

Google Summer of Code 2015



Anshu Aviral

Msc. Physics and BE Computer Science
BITS Pilani, Goa Campus
Aviral.2815@gmail.com

Riccardo de Maria



Beams Department (BE)
CERN, Geneva
Riccardo.De.Maria@cern.ch

Create a Standalone Tracking Library

<https://github.com/SixTrack/SixTrackLib>



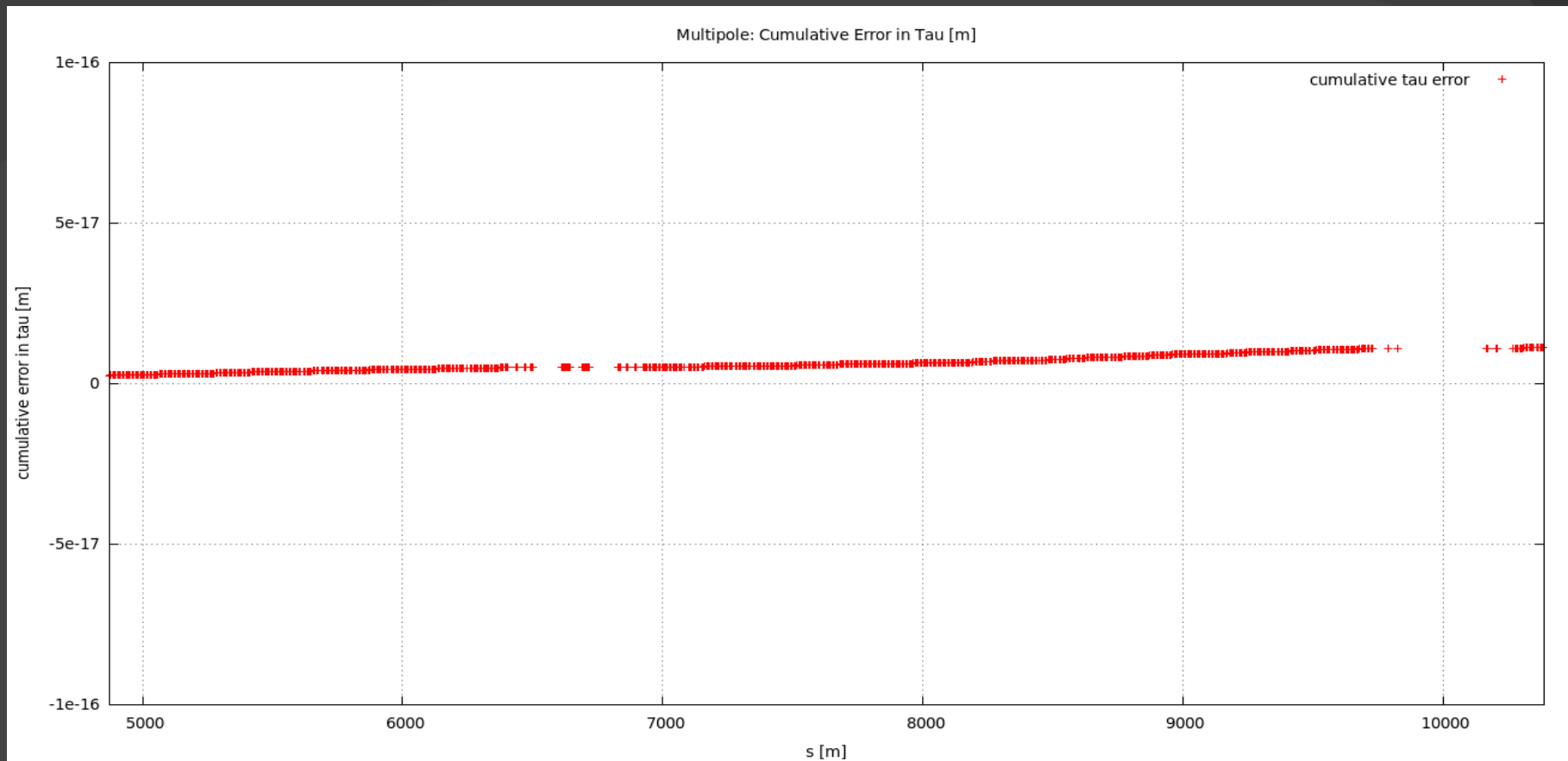
GsoC '15 Deliverables

- 
- Implemented the Generic, 2D and Sixtrack maps in accordance to decided convention
 - Tested and Benchmarked the following Sixtrack maps and functions :
 - Multipole Map
 - Drift and Drift Exact Maps
 - Block and Align Maps
 - Errf function
 - Wzsub function
- 

Code Organization

```
1 #include "hmacro.h"
2 #include <math.h>
3 #include <stdio.h>
4
5 #define kick2d_TYPE 1
6 #define kick2d_float_k 0
7 #define kick2d_int_o 0
8 #define kick2d_int_i 1
9 #define kick2d_int_j 2
10
11 inline void kick2d_track(INT i, INT j, FLOAT k, INT o, FLOAT coordf[]){
12     coordf[j]+=k*pow(coordf[i],o);
13 }
14
15 INT kick2d_single(INT elemi[], FLOAT elemf[], INT elemid, INT parti[], FLOAT partf[], INT partid, INT partn){
16     ELEMINIT;
17     GETATTRF(kick2d,k);
18     GETATTRI(kick2d,i);
19     GETATTRI(kick2d,j);
20     GETATTRI(kick2d,o);
21     INITPARTF;
22     kick2d_track(i,j,k,o,GETPARTF(partid));
23     return 1;
24 }
25
26 INT kick2d_map(INT elemi[], FLOAT elemf[], INT elemid, INT parti[], FLOAT partf[], INT partid, INT partn){
27     ELEMINIT;
28     GETATTRF(kick2d,k);
29     GETATTRI(kick2d,i);
30     GETATTRI(kick2d,j);
31     GETATTRI(kick2d,o);
32     INITPARTF;
33     for(;partid<partn;partid++){
34         kick2d_track(i,j,k,o,GETPARTF(partid));
35     };
36     return 1;
37 }
```

