

Status of ATLAS Simulation

Zach Marshall (Penn)

On behalf of the ATLAS Collaboration

G4 Tech Forum

26 March 2013

Current Production

- ATLAS is still in “MC12” production mode
 - ~3B events with G4 sim (many w/shower libraries in the forward calo)
 - ~3.5B events with fast calorimeter sim (G4 in tracker and muon sys)
- Production using G4 9.4 + (official and private) patches
 - Supporting both 32 and 64-bit, gcc 4.3, SLC5
 - Expect to stay in this configuration or similar through early 2014
- Many code migrations on the horizon
 - Expect a move in early 2014 to G4 9.6
 - Expect a move in next few months to gcc 4.7, C++11, SLC6, 64-bit only (possibly), CLHEP 2.1 (no internal G4 CLHEP to be consistent with reco)
 - Testing most of these platforms already, and some are quite advanced
 - Also testing icc, clang, mac builds (but no production plans yet)
- We know moving to G4 9.6 ‘requires’ moving to FTFP_BERT
 - Testing underway, first results soon. Trying this physics list move in G4 9.4 first, to decouple with the move to G4 9.6
- G4MT Trial / prototype in testing, almost running normally
 - Development within the context of our new ISF
 - Likely to become more mainline when G4MT is integrated into G4

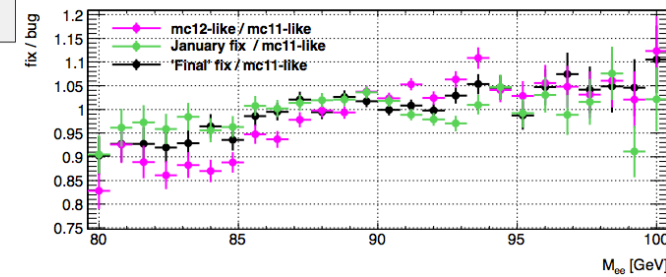
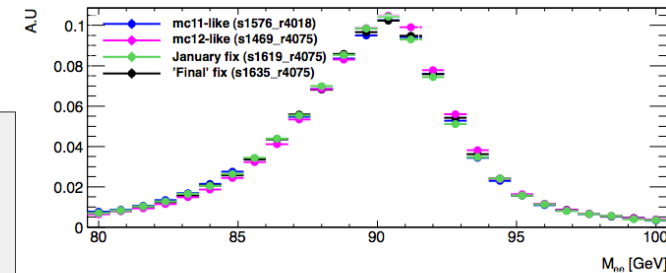
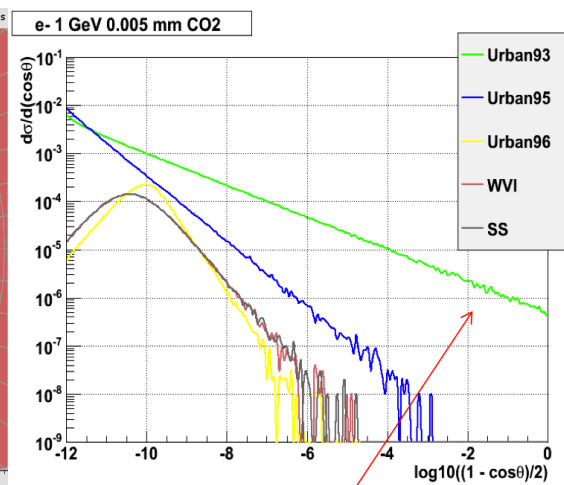
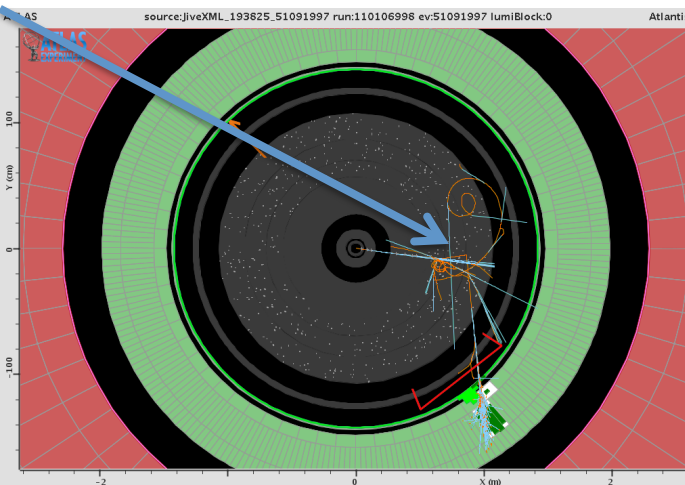
Bugs and Crashes in G4 9.4

