

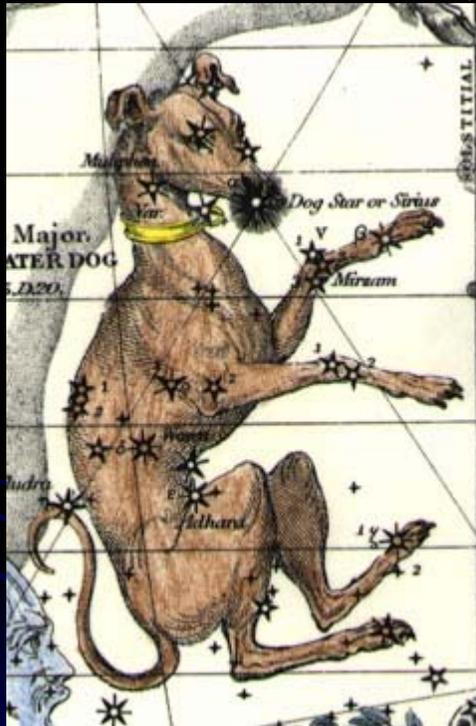
Ληλαψ 3

Fast Detector Simulation Using Lelaps

Overview

- Introduction
- History
 - CEPack Tool Kit
 - Lelaps standalone
- CEPack
 - Geometry and Materials
 - Tracking, Multiple Scattering and dE/dx
 - Electromagnetic and Hadronic Shower Parameterization
- Using CEPack inside Geant4
- Standalone CEPack: Lelaps
- Performance
- Future

Introduction: Lelaps



Lelaps (“storm wind”) was a dog with such speed that, once set upon a chase, he could not fail to catch his prey. Having forged him from bronze, Hephaestus gave him to Zeus, who in turn gave him to Athena, the goddess of the hunt. Athena gave Lelaps as a wedding present to Procris, daughter of Thespius, and the new bride of famous hunter Cephalus.

A time came when a fox created havoc for the shepherds in Thebes. The fox had the divine property that its speed was so great that it could not be caught. Procris sent Lelaps to catch the fox. But because both were divine creatures, a stalemate ensued, upon which Zeus turned both into stone. Feeling remorse, Zeus elevated Lelaps to the skies, where he now shines as the constellation Canis Major, with Sirius as the main star.

Introduction (continued)

Clarification:

No need to infer that Lelaps (the program) is a dog!

Rather, think of “stellar” performance!

