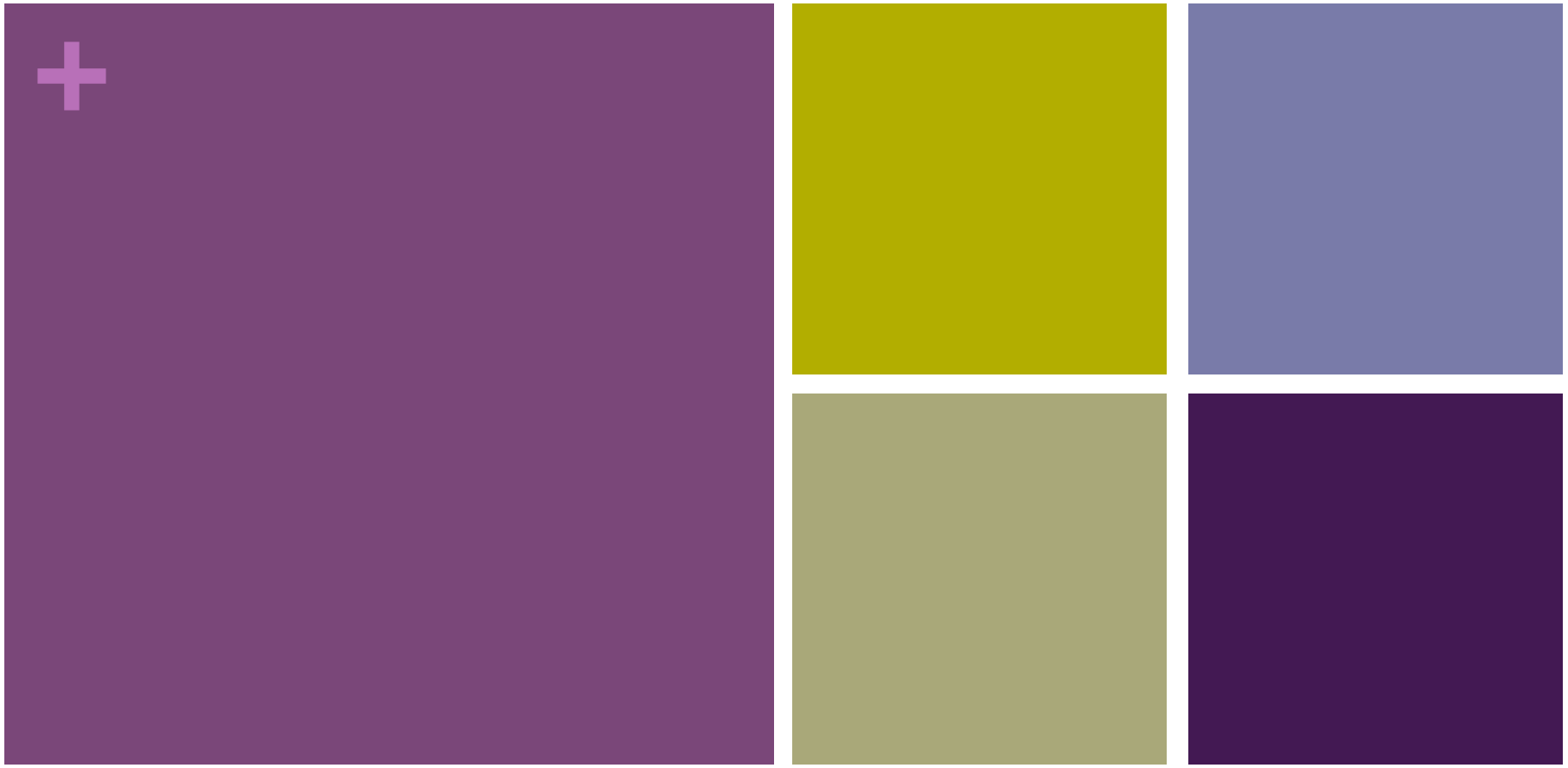


Geant4 plans

J. Apostolakis

Topics

- Geometry
- Propagation in Field
- Multi-threading

- Focus ‘technical’ parts – all parts except physics modeling

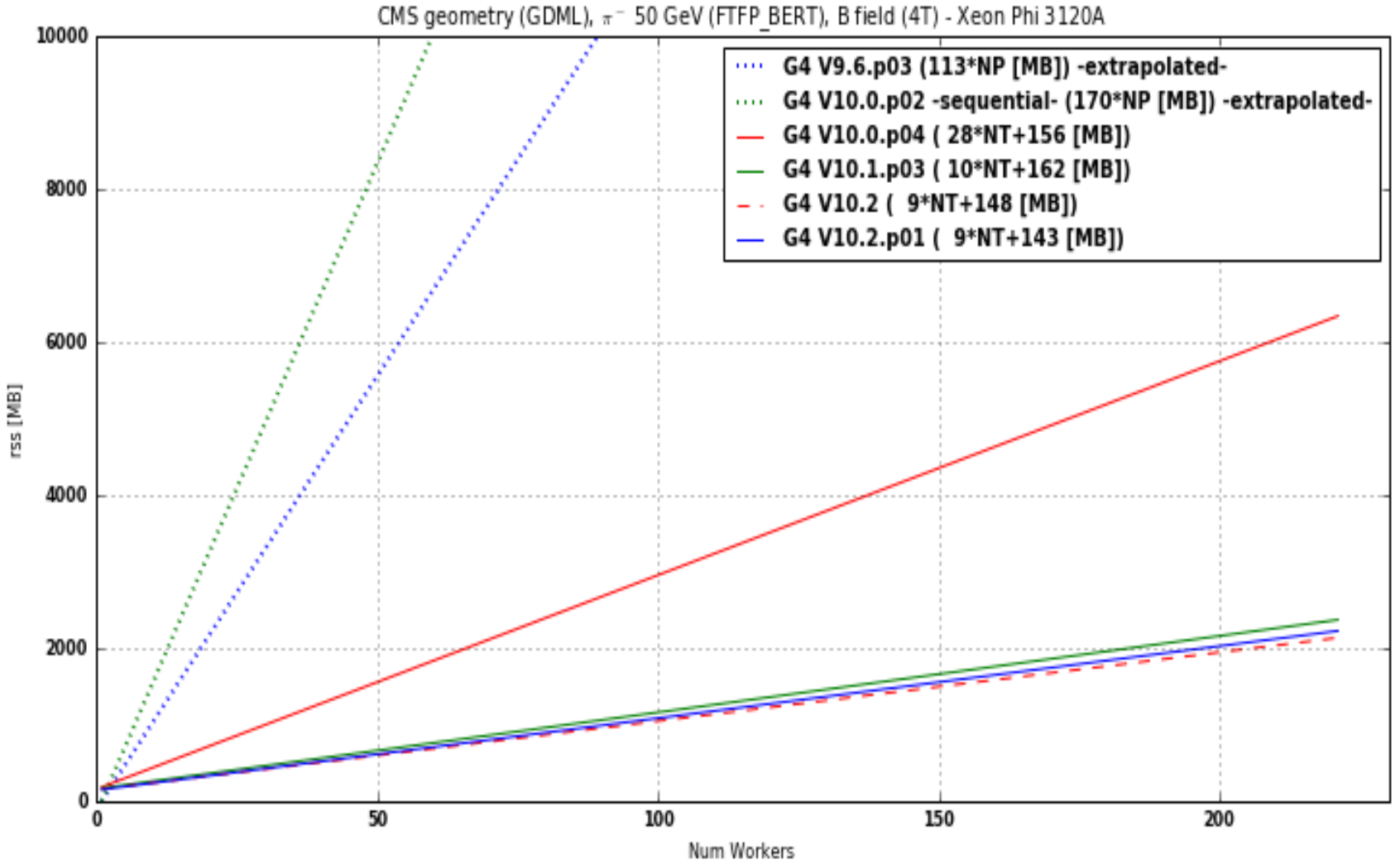
Last and next Geant4 release

- Release 9.6 (released Dec 2012)
  - Last patch - 30 January 2015 (patch-04)
- Release 10.1 (released 6 December 2014)
  - Patch-03, released 5 February 2016 (source archive)
  - Expect ATLAS will use it in next MC version
- Latest release Geant4 10.2 (Dec. 2015, notes , highlights)
  - patch-01, released 26 February 2016 (download latest)
- Beta release 10.3-beta (scheduled 30 June 2016)
- Full release 10.3 (scheduled 3 December 2016)

