Delaunay Triangulations on the GPU

05/09/2025

Patryk Drozd

MSc for High-performance Computing

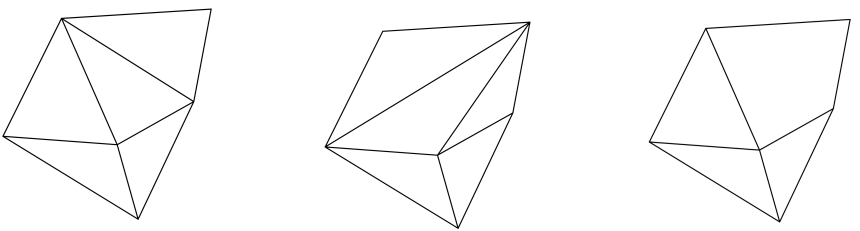
drozdp@tcd.ie

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

What is a Triangulation

Definition 1.

A *triangulation* of a planar point set P is a subdivision of the plane determined by a maximal set of non-crossing edges whose vertex set is P.



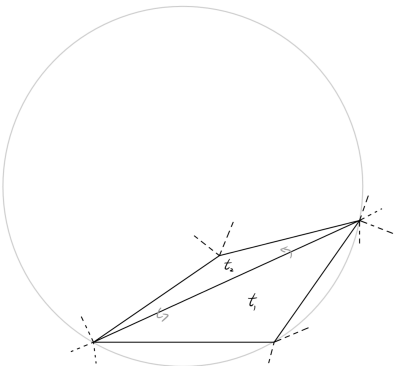
(a) (b) (c)

Figure 1: Examples of two triangulations (a) (b) on the same set of points. In (c) an illustration of a non maximal set of edges.

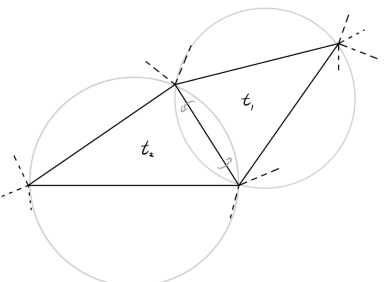
What is a Delaunay Triangulation

Definition 2.

A *Delaunay triangulation* of a point set P is a triangulation such that the circumsphere of any triangle in the triangulation does not contain any other point in P.

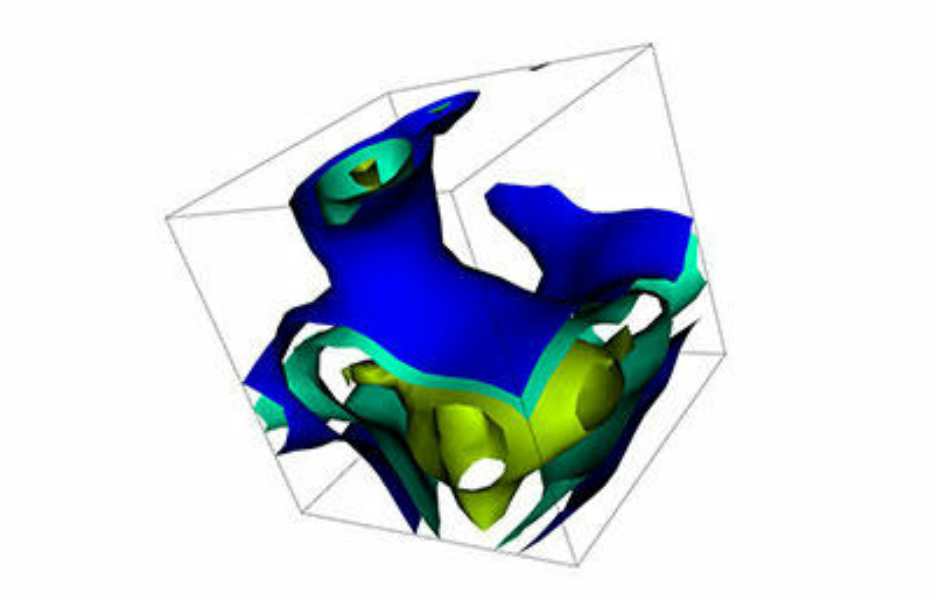


(a)



(b)