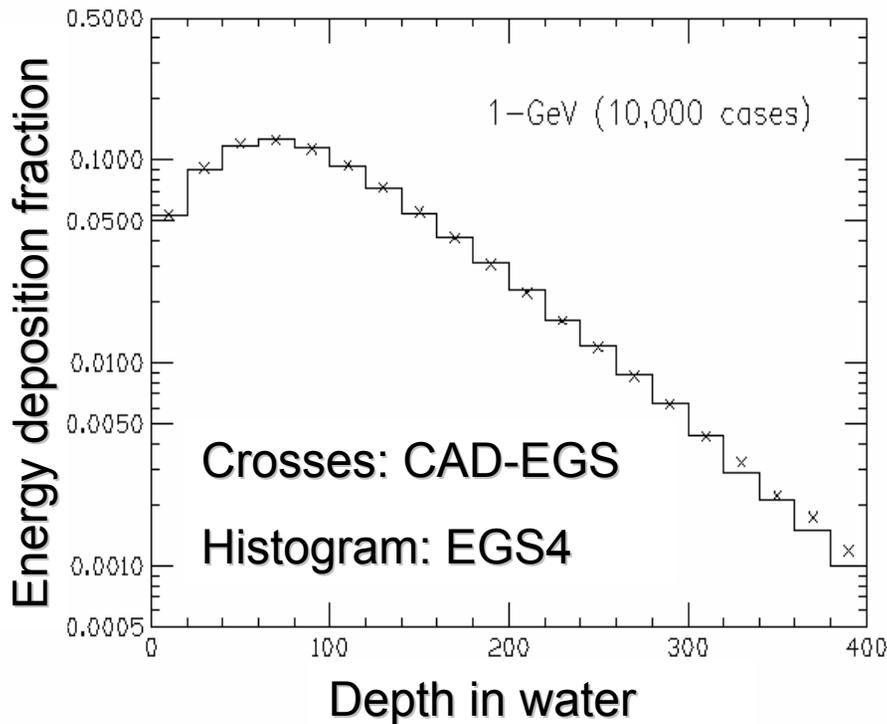
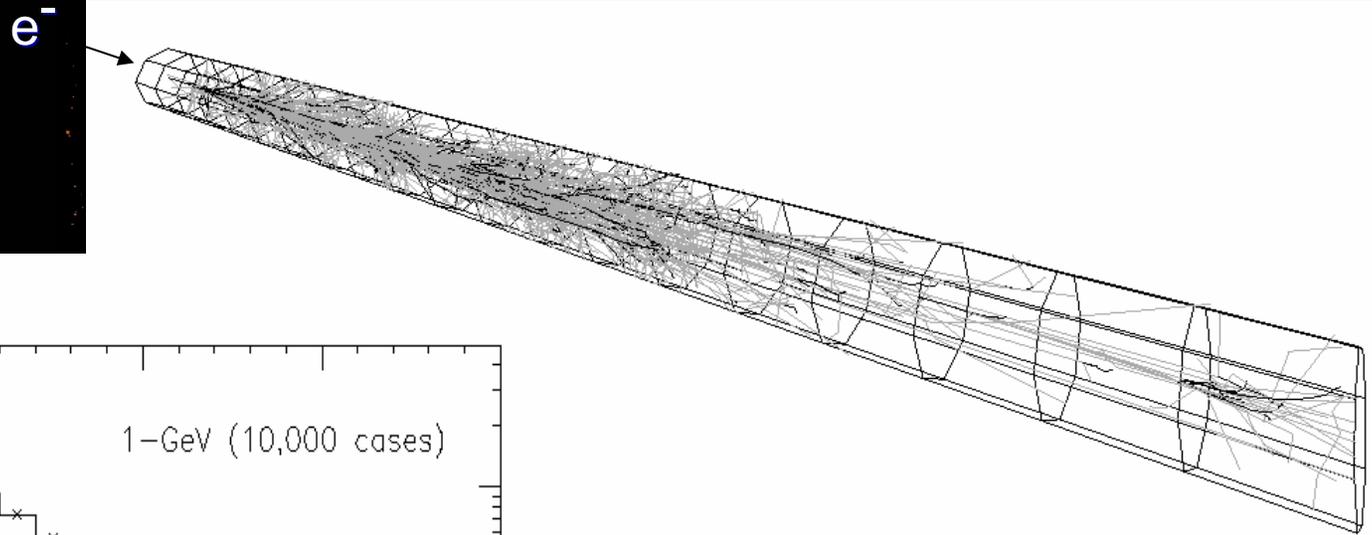
Introduction (continued)

- Lelaps consists of a set of C++ class libraries and a main program, which itself is called lelaps.
- The main library is called CEPack, which contains the actual simulation tool kit. It uses only one utility library, “vec”.
- Other utility libraries include: IStdHep, vlutil, plotpp, pl3.
- Main programs (lelaps) have been written for BaBar and for the Large Detector of LCD.
- CEPack can also be used in conjunction with Geant4 parameterized volumes. In this way it is integrated in BaBar’s Moose (formerly known as Bogus) Geant4-based simulation.
- The standalone version for LCD reads StdHep generator files and produces SIO output files that can be read by JAS and the LCD version of Wired.

History

- 1997 CEPack 0.x: CAD-EGS pilot project to import geometries into EGS4 from standard 3D file format (VRML). CEPack only does geometry.
 1. "The CAD-EGS Project: Using CAD geometries in EGS4", W.G.J. Langeveld, W.R. Nelson, SLAC TN 97-001, July 1997.
 2. "The CAD-EGS Project: Using CAD geometries in EGS4", W.G.J. Langeveld, J.C. Liu, W.R. Nelson, Proceedings of the First International Workshop on EGS4, August 26--29, 1997, KEK, Tsukuba, Japan, page 75.

