Geant4 10.6 beta

Choosing a Physics List

Geant4 Tutorial at Chalk River

Dennis Wright (SLAC)

30 August 2019

Outline

- Review
  - Physics lists, reference physics lists, naming convention
- Some application-based recommendations
- Using validation to choose your physics list
  - Example

Physics List

- An object responsible for:
  - specifying all particles to be used in a simulation application
  - specifying physics processes and assigning them to each particle type

- One of three mandatory objects that the user must provide to the G4RunManager in any application
  - tells run manager what physics needs to be invoked and when

- Provides a very flexible way to set up the physics environment
  - user can choose and specify particles he wants
  - user can choose the physics (processes) to assign to each particle

- BUT, user must have a good understanding of the physics required to describe the problem
  - omission of relevant particles and/or physics interactions could lead to poor modeling results

Reference Physics Lists

- Also called "Production Physics Lists"
  - used by larger groups like ATLAS, CMS, etc.
  - well-maintained and tested
  - very stable: not often changed and usually updated only for bug fixes
  - extensively validated inside and outside of Geant4
  - FTFP_BERT, QGSP_BERT, QGSP_FTFP_BERT_EMV, FTFP_BERT_HP, QGSP_BIC_EMY, QGSP_BIC_HP, QBBC, Shielding
- Caveats:
  - these are provided as a "best guess" of the physics needed in certain use cases
  - intended as templates or starting points
  - if you decide to use them, you are responsible for validating them for your application
  - this may mean adding or removing physics or changing settings

Reference Physics List Naming Convention

- Hadronic options
  - "QGS" Quark Gluon String model (>~ 15 GeV)
  - "FTF" FRITIOF String Model (>~ 5 GeV)
  - "BIC" Binary Cascade model (<~ 10 GeV)
  - "BERT" Bertini Cascade model (<~ 15 GeV)
  - "P" G4Precompound model used for de-excitation
  - "HP" High precision particle (neutrons and some charged particles (<~ 20 MeV)

- Electromagnetic options
  - no suffix: standard EM (default G4EmStandardPhysics constructor)
  - "EMV" G4EmStandardPhysics_option1 (fast, less precise)
  - "EMY" G4EmStandardPhysics_option3 (precise, used for medical and space)
  - "EMZ" G4EmStandardPhysics_option4 (most precise, slower)

- Name decoding: String_Cascade_Neutron_EM

- Complete list of pre-packaged physics lists (with detailed descriptions) at
  - http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/PhysicsListGuide/html/index.html

Recommended Physics Lists for Some Applications

- FTFP_BERT
  - HEP applications
  - FRITIOF string + Precompound deexcitation + Bertini cascade + standard EM
  - can add _HP if neutron flux is important
- Shielding
  - for shielding and space applications
  - very similar to FTFP_BERT_HP, but with better ion-ion interactions
- QGSP_BIC_HP
  - medical applications
  - Quark Gluon String + FRITIOF + Precompound + Binary cascade + high precision neutron + standard EM
  - can add best precision EM by appending “EMZ”

Changing the EM Physics in a Reference Physics List

- QGSP_BIC_HP_EMZ
  - QGSP_BIC_HP is a reference physics list with standard EM
  - you can change this by using the G4PhysListFactory
    - knows all available reference physics lists and makes EM substitutions easy

```
// IM YOUR MAIN APPLICATION
//
  // create your run manager
#ifdef G4MULTITHREADED
  G4MTRunManager* runManager = new G4MTRunManager;
  // number of threads can be defined via macro command
  runManager->SetNumberOfThreads(4);
#else
  G4RunManager* runManager = new G4RunManager;
#endif
  //
  // create a physics list factory object that knows
  // everything about the available reference physics lists
  // and can replace their default EM option
  G4PhysListFactory physListFactory;
  // obtain the QGSP_BIC_HP_EMZ reference physics lists
  // which is the QGSP_BIC_HP refrence list with opt4 EM
  const G4String plName = "QGSP_BIC_HP_EMZ";
  G4VModularPhysicsList* pList = physListFactory.GetReferencePhysList(plName);
  // (check that pList is not nullptr, that I skipp now)
  // register your physics list in the run manager
  runManager->SetUserInitialization(pList);
  // register further mandatory objects i.e. Detector and Primary-generator
  ...
```