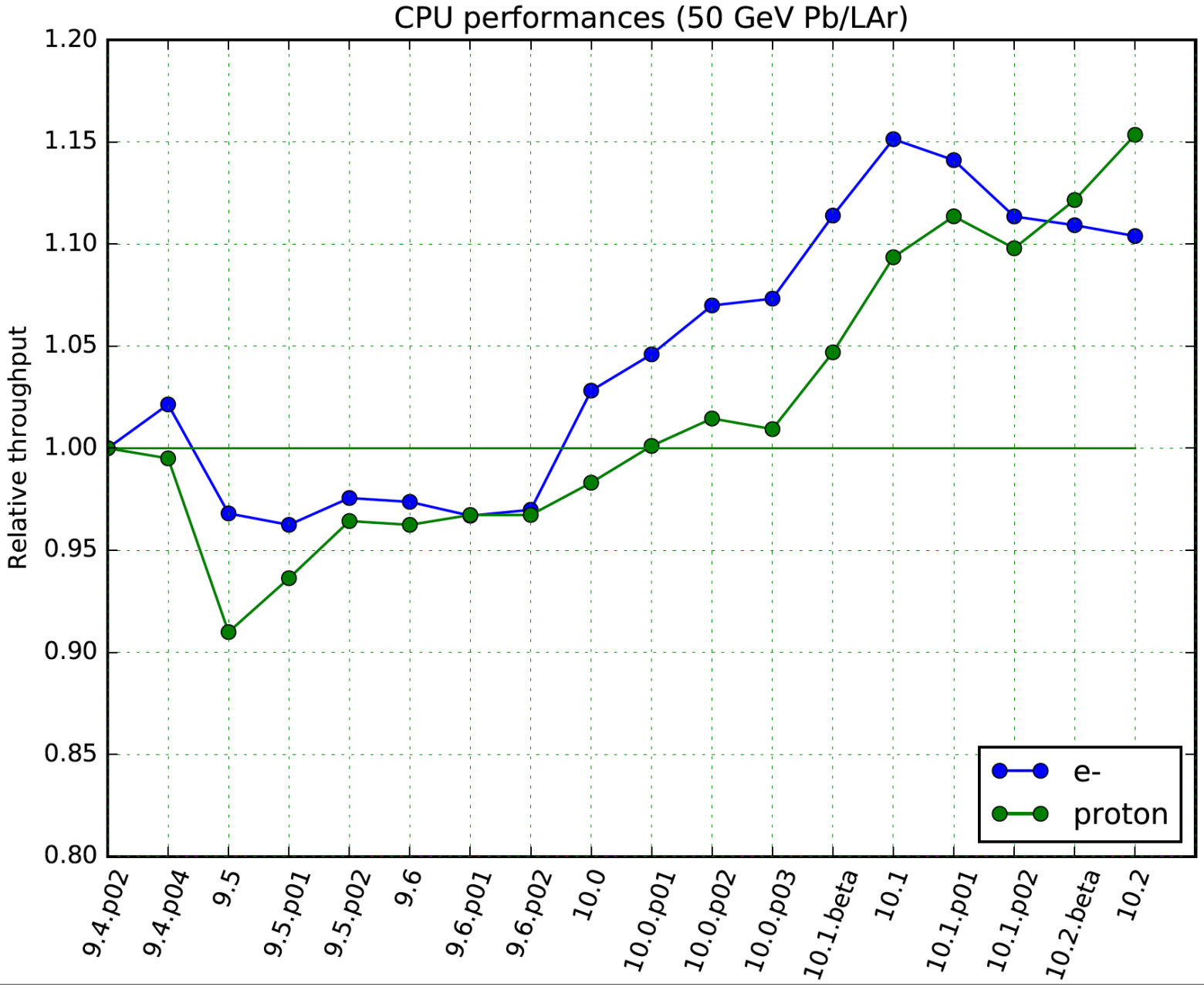
Recent evolution

A few highlights from Release 10.2

Memory consumption



Computing performance in sequential mode



New RNG class

- New class of pseudo-random number generator 'MIXMAX'
  - Provided in most recent releases of CLHEP (since 2.3.1)
  - Theory-based guarantees of quality similar to RANLUX at higher/ highest quality levels
  - Very fast de-correlation of most nearby sequences
  - CPU speed: as fast as Merseinne Twister (in which nearby sequences stay together for many steps)

USolids: new Solids modeler

- USolids project to merge development of Shapes / Solids for Geant4 and Root (started ca 2011)
- Geant4 10.1 included a first internal release of USolids
  - an old snapshot
  - Installation choice: Usolids to replace some or all G4 solids
- Geant4 10.2: can use external USolids installation
  - USolids extended with vector interface for GeantV into 'VecGeom' – common Solids library (+navigation)
  - New vector-capable implementations for most methods/solids

VecGeom – the future of Solids

- Potential - try out USolids/VecGeom, the future Solids library
  - Current version is tested using CMS geometry GDML (in GeantV)
  - Needs further fixes and improvements to become robust enough for production use
- Your involvement is sought:
  - first providing up-to-date Atlas GDML geometry file,
  - Before long trying out upcoming VecGeom releases with ATLAS simulation!
    - CMS and ALICE have expressed interest to evaluate it for its performance improvement potential in Geant4

Propagation in Field

Plans for next release - Geant4 10.3

Plans for 10.3-beta (June) & full release 10.3(Dec. 2016)

- Full list of proposed 2016 work plans at http://geant4.web.cern.ch/geant4/support/planned_features.shtml
- Makoto's talk at March Geant4 Tech Forum summarized key development items in non-physics part of the work plan.
- For all categories :
  - Implement use of C++11 constructs in key areas
- Next slides focus on a couple of areas

Geometry & transport