History – CAD-EGS



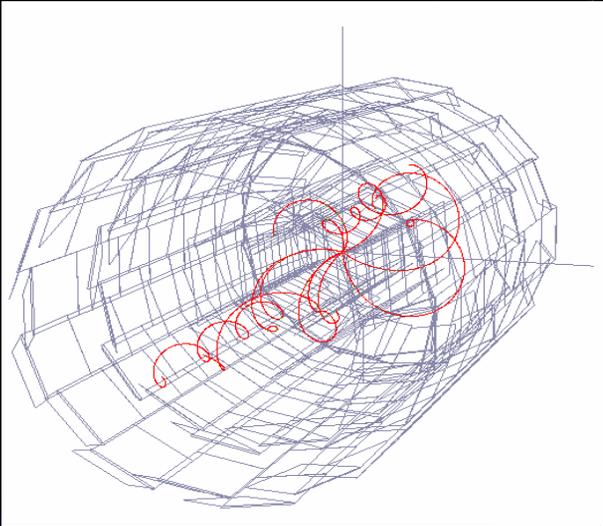
Crannel benchmark: 1 GeV electron beam in a 4 m long water cylinder, radius 12 cm, sampled in 20 segments.

History (continued)

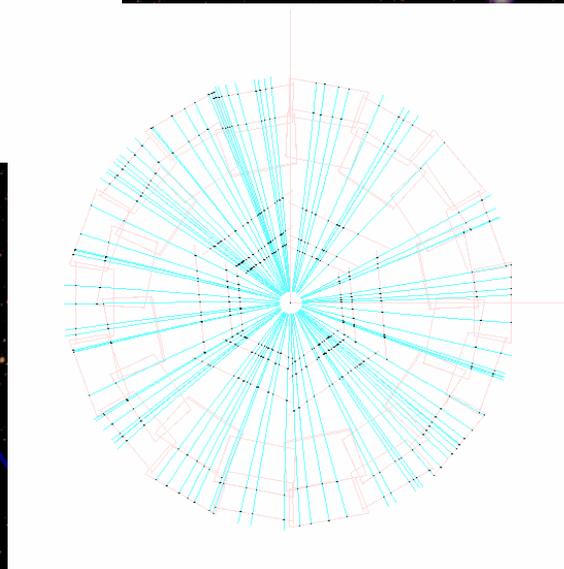
- 1997 CEPack 0.x: CAD-EGS pilot project to import geometries into EGS4 from standard 3D file format (VRML). CEPack only does geometry.
 1. "The CAD-EGS Project: Using CAD geometries in EGS4", W.G.J. Langeveld, W.R. Nelson, SLAC TN 97-001, July 1997.
 2. "The CAD-EGS Project: Using CAD geometries in EGS4", W.G.J. Langeveld, J.C. Liu, W.R. Nelson, Proceedings of the First International Workshop on EGS4, August 26--29, 1997, KEK, Tsukuba, Japan, page 75.
- 1998 CEPack 1.0: Geant4 program with Parameterized Volumes for BaBar SVT and DCH.

CEPack does transport with dE/dx and multiple scattering.
- 1999 Integration into Bogus (CEPack 1.x)

History – CEPack 1.x in BaBar



BaBar Silicon Vertex Detector as modeled by CEPack. All 320 wafers are individually represented.



100 one-GeV muons, end view. Black dots represent hits.

