Recommendations for improving application performance (… but not presentation speed)

1. **Physics Based Biasing**
2. **Geometrical Biasing**
3. **Neutron_HP**
4. **Physics Tuning and Electromagnetic Physics**
5. **Geometry/Detector Description**
6. **Magnetic Field Tuning**
7. **User Actions**
8. **Configuration issues**
9. **Summary**

Alex Howard, CERN

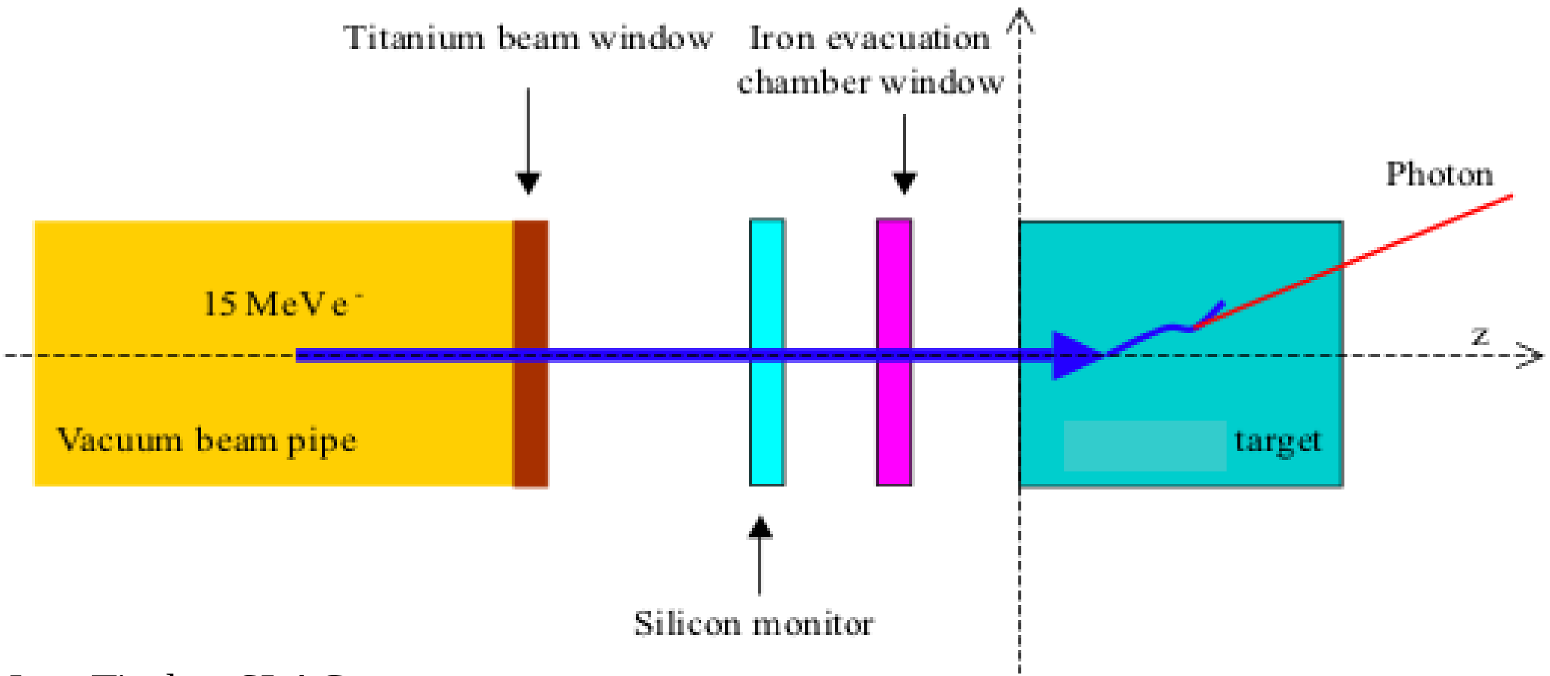
CPU Performance: Recommendations for improving application performance

Geant4 Users Workshop Hebden Bridge 14th September 2007

With input from Tatsumi Koi, Vladimir Ivanchenko, Jane Tinslay, Gabriele Cosmo, Tatiana Nikitina, Makoto Asai, John Apostolakis

Uniform Bremsstrahlung Splitting

- Thick target bremsstrahlung simulation
- Primary electron energy of 15.8 MeV
- Low energy physics list with 0.1mm cut
- Target materials: beryllium, aluminium and lead
- Photons are counted on a scoring sphere 1m from the target