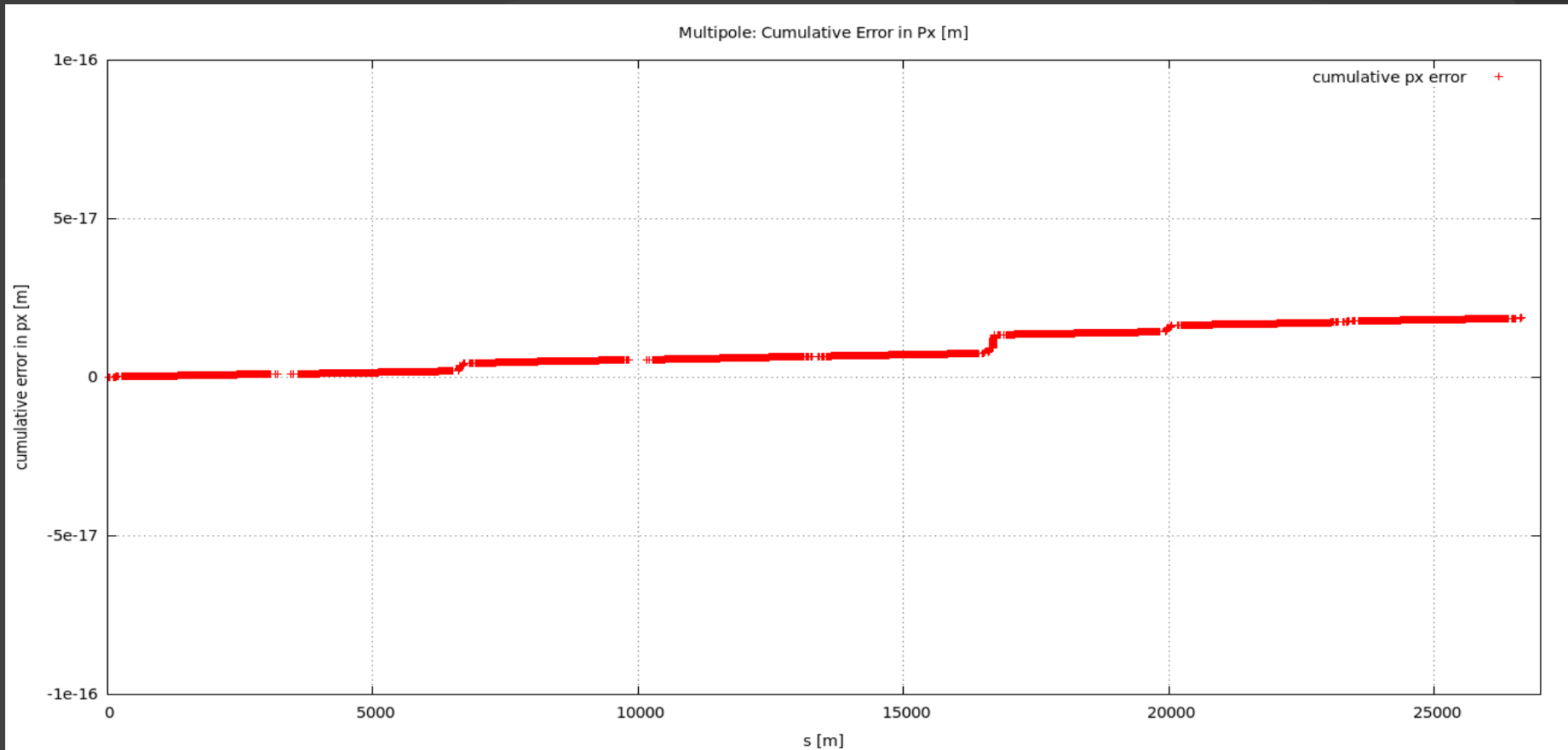
Benchmark Results : Multipole Map (tau error)



Mean Square Error : $2.6050040335e-20$



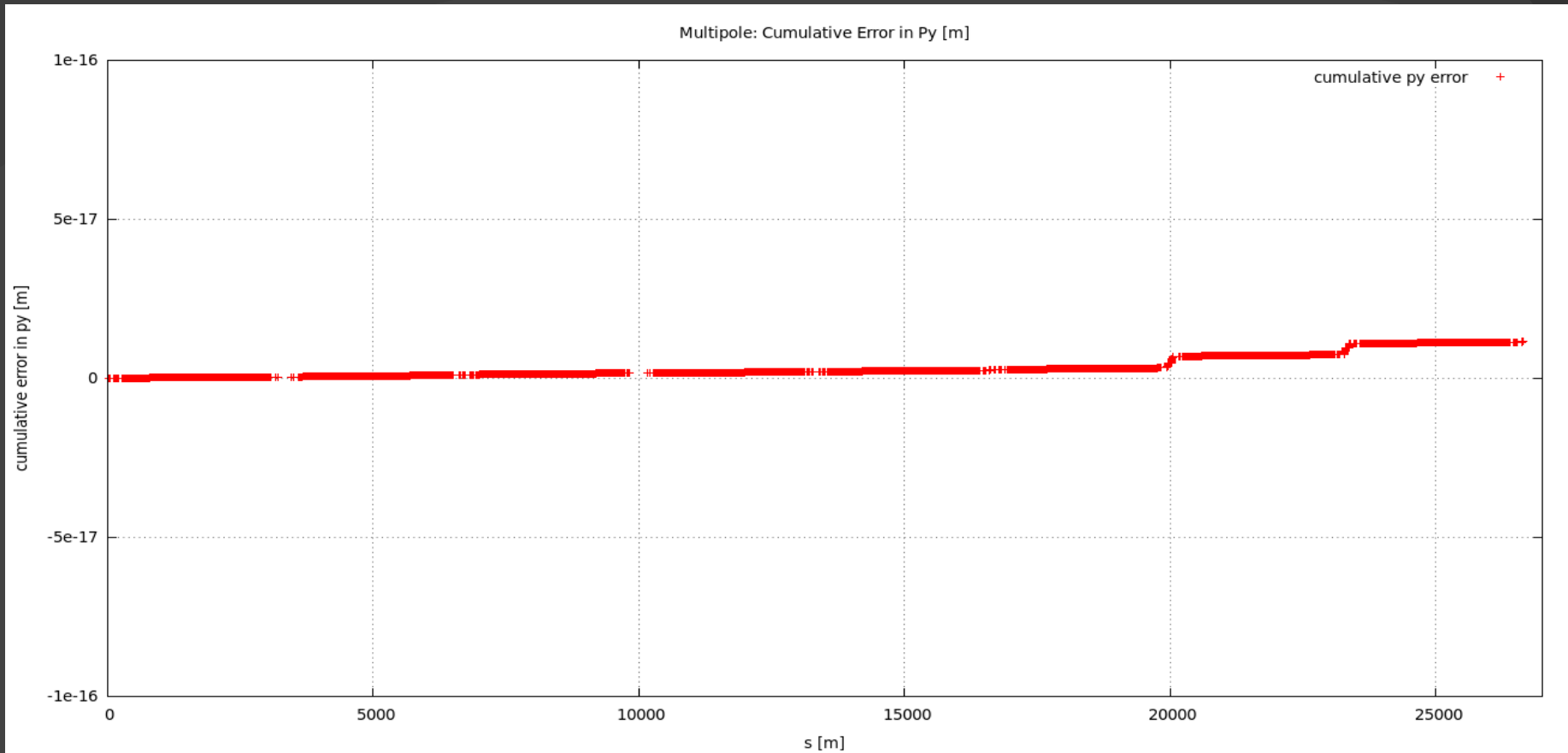
Benchmark Results : Multipole Map (px error)