Why Delaunay Triangulations



The Algorithm

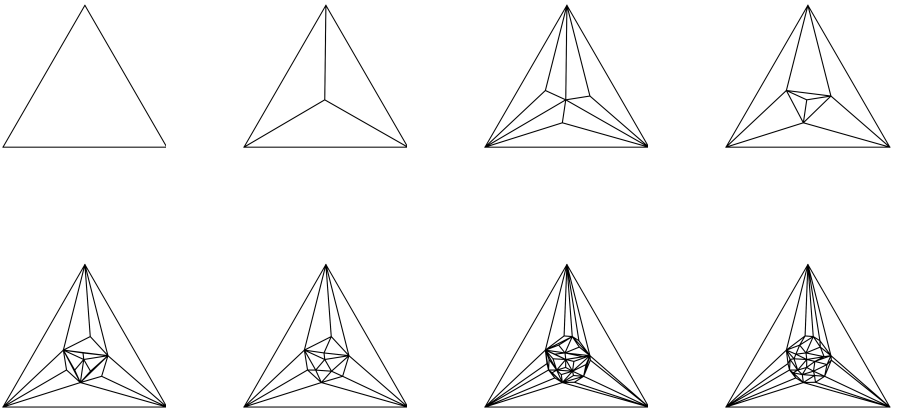
Parallel Algorithm

Algorithm 1: Parallel point insertion and flipping

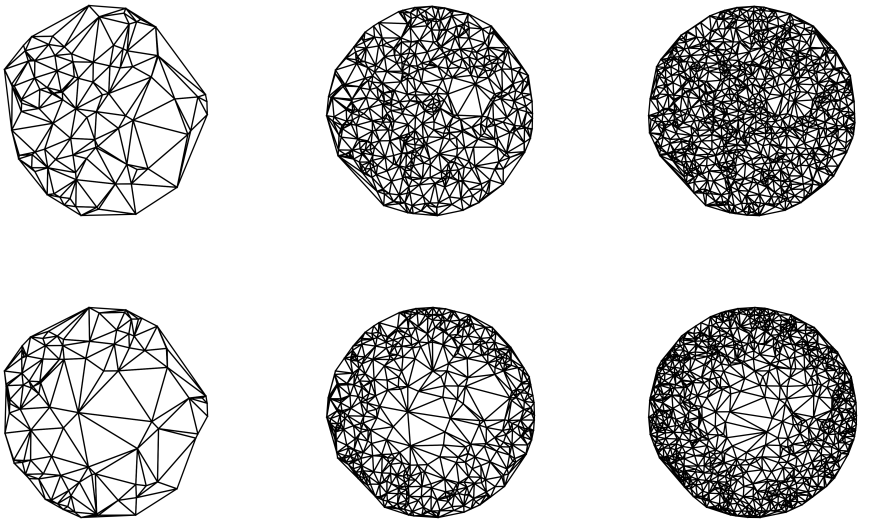
Data: A point set P
Out: Delaunay Triangulation T

Initialize T with a triangle t enclosing all points in P
Initialize locations of $p \in P$ to all lie in t
while there are $p \in P$ to insert
  for each $p \in P$ **do in parallel**
    choose $p_t \in P$ to insert if any
  for each $t \in T$ with p_t to insert **do in parallel**
    split t
  while there are illegal edges
    for each triangle $t \in T$ **do in parallel**
      mark whether it should be flipped
    for each triangle $t \in T$ in a configuration marked to flip **do in parallel**
      flip t
  update locations of $p \in P$
return T

