GEANT4 GPU Port:

Test Report

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

Version 0
March 19, 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	Code goes here	Description goes here

2.4.2 Accuracy

Table 5: Accuracy

Test #	CPU	GPU
??	CPU time	GPU time

2.4.3 Performance

2.5 GetPoint

2.5.1 Unit Tests

Table 6: Unit Tests

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

Table 7: Accuracy

Test #	CPU	GPU
2	CPU time	GPU time
3	CPU time	GPU time
4	CPU time	GPU time
5	CPU time	GPU time
6	CPU time	GPU time
7	CPU time	GPU time
8	CPU time	GPU time
9	CPU time	GPU time

2.5.2 Accuracy

2.5.3 Performance

2.6 GetX

2.6.1 Unit Tests

Table 8: Unit Tests

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

Table 9: Accuracy

Test #	CPU	GPU
10	CPU result	GPU result
11	CPU result	GPU result
12	CPU result	GPU result
13	CPU result	GPU result
14	CPU result	GPU result
15	CPU result	GPU result
16	CPU result	GPU result
17	CPU result	GPU result

Table 10: Unit Tests

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

Table 11: Accuracy

Test #	CPU	GPU
18	CPU result	GPU result
19	CPU result	GPU result
20	CPU result	GPU result
21	CPU result	GPU result
22	CPU result	GPU result
23	CPU result	GPU result
24	CPU result	GPU result
25	CPU result	GPU result

2.6.2 Accuracy

2.6.3 Performance

2.7 GetY

2.7.1 Unit Tests

2.7.2 Accuracy

2.7.3 Performance

2.8 GetXsec

2.8.1 Unit Tests

Table 12: Unit Tests

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

2.8.2 Accuracy

Table 13: Accuracy

Test #	CPU	GPU
26	CPU result	GPU result
27	CPU result	GPU result
28	CPU result	GPU result
29	CPU result	GPU result
30	CPU result	GPU result
31	CPU result	GPU result
32	CPU result	GPU result
33	CPU result	GPU result

2.8.3 Performance

2.9 SetData

2.9.1 Unit Tests

Table 14: Unit Tests

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

2.9.2 Accuracy

Table 15: Accuracy

Test #	CPU	GPU
34	CPU result	GPU result
35	CPU result	GPU result
36	CPU result	GPU result
37	CPU result	GPU result
38	CPU result	GPU result
39	CPU result	GPU result
40	CPU result	GPU result
41	CPU result	GPU result
42	CPU result	GPU result
43	CPU result	GPU result

Table 16: Unit Tests

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

2.9.3 Performance

2.10 SetEnergy

2.10.1 Unit Tests

2.10.2 Accuracy

Table 17: Accuracy

Test #	CPU	GPU
44	CPU result	GPU result
45	CPU result	GPU result
46	CPU result	GPU result
47	CPU result	GPU result
48	CPU result	GPU result
49	CPU result	GPU result
50	CPU result	GPU result
51	CPU result	GPU result
52	CPU result	GPU result
53	CPU result	GPU result

Table 18: Unit Tests

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

Table 19: Accuracy

Test #	CPU	GPU
54	CPU result	GPU result
55	CPU result	GPU result
56	CPU result	GPU result
57	CPU result	GPU result
58	CPU result	GPU result
59	CPU result	GPU result
60	CPU result	GPU result
61	CPU result	GPU result
62	CPU result	GPU result
63	CPU result	GPU result

2.10.3 Performance

2.11 SetXsec

2.11.1 Unit Tests

2.11.2 Accuracy

2.11.3 Performance

2.12 SetX

2.12.1 Unit Tests

Table 20: Unit Tests

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

Table 21: Accuracy

Test #	CPU	GPU
64	CPU result	GPU result
65	CPU result	GPU result
66	CPU result	GPU result
67	CPU result	GPU result
68	CPU result	GPU result
69	CPU result	GPU result
70	CPU result	GPU result
71	CPU result	GPU result
72	CPU result	GPU result
73	CPU result	GPU result

