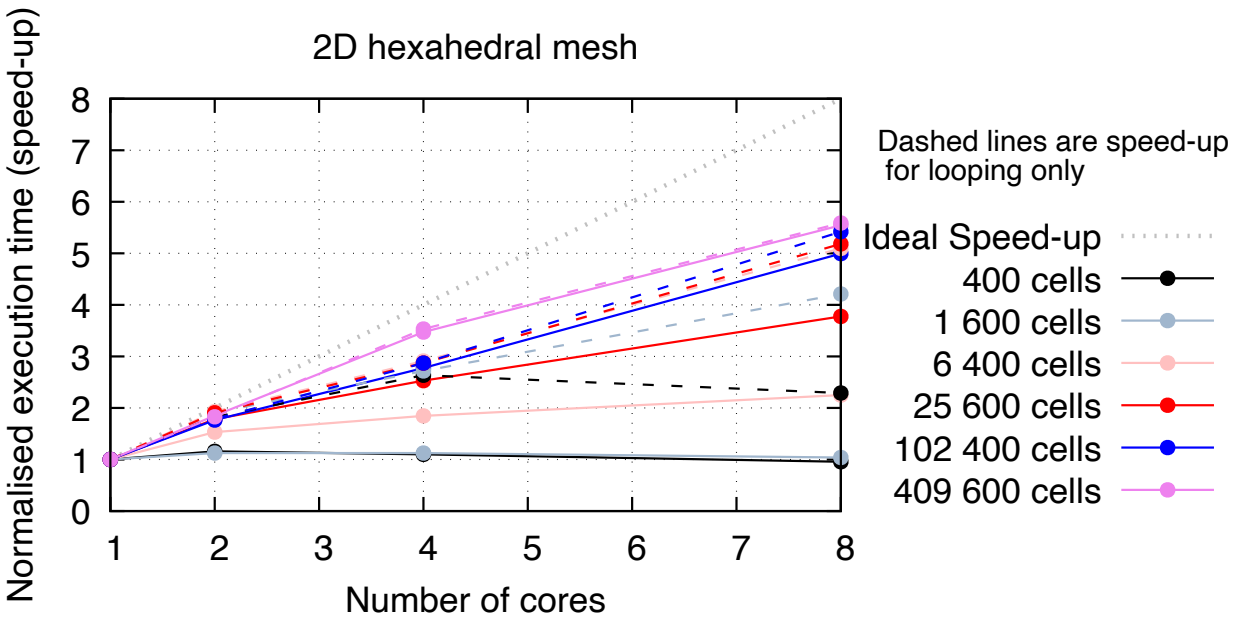
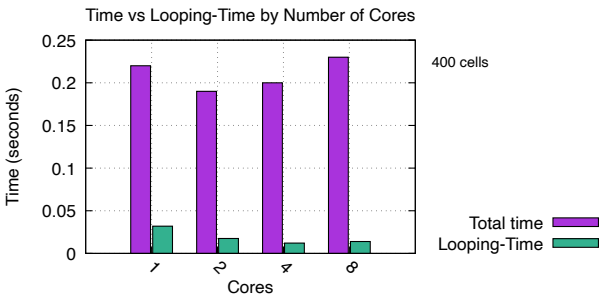
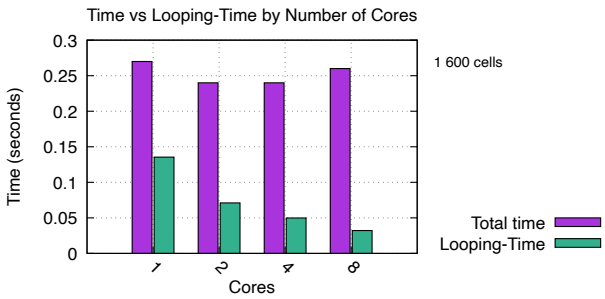
Results for testBuildStencilsParallel_v2

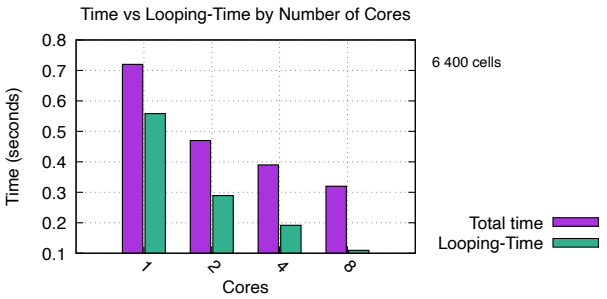
- Looping is performed 300 times (6 integration points per face)
    - In each loop field D is exchanged between processors
- Cubic interpolation (p=3) is assumed:
    - 20 cells in stencil in 2D
    - 100 cells in stencil in 3D
- Stencils are build using layer technique
- Speed-up is calculated by dividing time with time in serial
- Total time - looping time = stencil construction
- Looping time is max looping time over all processors

