GEANT4 GPU Port:

Test Report

Stuart Douglas – dougls2
Matthew Pagnan – pagnanmm
Rob Gorrie – gorrierw
Victor Reginato – reginavp

Version 0
March 17, 2016

Contents

Revision History

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

Table 1: Revision History

Description of Changes	Author	Date
Initial draft of document	Matt, Stuart, Rob, Victor	2016-03-15

List of Figures

Definitions and Acronyms

Table 2: Definitions and Acronyms

2 **Term**	**Description**
GEANT4	Open-source software toolkit used to simulate the passage of particles through matter
GEANT4-GPU	GEANT4 with some computations running on the GPU
GPU	Graphics processing unit, well-suited to parallel computing tasks
CPU	Computer processing unit, general computer processor well-suited to serial tasks
CUDA	Parallel computing architecture for general purpose programming on GPU, developed by NVIDIA
RHEL	Red Hat Enterprise Linux Server
OS X	Operating system developed by Apple

2

1 Introduction

1.1 Purpose of the Document

This document summarizes the testing and test conclusions of GEANT4-GPU. This document uses the implementation outlined in the test plan.

1.2 Scope of the Testing

1.3 Organization

In Section 4 we provide an introduction to this report. Section 5 describes the test cases which are carried out on each function. Section 6 describes system test cases that

were carried out by our team. In section 7 traceability matrices to requirements and modules are documented. Section 8 provides a summary of changes made in response to the testing results.

1.4 Note about Usability

GEANT4-GPU is a back end implementation of already existing GEANT4 modules. Therefore users will not be interacting with is directly. Since there is no direct user interaction with GEANT4-GPU. There are no usability test.

2 Module Unit Testing

2.1 Use of Automated Testing

Our testing is semi-automated. Due to the nature of this implementation we need to recompile GEANT4-GPU from GPU to CPU in order to get the CPU results to compare against the GPU results. We have a unit test file which preforms all our unit tests and writes the results into a file. The user will then have to manually recompile GEANT4-GPU with GPU acceleration off. Once the unit test file is run again another results file is generated. The comparing of the results is automated by feeding them to an application that we created that will compare the test results against each other. The program outputs a summary of any differences between the two results, if there are any.

