Entire Domain Advancing Layer Surface Mesh (EDAM-S) Generation

by

Jasmeet Singh

B. Tech, Indian Institute of Technology (BHU), Varanasi, 2015

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Masters in Applied Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Mechanical Engineering)

The University of British Columbia
(Vancouver)

December 2019

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Entire Domain Advancing Layer Surface Mesh (EDAM-S) Generation

submitted by **Jasmeet Singh** in partial fulfillment of the requirements for the degree of **Masters in Applied Science** in **Mechanical Engineering**.

Examining Committee:

Carl Ollivier-Gooch, Mechanical Engineering
Supervisor

XYZ, Mechanincal Engineering
Supervisory Committee Member

PQR, LMN Department
Supervisory Committee Member

Table of Contents

List of Tables

List of Figures

Glossary

****

To be updated

****

Chapter 3

Methodology Part 2: Advancing Several Layers

In the previous chapter, we talked about how the very first layer of the mesh is generated. Starting from the discussion of extruding one boundary vertex onto the sub-surface, we went on explained how all the vertices on the boundary curve of the sub-surface are extruded and subsequently, how the advancing front is recovered so as to move on to the next layer. We also talked about how extrusion length scaling at concave corners helps us avoid immediate front collapse and improves overall mesh quality. In this chapter, we are going to discuss some of the important subroutines which help complete the anisotropic surface mesh.

We will start by discussing the subroutine used to control the aspect ratio as we advance several layers in the mesh. Combining triangular mesh elements to quad elements would be discussed next. Subsequently, mesh smoothing and collision handling would be discussed.

3.0.1 Edge Collapse and Sub-surface Interior Improvement

As the advancing front moves into the surface interior, the layers grow in size. This is done for the purpose of giving a higher refinement at the boundary curves. As the size of the layer grows, the aspect ratio of the mesh elements generated decrease. Also, some of the vertices on the front might come so close to each other that the aspect ratio approaches unity. Growing the layers further with all the vertices on the front would lead to anisotropy in the orthogonal direction and/or front overlap. Hence, decimation of some of the front vertices is necessary to proceed with the next few layers.

Once we have recovered the advancing front by iterative edge swaps to connect the kid points in the mesh, we check for vertices on the front that are too close to each other relative to the extrusion length. For instance, vertices which are near a concave corner could encroach each other if they are not decimated. Another example would be vertices on the advancing front which are far away from surface

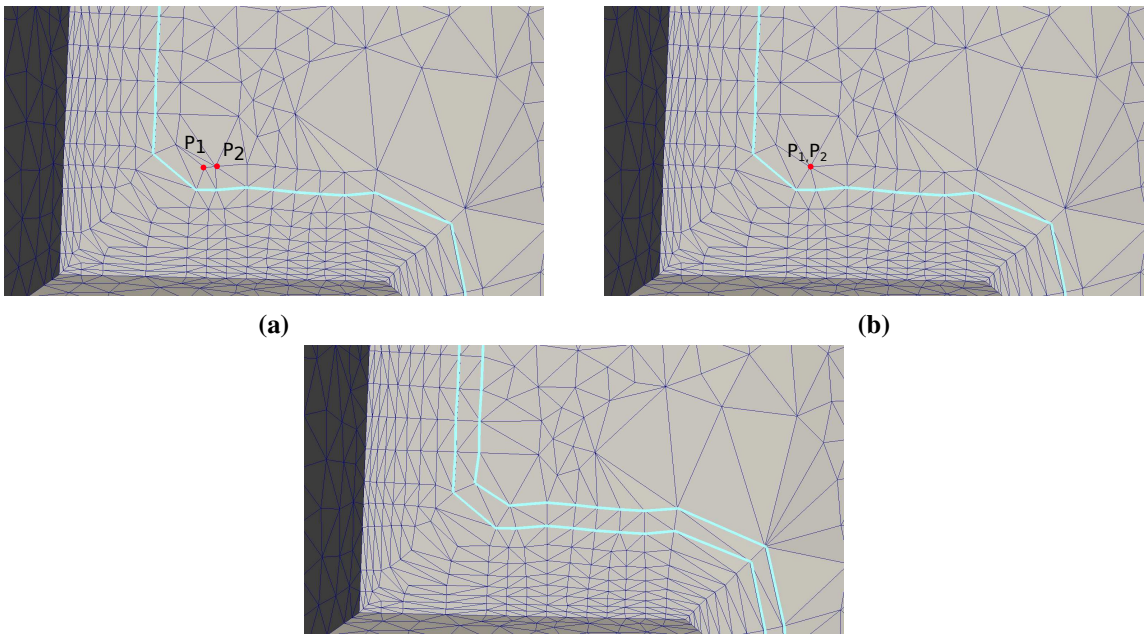

Figure 3.1: Edge collapse on an advancing front to avoid encroachment of points. In (a), two points in the kid layer P_1 and P_2 are sufficiently close to each other. Their parent layer is highlighted. If both the points advance to the next layer, then the next front would fail to recover. Hence, the edge between them is chosen to collapse. (b) shows the result of the edge collapse. The new location of both the points is the average position of their initial location. (c) shows how the next front looks like.