Mean Square Error : $6.9025851537e-21$



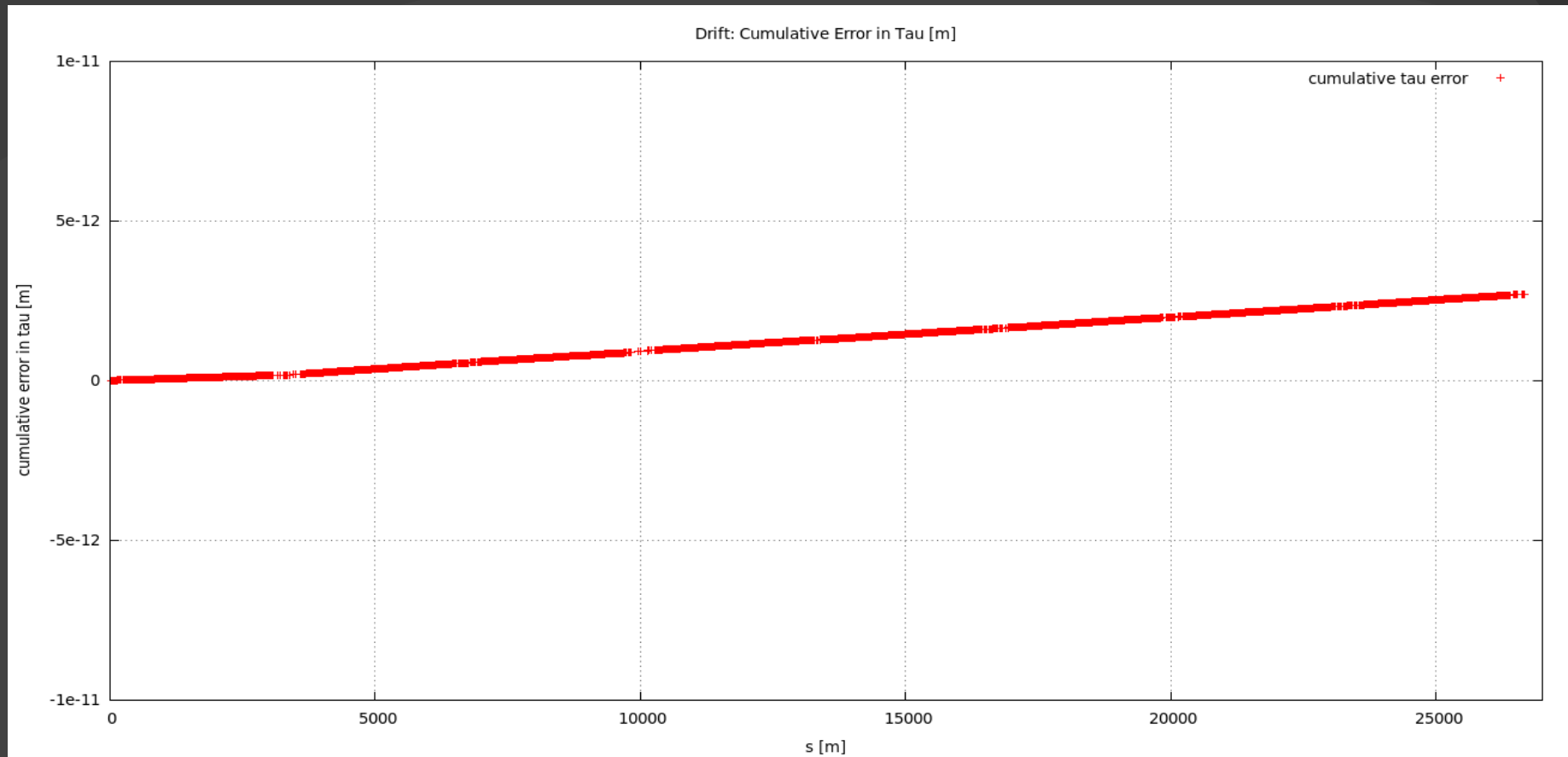
Benchmark Results : Multipole Map (py error)