- Rather complex subtractions cause problems
 - One example here – the “Bar connecting box”
 - Obvious sometimes (crashes)
 - May also be less obvious (particles and moving for km inside the volume, only crashes if a secondary is placed outside of the world)
 - So far worked around by re-implementing problem solids
 - Now able to dump ATLAS geometry to GDML and provide some volumes directly to the G4 team – debugging subtraction solids now
 - Have previously noted a dependence on placement – smells like a tolerance / numerical accuracy problem?
- Other bugs are more rare, but there are others around
 - Crash following large energy non-conservation
 - [Apparently] looping jobs that have not been fully diagnosed
 - [Accurate crash rate not yet available ☹️☹️. Upgrades to our production system will make these numbers easier to get out soon!]



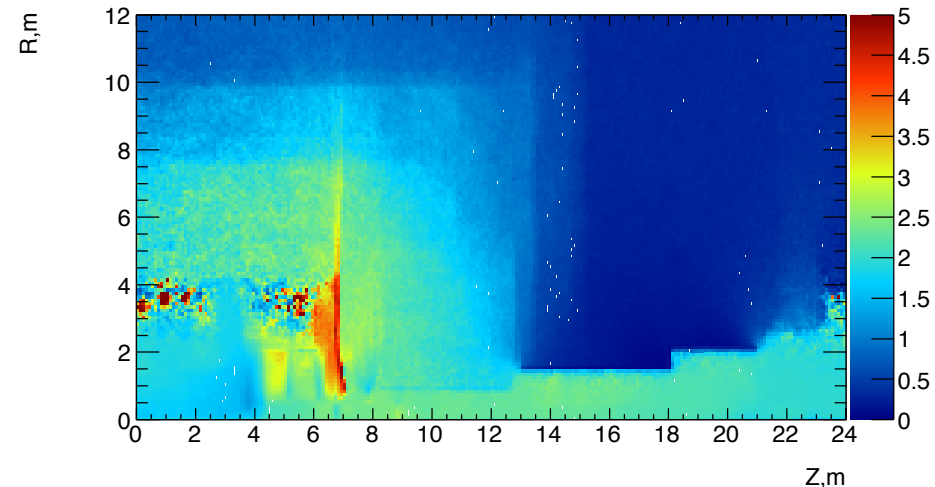
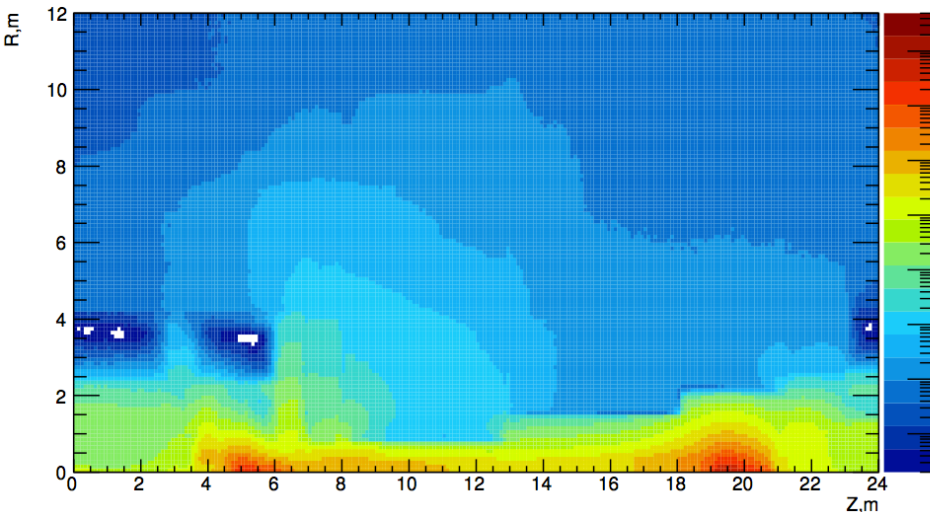
Electron Saga