boundaries because the extrusion length has grown by this point. Decimating vertices from the front which are at a substantial distance from surface boundaries helps prevent the cell aspect ratio, ie front edge length over extrusion length, from dropping below one.

To check for vertices to decimate, we iterate through the vertices in the front and identify the ones which are too close to their neighbours. Vertex decimation is done through the edge-collapse routine as described in [12]. The threshold edge length between two points on the front is set to be $2\tan(\pi/8)$ times the average extrusion length at those points. This value is set so as to minimize the normalized maximum deviation of angle from 90° for quad elements. Hence, the threshold ensures that the anisotropic properties are retained for several layers into the surface. All short edges on the front are collapsed using the edge-collapse algorithm. An example of an edge-collapse on the front is shown in Figure 3.1. Here, two points P_1 and P_2 are collapsed into a single point which forms a part of the next front on the surface. Validation checks are made before collapsing an edge. The resultant triangles should not have a deviations of more than 50° from the underlying surface.

As the advancing front marches into the surface, vertices in the interior of the surface immediately next to the front are decimated to make way for the advancing layers. Before extruding a point on the advancing front, we check if any point in the surface interior with which it shares an edge agrees with

the following condition. If it does, we decimate the interior vertex.

$$d < max\left(l_1/\sqrt{2}, l_2/\sqrt{2}, c \times extrudeLength\right) \tag{3.1}$$

Here d is the distance between the point on the advancing front and the interior point, l_1 and l_2 are the lengths of adjacent edges of the point on the front, c is a constant whose value is set as 2 and $extrudeLength$ is the extrusion length at the vertex on the front. This condition ensures decimation of vertices in the surface interior which are close to the advancing front and avoids any encroachment of surface interior vertices on the advancing layers. Topological and geometrical checks are done before an edge can be collapsed. These include a threshold for the ratio of area of generated triangles and a limit on the dihedral angle between the adjacent triangles created by the collapse. The area threshold is set to be 10^8. Also, edge-collapse is successful only if the triangles resulting from it are be within a limit of $\theta < 30^\circ$ from the surface(see Figure 2.5). The threshold for the maximum dihedral angle between any two adjacent triangles that result from the edge-collapse is set to 40°. The quality criterion used for edge collapse is again maximization of the minimum angle in the triangles thus produced. The best edge for collapse is chosen when decimating the interior vertices using this quality criterion. This is in contrast to the vertex decimation on the advancing front where the candidate edge for collapse is already identified. An illustrative example for surface interior vertex decimation can be seen in Figure 3.2.

After we have a recovered the front and decimated encroaching vertices in the surface interior as well as on the advancing front through edge collapse, we queue up the immediate interior edges of the surface and swap them for maximizing mesh quality. This step is included so that we have a good interior triangulation at each step of the advancing layer routine. This step ensures that we do not have bad triangles causing problems as we continue to march.