Mean Square Error : $5.4699917283e-21$



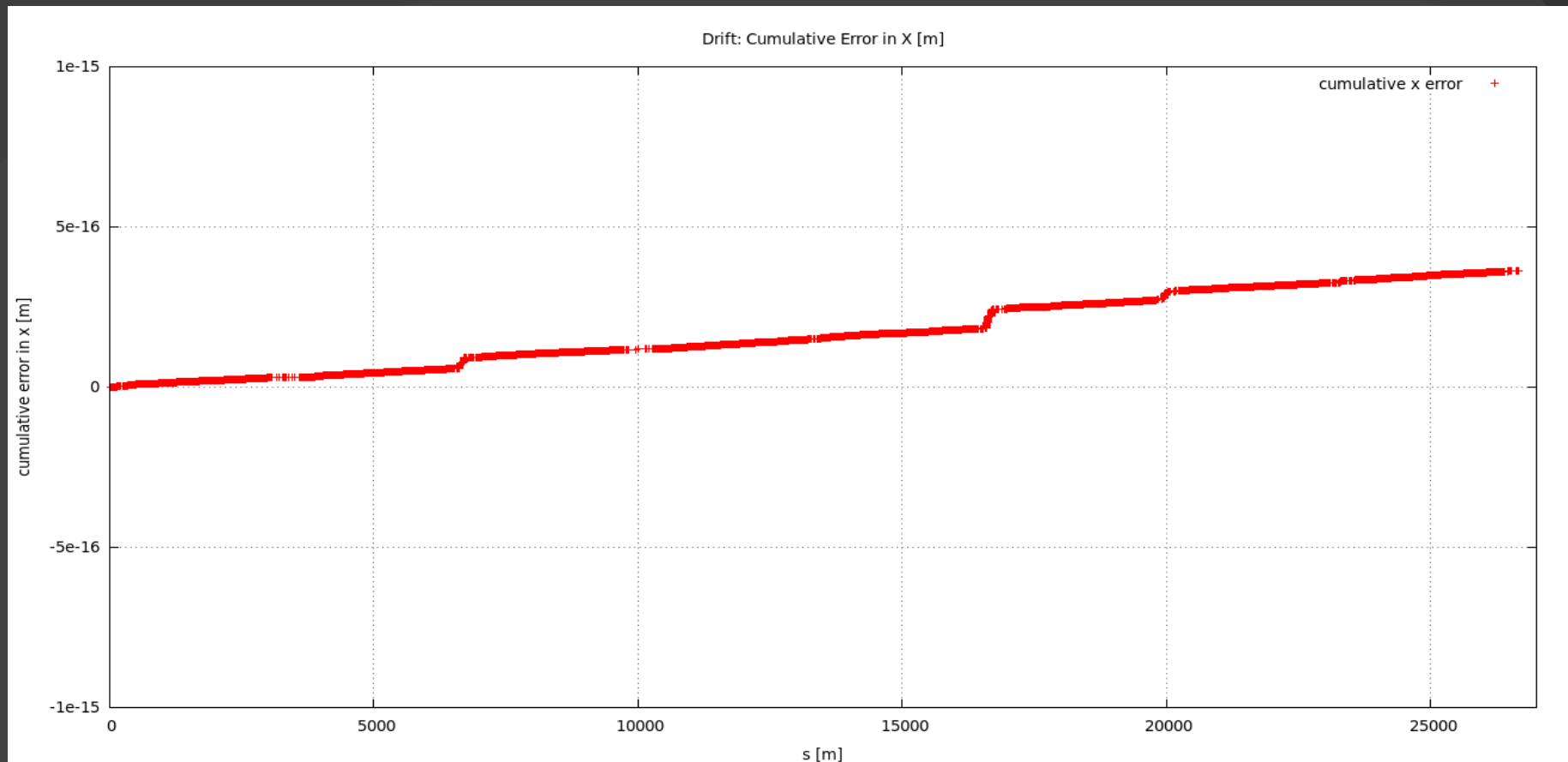
Benchmark Results : Drift Map (tau error)



