GEANT4 GPU Port:

Test Report

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

Version 0
March 20, 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, Rob, Victor, Stuart	2016-03-18
Template of document	Matt	2016-03-15

List of Figures

Tables and figures for specific unit tests have been omitted in order to keep this document readable.

Definitions and Acronyms

Table 2: Definitions and Acronyms

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

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 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.

1.5 Robustness

The GEANT4-GPU functions are meant to mimic the already existing GEANT4 functions. Therefore the GEANT4-GPU functions must also mimic the the robustness of the GEANT4 functions. The accuracy section for unit tests has several unit tests designed to test the robustness of the functions.

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 General variables used for Unit Testing

The following are variables that are used for multiple unit tests. Instead of defining them again for each unit test they are defined here only once. Other variables used for specific unit tests will be defined in their respective unit test sections
For all unit tests:

Table 3: General Unit Test Variables

Name #	Type	**Value**
n	G4double	length of the G4ParticleHPVector
r0	G4double	0.000512
r1	G4double	1.58
r2	G4double	513.18
Empty	G4ParticleHPVector	Empty
A	G4ParticleHPVector	66 entries
B	G4ParticleHPVector	1509 entries
C	G4ParticleHPVector	8045 entries
D	G4ParticleHPVector	41854 entries
E	G4ParticleHPVector	98995 entries
F	G4ParticleHPVector	242594 entries

2.3 Note about Performance testing

Tests on vectors A - F all behave the same. Showing accuracy for vectors A - F does not provide any extra useful information. Therefore only unit tests on vector D will be shown in the Unit Tests and Accuracy sections. Unit test interfaces for the other vectors will be omitted from this document in order to make it more readable. The

unit tests were still performed on the other vectors. These unit tests on vectors of different length are done to show how increasing the size of the vector increases the execution time of some functions

2.4 OperatorEquals

2.4.1 Unit Tests

Table 4: Unit Tests

Test #	Code	**Description**
1	Empty = Empty2	Assign an empty vector to another empty vector
2	D = Empty	Assign an empty vector to regular vector
3	Empty = D	Assign a vector to an empty vector
4	D = D	Assign a vector to itself
5	D = F	Assign a vector to another vector

2.4.2 Accuracy

Table 5: Accuracy

Test #	Status
1	Pass
2	Pass
3	Pass
4	Pass
5	Pass

2.4.3 Performance

2.5 GetPoint

2.5.1 Unit Tests

Table 6: Unit Tests

Test #	Code	**Description**
6	Empty.GetPoint(-1)	Get a point at a negative index from an empty vector
7	Empty.GetPoint(0)	Get a point at a the first index from an empty vector
8	Empty.GetPoint(1)	Get a point at an index out of bounds from an empty vector
9	D.GetPoint(-1)	Get a point at a negative index
10	D.GetPoint(0)	Get a point at a the first index
11	D.GetPoint(n/2)	Get a point at an index within the vector
12	D.GetPoint(n-1)	Get a point at the last index
13	D.GetPoint(n)	Get a point at an index our of bounds

2.5.2 Accuracy

Table 7: Accuracy

Test #	Status
6	Pass
7	Pass
8	Pass
9	Pass
10	Pass
11	Pass
12	Pass
13	Pass

2.5.3 Performance

2.6 GetX

2.6.1 Unit Tests

Table 8: Unit Tests

Test #	Code	**Description**
14	Empty.GetX(-1)	Set an xSec at a negative index of an empty vector
15	Empty.GetX(0)	Set an xSec at a the first index of an empty vector
16	Empty.GetX(1)	Set an xSec at an index out of bounds of an empty vector
17	D.GetX(-1)	Set an xSec at a negative index
18	D.GetX(0)	Set an xSec at a the first index
19	D.GetX(n/2)	Set an xSec at an index within the vector
20	D.GetX(n-1)	Set an xSec at the last index
21	D.GetX(n)	Set an xSec at an index our of bounds

2.6.2 Accuracy

Table 9: Accuracy

Test #	Status
14	Pass
15	Pass
16	Pass
17	Pass
18	Pass
19	Pass
20	Pass
21	Pass

2.6.3 Performance

2.7 GetY

2.7.1 Unit Tests

Table 10: Unit Tests

Test #	Code	**Description**
22	Empty.GetY(-1)	Get a point at a negative index of an empty vector
23	Empty.GetY(0)	Get a point at a the first index of an empty vector
24	Empty.GetY(1)	Get a point at an index out of bounds of an empty vector
25	D.GetY(-1)	Get a point at a negative index
26	D.GetY(0)	Get a point at a the first index
27	D.GetY(n/2)	Get a point at an index within the vector
28	D.GetY(n-1)	Get a point at the last index
29	D.GetY(n)	Get a point at an index our of bounds