2.2 PerformInitialization

2.2.1 Unit Tests

Table 3: Unit Tests

Test #	Code	**Description**
1	Code goes here	Description goes here

2.2.2 Accuracy

Table 4: Accuracy

Test #	CPU	GPU
1	CPU time	GPU time

2.2.3 Performance

2.3 OperatorEquals

2.3.1 Unit Tests

Table 5: Unit Tests

Test #	Code	**Description**
2	Code goes here	Description goes here

2.3.2 Accuracy

Table 6: Accuracy

Test #	CPU	GPU
2	CPU time	GPU time

2.3.3 Performance

2.4 GetPoint

2.4.1 Unit Tests

Table 7: Unit Tests

Test #	Code	**Description**
3	Code goes here	Description goes here

2.4.2 Accuracy

Table 8: Accuracy

Test #	CPU	GPU
3	CPU time	GPU time

2.4.3 Performance

2.5 GetX

2.5.1 Unit Tests

Table 9: Unit Tests

Test #	Code	**Description**
4	Code goes here	Description goes here

2.5.2 Accuracy

Table 10: Accuracy

Test #	CPU	GPU
4	CPU time	GPU time

2.5.3 Performance

2.6 GetY

2.6.1 Unit Tests

Table 11: Unit Tests

Test #	Code	**Description**
5	Code goes here	Description goes here

2.6.2 Accuracy

Table 12: Accuracy

Test #	CPU	GPU
5	CPU time	GPU time

2.6.3 Performance

2.7 GetXsec

2.7.1 Unit Tests

Table 13: Unit Tests

Test #	Code	**Description**
6	Code goes here	Description goes here

2.7.2 Accuracy

Table 14: Accuracy

Test #	CPU	GPU
6	CPU time	GPU time

2.7.3 Performance

2.8 SetData

2.8.1 Unit Tests

Table 15: Unit Tests

Test #	Code	**Description**
7	Code goes here	Description goes here

2.8.2 Accuracy

Table 16: Accuracy

Test #	CPU	GPU
7	CPU time	GPU time

2.8.3 Performance

2.9 SetX

2.9.1 Unit Tests

Table 17: Unit Tests

Test #	Code	**Description**
8	Code goes here	Description goes here

2.9.2 Accuracy

Table 18: Accuracy

Test #	CPU	GPU
8	CPU time	GPU time

2.9.3 Performance

2.10 SetY

2.10.1 Unit Tests

Table 19: Unit Tests

Test #	Code	**Description**
9	Code goes here	Description goes here

2.10.2 Accuracy

Table 20: Accuracy

Test #	CPU	GPU
9	CPU time	GPU time

2.10.3 Performance

2.11 Init

2.11.1 Unit Tests

Table 21: Unit Tests

Test #	Code	**Description**
10	Code goes here	Description goes here

2.11.2 Accuracy

Table 22: Accuracy

Test #	CPU	GPU
10	CPU time	GPU time

2.11.3 Performance

2.12 CleanUp

2.12.1 Unit Tests

Table 23: Unit Tests

Test #	Code	**Description**
11	Code goes here	Description goes here

2.12.2 Accuracy

Table 24: Accuracy

Test #	CPU	GPU
11	CPU time	GPU time

2.12.3 Performance

2.13 SampleLin

2.13.1 Unit Tests

Table 25: Unit Tests

Test #	Code	**Description**
12	Code goes here	Description goes here

2.13.2 Accuracy

Table 26: Accuracy

Test #	CPU	GPU
12	CPU time	GPU time

2.13.3 Performance

2.14 Integrate

2.14.1 Unit Tests

Table 27: Unit Tests

Test #	Code	**Description**
13	Code goes here	Description goes here

2.14.2 Accuracy

Table 28: Accuracy

Test #	CPU	GPU
13	CPU time	GPU time

2.14.3 Performance

2.15 IntegrateAndNormalise

2.15.1 Unit Tests

Table 29: Unit Tests

Test #	Code	**Description**
14	Code goes here	Description goes here

2.15.2 Accuracy

Table 30: Accuracy

Test #	CPU	GPU
14	CPU time	GPU time

2.15.3 Performance

2.16 Times

2.16.1 Unit Tests

Table 31: Unit Tests

Test #	Code	**Description**
15	Code goes here	Description goes here

2.16.2 Accuracy

Table 32: Accuracy

Test #	CPU	GPU
15	CPU time	GPU time

2.16.3 Performance

2.17 GetXsecBuffer

2.17.1 Unit Tests

Table 33: Unit Tests

Test #	Code	**Description**
16	Code goes here	Description goes here

2.17.2 Accuracy

Table 34: Accuracy

Test #	CPU	GPU
16	CPU time	GPU time

2.17.3 Performance

2.18 Dump

2.18.1 Unit Tests

Table 35: Unit Tests

Test #	Code	**Description**
17	Code goes here	Description goes here

2.18.2 Accuracy

Table 36: Accuracy

Test #	CPU	GPU
17	CPU time	GPU time

2.18.3 Performance

2.19 ThinOut

2.19.1 Unit Tests

Table 37: Unit Tests

Test #	Code	**Description**
18	Code goes here	Description goes here

2.19.2 Accuracy

Table 38: Accuracy

Test #	CPU	GPU
18	CPU time	GPU time

2.19.3 Performance

2.20 Sample

2.20.1 Unit Tests

Table 39: Unit Tests

Test #	Code	**Description**
19	Code goes here	Description goes here

2.20.2 Accuracy

Table 40: Accuracy

Test #	CPU	GPU
19	CPU time	GPU time

2.20.3 Performance

2.21 Check

2.21.1 Unit Tests

Table 41: Unit Tests

Test #	Code	**Description**
20	Code goes here	Description goes here

2.21.2 Accuracy

Table 42: Accuracy

Test #	CPU	GPU
20	CPU time	GPU time

2.21.3 Performance

2.22 GetVectorLength

2.22.1 Unit Tests

Table 43: Unit Tests

Test #	Code	**Description**
21	Code goes here	Description goes here

2.22.2 Accuracy

Table 44: Accuracy

Test #	CPU	GPU
21	CPU time	GPU time

2.22.3 Performance

2.23 GetLabel

2.23.1 Unit Tests

Table 45: Unit Tests

Test #	Code	**Description**
22	Code goes here	Description goes here

2.23.2 Accuracy

Table 46: Accuracy

Test #	CPU	GPU
22	CPU time	GPU time

2.23.3 Performance

2.24 GetIntegral

2.24.1 Unit Tests

Table 47: Unit Tests

Test #	Code	**Description**
23	Code goes here	Description goes here

2.24.2 Accuracy

Table 48: Accuracy

Test #	CPU	GPU
23	CPU time	GPU time

2.24.3 Performance

2.25 SetPoint

2.25.1 Unit Tests

Table 49: Unit Tests

Test #	Code	**Description**
24	Code goes here	Description goes here

2.25.2 Accuracy

Table 50: Accuracy

Test #	CPU	GPU
24	CPU time	GPU time

2.25.3 Performance

2.26 SetInterpolationManager

2.26.1 Unit Tests

Table 51: Unit Tests

Test #	Code	**Description**
25	Code goes here	Description goes here

2.26.2 Accuracy

Table 52: Accuracy

Test #	CPU	GPU
25	CPU time	GPU time

2.26.3 Performance

2.27 SetScheme

2.27.1 Unit Tests

Table 53: Unit Tests

Test #	Code	**Description**
26	Code goes here	Description goes here

2.27.2 Accuracy

Table 54: Accuracy

Test #	CPU	GPU
26	CPU time	GPU time

2.27.3 Performance

2.28 Merge

2.28.1 Unit Tests

Table 55: Unit Tests

Test #	Code	**Description**
26	Code goes here	Description goes here

2.28.2 Accuracy

Table 56: Accuracy

Test #	CPU	GPU
26	CPU time	GPU time

2.28.3 Performance

3 Specific System Tests

3.1 Summary of Tests Performed

3.2 System Tests Results

4 Traceability

The following section is used to highlight the relations of implemented test cases to requirements and modules. In doing so, we hope to draw clear reasoning upon the inclusion of such tests.

4.1 Requirements

Below is a traceability table outlining test cases and the requirements they are related to:

Table 57: Tests and Requirements Relationship

Test	Description	Requirement
1	Performance test of functions	requirement
2	InitializeVector	requirement
3	SettersandGetters	requirement
4	GetXSec	requirement
5	ThinOut	requirement
6	Merge	requirement
7	Sample	requirement
8	GetBorder	requirement
9	Integral	requirement
10	Times	requirement
11	Assignment	requirement

4.2 Modules

Similarly, the following is a traceability table explicitly relating test cases to modules:

Table 58: Tests and Modules Relationship

Test	**Description**	**Module**
1	Performance test of functions	module
2	InitializeVector	module
3	SettersandGetters	module
4	GetXSec	module
5	ThinOut	module
6	Merge	module
7	Sample	module
8	GetBorder	module
9	Integral	module
10	Times	module
11	Assignment	module

5 Changes after Testing