Mean Square Error : $6.9959249335e-16$



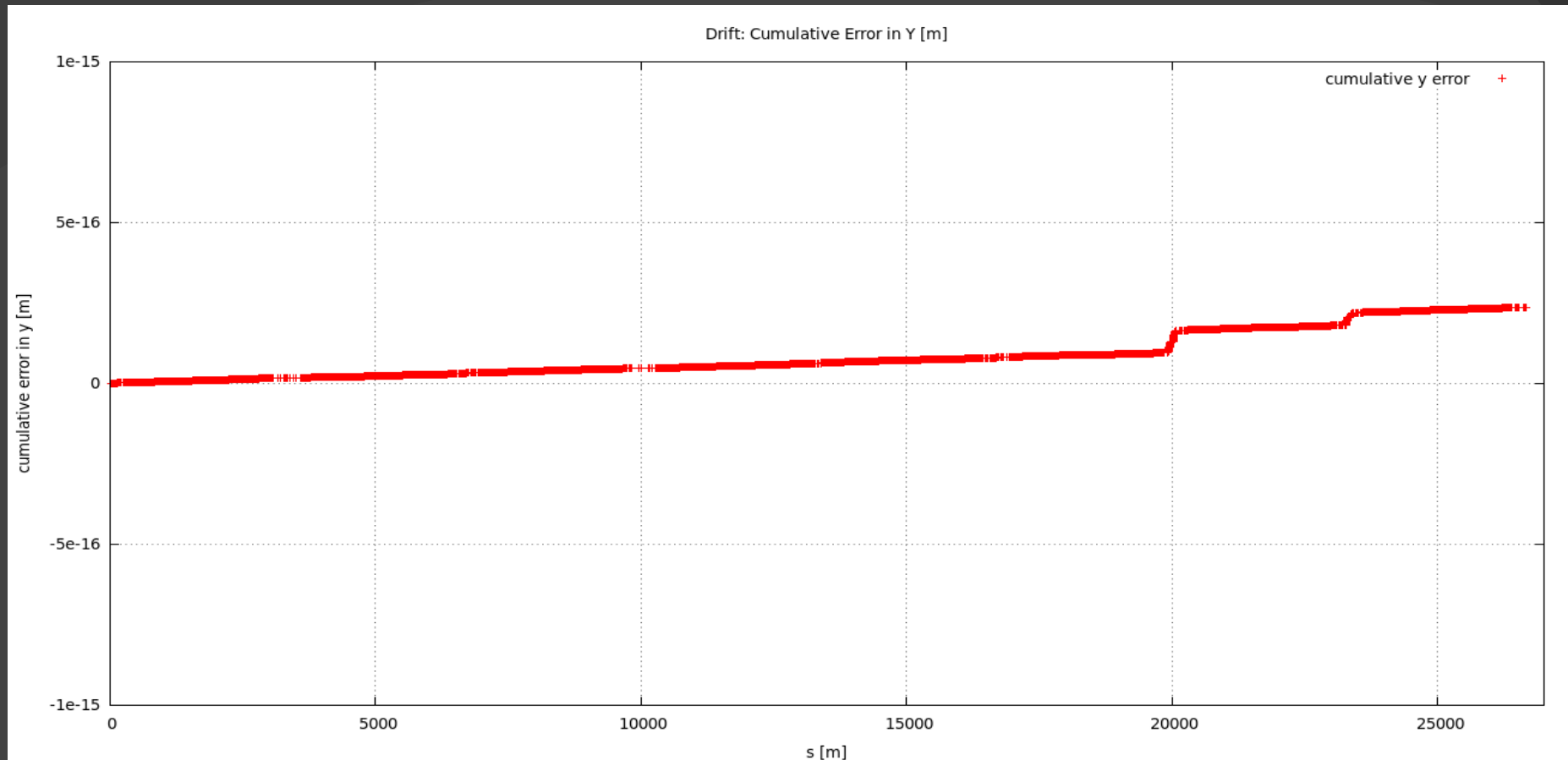
Benchmark Results : Drift Map (x error)