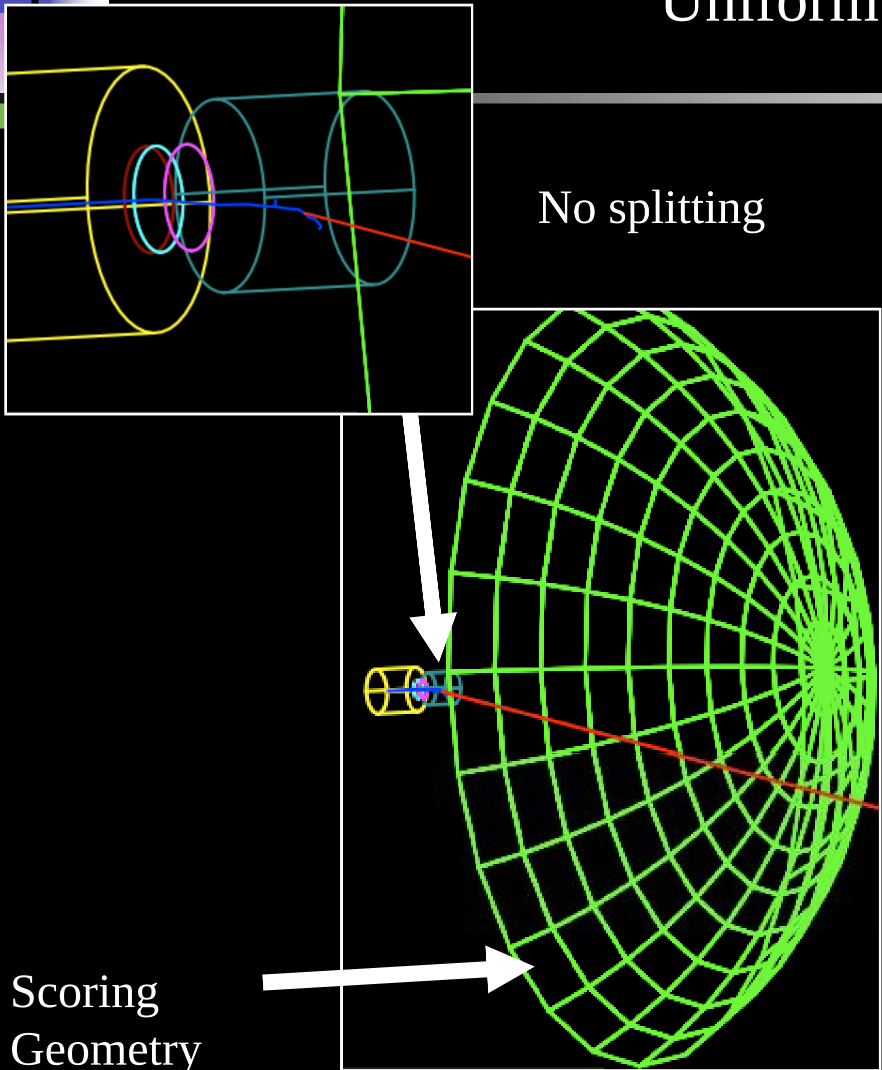


- Using G4WrapperProcess, speed up simulation using the uniform bremsstrahlung splitting with Russian roulette variance reduction technique: A.F. Bielajew et al., SLAC-PUB-6499 (1994)

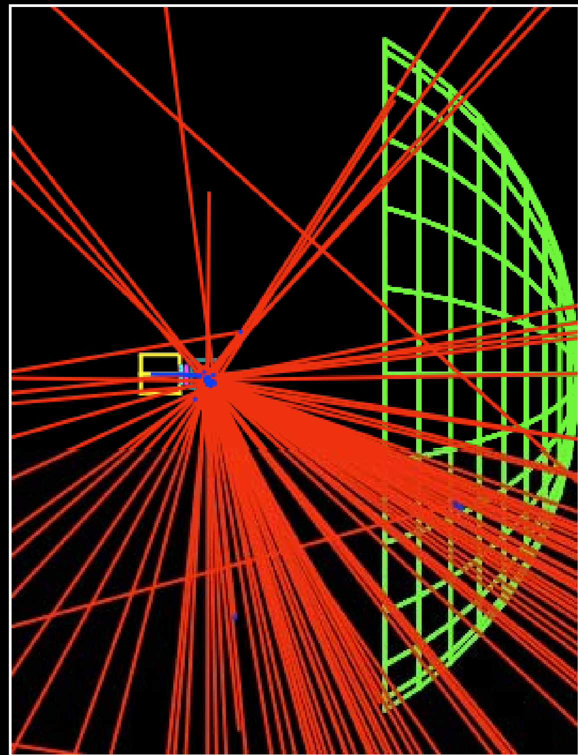
- Bremsstrahlung splitting
  - Reduce time spent tracking electrons
  - Do regular electron transport until bremsstrahlung interaction occurs
  - Then generate 100 unique secondary photons instead of just one
  - Apply relative weight correction of 1/100 to photons
- Russian roulette
  - Population control
    - Don’t want too many low weight particles
  - Keep only 1/100 charged secondaries with their weight increased by a factor of 100
  - Net effect is that all photons have relative weight of 1/100, while all electrons have same weight

