Computing Geodesics in Parallel

ME570X

4/24/2018

Nathan Scheirer

Background

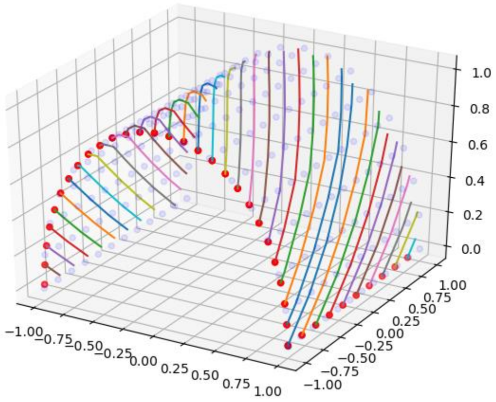
- A geodesic is a generalization of the notion of a “straight line” to “curved spaces.” [1]
  - Defined as a curve whose tangent vectors remain parallel as it progresses along a surface.
- Existing computational methods:
  - Gradient descent/Midpoint method for single geodesic lines [2]
  - Fast Marching Method [3]
  - Fast Iterative Method [4]
  - Fast Sweeping Method [5]

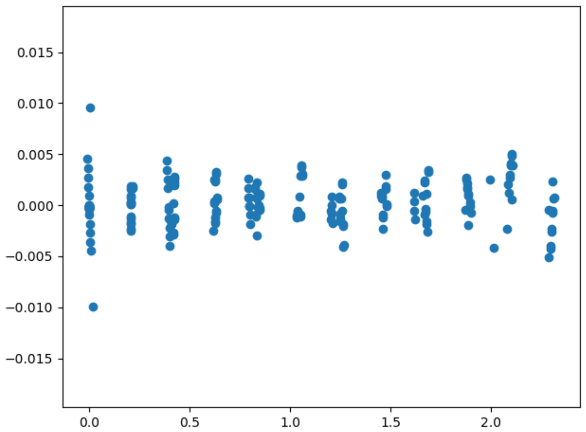
Problem Statement

- When creating a parameterization of a surface based on geodesics, especially on large meshes, the required number of geodesics to be computed can be quite large.
- Therefore, the use of a geodesic kernel executed on the GPU could speed up the calculation of the geodesics required for parameterization.