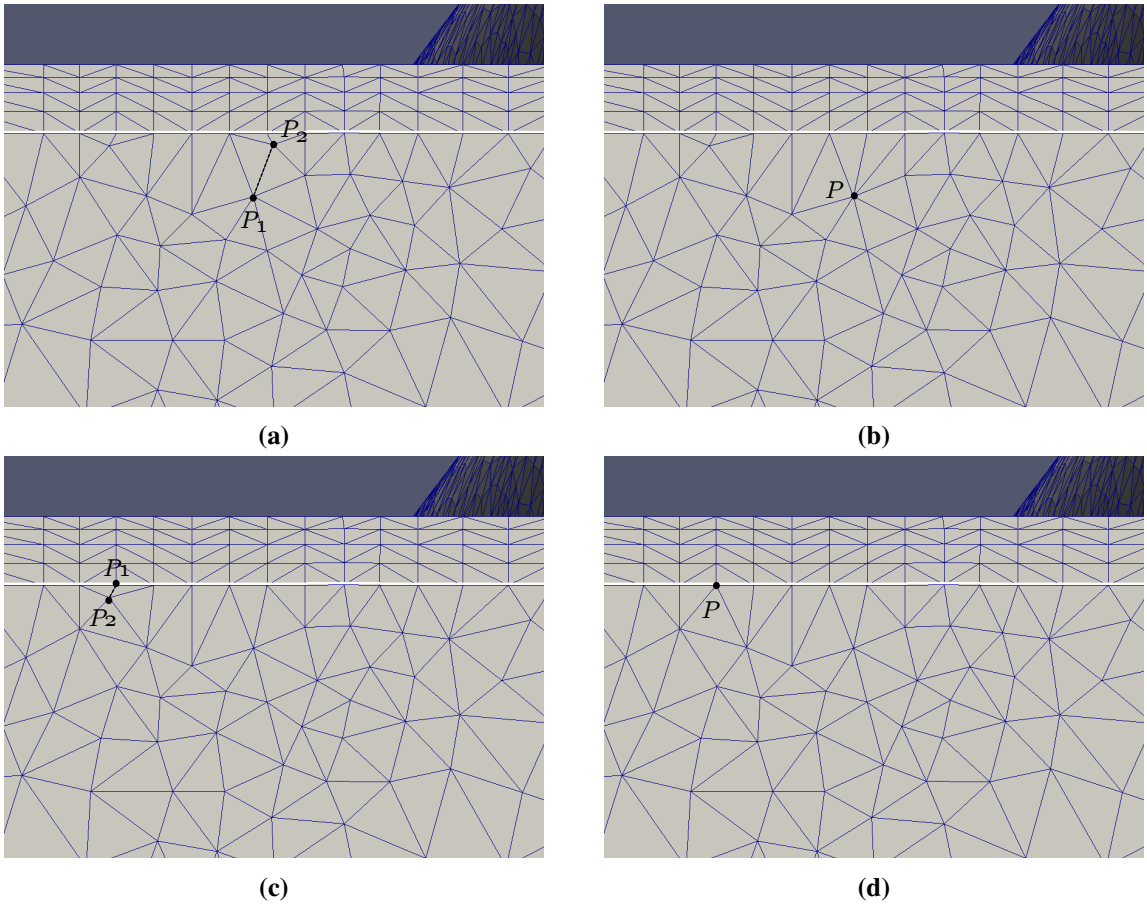

Figure 3.2: Interior vertex decimation through edge collapse. The highlighted line shows the advancing front. In (a), vertex P_2 is about to encroach the front. Hence, the best vertex for collapse is chosen among its neighbours. The best vertex for edge collapse here is P_1. Hence, P_2 is collapsed on to P_1. The connectivity after the edge collapse is shown in (b) where vertex P represents the collapsed vertex. Similarly, in (c), vertex P_2 is about to encroach the advancing front and is collapsed onto vertex P_1 which is on the advancing front itself. The new connectivity is shown in (d) where all the possibly encroaching vertices for the advancing layer are decimated.

Bibliography

[1] 2019 most common 3d file formats. https://all3dp.com/3d-file-format-3d-files-3d-printer-3d-cad-vrml-stl-obj/. Accessed: 2020-01-12. → pages x, 24

[2] Mesh generation software list. http://www.robertschneiders.de/meshgeneration/software.html/. Accessed: 2010-01-12. → page 25

[3] M. Aftosmis, D. Gaitonde, and T. S. Tavares. On the accuracy, stability, and monotonicity of various reconstruction algorithms for unstructured meshes. 1994. → pages 7, 17

[4] T. D. Blacker and M. B. Stephenson. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4):811–847, 1991. → pages ix, 7, 8

[5] M. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *International Journal for Numerical Methods in Fluids*, 25(4): 475–491, 1997. → pages ix, 11, 12

[6] H. Chen and J. Bishop. Delaunay triangulation for curved surfaces. *Meshing Roundtable*, pages 115–127, 1997. → page 16

[7] J.-C. Cuillière. An adaptive method for the automatic triangulation of 3d parametric surfaces. *Computer-aided design*, 30(2):139–149, 1998. → page 16

[8] E. F. D'Azevedo and R. B. Simpson. On optimal triangular meshes for minimizing the gradient error. *Numerische Mathematik*, 59(1):321–348, 1991. → page 4

[9] P.-J. Frey and F. Alauzet. Anisotropic mesh adaptation for cfd computations. *Computer methods in applied mechanics and engineering*, 194(48-49):5068–5082, 2005. → page 9

[10] R. V. Garimella and M. S. Shephard. Boundary layer mesh generation for viscous flow simulations. *International Journal for Numerical Methods in Engineering*, 49(1-2):193–218, 2000. → pages ix, 12, 14