Uniform Bremsstrahlung Splitting

No splitting



Splitting factor = 100



Scoring Geometry

- Estimate biasing efficiency:

$$Efficiency = \frac{CPU\,time\,to\,generate\,10\,million\,unbiased\,events}{CPU\,time\,to\,generate\,0.1\,million\,biased\,events}$$

	Beryllium	**Aluminium**	**Lead**
Efficiency	**11.8**	**6.6**	**3.1**

- Efficiency drops off as a function of atomic number since more bremsstrahlung photons interact in higher atomic number material
- Speedup is reasonable

Geometrical Biasing (1)

Example measurement

One example of the simulation: 68 MeV protons on ^{7}Li target, 200 cm concrete shield, measure neutron flux on beam axis

processing times:

analog	non-analog
400 hours	100 hours

Estimation of the gain in computing time (from FOM $= \frac{1}{R^2 T}$):

$$G = \frac{\text{FOM}_{\text{imp.}}}{\text{FOM}_{\text{ana.}}}$$

Interpretation: For adequate statistics G is the factor the analog calculation would have to run longer than the non-analog calculation for the same precision.

Geometrical Biasing (2)

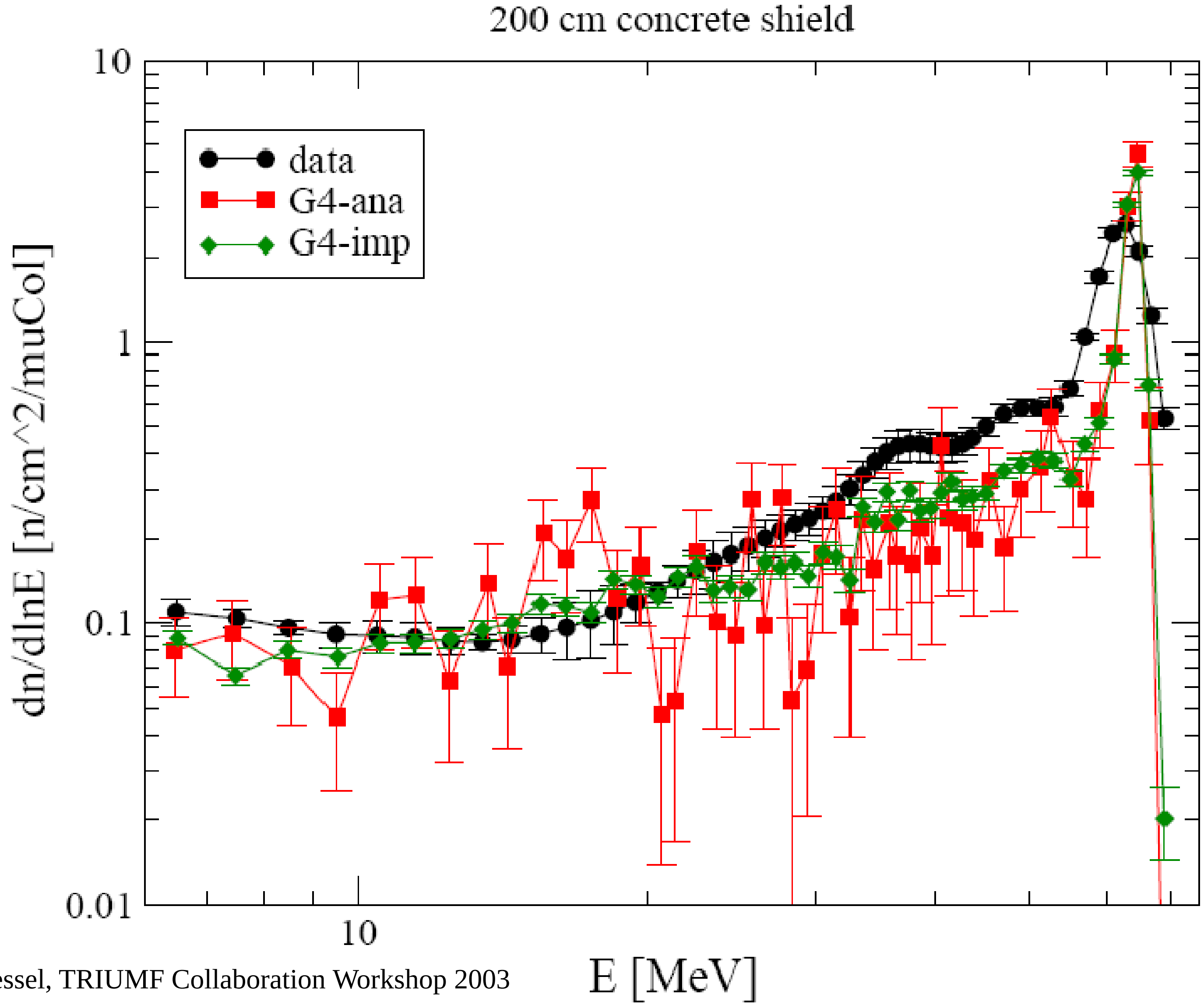
Result of TIARA simulation

Exp. or Sim.	10 - 60 MeV			
	Flux $[\frac{n}{cm^2 \mu C}]$	R [%]	$\frac{Sim.}{Exp.}$	G
Exp	5.17E-01	–	–	89
G4-imp	3.41E-01	1.33	0.66	
G4-ana	3.37E-01	6.26	0.65	
	60 - 70 MeV			
Exp	2.80E-01	–	–	59
G4-imp	2.71E-01	1.81	0.97	
G4-ana	2.86E-01	6.92	1.02	

Analog needs 89 times longer for same precision (10 - 60 MeV)

Geometrical Biaising (3)

