

## AbstractID: 13962 Title: Implementation of modular phase space IO in Geant4 with enhanced latch capability

Purpose: Write general-purpose, modular code for phase space input and output in Geant4 including latch and npass.

Method and Materials: Geant4.9.3 does not include the ability to read or write phase space files. These files are useful for dividing the simulation into parts, for example, into patient-independent and patient-specific parts. Stand-alone classes inheriting from G4VPrimitiveScorer and G4VPrimaryGenerator were written for writing and reading phase space in IAEA format. Latch mechanisms are not stand-alone and require implementation of (provided) G4User classes for tracking.

Results: Phase space I/O is implemented in Geant4. For writing, several types of surfaces are allowed, including planes, spherical sections, cylindrical sections, surfaces enclosing a volume. Multiple surfaces of any orientation and geometry are allowed. The concept of latch in BEAMnrc (particle interacted in a region) was expanded to include: particle interacted in a region; particle passed through a region; similarly for ancestors; type of interaction in particle's history; type of ancestor particle; and user-defined. The user defines the interactions, particles, and regions of interest via get/set functions. As in BEAMnrc, an integer npass records the number of times a particle crosses a phase space surface, which is particularly valuable in studies of backscatter. The MGH proton treatment facility was simulated with phase space files recorded at the phantom surface. Spectra using the various latch cuts were generated, demonstrating the capabilities of latching on regions, interactions, or ancestors.

Conclusions: Phase space I/O capabilities have been implemented for Geant4. The classes are easy to incorporate into existing code, allowing writing on different surface geometries, and extend the Latch functionality of BEAMnrc.

Support: Support from NIH grant 1R01CA140735-01.