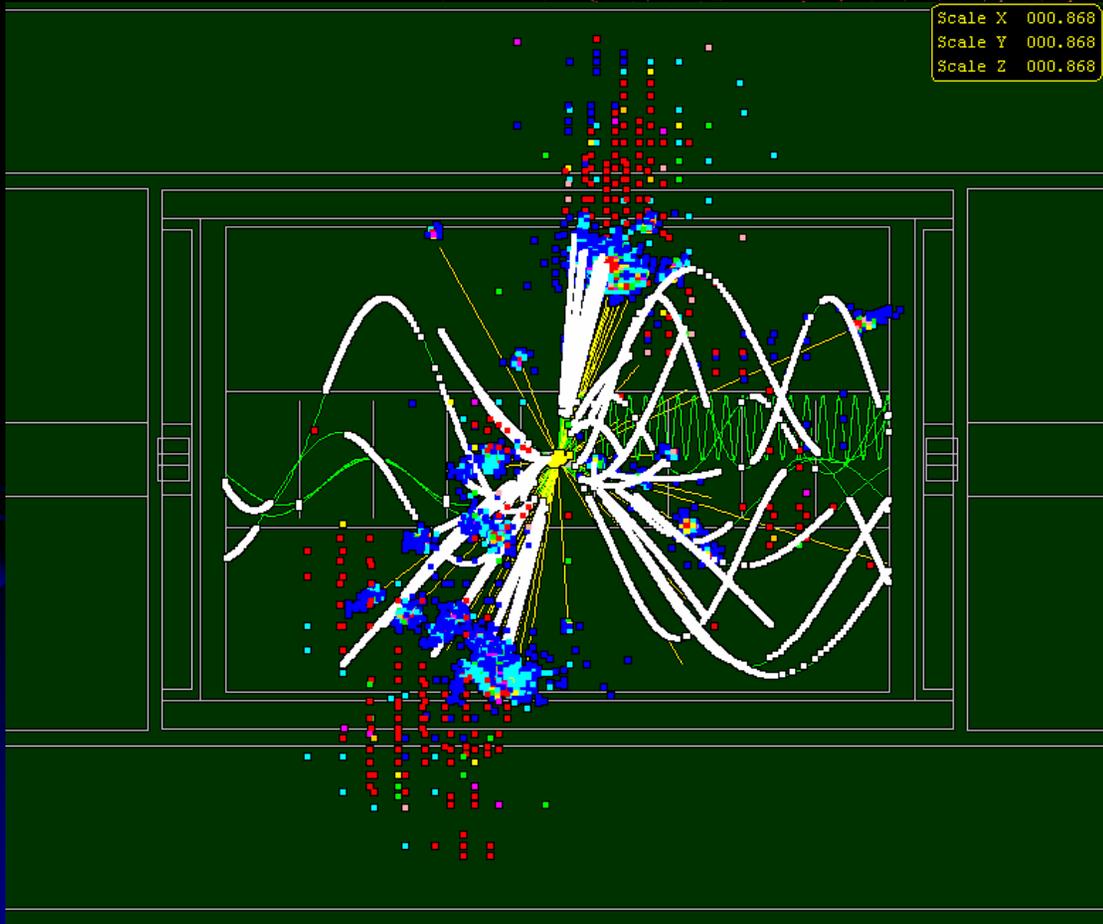
(BaBar Drift Chamber was added later.)

History (continued)

- 1997 CEPack 0.x: CAD-EGS pilot project to import geometries into EGS4 from standard 3D file format (VRML). CEPack only does geometry.
 1. "The CAD-EGS Project: Using CAD geometries in EGS4", W.G.J. Langeveld, W.R. Nelson, SLAC TN 97-001, July 1997.
 2. "The CAD-EGS Project: Using CAD geometries in EGS4", W.G.J. Langeveld, J.C. Liu, W.R. Nelson, Proceedings of the First International Workshop on EGS4, August 26--29, 1997, KEK, Tsukuba, Japan, page 75.
- 1998 CEPack 1.0: Geant4 program with Parameterized Volumes for BaBar SVT and DCH.

CEPack does transport with dE/dx and multiple scattering.
- 1999 Integration into Bogus (CEPack 1.x)
- 2000 BaBar fast simulation shelved
- 2002 Lelaps stand-alone program for LCD (CEPack 2.x)
- 2002 BaBar revived fast simulation, CEPack 2.x used in Moose
- 2003 Lelaps 3.x for LCD using CEPack 3.x. Upgrade BaBar to CEPack 3.x, CEPack does parameterized shower simulation.

The Present – Lelaps and CEPack 3.x



Wired picture of an $ee \rightarrow ZZ$ event as simulated by Lelaps for the LCD Large Detector model.

White dots represent track hits. The blue showers are in the EM calorimeter, the red ones in the hadron calorimeter.