68 MeV p->Li

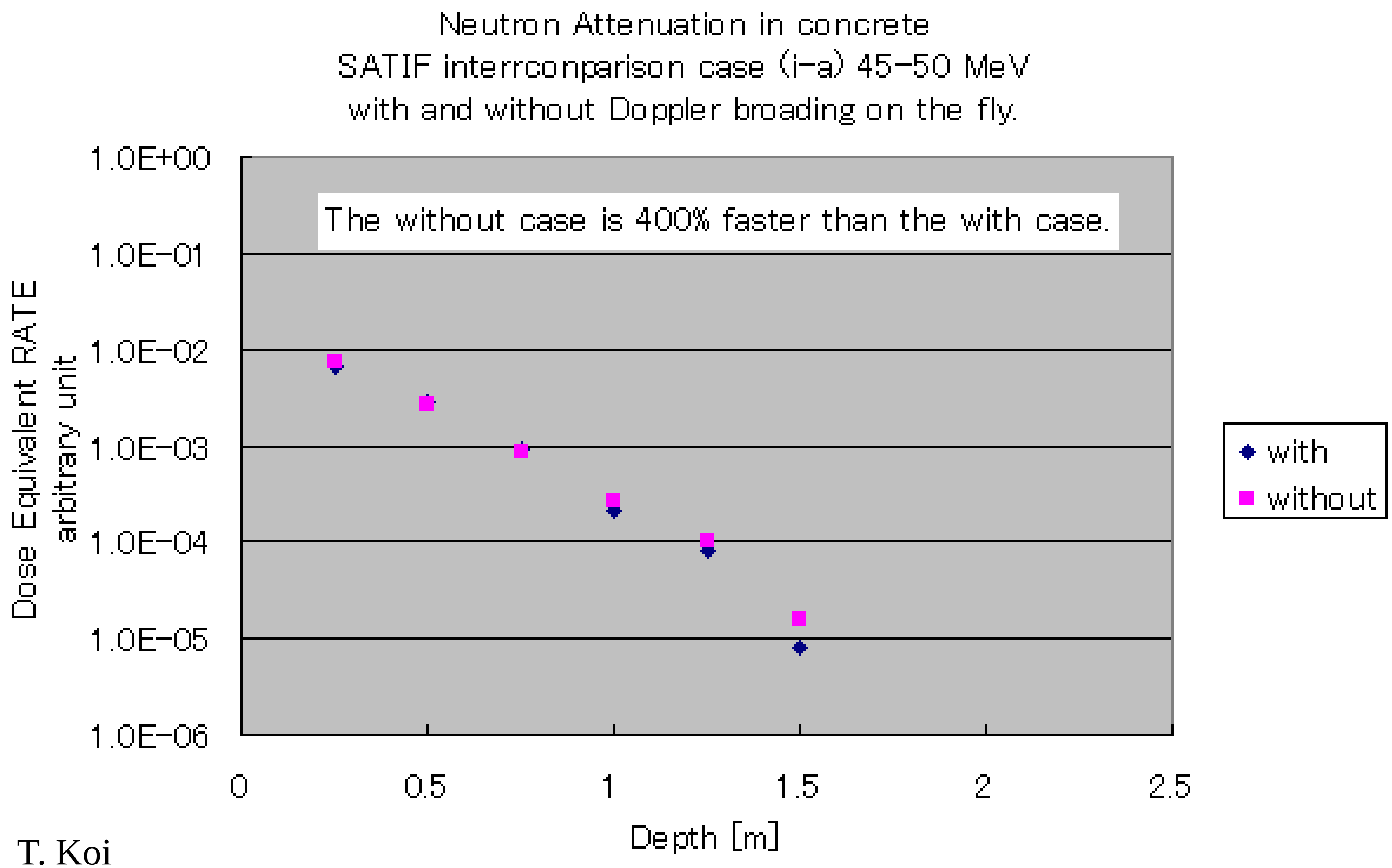


Neutron_HP - Switching off Doppler Broadening (1)

- Since v9.0 you can switch off Doppler Broadening in neutron_hp – due to the thermal motion of the target nucleus.
- The broadening is used to correct the G4NDLdata, which are at zero temperature. This must be done only for appropriate applications.
- To switch broadening off, the environment variable `G4NEUTRONHP_NEGLECT_DOPPLER` must be set
- Note the thermal motion of the nucleus is still taken into account in the FinalState Generator.
- This option provides a significant CPU performance advantage
- It is not suitable for simulation in which low energy (thermal) neutrons are important, as related observables