Table 22: Unit Tests

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

Table 23: Accuracy

Test #	CPU	GPU
74	CPU result	GPU result
75	CPU result	GPU result
76	CPU result	GPU result
77	CPU result	GPU result
78	CPU result	GPU result
79	CPU result	GPU result
80	CPU result	GPU result
81	CPU result	GPU result
82	CPU result	GPU result
83	CPU result	GPU result

Table 24: Unit Tests

Test #	Code	**Description**
84	Empty.Init()	Init an empty Vector
85	D.Init()	Init a Vector

Table 25: Accuracy

Test #	CPU	GPU
84	CPU result	GPU result
85	CPU result	GPU result

2.12.2 Accuracy

2.12.3 Performance

2.13 SetY

2.13.1 Unit Tests

2.13.2 Accuracy

2.13.3 Performance

2.14 Init

2.14.1 Unit Tests

2.14.2 Accuracy

2.14.3 Performance

2.15 SampleLin

2.15.1 Unit Tests

Table 26: Unit Tests

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

Table 27: Accuracy

Test #	CPU	GPU
86	CPU result	GPU result
87	CPU result	GPU result

2.15.2 Accuracy

2.15.3 Performance

2.16 Integrate

2.16.1 Unit Tests

Table 28: Unit Tests

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

2.16.2 Accuracy

Table 29: Accuracy

Test #	CPU	GPU
88	CPU result	GPU result
89	CPU result	GPU result

2.16.3 Performance

2.17 IntegrateAndNormalise

2.17.1 Unit Tests

Table 30: Unit Tests

Test #	Code	Description
90	Empty.IntegrateAndNormalise()	Integrate and normalize an empty Vector
91	D.IntegrateAndNormalise()	Integrate normalize a Vector

2.17.2 Accuracy

Table 31: Accuracy

Test #	CPU	GPU
30	CPU result	GPU result
30	CPU result	GPU result

2.17.3 Performance

2.18 Times

2.18.1 Unit Tests

Table 32: Unit Tests

Test #	Code	**Description**
92	Empty.Times(-1)	Times an empty vector by a negative factor
93	Empty.Times(0)	Times an empty vector by zero
94	Empty.Times(1)	Times an empty vector by 1
95	Empty.Times(r1)	Times an empty vector by a random factor
96	D.Times(-1)	Times a vector by a negative factor
97	D.Times(0)	Times a vector by zero
98	D.Times(1)	Times a vector by 1
99	D.Times(r1)	Times a vector by a random factor

2.18.2 Accuracy

Table 33: Accuracy

Test #	CPU	GPU
92	CPU result	GPU result
93	CPU result	GPU result
94	CPU result	GPU result
95	CPU result	GPU result
96	CPU result	GPU result
97	CPU result	GPU result
98	CPU result	GPU result
99	CPU result	GPU result

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**
100	D.GetXsecBuffer(normalBuff, -1)	buffer with a negative size
101	Empty.GetXsecBuffer(emptyBuff, 0)	Empty buffer of xSec queries to an empty vector
102	Empty.GetXsecBuffer(normalBuff, 1000)	Normal buffer of xSec queries to an empty vecto
103	D.GetXsecBuffer(emptyBuff, 0)	Empty buffer of xSec queries
104	D.GetXsecBuffer(smalllBuff, 50)	Small number of queries
105	D.GetXsecBuffer(normalBuff, 1000)	Normal case
106	D.GetXsecBuffer(highBuff, 10000)	Large number of queries
107	D.GetXsecBuffer(negBuff, 1000)	Buffer of negative xSec queries
108	D.GetXsecBuffer(emptyBuff, 1000)	Buffer of zeros
109	D.GetXsecBuffer(highBuff, 0)	Buffer of high valued xSec queries