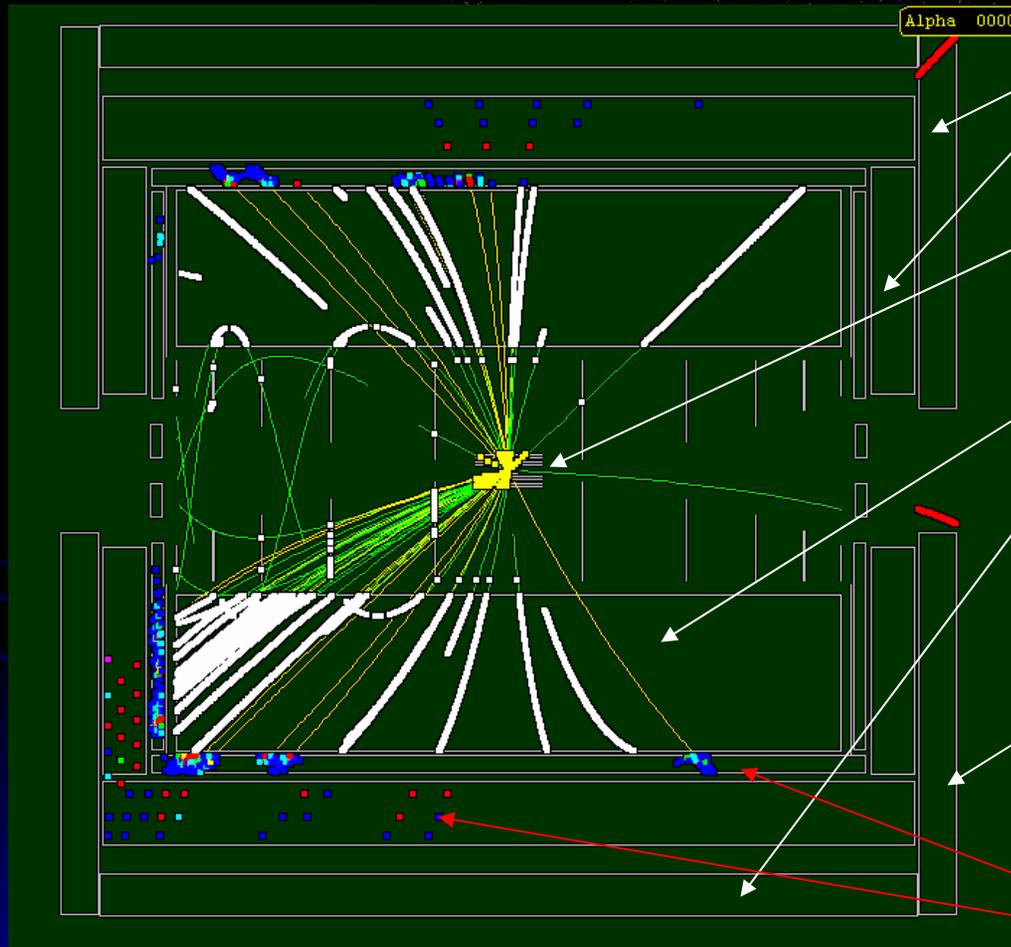
Caution

- The code has evolved beyond the original design, and is in fact still evolving.
- Because of its history, there are places with a lack of consistency in naming conventions.
- The interfaces are still not frozen.
- Therefore, don't depend on the details of what follows.
Things WILL change!

Geometry in CEPack

- Geometries are constructed using CENodes (which may contain a list of subnodes).
- A number of common CENodes are predefined
 - cylinders, cones, boxes, spheres
- Transformations may be applied to CENodes in order to position and orient them. Arbitrary affine transformations are allowed.
- CENodes need to provide methods for tracks entering and exiting, and for determining whether points are inside them or not.
- CENodes may be assigned a numeric id and subid.
- CENodes may implement a method to compute a subid from a location.
 - May be used to compute calorimeter segmentation.
- CENodes do not have to be 3D objects.
 - Several predefined CENodes consist of one or more 2D surfaces.

Geometry in CEPack (continued)



- Most objects in the LD are cylinders.
- There are some conical masks (not drawn).
- TPC and barrel muon detector each contain a CENode with a set of concentric cylinder surfaces.
- Muon endcaps use a CENode with a set of cylindrical slices.
- Calorimeters use subid calculation for their segmentation.

Geometry in CEPack (continued)

- Example: create a lead pipe (“double-walled cylinder”)
 - `CENode *ce = new CEDWCylinder(r1, r2, length, nlayers, radially);`
 - `ce->rotate_around_x(radians(90.0)); // in Z`
 - `ce->translate(vec(0, 0, offset)); // if needed`
 - `ce->material = CEElement("Pb");`
- Set type of object (Mask, Tracker, EM or Hadron calorimeter) and whether to report hits:
 - `ce->setflags(CEN_MASK | CEN_DETECTOR);`
 - `ce->setid(666);`
 - `ce->set_samplingFraction(0.02); // if a calorimeter`

Materials in CEPack: Elements

- Three types of materials: CEElement, CECompound and CEMixture.
- CEElement: basic knowledge of all the elements is built in.
 - CEElement Helium("He"), Lead("Pb"), Oxygen("O2");
 - Knows that helium and oxygen are gasses and lead a solid
- Can specify pressure and temperature for gasses, density for solids.
 - CEElement Argon("Ar", 5.0, 298.15); // in atm and K
 - CEElement LowDensityCarbon("C", 0.01); // in g/cm³
- May use common English name:
 - CEElement Tungsten("Tungsten");

