# GEANT-4 GPU Port:

## Design Document: System Architecture

Stuart Douglas – 1214422
Matthew Pagnan – 1208693
Rob Gorrie – 1222547
Victor Reginato – 1209975

January 6, 2016

# Table of Contents

# 1 Introduction

## 1.1 Purpose

The purpose of GEANT4-GPU is to reduce the computation times of particle simulations using the GEANT4 simulation library.

## 1.2 Description

The project aims to improve the computation times of Geant4 particle simulations by running certain parallel operations on a GPU. GEANT4-GPU will be a fork of the existing Geant4 system with the additional option for users with compatible hardware to run operations on the GPU for improved performane. This functionality will be available on Mac, Linux and Windows operating systems running on computers with NVIDIA graphics cards (G8X series or later).

The design strategy for the project will be based on taking a specific, computationally heavy class from Geant4 and creating a class that fulfills the same interface but that runs on the GPU. This will be repeated for many classes until the project's deadline has been reached. The user will have the option of using the existing classes (running on the CPU) or the new ones (running on the GPU).

## 1.3 Scope

The scope of the project in terms of modules to port to the GPU will be limited to physics simulations run by the McMaster Engineering Physics department.

# Revision History

All major edits to this document will be recorded in the table below.

Table 1: Revision History

| Description of Changes | Author | Date |
|---|---|---|
| Set up sections and filled out Introduction section | Matthew | 2015-12-15 |

# 2 Anticipated and unlikely changes

## 2.1 Likely Changes

## 2.2 Unlikely Changes

# 3 Module Hierarchy

# 4 Connection between requirements and design

# 5 Traceability matrices

## 5.1 Note About G4 variables

Geant4 uses its own basic types for standard C++ types (G4int, G4bool, G4double, etc). These types are currently just `typedefs` to the respective type as defined in the system libraries.

# 6 MIS of NeutronHPDataPoint

## 6.1 Interface Syntax

## 6.2 Exported Access Programs

## 6.3 Interface Semantics

### 6.3.1 State Variables

- energy : G4Double

- xSec : G4Double

### 6.3.2 Environment Variables

There are no environment variables for this Module.

### 6.3.3 Assumption

### 6.3.4 Access Program Semantics

# 7 MIS of NeutronHPVector

## 7.1 Interface Syntax

## 7.2 Exported Access Programs

## 7.3 Interface Semantics

### 7.3.1 State Variables

- G4NeutronHPInterPolator : theLine

- totalIntegral : G4double

- G4NeutronHPDataPoint * : theData

- G4InterpolationManager : theManager

- G4double * : theIntegral

- G4int : nEntries

- G4int : nPoints

- G4double : label

- G4NeutronInterpolator : theInt

- G4int : Verbose

- G4int : isFreed

- G4NeutronHPHash : theHash

- G4double : maxValue

- vector<G4double>: theBlocked

- vector<G4double>: theBuffered

- G4double : the15percentBorderCash

- G4double : the50percentBorderCash

### 7.3.2 Environment Variables

There are no environment variables for this Module.

### 7.3.3   Assumption

### 7.3.4   Access Program Semantics

# 8   MIS of CMake Files

## 8.1   Interface Syntax

## 8.2   Exported Access Programs

## 8.3   Interface Semantics

### 8.3.1   State Variables

- useCuda : Bool

### 8.3.2   Environment Variables

- NeutronHPVectorGPU.cu : cuda file with GPU code

### 8.3.3   Assumption

### 8.3.4   Access Program Semantics