Neutron_HP – switching off Doppler Broadening (2)

- **setenv G4NEUTRONHP_NEGLECT_DOPPLER=1**



T. Koi

Physics Tuning and Electromagnetic Physics

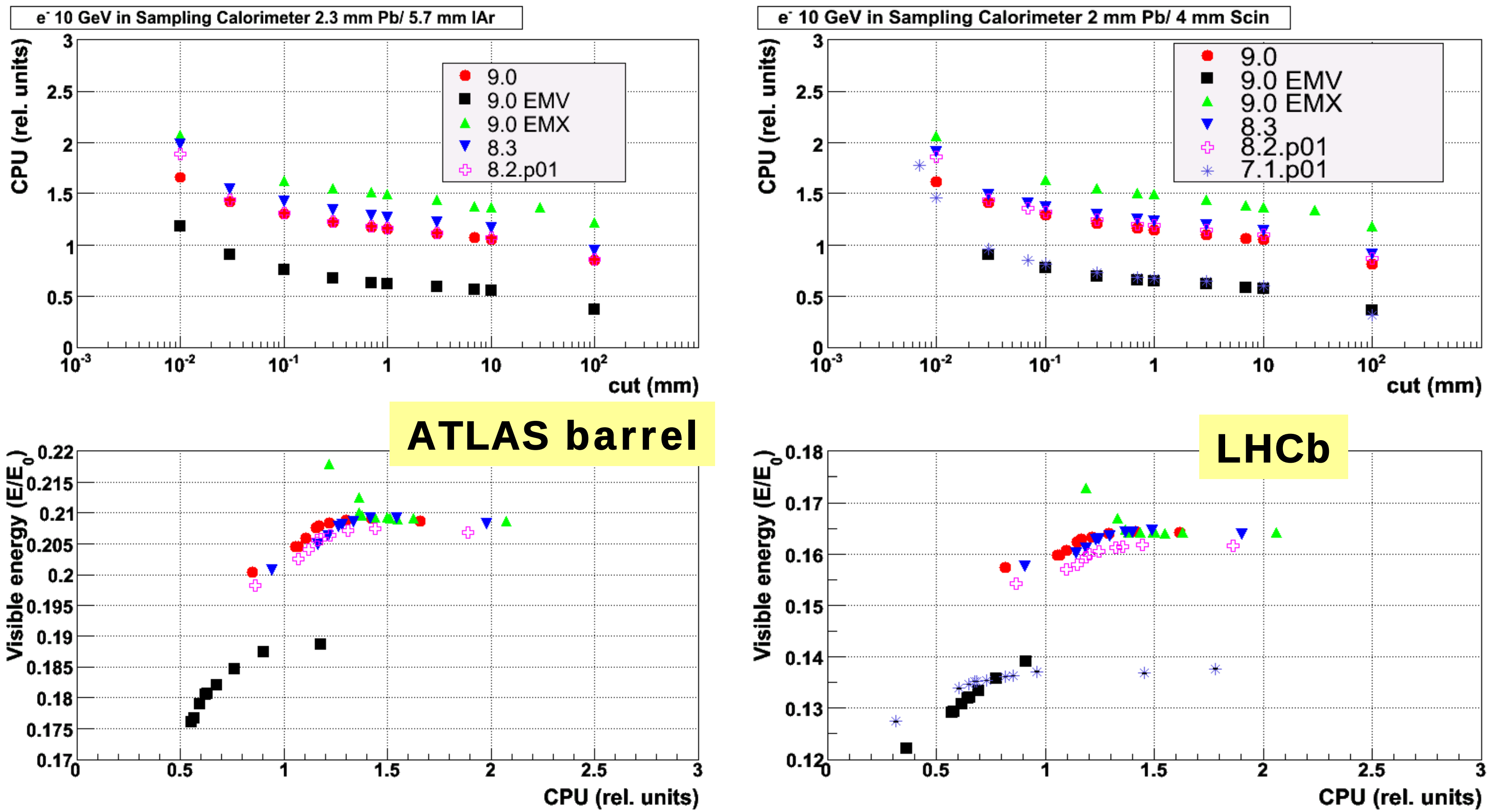
- Adequate physics list – what do you need for your use case/application?
- Choice of models – parametrized, FAST, theoretical driven, data-base driven energy ranges…
- Cuts in energy, particle type, range cuts
- Cuts by region
- Processes with infra-red divergence – what production cut do I need?
- EM settings?
  - E.g. Multiple scattering – model, parameters, range cut, choice of skin value, default values
- Initialisation time, data-base loading…