Visualization



Visualization



Challenges

GPU programming

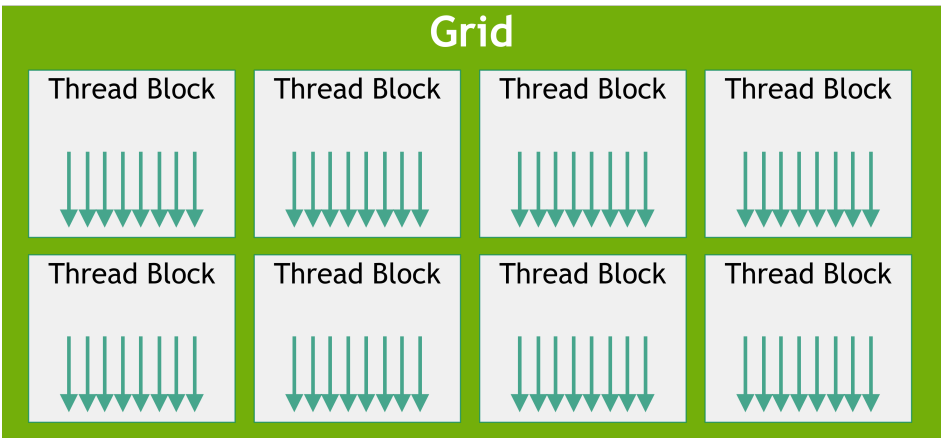
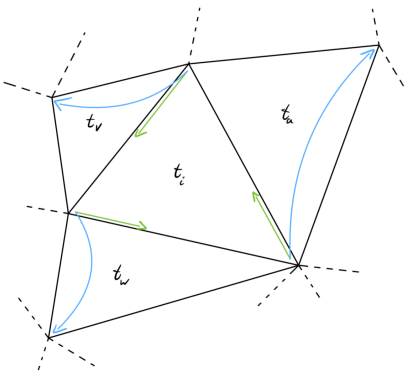
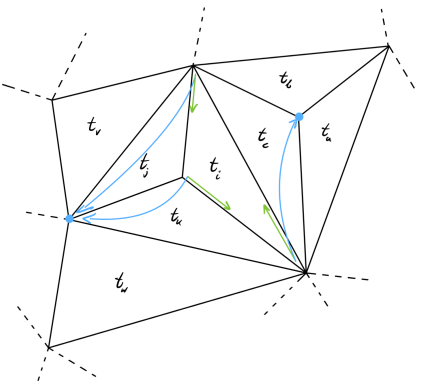


Figure 6: An illustration of the structure of the GPU programming model.

Parallel point insertion

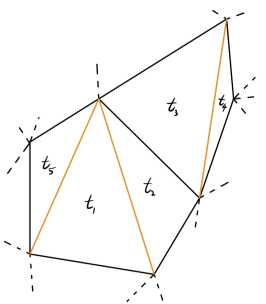


(a) Before insertion.

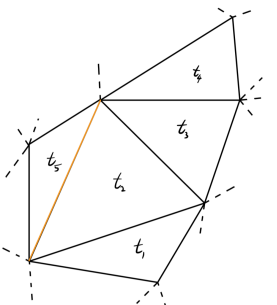


(b) After insertion.

