# Scientific Software for Simulation

I have a collider focus, and there are likely some issues for astrophysics, cosmology, and IF experiments that I don't know about.

"Scientific" == directly involves physics (particles, interactions)

1. Detector simulation (mainly GEANT) [particles with matter]
2. Event simulation (Pythia, etc.) [production of particles for 1.]
3. Lattice (nothing to say)
4. Accelerator modelling (nothing to say)

Gabe: "What are the major issues and challenges, what strategies should we pursue, are there things we should stop doing, etc."

# Detector Simulation

Major Issues:

1. LHC experiments expect 150 times more data. (more)
2. High luminosity environment, with up to 200 pileup interactions per event. (busier)
3. Hadron-hadron interaction models need to be tuned and improved, and systematic errors must be evaluated
4. Common codes have become fractured over time
5. Development of common “fast” (parametrized?) simulations

# Detector Simulation-2

Imperative to find solutions to speed up the generation and reconstruction of simulated events.
Speed is not going to come as a matter of course (buy faster machines)

Strategies to pursue: (mostly already being done)

1. Write “smarter” code (review and optimize) -- takes time, patience, focus
2. Exploit processor details:  multi threaded track-level (particle-level) parallelization, improved instruction pipelining, vectorization and single instruction multiple data (SIMD) architectures, and data locality.
3. Adopt tools from the generator tuning community to aid in model-fitting

# Event Simulation

Don’t forget that our goal is to understand/interpret the data.

Major issues:

1. Theoretical accuracy of codes must improve to match quantity of data
2. Codes must be able to run effectively on grid/Cloud/…
3. Predictions must be tuned to data with quantifiable error estimates
4. GENIE isn’t able to exploit many theoretical calculations
5. No common, common tools (Phase Space, ME generation)
6. Tuning greatly improved, but not quite automated, adaptive
7. No real plan to preserve codes
8. Not enough young MC authors, students, post-docs

# Event Simulation-2

Strategies to pursue:

1. General evaluation of codes for bottlenecks, inefficiencies, large memory footprints (e.g. some NNLO calculations have expressions that are too big)
2. Understand where GPUs, multi-threading, MPIs could be exploited
3. Increase communication between computing experts and MC authors (e.g. aMCatNLO has large number of negative weights -- is there a smarter algorithm?)
4. Develop accords with theorists on how to present calculations (what is the leverage to teach old dogs new tricks?)
5. Develop common (community-developed, parallelizable codes/algorithms (e.g. a replacement for VEGAS)

# Event Simulation-3

Things to stop doing:

1. Not redesigning code to exploit grid/Cloud, etc. (e.g. Madgraph grid-packs have a huge number of directories)
2. Saying that speed doesn’t matter, because GEANT is 1000x slower