2.7.2 Accuracy

Table 11: Accuracy

Test #	Status
22	Pass
23	Pass
24	Pass
25	Pass
26	Pass
27	Pass
28	Pass
29	Pass

2.7.3 Performance

2.8 GetXsec

2.8.1 Unit Tests

Table 12: Unit Tests

Test #	Code	**Description**
30	Empty.GetXsec(-1)	Get an xSec with a negative energy from an empty vector
31	Empty.GetXsec(0)	Get a xSec with an energy of zero from an empty vector
32	Empty.GetXsec(r1)	Get a xSec with a normal energy from an empty vector
33	D.GetXsec(-1)	Get a xSec with a negative energy
34	D.GetXsec(0)	Get a xSec with a zero energy
35	D.GetXsec(r0)	Get a xSec with a small energy
36	D.GetXsec(r1)	Get a xSec with a normal energy
37	D.GetXsec(r2)	Get a xSec with a large energy

2.8.2 Accuracy

Table 13: Accuracy

Test #	Status
30	Pass
31	Pass
32	Pass
33	Pass
34	Pass
35	Pass
36	Pass
37	Pass

2.8.3 Performance

2.9 SetData

2.9.1 Unit Tests

Table 14: Unit Tests

Test #	Code	**Description**
38	Empty.SetData(-1, r1, r2)	Set a point at a negative index of an empty vector
39	Empty.SetData(0, r1, r2)	Set a point at a the first index of an empty vector
40	Empty.SetData(1, r1, r2)	Set a point at an index out of bounds of an empty vector
41	D.SetData(-1, r1, r2)	Set a point at a negative index
42	D.SetData(0, r1, r2)	Set a point at a the first index
43	D.SetData(n/2, r1, r2)	Set a point at an index within the vector
44	D.SetData(n-1, r1, r2)	Set a point at the last index
45	D.SetData(n, r1, r2)	Set a point at an index our of bounds
46	D.SetData(0, -1, -1)	Set a point with a negative energy and xSec
47	D.SetData(0, 0, 0)	Set a point with a zero energy and xSec

2.9.2 Accuracy

Table 15: Accuracy

Test #	Status
38	Pass
39	Pass
40	Pass
41	Pass
42	Pass
43	Pass
44	Pass
45	Pass
46	Pass
47	Pass

2.9.3 Performance

2.10 SetEnergy

2.10.1 Unit Tests

Table 16: Unit Tests

Test #	Code	**Description**
48	Empty.SetEnergy(-1, r1)	Set an energy at a negative index of an empty vector
49	Empty.SetEnergy(0, r1)	Set an energy at a the first index of an empty vector
50	Empty.SetEnergy(1, r1)	Set an energy at an index out of bounds of an empty vector
51	D.SetEnergy(-1, r1)	Set an energy at a negative index
52	D.SetEnergy(0, r1)	Set an energy at a the first index
53	D.SetEnergy(n/2, r1)	Set an energy at an index within the vector
54	D.SetEnergy(n-1, r1)	Set an energy at the last index
55	D.SetEnergy(n, r1)	Set an energy at an index our of bounds
56	D.SetEnergy(0, -1)	Set an energy at an index within the vector to a negative value
57	D.SetEnergy(0, 0)	Set an energy at an index within the vector to a zero value

2.10.2 Accuracy

Table 17: Accuracy

Test #	Status
48	Pass
49	Pass
50	Pass
51	Pass
52	Pass
53	Pass
54	Pass
55	Pass
56	Pass
57	Pass

2.10.3 Performance

2.11 SetXsec

2.11.1 Unit Tests

Table 18: Unit Tests

Test #	Code	**Description**
58	Empty.SetXsec(-1, r1)	Set an xSec at a negative index of an empty vector
59	Empty.SetXsec(0, r1)	Set an xSec at a the first index of an empty vector
60	Empty.SetXsec(1, r1)	Set an xSec at an index out of bounds of an empty vector
61	D.SetXsec(-1, r1)	Set an xSec at a negative index
62	D.SetXsec(0, r1)	Set an xSec at a the first index
63	D.SetXsec(n/2, r1)	Set an xSec at an index within the vector
64	D.SetXsec(n-1, r1)	Set an xSec at the last index
65	D.SetXsec(n, r1)	Set an xSec at an index our of bounds
66	D.SetXsec(0, -1)	Try to set a negative xSec
67	D.SetXsec(0, 0)	Try to set a zero xSec

2.11.2 Accuracy

Table 19: Accuracy

Test #	Status
58	Pass
59	Pass
60	Pass
61	Pass
62	Pass
63	Pass
64	Pass
65	Pass
66	Pass
67	Pass

