Geometric-aware Partitioning on Large-scale Data for Parallel Quad Meshing

Wuyi Yu, Qin Jim Chen, Jian Tao, Xin Li (xinli@lsu.edu)
Center for Computation and Technology, Louisiana State University

INTRODUCTION

Finite Element Mesh Generation is important in scientific and engineering computing. In many problems, geometric data need to be first discretized into meshes, whose quality dictates the stability and efficiency of the subsequent simulation and analysis tasks.
2D data can be easily discretized using unstructured triangular meshes. Converting them to structured quadrilateral (quad) meshes can often improve the efficiency of storage and computation. However, automatic generation of high-quality structured mesh is often difficult, especially when in many scientific tasks, data are extremely big and complicated. We aim to develop effective parallel meshing algorithms to solve this problem using HPC. This work can benefit many scientific tasks in improved performance in storage, simulation, and modeling.

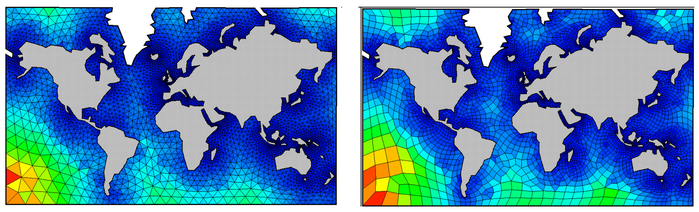


Fig.1 Left: Irregular (Triangle) Mesh, Right: Regular (Quadrilateral, or Quad) Mesh
Image Courtesy of Argus One (www.argusone.com)

Parallel Mesh Generation Pipeline:
1. Decomposition: Partition the given region into a set of sub-regions
2. Mesh Generation: Tessellate the sub-regions in parallel
3. Post-processing: Merge and refine the composed meshes

Partitioning quality dictates the whole pipeline's efficiency and meshing quality.

Criteria: For efficient parallel mesh generation, the decomposition should have:
1. Balanced (similar) subregion areas
2. Small partition boundary length
3. Nice subregion geometric property (the corner angles of subregions should be close to $\frac{k\pi}{2}, k = 0,1,2,3$)

METHOD

Basic Idea:
Existing partitioning algorithms such as METIS ([2]) only model criteria 1 and 2 (parallel efficiency), while rigorously formulating criterion 3 is often very expensive. We observe that: if the cells of an initial graph tessellation has near-square geometry, then the partition performed on the dual graph of this tessellation have desirable corner angles.

Our Proposed Two-Step Partitioning:
1. L_∞-norm Centroidal Voronoi Tessellation → initial tessellation with near-square cells
2. Graph Partitioning on L_∞-CVT tessellation → to obtain good load balancing and minimized communication costs for Parallel Efficiency

L_∞-norm Centroidal Voronoi Tessellation (L_∞-CVT):
A L_∞-CVT[1] of a given set of distinct sites $\boldsymbol{X} = \{\boldsymbol{x_i}\}_{i=1}^n$ is defined b the Voronoi cells which minimizes the Voronoi energy:

$$F(\mathbf{X}) = F([\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]) = \sum_i \int_{\Omega_i \cap \Omega} \|\mathbf{y} - \mathbf{x}_i\|_p^p d\mathbf{y}$$

The trajectory of L_p (x,y) = 1 with different p values. With the increase of p the trajectory gradually approximates the unit square L_∞(x,y) = 1 gives a square. We use $p = 8$ to get a good approximation of L_∞ while keeping the energy smooth. Fig.2 is from [3], for more details please refer to our paper[3].

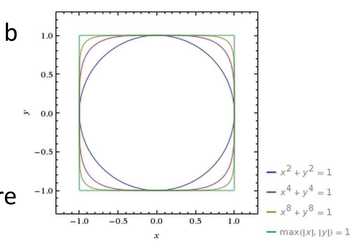


Fig. 2

Graph Partitioning:

On dual graph of CVT tessellation, for each node i, assign an indicator $x_i = \begin{cases} 1, & i \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$

The partitioning should be optimized via three criteria:

Balanced Sub-region Areas: E_A, $\min |\sum_{i \in V} x_i w_i - \frac{A}{k}|$,
where w_i is vertex weight (subregion area), A is the total area, k is the number of sub-regions.

Small Communication Cost: E_C, $\min \sum_{ij \in E} (x_i - x_j)^2 w_{ij}$,
where w_{ij} is weight (length, i.e., communication cost) defined on edge $[i, j]$.

Local Mesh Generation: We use the advancing front algorithm to generate regular mesh.

1. Starting from the boundary as the paving front
2. Check if can close the region → if closed, finish
3. Generate the regular cells on the paving front
4. If cells intersected, merge them.
5. Update the paving front , go to 2.

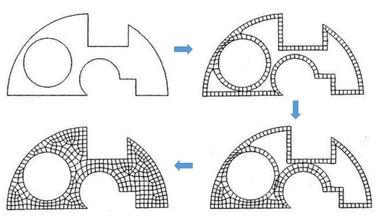


Fig.3 Advancing Front Algorithm.
[Image courtesy of Blacker]

EXPERIMENTAL RESULTS

We run this meshing algorithm using the LSU SuperMIC cluster, which consist of 1024 computing nodes where each node has two 2.6GHz 8-Core Xeon 64-bit Processors and 32GB memory.