Materials in CEPack: Compounds

- Compounds specified by (simple) chemical formula and density (for solids and liquids) or pressure and temperature (gasses):
 - CECompound Na₂O("Na₂O", 2.27);
 - CECompound CaO("CaO", 3.34);
 - CECompound Al₂O₃("Al₂O₃", 3.97);
 - CECompound SiO₂("SiO₂", 2.196);
 - CECompound MgO("MgO", 3.6);
 - CECompound Epichlorohydrin("C₃H₅ClO", 1.18);
- May use floating point:
 - CECompound YBaCuO("YBa₂Cu₃O_{6.6}", density);
- May give it a convenient name:
 - CECompound Epichlorohydrin("Epoxy", "C₃H₅ClO", 1.18);
- May not use parentheses in formula.

Materials in CEPack (continued)

- Mixtures can be created by adding defined materials either by volume or by weight:
 - CEMixture G10("G10");
 - G10.addByWeight(Na2O, 17.0);
 - G10.addByWeight(CaO, 5.0);
 - G10.addByWeight(Al2O3, 1.0);
 - G10.addByWeight(SiO2, 73.0);
 - G10.addByWeight(MgO, 4.0);
 - G10.addByWeight(Epichlorohydrin, 54.0);
- The “fractions” do not have to add up to 1 (or 100%) – with each addition, the fractions are renormalized.
- Use CEGasMixture or CECondensedMixture for mixtures of gasses and solids in order to specify whether the result is a gas or solid.

Materials in CEPack (continued)

- All needed quantities are calculated automatically
 - Constants needed for multiple scattering and energy loss
 - Radiation lengths (Tsai, PDG)
 - Interaction lengths (from a fit to element data)
 - Other constants needed for shower parametrization

Tracking, Multiple Scattering and dE/dx in CEPack

- Tracking is performed by taking steps along a linear trajectory with endpoints on a helix, such that the sagitta stays below a certain (settable) maximum.
- CENodes have bounding spheres (or bounding cylinders).
 - When computing distances to CENodes, only “relevant” CENodes are considered.
- After each step, the amount of material traversed is checked: if enough material was traversed, multiple scattering and energy loss is performed and track parameters and list of relevant CENodes are recalculated.
- When an intersection occurs within a step, the fractional step is executed, the CENode is entered, and the remaining fraction of the step follows.

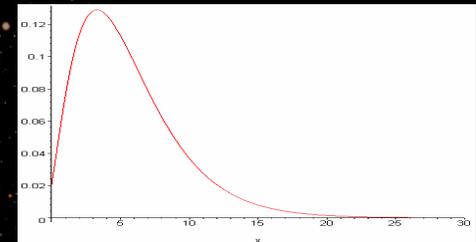
Multiple Scattering and dE/dx (continued)

- Multiple scattering is performed using the algorithm of Lynch and Dahl.
 - Gerald R. Lynch and Orin I. Dahl, Nucl. Instr. And Meth. B58 (1991) 6.
- Material is “saved up” along the track until there is enough.
- dE/dx is calculated using the methods by Sternheimer and Peierls.
 - R.M. Sternheimer and R.F. Peierls, Phys. Rev. B3 (1971) 36810
- All constants precalculated by the material classes.

Electromagnetic Shower Parameterization in CEPack

- Electromagnetic showers are parameterized using the algorithms of Grindhammer and Peters.
 - G. Grindhammer and S. Peters, arXiv:hep-ex/0001020v1 (2000)
(Paper is a 1993 conference contribution, submitted by request to the archive in 2000).
- Calorimeters are treated as homogeneous media, with longitudinal shower profile given by a gamma distribution (t in radiation lengths):

$$\frac{1}{E} \frac{dE(t)}{dt} = f(t) = \frac{(\beta t)^{\alpha-1} \beta e^{-\beta t}}{\Gamma(\alpha)}$$

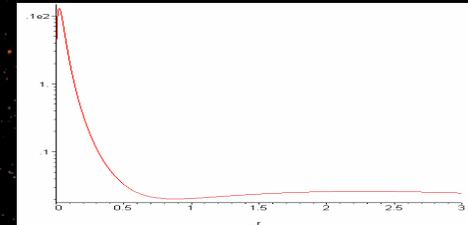


- Coefficients α and β depend on the material (Z) and energy.
- The profiles are fluctuated, and correlations between α and β are taken into account.