- Profiling and optimization of multiple navigation
- Separate safety computation from navigator class
- Check the evolving use of navigation ‘services’ by the multiple-scattering process
- Introduce a scaled shape construct
- Complete implementation of the Unified Solids library with progressive adoption of shapes from VecGeom

Propagation in Field

Propagation in field - Intro

- Field propagation is a significant consumer of CPU
  - Order 5-20% of CPU in integration of motion
  - Depends greatly on complexity of field evaluation
- Field evaluations generally are amongst the costliest methods in G4 simulation
  -

Field propagation

- Field – typically expensive to evaluate
- Equation of Motion [usually trivial $d\mathbf{p}/ds = \mathbf{F} = q\,(\,\mathbf{v} \times \mathbf{B}\,)$]
- Integration Method – '**Stepper**' estimates end-value (x,p) & error
  - Existing steppers - types of Runge-Kutta ODE integr. methods
  - Stages: # of derivative/field evaluations per step (including at start)
  - Order – last 'good' expansion polynomial order
- Driver – enforces error control
- ChordFinder – breaks trajectory into pieces with saggita < limit ('miss distance')

Integrating ODE of motion

- Old type methods – no 'native' error estimation. Error: split step in half
  - ClassicalRK4 – Stages: 4, Order: 4. With error estimation: 11 field evaluations
  - Heum – stages: 3, order 3 ; with error estimate: 7 field evals
- Embedded RK method – error estimation using difference of 2 methods
  - CashKarpRKF45 – stages: 6, order 4 & 5 [Aside: fix included in 10.3-beta]
- Specialised steppers only for pure magnetic field
  - Helix-based steppers – uses known exact solution for constant field
    - 'ExactHelix' – 1 evaluation => no error
    - 'HelixExplicitEuler' – 2 evals, simple error estimate
  - ATLAS's Nystrom stepper (G4AtlasRK4): 3 field evaluations per (trial) step
    - Relies on custom fast method of estimating error, using only a numerical estimate of fourth derivative of position/momentum