[11] P.-L. George and H. Borouchaki. Delaunay triangulation and meshing. 1998. → page 16

[12] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. Technical report, Washington Univ Seattle Dept OF Computer Science And Engineering, 1994. → page 39

[13] Y. Ito and K. Nakahashi. Unstructured mesh generation for viscous flow computations. In *IMR*, pages 367–377, 2002. → pages ix, 13, 14

[14] Y. Ito, A. M. Shih, B. K. Soni, and K. Nakahashi. Multiple marching direction approach to generate high quality hybrid meshes. *AIAA journal*, 45(1):162–167, 2007. → pages ix, 15

[15] G. Kunert. Toward anisotropic mesh construction and error estimation in the finite element method. *Numerical Methods for Partial Differential Equations: An International Journal*, 18(5): 625–648, 2002. → page 11

[16] T. Lan and S. Lo. Finite element mesh generation over analytical curved surfaces. *Computers & Structures*, 59(2):301–309, 1996. → pages x, 16, 18

[17] Y. Lee and C. K. Lee. A new indirect anisotropic quadrilateral mesh generation scheme with enhanced local mesh smoothing procedures. *International journal for numerical methods in engineering*, 58(2):277–300, 2003. → page 12

[18] X. Li and W. Huang. An anisotropic mesh adaptation method for the finite element solution of heterogeneous anisotropic diffusion problems. *Journal of Computational Physics*, 229(21): 8072–8094, 2010. → pages ix, 11, 12

[19] R. Löhner. Matching semi-structured and unstructured grids for navier-stokes calculations. In *11th Computational Fluid Dynamics Conference*, page 3348, 1993. → page 12

[20] A. Loseille and R. Löhner. On 3d anisotropic local remeshing for surface, volume and boundary layers. In *Proceedings of the 18th International Meshing Roundtable*, pages 611–630. Springer, 2009. → page 15

[21] D. Mavriplis. Unstructured grid techniques. *Annual Review of Fluid Mechanics*, 29(1):473–514, 1997. → pages 7, 17

[22] D. J. Mavriplis. Adaptive mesh generation for viscous flows using triangulation. *Journal of computational Physics*, 90(2):271–291, 1990. → pages ix, 11

[23] K. Nakahashi. Fdm-fem zonal approach for viscous flow computations over multiple-bodies. In *25th AIAA Aerospace Sciences Meeting*, page 604, 1987. → pages 11, 12

[24] S. J. Owen. A survey of unstructured mesh generation technology. In *IMR*, pages 239–267, 1998. → page 16

[25] N. M. Patrikalakis and T. Maekawa. *Shape interrogation for computer aided design and manufacturing*. Springer Science & Business Media, 2009. → page 23

[26] S. Pirzadeh. Unstructured viscous grid generation by the advancing-layers method. *AIAA journal*, 32(8):1735–1737, 1994. → page 12

[27] O. Sahni, K. E. Jansen, M. S. Shephard, C. A. Taylor, and M. W. Beall. Adaptive boundary layer meshing for viscous flow simulations. *Engineering with Computers*, 24(3):267, 2008. → pages ix, 13

[28] K. Shimada, A. Yamada, T. Itoh, et al. Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles. In *6th International Meshing Roundtable*, pages 375–390, 1997. → page 11

[29] J. R. Tristano, S. J. Owen, and S. A. Canann. Advancing front surface mesh generation in parametric space using a riemannian surface definition. In *IMR*, pages 429–445, 1998. → page 16

[30] J. Tu, G.-H. Yeoh, and C. Liu. Chapter 6 - practical guidelines for cfd simulation and analysis. In J. Tu, G.-H. Yeoh, and C. Liu, editors, *Computational Fluid Dynamics (Second Edition)*, pages 219 – 273. Butterworth-Heinemann, second edition edition, 2013. ISBN 978-0-08-098243-4. doi:https://doi.org/10.1016/B978-0-08-098243-4.00006-8. URL http://www.sciencedirect.com/science/article/pii/B9780080982434000068. → pages 4, 10

[31] N. Viswanath, K. Shimada, and T. Itoh. Quadrilateral meshing with anisotropy and directionality control via close packing of rectangular cells. *world wide web*, 10:12, 2000. → pages ix, 12

[32] S. Yamakawa and K. Shimada. High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing. In *IMR*, pages 263–274. Citeseer, 2000. → pages ix, 13, 14

[33] J. Zhu, O. Zienkiewicz, E. Hinton, and J. Wu. A new approach to the development of automatic quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32 (4):849–866, 1991. → page 7