Electromagnetic Shower Parameterization (continued)

- For each step of one radiation length, a radial profile is computed of the form:

$$\frac{1}{dE(t)} \frac{dE(r,t)}{dr} = f(r) = p \frac{2rR_C}{(r^2 + R_C^2)^2} + (1-p) \frac{2rR_T}{(r^2 + R_T^2)^2}$$



with R_C the median radius (in units of Molière radius) of core component of the shower, R_T the median radius of tail component and p the fraction of “core” in the shower.

- R_C , R_T and p are functions of Z , shower depth t and E .
- Energy $dE(t)$ is divided into spots (another gamma distribution with parameters depending on t). Spots are thrown in r according to the profile above, uniformly in ϕ and also uniformly between t and $t+1$.
- Roughly, about 400 spots are generated per GeV of shower energy.
- Spots are reported as “hits”.

Hadronic Shower Parameterization in CEPack

- Hadronic showers are parameterized using code that is similar to the code for electromagnetic showers, with some modifications:
- The location where the shower starts is simulated using an exponential law with attenuation given by the interaction length.
- The longitudinal profile uses the Bock parameterization.
 - R.K. Bock, T. Hansl-Kozanecka and T.P. Shah, Nucl. Instr. And Meth. 186 (1981) 533.
- A combination of two gamma distributions, one using radiation lengths and the other interaction lengths, is used.
- Bock parameterization does not specify radial profiles. For the moment we use the radial profiles from Grindhammer & Peters (for EM showers) but with radiation lengths replaced by interaction lengths. The parameters need to be adjusted.