Mean Square Error : $1.3624436040e-19$



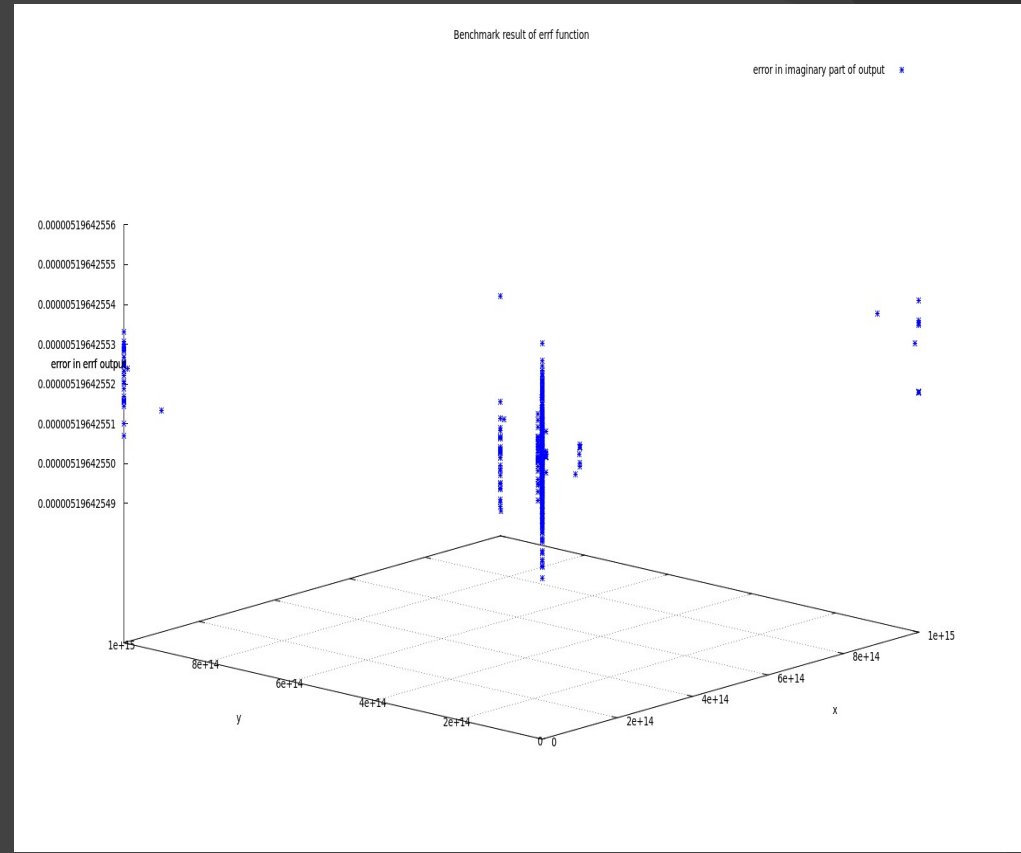
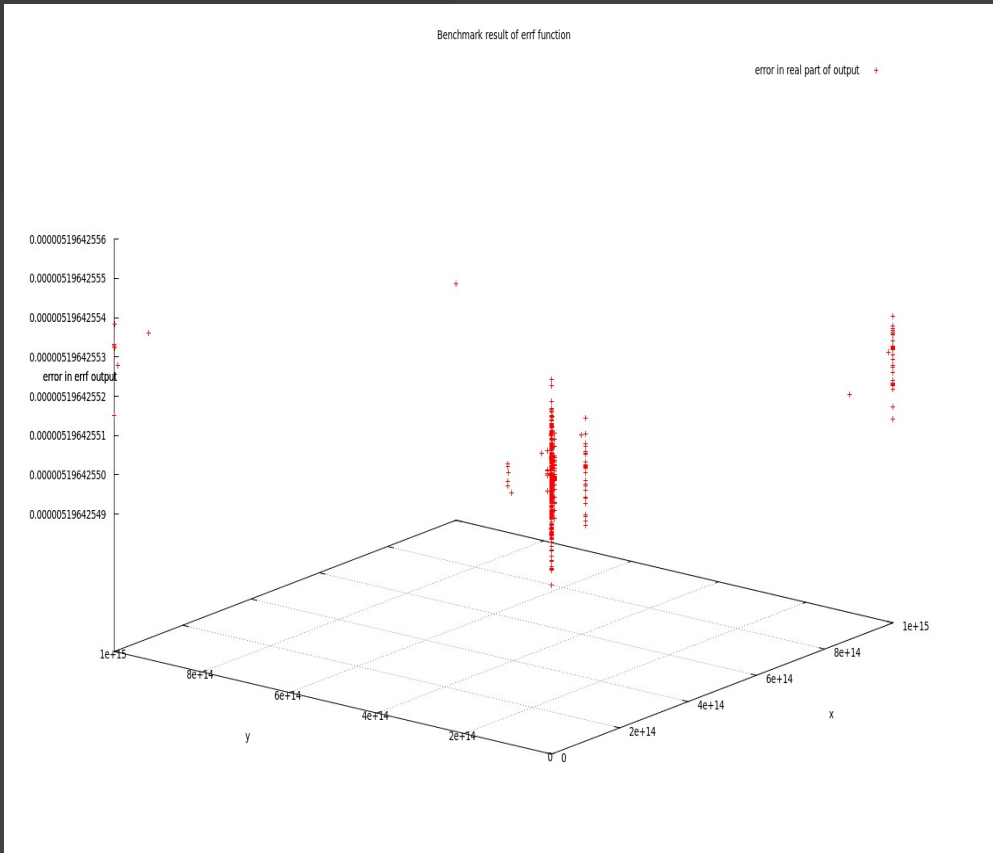
Benchmark Results : Drift Map (y error)



Mean Square Error : $1.2113108619 \times 10^{-19}$




Benchmark Results : Erff function (percentage error in output)



Mean Square Error in real part: $2.5217084049e-08$
Mean Square Error in imag part: $4.0105380294e-09$



Future Work

- 
- Benchmark the remaining map: *beambeam*
 - Build a GPU implementation using CUDA/OpenCL
 - Incorporate Differential Algebra Library for higher dimensional vectors



Thank You



Questions ?

