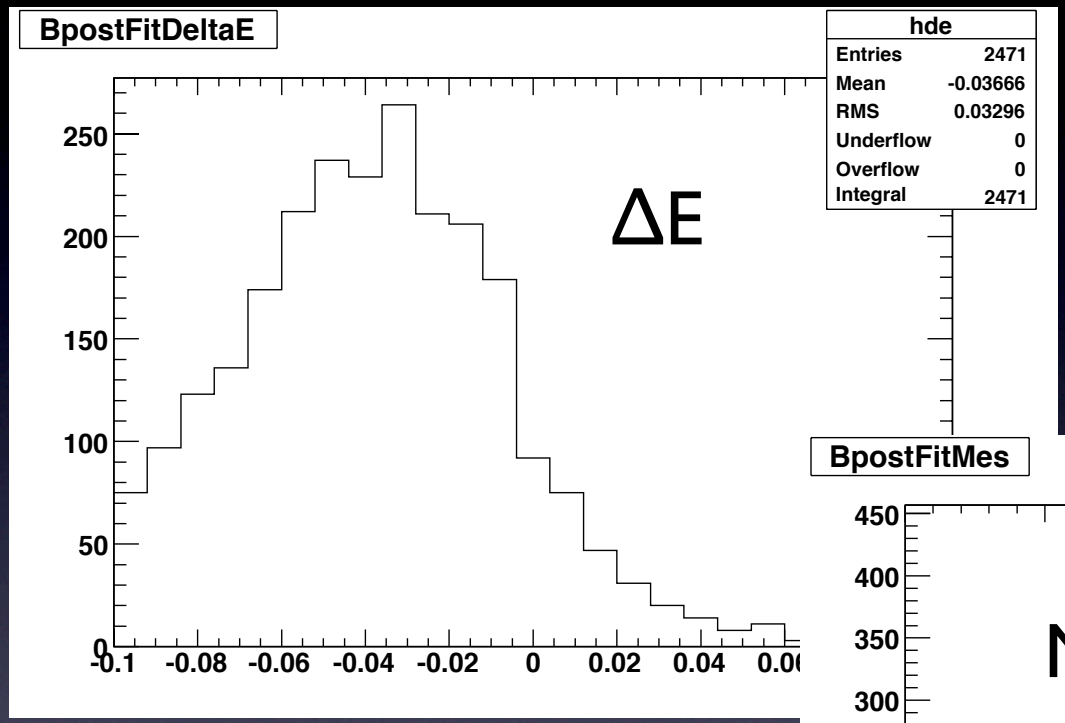
Tracking Results

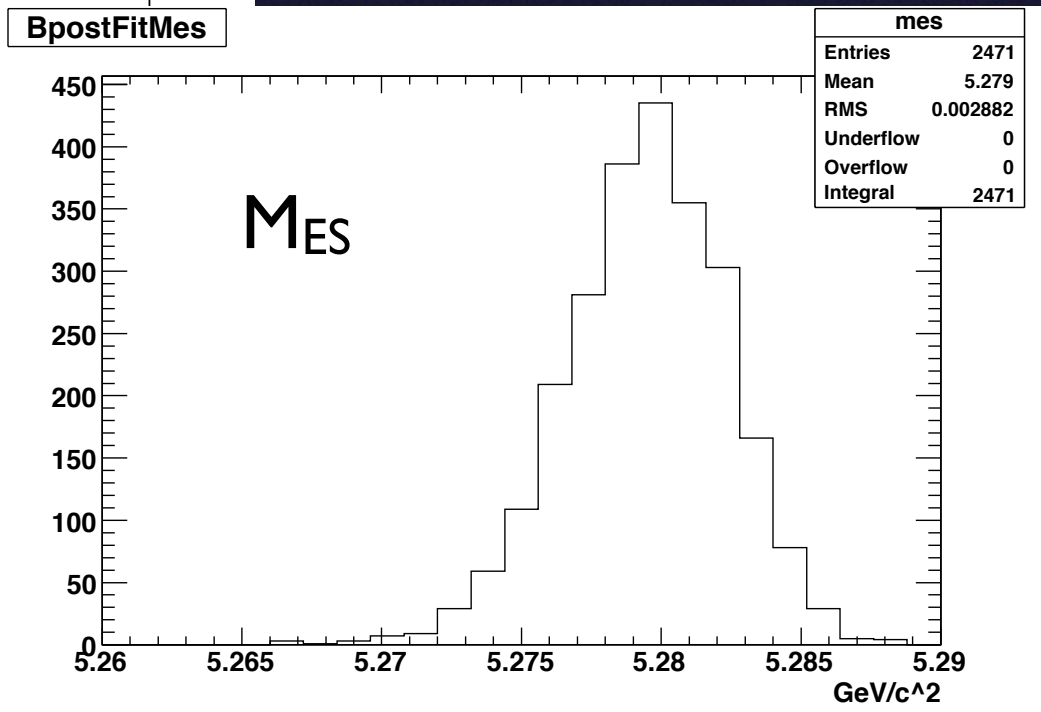


Produced with standard BaBar tracking diagnostic code



B→π+π- Results





Plots by G. Simi

Development Priorities

- Implement plane, cone geometries
- Svt testbeam configuration
- Clean package dependencies, split sequences (G. Simi, M. Rama)
- Hit confusion
- Dch wire material simulation
- Realistic scattering and resolution functions
- Visualization (detector and tracks)
- MC truth matching
- Migrate code from BaBar to SuperB
- SuperB prototype configuration

Configuration Upgrade

- Plan to replace current format with 'GDML'-like xml language
  - Use GDML to describe geometry and material
  - Add extensions to describe sensor segmentation, response functions
    - Used to compute hit confusion, resolution, efficiency
- Derive FastSim from FullSim description?
  - requires code to compress complexity
- Copy or learn from ILC implementation?

Conclusions

- A major upgrade of BaBar fastsim has been started
- Tracking code is implemented, being tested and extended
- Outer detector code is under development
- Capabilities and performance of upgraded fastsim should meet SuperB TDR needs
- A ‘beta’ public release using the SuperB subversion repository should be ready this summer