

Proposed Change to the Handling of AtRest Physics Processes

Tom Roberts

Muons, Inc.

Current Algorithm / Proposal

- At present (Geant4 9.6), only a single step is taken for a track with status=`fStopButAlive`. The process returning the shortest GPIL has its `Dolt` function called; then the track is killed.
 - Bug 1397 indicates a problem with this.
- Proposal: Change the `AtRest` processing for tracks with status=`fStopButAlive` to be similar to the `PostStep` processing for `fAlive` tracks:
 - Multiple steps are taken until the track status is some value other than `fAlive` or `fStopButAlive`.
 - The process returning the shortest GPIL has its `Dolt` function called, as well as every process that forced its `Dolt` to be called.
 - There is no analog to “`AlongStep`” for stopped tracks.

Use Cases - 1

1. For collective beam effects (e.g. space charge), G4beamline has a modified RunManager that tracks all members of a `std::vector<G4Track*>` “simultaneously”:
 - Each member is tracked individually as usual, until it has `status!=fAlive`.
 - Every particle has a process that suspends the track when the next time step is reached (it uses the track’s velocity to convert time remaining to a GPIL distance). Each instance of this process gets the step time from the RunManager.
 - After looping over the vector, all members have either reached the time step or are dead:
 1. Collective physics processes are invoked on the vector.
 2. Tracks with `status=fSuspend` are changed to `fAlive`.
 3. The step time is incremented and we go around the loop again.
 - **This does not work for tracks that are stopped.**

Use Cases - 2

2. For an inverse cyclotron, which uses frictional cooling to cool and stop a muon beam, tracks stop in the absorber gas, but must be kept active until the extraction E field turns on.
 - Each particle has a process that changes status from `fStopButAlive` to `fAlive`; it also gives the track a tiny KE and momentum direction along the E field. This process is told when the E field will turn on, and its direction.
 - Decays and `MuMinusCaptureAtRest` must work properly.
 - This must work in both normal tracking mode and in collective mode (previous slide).

Implications

- I believe that few changes to existing physics processes will be required.
 - Most negative particles have two AtRest processes, Decay and *CaptureAtRest; they compete and the proposed change should work just as it does now (if the TimeStepLimiter determined the AtRest step size, the other Dolt functions won't be called).
 - Bug 1397 probably means this does not work correctly.
 - Most positive particles have a single AtRest process, Decay; stable particles have none.
 - If you disable Decay, when a mu+ stops you get an error from the SteppingManager “No AtRestDolt process is active!”
 - But somehow a proton can stop without any AtRest process – this may need to be revised, depending on how it is implemented.
 - Neutral particles are each a special case and need to be examined for AtRest behavior.