2.11.3 Performance

2.12 SetX

2.12.1 Unit Tests

Table 20: Unit Tests

Test #	Code	**Description**
68	Empty.SetX(-1, r1)	Set an energy at a negative index of an empty vector
69	Empty.SetX(0, r1)	Set an energy at a the first index of an empty vector
70	Empty.SetX(1, r1)	Set an energy at an index out of bounds of an empty vector
71	D.SetX(-1, r1)	Set an energy at a negative index
72	D.SetX(0, r1)	Set an energy at a the first index
73	D.SetX(n/2, r1)	Set an energy at an index within the vector
74	D.SetX(n-1, r1)	Set an energy at the last index
75	D.SetX(n, r1)	Set an energy at an index our of bounds
76	D.SetX(0, -1)	Set a negative energy
77	D.SetX(0, 0)	Set a zero energy

2.12.2 Accuracy

Table 21: Accuracy

Test #	Status
68	Pass
69	Pass
70	Pass
71	Pass
72	Pass
73	Pass
74	Pass
75	Pass
76	Pass
77	Pass

2.12.3 Performance

2.13 SetY

2.13.1 Unit Tests

Table 22: Unit Tests

Test #	Code	**Description**
78	Empty.SetY(-1, r1)	Set an xSec at a negative index of an empty vector
79	Empty.SetY(0, r1)	Set an xSec at a the first index of an empty vector
80	Empty.SetY(1, r1)	Set an xSec at an index out of bounds of an empty vector
81	D.SetY(-1, r1)	Set an xSec at a negative index
82	D.SetY(0, r1)	Set an xSec at a the first index
83	D.SetY(n/2, r1)	Set an xSec at an index within the vector
84	D.SetY(n-1, r1)	Set an xSec at the last index
85	D.SetY(n, r1)	Set an xSec at an index our of bounds
86	D.SetY(0, -1)	Set a negative xSec
87	D.SetY(0, 0)	Set a zero xSec

2.13.2 Accuracy

Table 23: Accuracy

Test #	Status
78	Pass
79	Pass
80	Pass
81	Pass
82	Pass
83	Pass
84	Pass
85	Pass
86	Pass
87	Pass

2.13.3 Performance

2.14 Init

2.14.1 Unit Tests

Table 24: Unit Tests

Test #	Code	**Description**
88	Empty.Init()	Init an empty Vector
89	D.Init()	Init a Vector

2.14.2 Accuracy

Table 25: Accuracy

Test #	Status
88	Pass
89	Pass

2.14.3 Performance

2.15 SampleLin

2.15.1 Unit Tests

Table 26: Unit Tests

Test #	Code	**Description**
90	Empty.SampleLin()	Sample an empty Vector
91	D.SampleLin()	Sample a Vector

2.15.2 Accuracy

Table 27: Accuracy

Test #	CPU	GPU
90	CPU result	GPU result
91	CPU result	GPU result

2.15.3 Performance

2.16 Integrate

2.16.1 Unit Tests

Table 28: Unit Tests

Test #	Code	**Description**
92	Empty.Integrate()	Integrate an empty Vector
93	D.Integrate()	Integrate a Vector

2.16.2 Accuracy

Table 29: Accuracy

Test #	Status
92	Pass
93	Pass

2.16.3 Performance

2.17 IntegrateAndNormalise

2.17.1 Unit Tests

Table 30: Unit Tests

Test #	Code	**Description**
94	Empty.IntegrateAndNormalise()	Integrate and normalize an empty Vector
95	D.IntegrateAndNormalise()	Integrate normalize a Vector

2.17.2 Accuracy

Table 31: Accuracy

Test #	Status
30	Pass
30	Pass

2.17.3 Performance

2.18 Times

2.18.1 Unit Tests

Table 32: Unit Tests

Test #	Code	**Description**
96	Empty.Times(-1)	Times an empty vector by a negative factor
97	Empty.Times(0)	Times an empty vector by zero
98	Empty.Times(1)	Times an empty vector by 1
99	Empty.Times(r1)	Times an empty vector by a random factor
100	D.Times(-1)	Times a vector by a negative factor
101	D.Times(0)	Times a vector by zero
102	D.Times(1)	Times a vector by 1
103	D.Times(r1)	Times a vector by a random factor

2.18.2 Accuracy

Table 33: Accuracy

Test #	Status
96	Pass
97	Pass
98	Pass
99	Pass
100	Pass
101	Pass
102	Pass
103	Pass

2.18.3 Performance

2.19 GetXsecBuffer

Table 34: General Unit Test Variables

Name	**Size**	Description
emptyBuff	0	Array with no queries
singleBuff	1	Array with a single query
smallbuff	50	Array with a small number of queries
normalBuff	1000	Array with a moderate number of queries
largeBuff	10000	Array with a large amount of queries
negBuff	50	Array of queries with negative values
zeroBuff	50	Array of queries with values of zero
highBuff	50	Array of queries with values larger than the highest energy in the vector