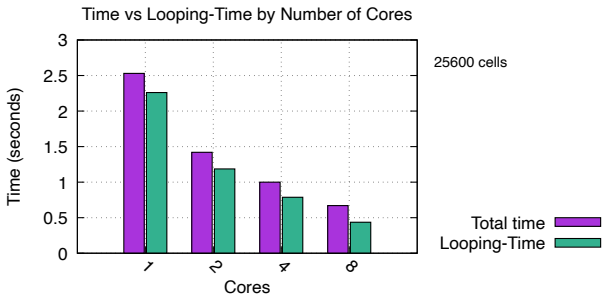
2D case

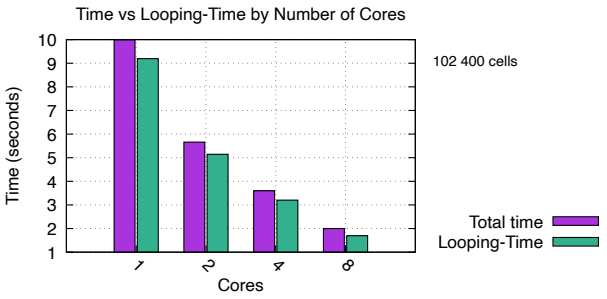


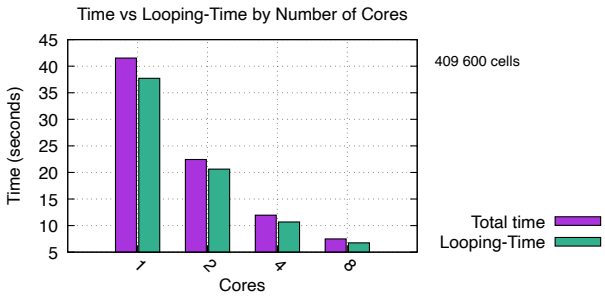




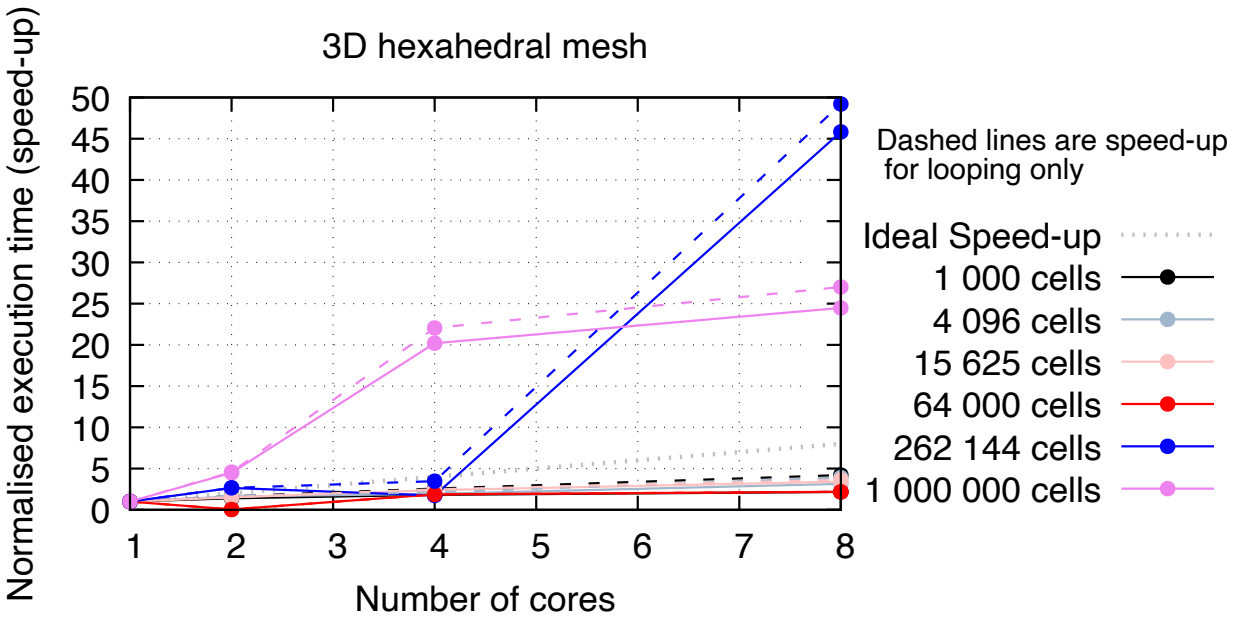




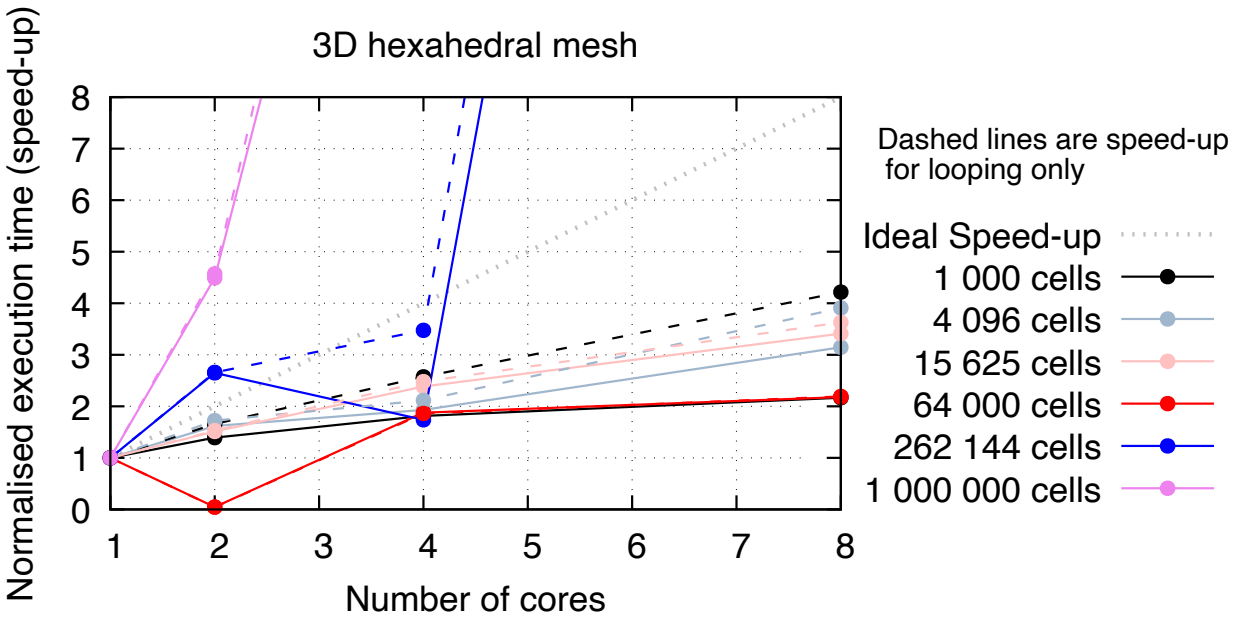