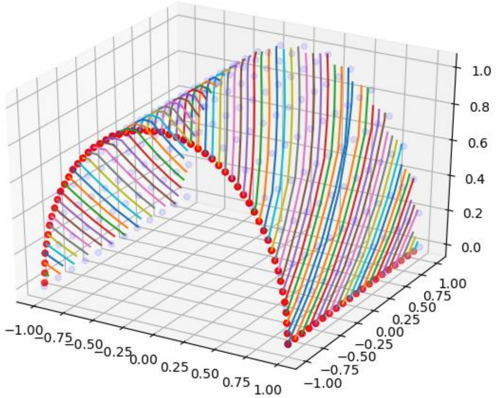
Number of Geodesics

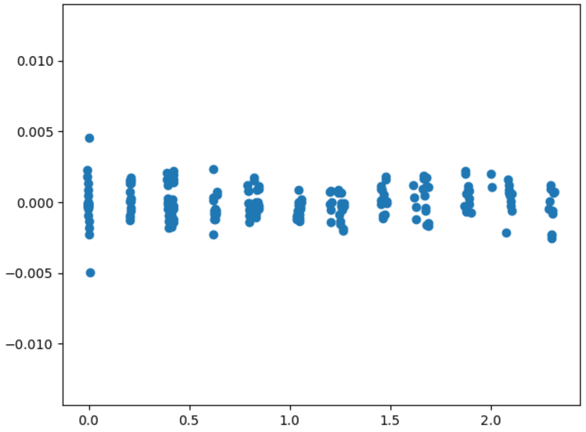
47



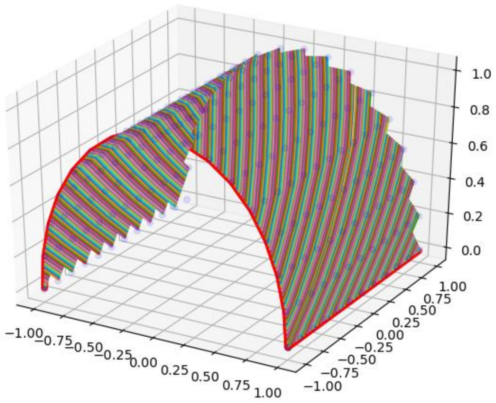


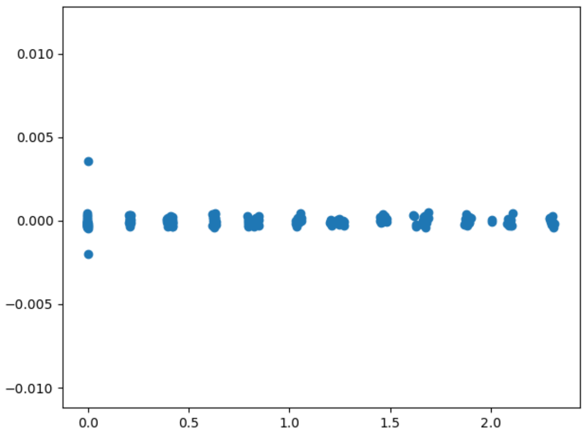
74





371





Proposed Solution

- A sequential process has already been developed in Python
  - Can compute approximately 700 geodesics in about 1.5 minutes
- Method:
  - For each geodesic, dedicate one thread to perform the sequential algorithm
  - All geodesics will be calculated at the same time
- Implementation:
  - Use Numba to accelerate the sequential python code
  - Algorithm on the following slides

General Algorithm

- Given a starting point, fiber direction, surface normal, fiber interval, and the fiber material properties
- All fiber start points are calculated using geodesics running in each valid principle direction (i.e. x, y, z, -x, -y, -z)
- For each start point find initial facet and the rotated fiber direction for that element (this saves computation time in the kernel)
- Send start points, elements, and fiber directions to the geodesic kernel
- Store the points of intersection between the geodesic and each element edge
- Cleanup and interpolate any missed vertices

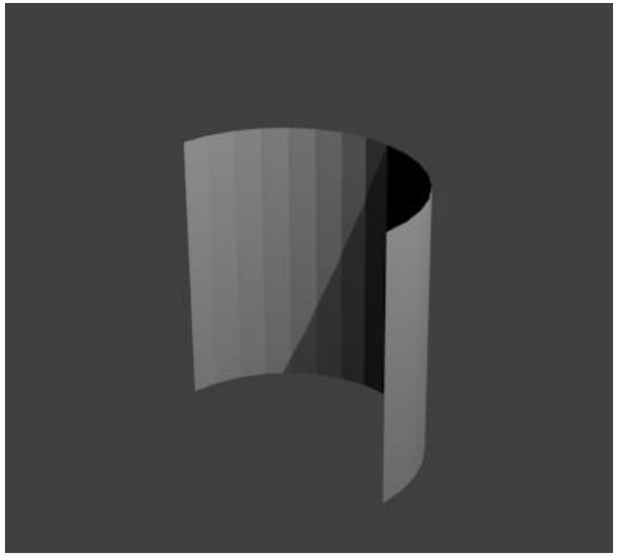
Geodesic Kernel

- For each start point:
  - Check current element
    - If not in list of traveled faces, add it
  - Traverse element:
    - Find neighboring elements
    - Convert current element vertices and fiber direction to barycentric coordinates
    - Determine the next edge to be crossed given the current intersection point and fiber direction
    - Find which element shares the crossed edge
    - Convert the intersection point back into 3d coordinates
      - Set current point to new intersection point
    - Rotate current fiber direction into plane with the next element
  - Repeat until a model edge is found