How to optimize CPU usage for EM applications

- CPU of a job is strongly correlated with number of simulation steps
  - Production thresholds (cuts)
  - Step limitations from energy loss processes
  - Step limitation from multiple scattering process
  - User step limitation
- CPU optimisation is (nearly) always a compromise with accuracy

V.Ivanchenko

Visible energy in EM calorimeters and CPU performance

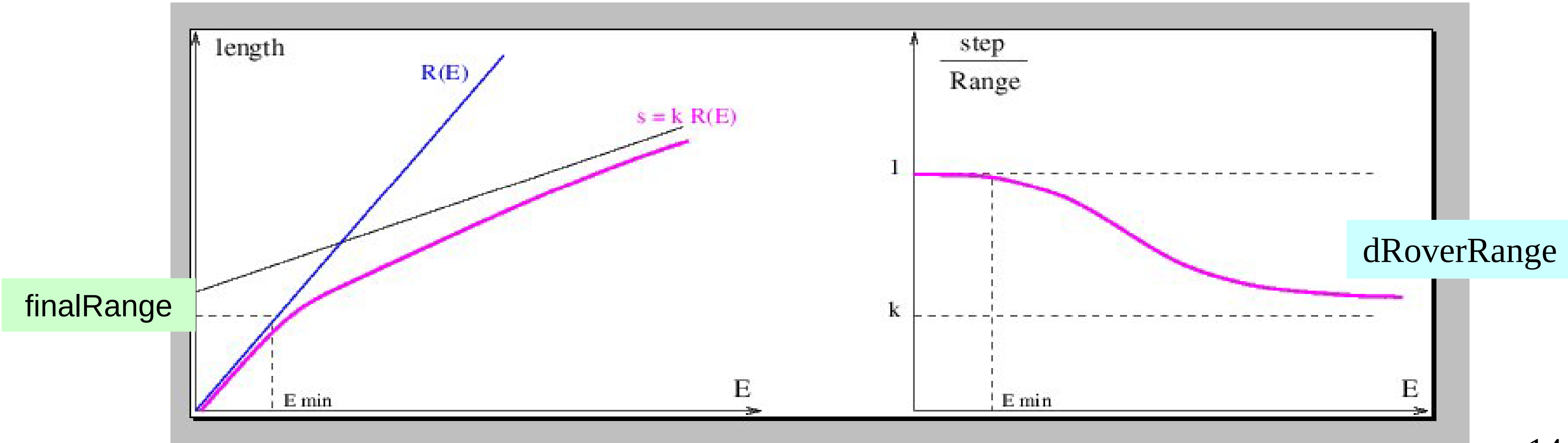


V.Ivanchenko et al., CHEP'07, Victoria, Canada, Sept. 2-7, 2007

Step limitation from continuous energy loss

- The cross sections depend of the energy. The step size must be small enough to ensure a small fraction of energy loss along the step : $$\frac{\mathrm{step}}{\mathrm{Range(E)}} \leq \mathrm{dRoverRange}$$
- This constraint must be relaxed when E → 0