Field propagation

- 10.3-beta introduce faster and higher-order RK steppers – several embedded RK method with robust error estimation (can use as 'drop-in' in older G4 ver):
  - **BogackiShampine23** – stages: 3, orders 3/2, FSAL, Dense
  - BogackiShampine45 – stages: 7, orders 5/4, FSAL, Dense
  - **DormandPrince** 7 4/5 – stages: 7, orders 5/4, FSAL, Dense
- There are the methods used in the 'industry standard' RK ODE codes – DOPRI5 (Hairer), RKSUITE (Shampine et. Al) – and also the basis of MATLAB and other problem solving suites for general, non-stiff ODEs
  - Robust error estimation, fewest field evaluations for required accuracy
- Advanced capabilities (available, but need changes further up to use)
  - Provides derivatives at final point – **First Same As Last (FSAL)**
    - Avoids call for derivative at start of next step (if continuing)
    - Potential return also the final point's field value
  - **Dense** output: obtain estimate of 'y' integrand at any intermediate value of 'x' – uses few calls to field once for all intermediates

Field: issues and unused capabilities

- Issue of robustness – track which misses boundary of calorimeter, and keeps going thinking it is still inside
  - Seen in 1 of 10^6 events on Berkley supercomputer – reproducible
  - First the track gets stuck at boundary of Calo for ~20 steps
- Our understanding is that the most recent set of fixes reduced the frequency of this problem (provided in G4 10.2, retrofitted into 10.1 patch 03)

Field: unused capabilities

- Can change 'stepper' method (& accuracy parameters) depending on track (particle type, momentum, possibly also location) using user-exposed method (deriving from G4FieldManager)
- Improved CPU / accuracy compromise by selecting different target for different tracks
  - higher requested accuracy (& potentially high-order stepper) for 'signal'/important tracks,
  - Very low accuracy (& maybe even simpler method) for the least interesting tracks
  - Medium accuracy for the rest (likely using existing choice).

Multi-threading

Multithreading: topics

- Technical topics related to
  - Integration in experiment frameworks
  - Robustness & repeatability
  - Computing performance
- Concretely:
  - Multithreading and adapting to parallel frameworks
  - Adding Physics Lists
  - Improving propagation in field

Multithreading

- G4UserWorkerInitialization
- G4MTRunManagerKernel.cc (line 105) contains the recipe to configure a thread
  - Calls methods that undertake TLS setup
  - No POSIXthread specific calls
  - Method re-engineered in 10.2 with granularity of calls
  - Migrated to use workspaces in 10.1

MT: Avoiding confusion

- We provide Geant4 out of the box with 'pthreads' initialiation of worker threads – as a solution for standalone programs and the ordinary users (not big experiments!)
- The elements required to work with other //-isation systems:
  - Initialize in a 'Master' thread before any 'Worker' is initialised and asked to do work
  - Ensure that the Master thread does not participate in work
  - Initalise Geant4 workers' TLS area
- Adapting G4 to a new framework
  - Figuring out how to deal with the master ('shoe-horning')
    - (Frameworks already have 'non-worker' threads for I/O, ..)
  - Adaptation of the remaining established recipe

MT: Under development

- Compete migration to use of workspaces
  - Enables reduction of memory (in case not all threads are used for G4 simulation). Expected savings is ~ 10 MB / un-used thread
  - Second example of integration of Geant4 with TBB, using workspaces.
- Add capability to start or terminate a thread between Geant4 runs
  - E.g. use 4 threads in run 0, next 5 threads in run 1,then 3 for run 3.
- Currently Scheduled for inclusion in Geant4 10.3 (Dec 2016) - but there is potential for slippage

Adding physics lists

- To address issues in Jet Energy Scale in ATLAS and CMS, we have proposed a variant Physics List: FTFP_BERT_ATL
  - Today this requires additional source code compiled into Geant4, - in order to enable its use via the G4PhysicsFactory
- In the future it would be an improvement to enable the creation of a new Physics List entirely in user code
  - A mechanism to enable this is under development (by Fermilab colleagues)
  - Potentially ready for inclusion in Geant4 10.3