- Roughly 18 months ago, found rare cases of HUGE scatters of electrons
 - 10 GeV electrons turning 110° in a small step in gas
- Caused by using a parameterization outside its range of applicability
 - Also influences the $Z \rightarrow ee$ line shape, which is concerning for physics!
- Simple fix to cut off parameterization was not sufficient
- Has caused us to investigate G4 MCS and brem models
 - Now attempting to run several different configurations of G4 EM physics with different MCS and Brem models included
 - Will report back soon to developers with which changes significantly affect line shapes, shower shapes, data/MC scale factors, etc



Cavern Background

- We've been exploring the implementation of a cavern background simulation entirely in G4
 - FLUKA and other programs have historically dominated this area
- Testing revealed some geometry features which are now fixed
 - Of course the parts of the geometry that are relevant to cavern background are not all the same as those relevant to, e.g., Higgs physics
 - Work on boronated polyethylene and electronics boxes to improve agreement
 - Below are neutron fluxes in QGSP_BERT_HP in kHz/cm² and the ratio from before and after one of the latest geometry changes (and a change from 7 TeV to 8 TeV)
- Comparisons are now more favorable, and we'll soon compare to data
 - Expect another useful round when we get to G4 9.6 and the new data libraries



Bugs and Crashes in G4 9.6

- Saw some strange behavior in the calorimetry
 - Particles seemed to get lost and step into the mother volume
- Traced back to strange behavior in EM Physics and navigation
 - Occasionally particles seem to have a conflict when they arrive at the edge of a boundary at almost the same time that they are meant to undergo some physics process
 - Moves the particle up in the geometry hierarchy?
- G4 can do what ever it likes in terms of geometry navigation, of course. This worries us for a few reasons:
 - We can't prove (yet) that it doesn't miss a step in the LAr because of this
 - We can't prove that this isn't a waste of sim time
 - We can't prove that similar things aren't happening in many places, and that we only notice these few cases that are obvious
- Once this is resolved, we see G4 9.6 as ready for computing and physics validation (barring some other problem, of course)

Integrated Sim Framework (ISF)

- ISF uses an ATLAS particle stack, handing particles to G4 or fast simulation according to user definitions
- It is moving into production soon for *all* ATLAS simulation
 - Much of this production will be running a “G4 for everything” config
- Current implementation uses one G4Event per particle
 - Allows us to do either Athena event level parallelism or G4 event level parallelism (sub-event parallelism)
 - However, overheads can be costly
 - Moving to an implementation that can take multiple particles and construct a G4Event from them
 - This implementation requires a distinction between G4Events (one or a few particles) and ATLAS/Athena events (potentially many G4Events)
 - Requires modification to event-level actions (e.g. sensitive detectors, user actions, event analysis)
- Considering taking over the G4 event loop instead
 - Would implement a the event-loop that G4RunManager calls
 - Will discuss more with the G4 team if this becomes a serious effort

More Plans for the Year

- Significant changes to simulation infrastructure will be attempted during LS1
 - Of course, this is not a “simulation shut-down”, so we have to keep a production version running at all times (quite challenging)
 - For example: use of hadronic physics modules in G4 for modeling hadronic interactions in fast sim (calls directly to G4 modules!)
- Will be re-visiting benchmarking, bottlenecks, etc in light of all the changes that have been made in the last few years
 - Last detailed benchmarking effort was several years ago now!
 - Since we have significantly improved the magnetic field access (stepper improvements, caching within the mesh granularity, call-chain shortening), which was one of the major hot spots
 - Of course, it only makes sense to benchmark a version once it is roughly bug-free and physics-validated!
 - This work will include work on “standard” performance improvements as well as vectorization, parallelization, multi-threading, etc etc