Table 36: Accuracy

Test #	CPU	GPU
100	CPU result	GPU result
101	CPU result	GPU result
102	CPU result	GPU result
103	CPU result	GPU result
104	CPU result	GPU result
105	CPU result	GPU result
106	CPU result	GPU result
107	CPU result	GPU result
108	CPU result	GPU result
109	CPU result	GPU result

Table 37: Unit Tests

Test #	Code	**Description**
110	Empty.Dump()	Dump an empty Vector
111	D.Dump()	Dump a Vector

Table 38: Accuracy

Test #	CPU	GPU
110	CPU result	GPU result
111	CPU result	GPU result

2.19.2 Accuracy

2.19.3 Performance

2.20 Dump

2.20.1 Unit Tests

2.20.2 Accuracy

2.20.3 Performance

2.21 ThinOut

2.21.1 Unit Tests

Table 39: Unit Tests

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

2.21.2 Accuracy

Table 40: Accuracy

Test #	CPU	GPU
112	CPU result	GPU result
113	CPU result	GPU result
114	CPU result	GPU result
115	CPU result	GPU result
116	CPU result	GPU result
117	CPU result	GPU result

Table 41: Unit Tests

Test #	Code	Description
118	Empty.Sample()	Sample an empty Vector
119	D.Sample()	Sample a Vector

2.21.3 Performance

2.22 Sample

2.22.1 Unit Tests

2.22.2 Accuracy

Table 42: Accuracy

Test #	CPU	GPU
118	CPU result	GPU result
119	CPU result	GPU result

2.22.3 Performance

2.23 GetVectorLength

2.23.1 Unit Tests

Table 43: Unit Tests

Test #	Code	Description
120	Empty.GetVectorLength()	Get the length of an empty vector
121	D.GetVectorLength()	Get the length of a vector

2.23.2 Accuracy

Table 44: Accuracy

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

2.23.3 Performance

2.24 GetIntegral

2.24.1 Unit Tests

Table 45: Unit Tests

Test #	Code	**Description**
122	Code goes here	Description goes here

2.24.2 Accuracy

Table 46: Accuracy

Test #	CPU	GPU
122	CPU time	GPU time

2.24.3 Performance

2.25 SetPoint

2.25.1 Unit Tests

- "rPoint" is a random G4ParticleHPDataPoint
- "nPoint" is a negative G4ParticleHPDataPoint
- "zPoint" is a zero G4ParticleHPDataPoint

Table 47: Unit Tests

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

Table 48: Accuracy

Test #	CPU	GPU
123	CPU result	GPU result
124	CPU result	GPU result
125	CPU result	GPU result
126	CPU result	GPU result
127	CPU result	GPU result
128	CPU result	GPU result
129	CPU result	GPU result
130	CPU result	GPU result
131	CPU result	GPU result
132	CPU result	GPU result

Table 49: Unit Tests

Test #	Code	**Description**
133	Code goes here	Description goes here

Table 50: Accuracy

Test #	CPU	GPU
133	CPU time	GPU time

Table 51: Unit Tests

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

2.25.2 Accuracy

2.25.3 Performance

2.26 Merge

2.26.1 Unit Tests

2.26.2 Accuracy

2.26.3 Performance

2.27 Get15percentBorder

2.27.1 Unit Tests

2.27.2 Accuracy

Table 52: Accuracy

Test #	CPU	GPU
134	CPU result	GPU result
135	CPU result	GPU result

2.27.3 Performance

2.28 Get50percentBorder

2.28.1 Unit Tests

Table 53: Unit Tests

Test #	Code	**Description**
136	Empty.Get50percentBorder()	Get 50 percent Border of an empty vector
137	D.Get50percentBorder()	Get 50 percent Border of a vector

Table 54: Accuracy

Test #	CPU	GPU
136	CPU result	GPU result
137	CPU result	GPU result

2.28.2 Accuracy

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 55: 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 56: 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