Results

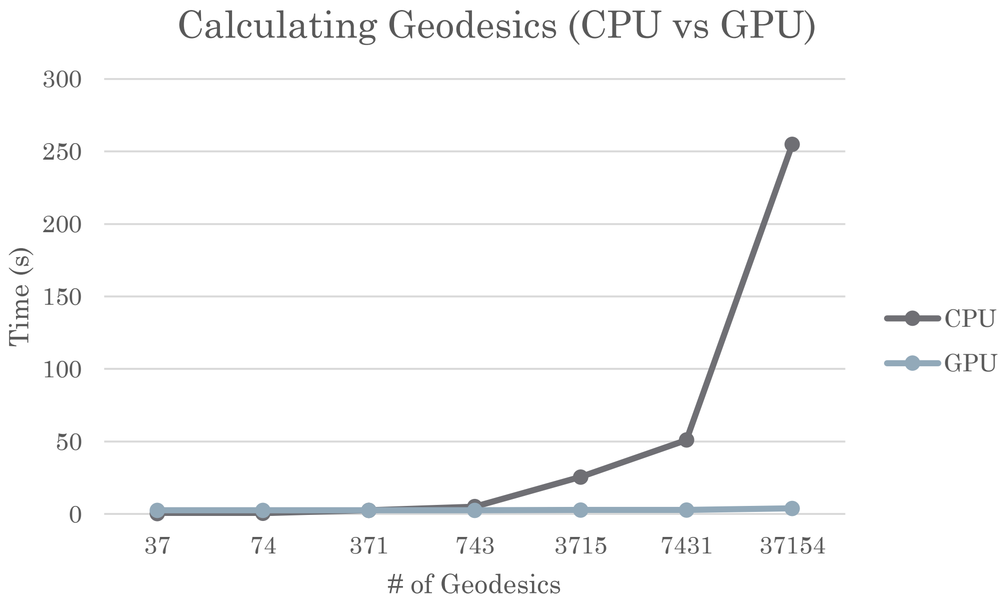
GPU: GTX 970
4 GB Memory
1664 CUDA Cores
1329 MHz (1.33 GHz)

CPU: i7-6700K
4 Cores
4.0 GHz



Model: Cylinder (204 vertices)

	CPU (s)	GPU (s)*	
Number of geodesics	1 Thread	[B]/[T per B]/[T]	
37	0.257	[3]/[16]/[48]	2.53
74	0.508	[5]/[16]/[80]	2.516
371	2.524	[24]/[16]/[384]	2.531
743	5.083	[47]/[16]/[752]	2.562
3715	25.526	[233]/[16]/[3728]	2.685
7431	50.971	[465]/[16]/[7440]	2.689
37154	254.942	[2323]/[16]/[37168]	3.793
* [Blocks]/[Threads per block]/[Threads]			



Results Comparison

- Iterative Method [6]:

	Number of vertices	Original method		Using a heap	
		Changes	Time	Changes	Time
1	49	60649	0.60	38089	0.65
2	29	37021	0.37	20469	0.35
3	46	126303	1.22	48478	0.82
4	103	1291478	12.58	549298	9.37
5	126	481039	4.85	80585	1.40
6	75	161026	1.60	51123	0.88

[6]

- Presented solution:

	CPU (s)	GPU (s)*	
Number of geodesics	1 Thread	[B]/[T per B]/[T]	
37	0.257	[3]/[16]/[48]	2.53
74	0.508	[5]/[16]/[80]	2.516
371	2.524	[24]/[16]/[384]	2.531
743	5.083	[47]/[16]/[752]	2.562
3715	25.526	[233]/[16]/[3728]	2.685
7431	50.971	[465]/[16]/[7440]	2.689
37154	254.942	[2323]/[16]/[37168]	3.793

* [Blocks]/[Threads per block]/[Threads]

Thank you,
Questions?

References

- [1] https://en.wikipedia.org/wiki/Geodesic
- [2] https://cs.stanford.edu/~jbaek/18.821.paper2.pdf
- [3] http://www.pnas.org/content/pnas/95/15/8431.full.pdf
- [4] https://ac.els-cdn.com/S1877050914003470/1-s2.0-S1877050914003470-main.pdf?_tid=f42aa592-bbef-46ac-adfd420e1f461a91&acdnat=1520357194_81c8eb277732e9cd13dd8a0ea576b095
- [5] https://ac.els-cdn.com/S002199911200722X/1-s2.0-S002199911200722X-main.pdf?_tid=caae28d8-c7f6-4513-99de840c0256c6b3&acdnat=1520358269_196d67c46972d33edec65f8524634f7f
- [6] https://www.sciencedirect.com/science/article/pii/S0097849305001299