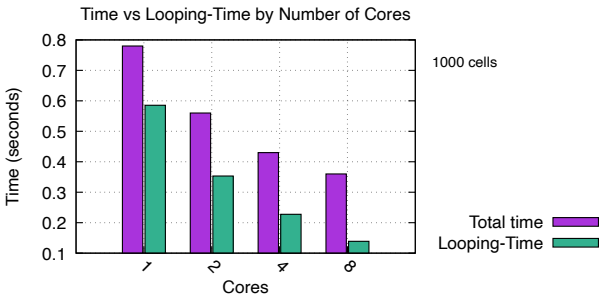


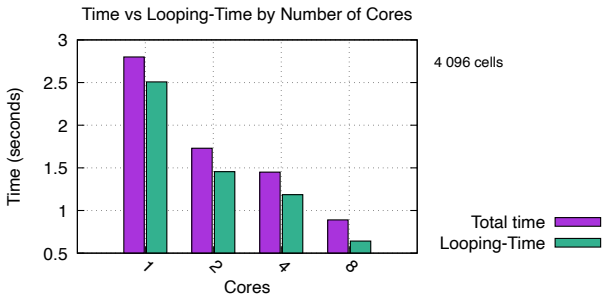
3D case

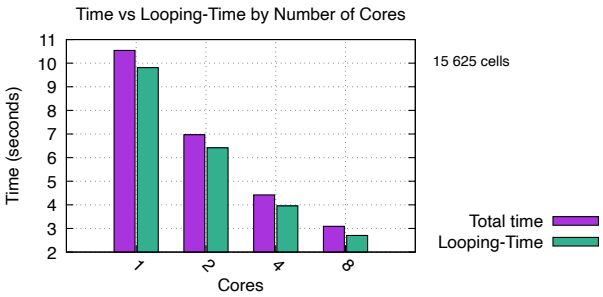


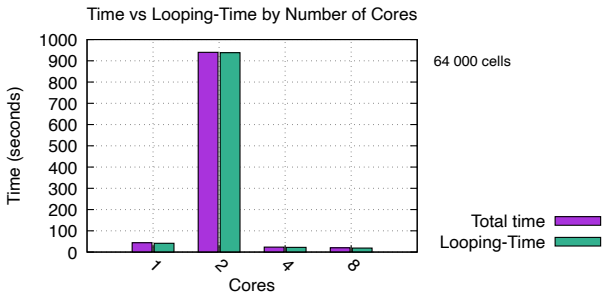
Same diagram with limited y axis:

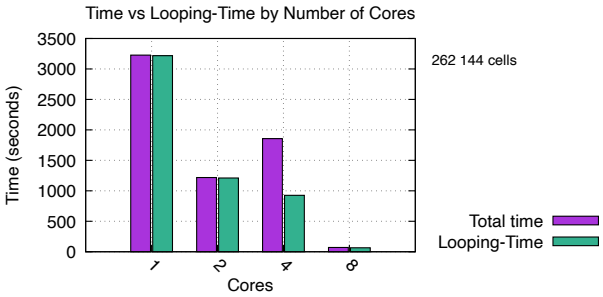


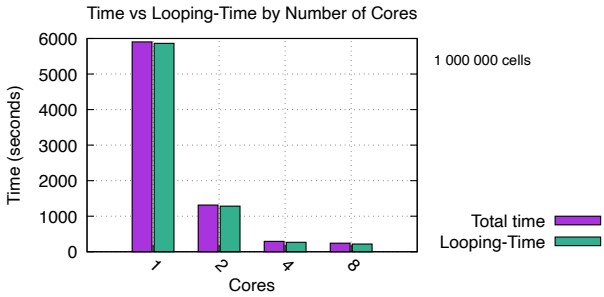








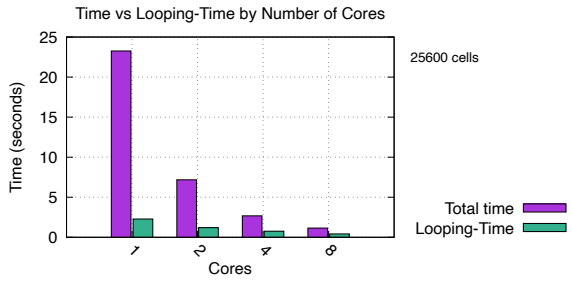


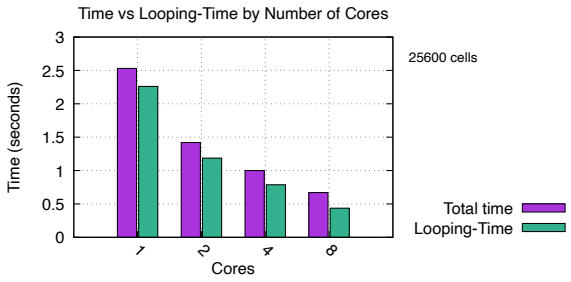


Some conclusions:

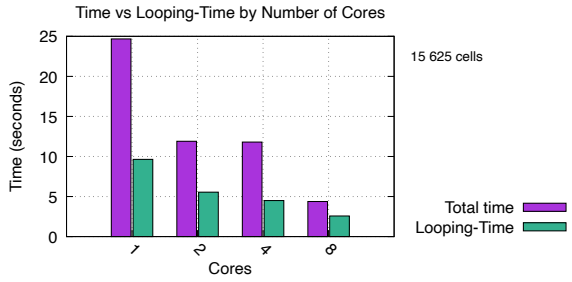
- Building the stencil using indexedOctree is slower. In 2D indexedOctree is extremely slow. For diagrams in paper i was using idexedOctree - according to these results we will improve timings a lot with layer technique.

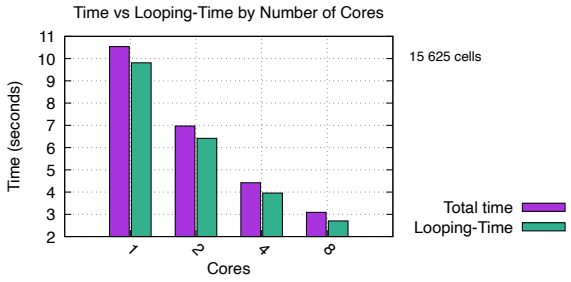
Timings for 2D case - (**left octree, right layer technique**)





Timings for 3D case - (**left octree, right layer technique**)





- I am observing some unexplained anomalies in 3D, with extreme speed-ups or slow-downs.