Choosing Your Physics List

- Ideally, user has a good understanding of the physics relevant to a given application
  - user can then build his own or decide on a pre-built one
  - in either case the physics list must be validated for the application
  - during the validation, some changes to the physics list may be required
- If your application fits within a well-defined area (e.g. medical)
  - used may choose a physics list used in that area as a starting point
  - validation, once again, is required
- Procedure that always works, but is time-consuming
  - start with most accurate physics (e.g. EMZ for EM)
  - run the simulation with lower statistics to obtain the most accurate result
  - if desired, choose a less accurate, but faster, physics list and run some more simulation statistics
  - modify in a granular way the first physics list using the most accurate results as a guide

Validation

- Physics list must always be chosen based on how well its components perform in your specific case
  - always balance physics accuracy with CPU performance

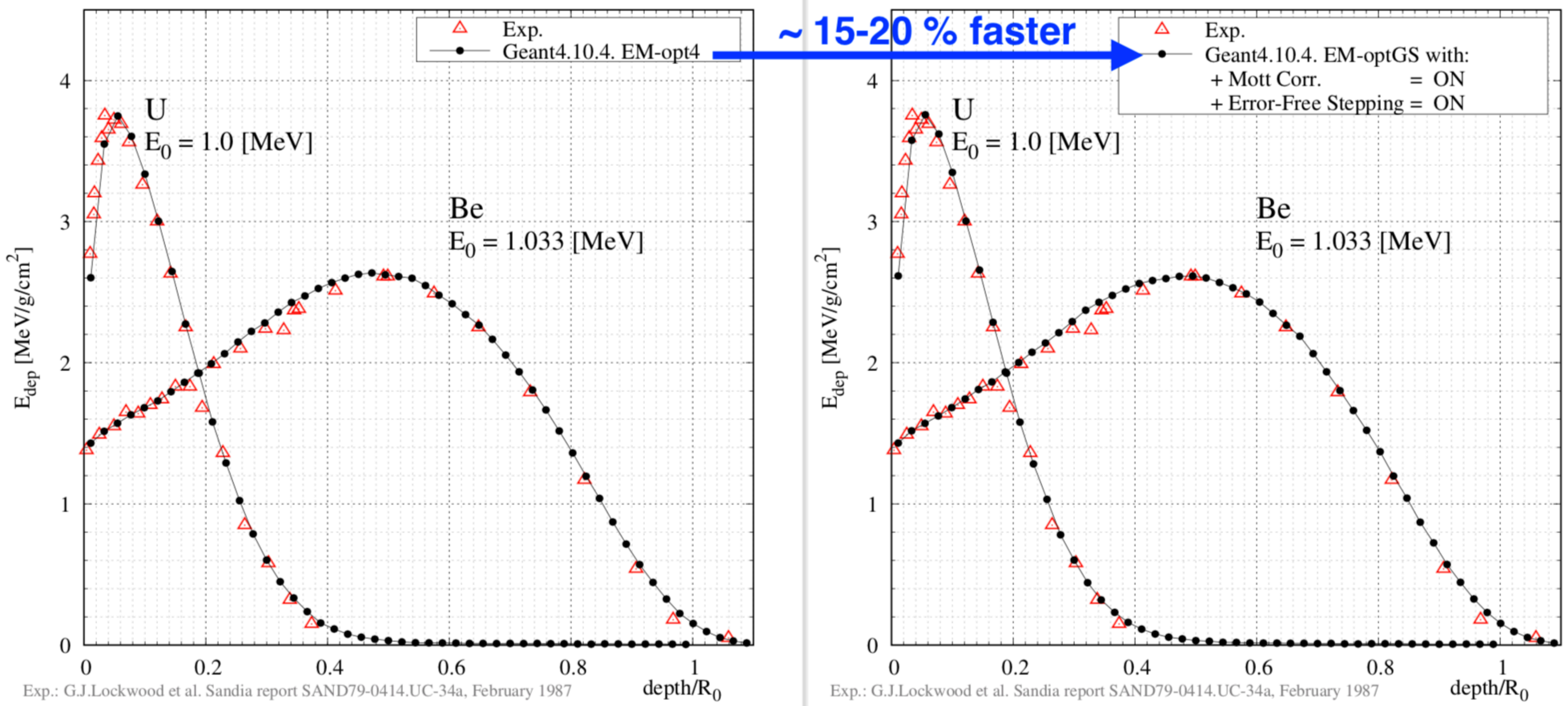
- Geant4 provides validation (comparison to data) for most of its physics codes
  - validation is a continuing task
  - performed at least with each release
  - more validations added with time