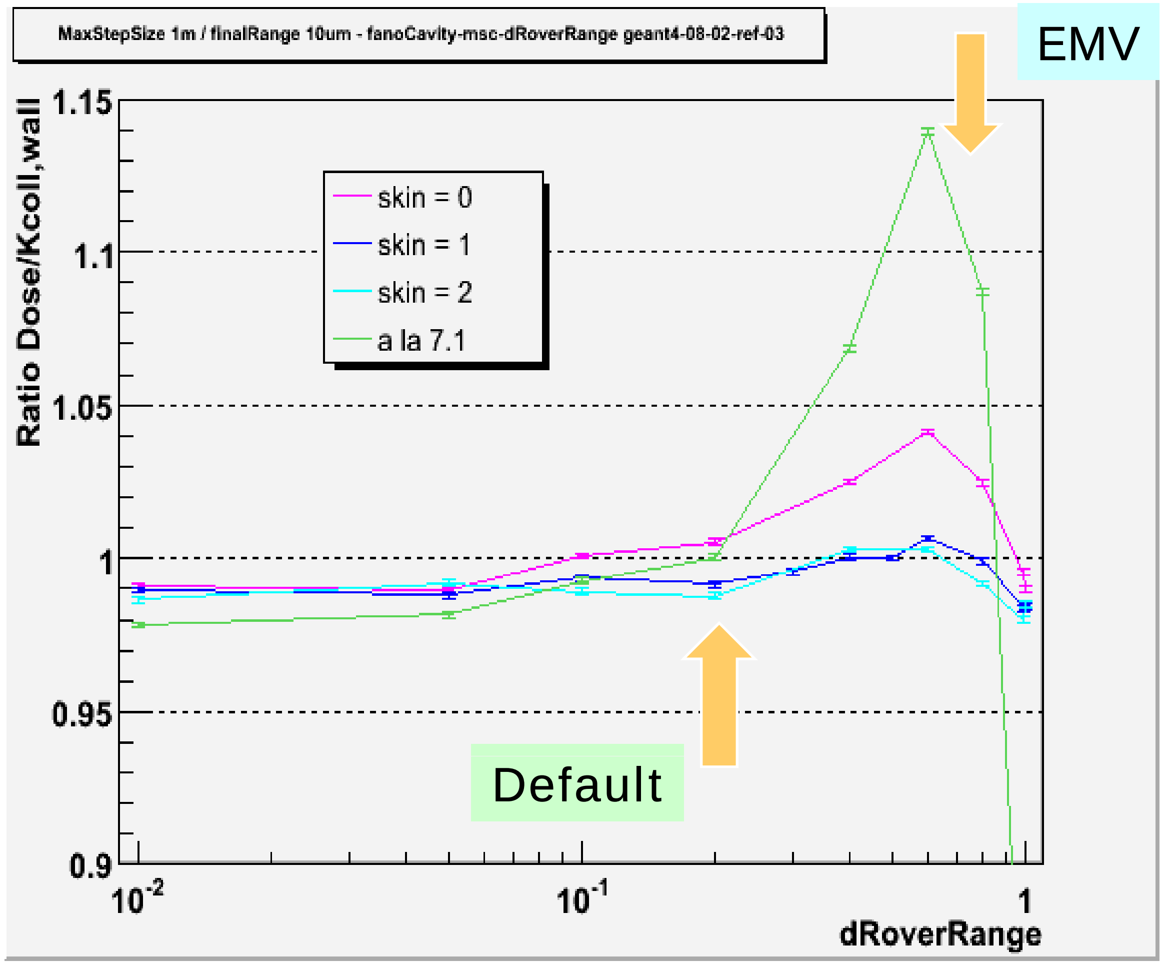
M.Maire



Fano Cavity test of e⁻ transport

S. Elles, M. Maire

- 1 MeV gamma beam in water with cavity of water-gas
- The absolute prediction of the dose deposition inside the cavity
- Significant deviation for EMV Physics List