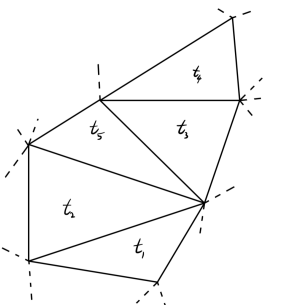
Parallel flipping



(a) Pass 0



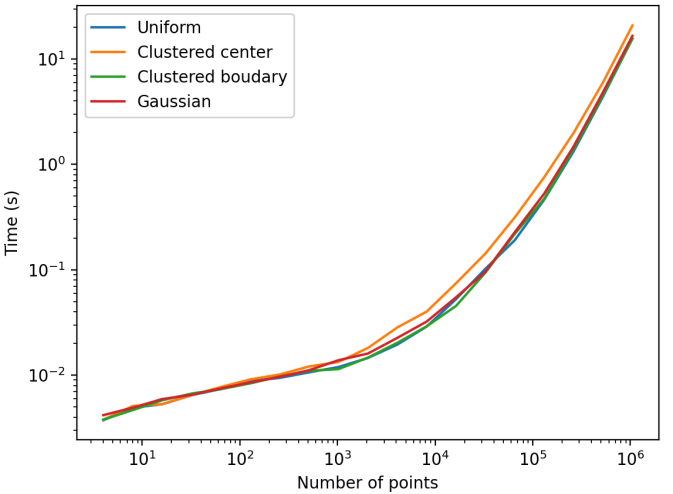
(b) Pass 1



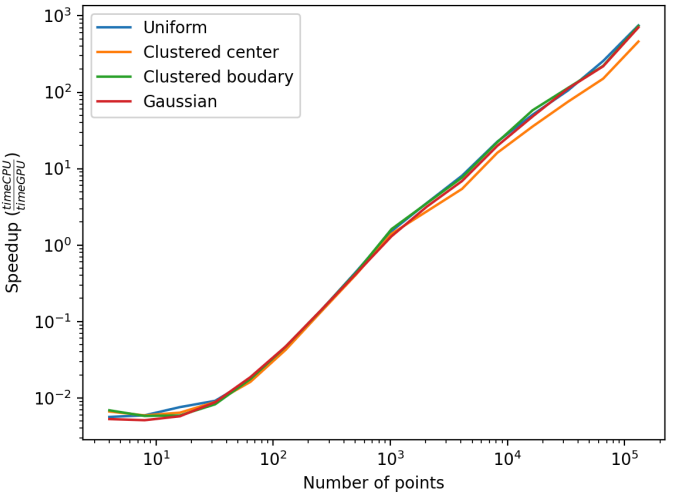
(c) Pass 2

Analysis

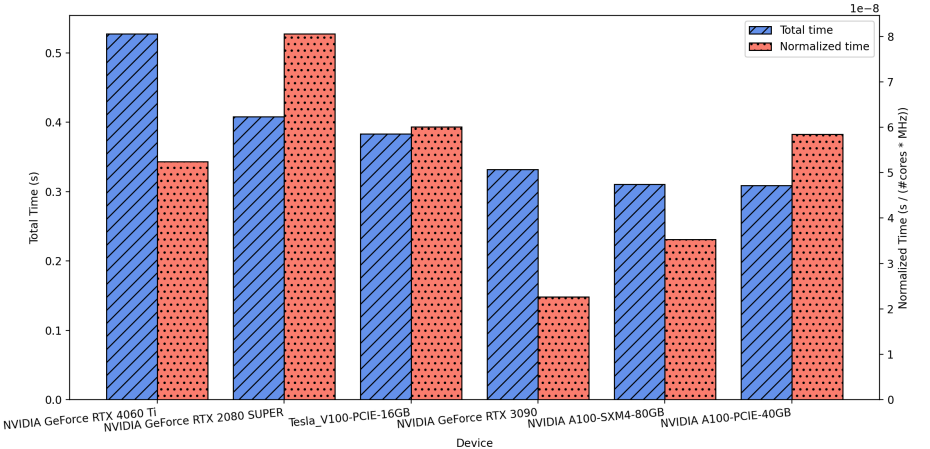
Timing


Uniform
Clustered center
Clustered boudary
Gaussian
Time (s)
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
10^{6}
Number of points

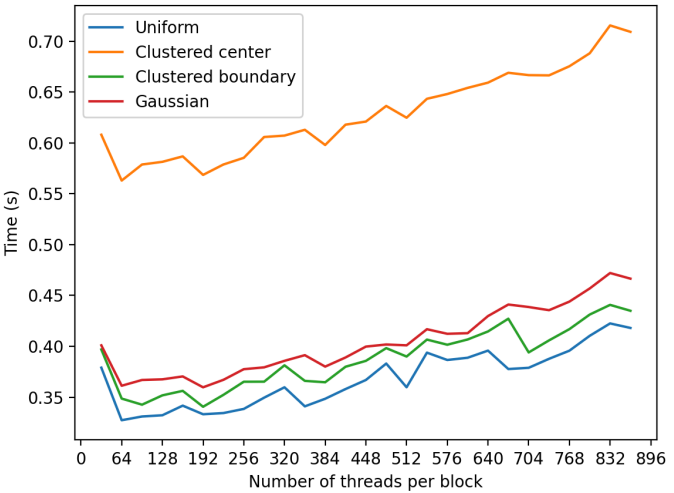
Speedup


Uniform
Clustered center
Clustered boudary
Gaussian
Speedup ($\frac{timeCPU}{timeGPU}$)
Number of points
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}

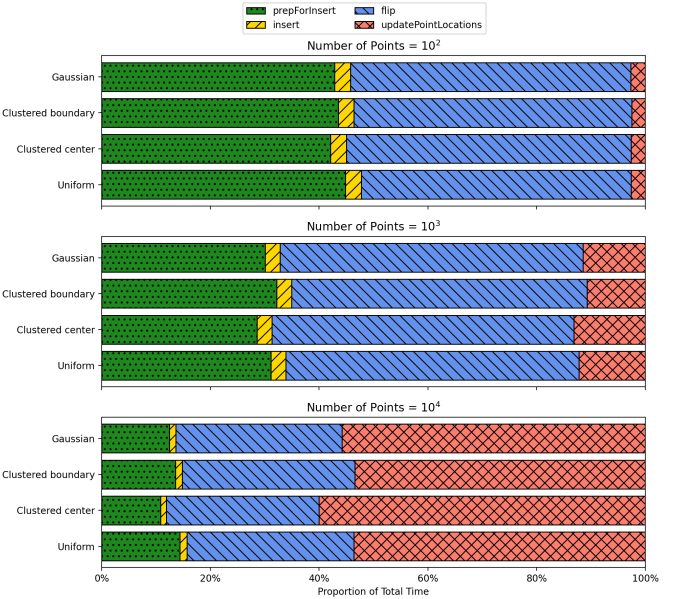
Scaling


1e−8
Total time
Normalized time
Total Time (s)
Normalized Time (s / (#cores * MHz))
0.0
0.1
0.2
0.3
0.4
0.5
0
1
2
3
4
5
6
7
8
NVIDIA GeForce RTX 4060 Ti
NVIDIA GeForce RTX 2080 SUPER
Tesla_V100-PCIE-16GB
NVIDIA GeForce RTX 3090
NVIDIA A100-SXM4-80GB
NVIDIA A100-PCIE-40GB
Device