2.19.1 Unit Tests

Table 35: Unit Tests

Test #	Code	**Description**
104	D.GetXsecBuffer(normalBuff, -1)	buffer with a negative size
105	Empty.GetXsecBuffer(emptyBuff, 0)	Empty buffer of xSec queries to an empty vector
106	Empty.GetXsecBuffer(normalBuff, 1000)	Normal buffer of xSec queries to an empty vecto
107	D.GetXsecBuffer(emptyBuff, 0)	Empty buffer of xSec queries
108	D.GetXsecBuffer(smalllBuff, 50)	Small number of queries
109	D.GetXsecBuffer(normalBuff, 1000)	Normal case
110	D.GetXsecBuffer(highBuff, 10000)	Large number of queries
111	D.GetXsecBuffer(negBuff, 1000)	Buffer of negative xSec queries
112	D.GetXsecBuffer(emptyBuff, 1000)	Buffer of zeros
113	D.GetXsecBuffer(highBuff, 0)	Buffer of high valued xSec queries

2.19.2 Accuracy

Table 36: Accuracy

Test #	Status
104	Pass
105	Pass
106	Pass
107	Pass
108	Pass
109	Pass
110	Pass
111	Pass
112	Pass
113	Pass

2.19.3 Performance

2.20 ThinOut

2.20.1 Unit Tests

Table 37: Unit Tests

Test #	Code	**Description**
114	Empty.ThinOut(r1)	ThinOut an empty Vector
115	D.ThinOut(-1)	ThinOut a Vector using a negative value
116	D.ThinOut(0)	ThinOut a Vector using a zero value
117	D.ThinOut(r0)	ThinOut a Vector using a small value
118	D.ThinOut(r1)	ThinOut a Vector using a normal value
119	D.ThinOut(r2)	ThinOut a Vector using a large value

2.20.2 Accuracy

Table 38: Accuracy

Test #	Status
114	Pass
115	Pass
116	Pass
117	Pass
118	Pass
119	Pass

2.20.3 Performance

2.21 Sample

2.21.1 Unit Tests

Table 39: Unit Tests

Test #	Code	**Description**
120	Empty.Sample()	Sample an empty Vector
121	D.Sample()	Sample a Vector

2.21.2 Accuracy

Table 40: Accuracy

Test #	CPU	GPU
120	CPU result	GPU result
121	CPU result	GPU result

2.21.3 Performance

2.22 SetPoint

2.22.1 Unit Tests

- “rPoint” is a random G4ParticleHPDataPoint
- “nPoint” is a negative G4ParticleHPDataPoint
- “zPoint” is a zero G4ParticleHPDataPoint

Table 41: Unit Tests

Test #	Code	**Description**
122	Empty.SetPoint(-1, rPoint)	Set a point at a negative index of an empty vector
123	Empty.SetPoint(0, rPoint)	Set a point at a the first index of an empty vector
124	Empty.SetPoint(1, rPoint)	Set a point at an index out of bounds of an empty vector
125	D.SetPoint(-1, rPoint)	Set a point at a negative index
126	D.SetPoint(0, rPoint)	Set a point at a the first index
127	D.SetPoint(n/2, rPoint)	Set a point at an index within the vector
128	D.SetPoint(n-1, rPoint)	Set a point at the last index
129	D.SetPoint(n, rPoint)	Set a point at an index our of bounds
130	D.SetPoint(0, nPoint)	Set a negative point
131	D.SetPoint(0, zPoint)	Set a zero point

2.22.2 Accuracy

Table 42: Accuracy

Test #	Status
122	Pass
123	Pass
124	Pass
125	Pass
126	Pass
127	Pass
128	Pass
129	Pass
130	Pass
131	Pass

2.22.3 Performance

2.23 Get15percentBorder

2.23.1 Unit Tests

Table 43: Unit Tests

Test #	Code	**Description**
132	Empty.Get15percentBorder()	Get 15 percent Border of an empty vector
133	D.Get15percentBorder()	Get 15 percent Border of a vector

2.23.2 Accuracy

Table 44: Accuracy

Test #	Status
132	Pass
133	Pass

2.23.3 Performance

2.24 Get50percentBorder

2.24.1 Unit Tests

Table 45: Unit Tests

Test #	Code	Description
134	Empty.Get50percentBorder()	Get 50 percent Border of an empty vector
135	D.Get50percentBorder()	Get 50 percent Border of a vector

2.24.2 Accuracy

Table 46: Accuracy

Test #	Status
134	Pass
135	Pass

2.24.3 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 47: 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 48: 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