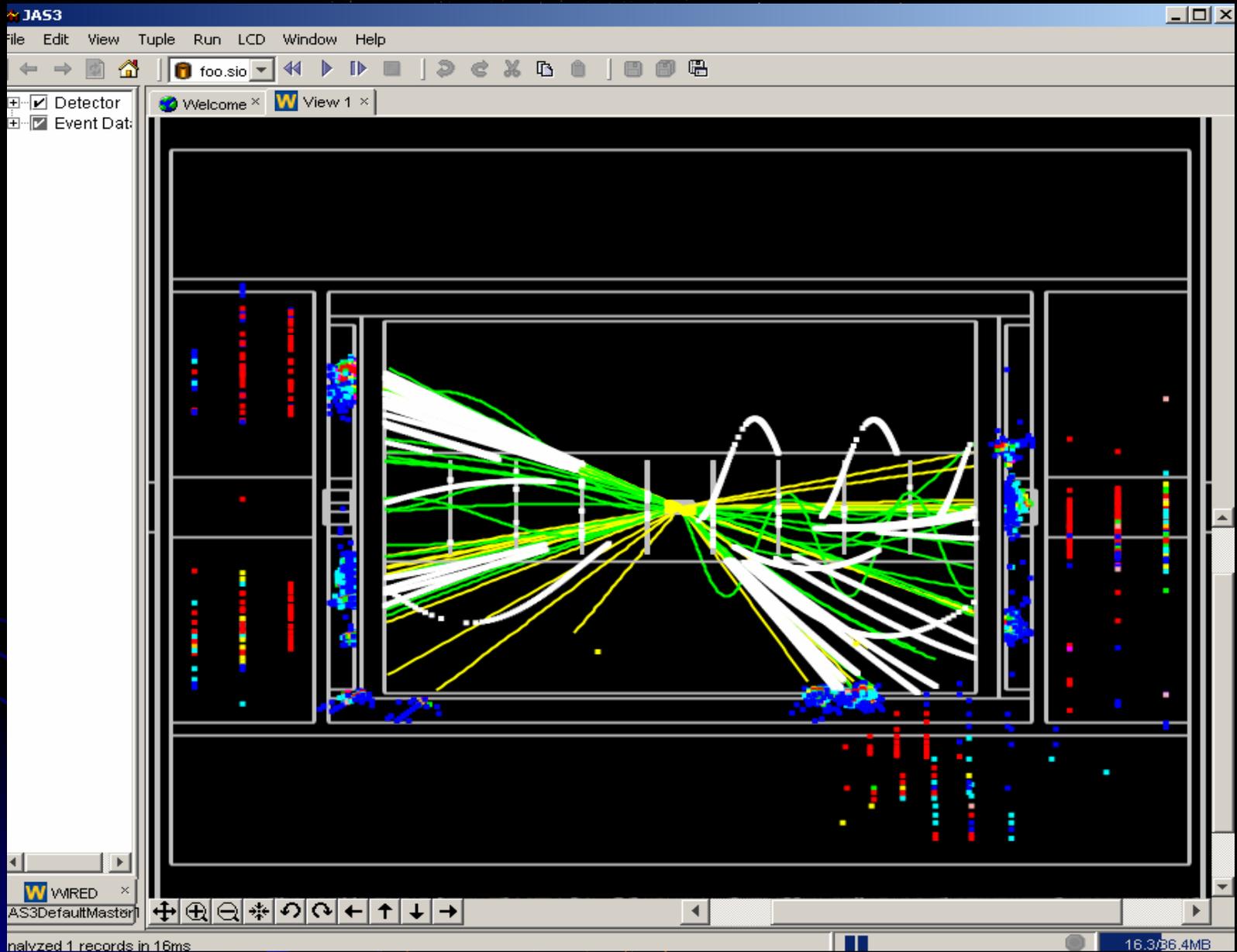
Using CEPack inside Geant4

- To use CEPack inside Geant4, create a Geant4 parameterized model, e.g.:
- In its setup(), create the CENode corresponding to the CEPack simulation with all its subnodes.
- In Doit(), convert from G4FastTrack track to CETrack, and call track.swim()
- By subclassing CETrack, all hits are reported using CETrack report_hit() method. Convert hits to your favorite format.
- Update G4FastStep (or call KillPrimaryTrack()):

```
fastStep.SetPrimaryTrackFinalPosition(  
    G4ThreeVector(track.p.getx() * m, track.p.gety() * m,  
    track.p.getz() * m), false);  
fastStep.SetPrimaryTrackFinalKineticEnergyAndDirection(  
    (track.energy - track.mass) * GeV,  
    G4ThreeVector(track.q.getx(), track.q.gety(),  
    track.q.getz()), false);
```

CEPack and Lelaps

- Lelaps for LCD is a standalone program which uses CEPack.
- It sets up the CEPack geometry (currently only for LD)
- It uses IStdHep (“StdHep light” – included in the distribution): a class library to read in generator event files in StdHep format. StdHep events are converted to SIO format and written out.
- Loops over events, creating CTracks. When hits are reported, they are written out in SIO format. For calorimeter hits, the “spots” are first accumulated and turned into energy depositions in individual calorimeter cells.



Performance

- 100 $ee \rightarrow ZZ$ events* at 500 GeV c.m. energy in the LD on a 1.4 GHz Pentium 4 running Linux (Noric09) gives the following performance (all numbers converted to 1 GHz processor speed):

	Overall time (sec at 1 GHz)	Time/event (msec at 1 GHz)	Time/track (msec at 1 GHz)
Tracking only (with dE/dx & MS)	23.6	236	3.95
Tracking + EM showers	28.8	288	4.82
Tracking + EM and hadronic showers	39.4	394	6.60
Same + SIO output file	77.3	773	12.9

* panpy-ZZ-500-001001-gen-1.stdhep (100 events, 5973 tracks)

Performance (continued)

- Tracking alone: 5 events per second (at 1 GHz).
 - This can probably be improved (it used to do better before the recent changes).
- Adding parameterized showering costs almost a factor 2.
 - No optimization attempted so far.
 - Hadronic showers take most of the time – probably unnecessarily so.
- SIO output file (10 MB, compressed) costs a lot, almost another factor of 2!
- Platform/machine dependent. Tracking only, time per event:

● Noric09 (1.4 GHz, gcc 2.95.3)	0.236 sec at 1 GHz
● Tersk08 (0.44 GHz, Solaris WS 6u1)	0.275 sec at 1 GHz
● Windows (2 GHz, cygwin with gcc 3.2)	0.458 sec at 1 GHz

Future

- Lelaps and CEPack interfaces are not yet frozen!
- New features planned for CEPack
 - Combinatorial geometry
 - Shower continuation into next volume
- New features planned for Lelaps:
 - Read in geometry from some “standard” file format
 - Write LCIO rather than SIO
- Old features need to be tested more thoroughly, known bugs fixed
 - In particular, need to test parameterized showers
- Review of code, improvements in consistency
- Revisit optimization



The End