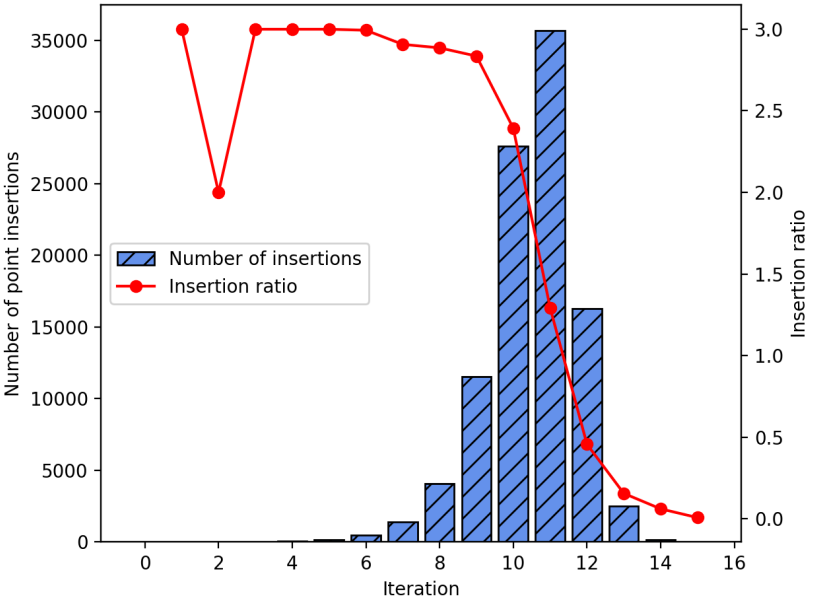
Block Size


Uniform
Clustered center
Clustered boundary
Gaussian
Time (s)
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0
64
128
192
256
320
384
448
512
576
640
704
768
832
896
Number of threads per block

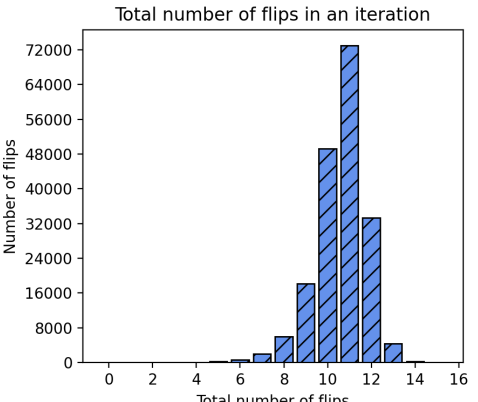
Profiling

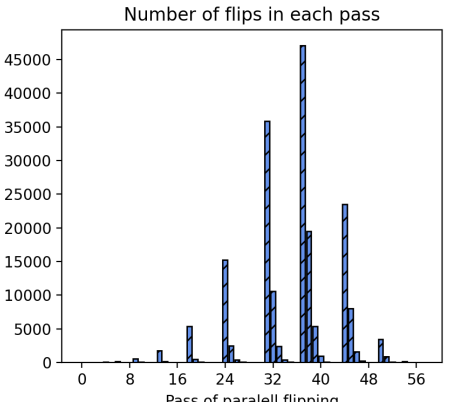

prepForInsert
insert
flip
updatePointLocations
Number of Points = 10^2
Number of Points = 10^3
Number of Points = 10^4
Gaussian
Clustered boundary
Clustered center
Uniform
0%
20%
40%
60%
80%
100%
Proportion of Total Time

Point Insertions



Flipping


Total number of flips in an iteration
Number of flips
72000
64000
56000
48000
40000
32000
24000
16000
8000
0
0
2
4
6
8
10
12
14
16
Total number of flips


Number of flips in each pass
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
0
8
16
24
32
40
48
56
Pass of paralell flipping

Conclusion

Thank You! Any Questions?