Multiple scattering options

- **G4MscStepLimitType**
  - Minimal - equivalent to the original algorithm, as implemented in Geant4 7.1 and earlier releases.
  - UseSafety - the current default, which makes use of the geometrical safety
  - UseDistanceToBoundary - the most advanced, recommended for accuracy in the cases where no magnetic field is set

- **Multiple scattering configuration**

```
/process/msc/StepLimit                    UseDistanceToBoundary
/process/msc/LateralDisplacement          true
/process/msc/RangeFactor                  0.02
/process/msc/GeomFactor                   3.5
/process/msc/Skin                         2
```

V.Ivanchenko

Some tips to consider

- We are making our best effort to improve the speed of Geant4 toolkit. But, since it is a toolkit, a user may also make the simulation unnecessarily slow.
- For general applications
  - Check methods which are invoked frequently, e.g. UserSteppingAction(), ProcessHits(), ComputeTransformation(), GetField() etc.
  - In such methods, avoid string manipulation, file access or cout, unnecessary object instantiation or deletion, or unnecessary massive polynomial calculation such as sin(), cos(), log(), exp().
- For relatively complex geometry or high energy applications
  - Kill unnecessary secondary particles as soon as possible.
  - Utilize G4Region for regional cut-offs, user limits.
  - For geometry, consider replica rather than parameterized volume as much as possible. Also consider nested parameterization.
  - Do not keep too many trajectories.
- For relatively simple geometry or low energy applications
  - Do not store the random number engine status for each event.

M. Asai

Geometry/Detector Description (1)

- The way in which geometry is defined in Geant4 allows the possibility to very optimised or clearly sub-optimal
- Care should be made on organisation of physical volumes
- The navigator "prefers" a hierarchical approach – i.e. a tree. Flat designs should be avoided, but the number of daughters per volume should also not become excessive
- Replicas are better than placements are better than parameterisations!
- Too many placements are not good
- Too complicated parameterisations should also be avoided
- Replicas are always good (if of regular geometry)
- Assemblies are convenient but are less performant than placements
- Choice of solids by complexity (polycons, polyhedra, traps, toruses, twisted solids, …)

Geometry/Detector Description (2)

- Definition of detector regions – can help significantly with cuts, fields etc…
- Geometry optimisation - can give memory and initialisation issues - scanning the tree
  - 3-D voxelisation (significant improvement, if possible)
  - Smartless factor – number of divisions, if you have a granular geometry with large spaces, optimisation will make things worse
  - Run/verbosity 2 should tell you the output of the geometry optimiser
- Choice of navigation (voxelised or not, navigation for regular structures (phantoms) - Pedro's talk earlier, …)
- Geometry correctness (avoid overlaps – can lead to stuck particles)
- Avoid rebuild of whole geometry in dynamical setups
- Switch off optimisation for part of geometry that's moving

Magnetic field tuning

- ConfigureForTrack()
  - Change parameters of propagation in field
  - accuracy vs. particle vs. energy
- Choice of steppers
  - E.g. G4HelixMixedStepper
  - G4RKG3_Stepper (calls the field one less time per step)
- Proper definition of zero field (e.g. example N02)
- Proper definition and access to user-defined field-maps
- Setting field by region
- Choosing algorithm by region
- Set local field
- Multiple scattering options (skin = 0 - default)

Tatiana Nikitina

User Actions

- It's not always appreciated that user actions can be used inappropriately – the UserSteppingAction is called every step UserTrackingAction for every new track…
- Use of allocators for hits/tracks…
- Sensitive detectors, hits collections
- Primitive scorers (could save memory and I/O)

Configuration Issues

- G4_NO_VERBOSE – could give a 15% speed up
  - Less info
  - Less error reporting
- Run-time commands to reduce/switch off verbosity
- Avoid G4DEBUG and use default optimised builds
- Use of a performant compiler (gcc 4.X series)
- Use of a performant system (Mac, iPhone, iPod, multi-cores)

- Static libraries (single executable) vs. shared
- Benchmark issues – care about system should be made:
  - Standalone system, local disk, single user, stability…
  - Avoid afs, virtual cpu's/hyperthreading, memory swapping
- Profiling can also help a lot

Summary/Final Remarks

- …This talk is by no means exhaustive
- There are a very large number of ways an application can be speeded up
- There are an equally large number of ways an application can be slowed down!
- The documentation and release notes can be very helpful
- Comparison with data and the application output should always be considered…