- Access these comparisons at the Geant4 website:
  - https://geant4.web.cer.ch
  - Click: Validation of Geant4
  - Choose Validation and Testing from the menu
  - today we'll look at the Geant4 GRID-based testing results portal

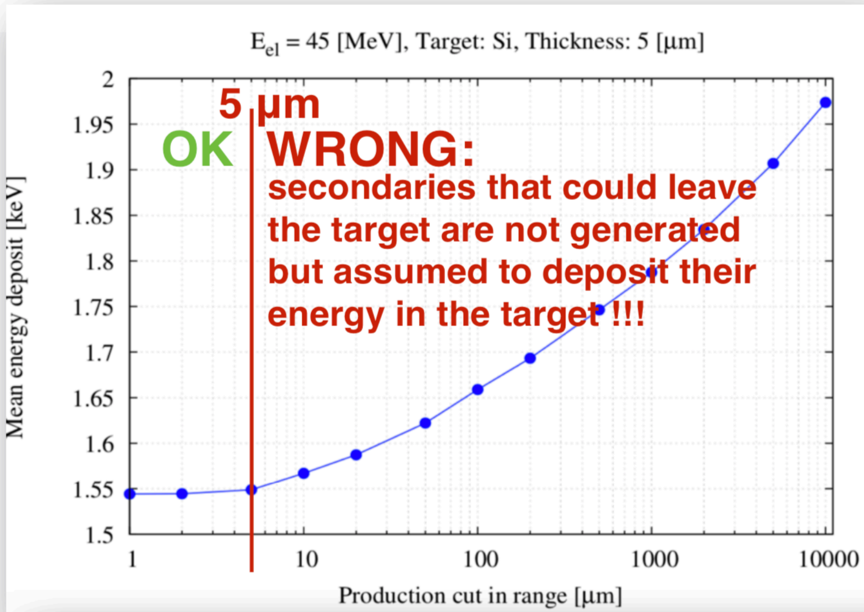
Validation Example

- Simulating (EM) depth dose profile
  - energy deposit by energetic electrons as a function of penetration depth in both lighter and heavier materials
  - use Geant4 validation results from the Geant4 GRID-based testing results portal, especially test37, to choose an initial physics list
  - then adjust initial reference physics list to achieve maximum physics performance while improving computational efficiency

Validation Example



And Always Check the Production Threshold



cut [μm]	mean E_{dep}	rms E_{dep}	prod. thres. [keV]		mean num. sec.	
			γ	e^-	γ	e^-
1	1.54423	0.000573911	0.99	0.99	0.0006811	0.1018230
2	1.54443	0.000583879	0.99	2.9547	0.0006843	0.0316897
5	1.54882	0.000605834	0.99	13.1884	0.0006857	0.0068261
10	1.56717	0.000665733	0.99	31.9516	0.0006730	0.0028232
20	1.58734	0.000743473	1.08038	47.8191	0.0006651	0.0018811
50	1.62223	0.000912408	1.67216	80.7687	0.0006557	0.0011304
100	1.65893	0.001108240	2.32425	121.694	0.0006518	0.0007536
200	1.69338	0.001342180	3.2198	187.091	0.0006465	0.000477
500	1.74642	0.001774670	5.00023	337.972	0.0006184	0.0002617
1000	1.78751	0.002219870	6.95018	548.291	0.0006054	0.0001622
2000	1.83440	0.002861020	9.66055	926.09	0.0005786	9.3e-05
5000	1.90700	0.004243030	14.9521	2074.3	0.0005427	4.07e-05
10000	1.97378	0.006036600	20.6438	4007.59	0.000521	2.22e-05

Compute the mean of the energy deposit (E_f-E_0) in the target