Parallel Computation Efficiency

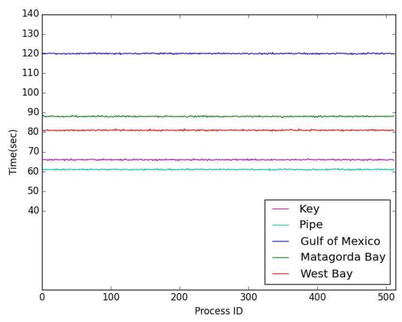


Fig.4(a) The load balance for 512 working processes of the meshing of our dataset

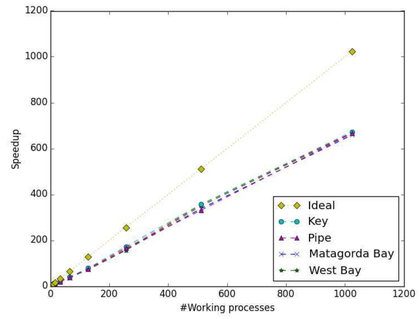
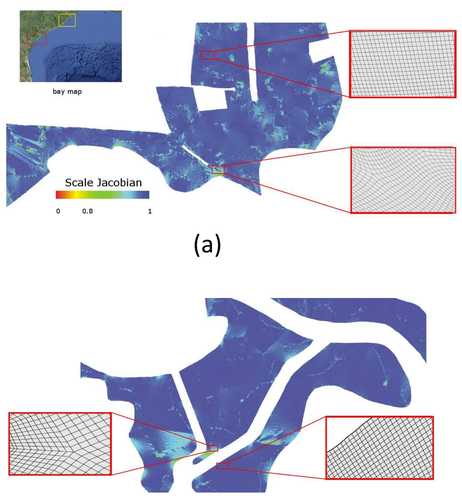


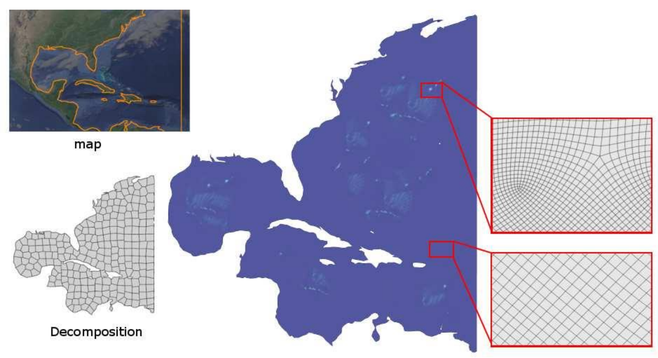
Fig.4(b) The parallel speedup of meshing: The speed up of our algorithm on different models ranges from 2.78 to 601.5 when using 4 to 1024 working processes.

Meshing Results



(a)

(b)



(c)

Fig.5 The quad meshes for West Bay (yellow region in bay map), Matagorda Bay (red region in bay map), and the Gulf of Mexico respectively. The quad meshes are color-coded in scaled Jacobian.

Compared with Directly using METIS to partition the mesh

Mesh Quality Comparison between METIS[2] and our algorithm. The three values are average, standard deviation and minimum of the scaled Jacobian of quad elements. Larger scaled Jacobian values indicates higher quality of the cell. Statistics in this table show that our partitioning algorithm leads to a better (about 50% improvement on the minimal scaled Jacobian) meshing result than that using METIS.

Model (N_S)	Sequential Algorithm	METIS	Our Method
Key (32)	0.97 / 0.13 / 0.39 / 35	0.93 / 0.27 / 0.23/ 411	0.98 / 0.13 / 0.38 / 216
Pipe (16)	0.97/ 0.04 / 0.31 / 34	0.94 / 0.09 / 0.23 / 128	0.97 / 0.04 / 0.30 / 50
Matagorda Bay (512)	-	0.93 / 0.16 / 0.21 / 413	0.97 / 0.04 / 0.35 /305
West Bay (512)	-	0.93 / 0.12 / 0.24 / 562	0.98 / 0.05/ 0.36 / 225
Gulf of Mexico (1024)	-	0.96 / 0.15 / 0.21 / 3491	0.98 / 0.04 / 0.36 / 3104

CONCLUSION

Results:
- Using the LSU SuperMIC cluster (1024 nodes), high-quality structured meshes with more than 5 billion elements can be generated within 30 seconds.

Ongoing work:
- Generalizing this parallel meshing pipeline onto curved 2D surfaces and 3D regions;
- Investigating parallel structured meshing algorithms with controlled singularity numbers and distributions.

ACKNOWLEDGEMENT

This research is partially supported by NSF IIS-1320959, IIS-1251095, CNS-1205682, and EPS-1010640, and Louisiana Board of Regents NSF-PFunds: LEQSF-EPS(2150)-PFUND-397.

REFERENCE

1. Lévy, Bruno, and Yang Liu. "Lp Centroidal Voronoi Tessellation and its applications." ACM Transactions on Graphics (TOG). Vol. 29. No. 4. ACM, 2010.
2. Karypis, George, and Vipin Kumar. "A fast and high quality multilevel scheme for partitioning irregular graphs." SIAM Journal on scientific Computing 20.1 (1998): 359-392.
3. Wuyi Yu and Xin Li "A Geometry-aware Data Partitioning Algorithm for Parallel Quad Mesh Generation on Large-scale 2D Regions", Proceedings of IMR25, 2015