An Alignment Algorithm for Anisotropic Meshes with Non-Uniform Flow Fields

Alper Üngör

Department of Computer Science,
University of Illinois at Urbana Champaign,
Urbana, IL, 61801
e-mail : ungor@cs.uiuc.edu

Abstract

Anisotropic meshes that are adapatvie both in size and shape are critical for numerical accuracy of various problems such as convection dominated problems. A good unstructured anisotropic meshing algorithm uses elements of properly chosen size, shape and direction that adapt to the geometry and flow field of the physical problem. Efficient generation of a high quality anisotropic mesh is a non-trivial problem. This paper presents an algorithm to align a mesh with respect to the desired anisotropy.

Introduction

Finite element methods (FEM) are popular due their sucess in solving numerically challenging engineering problems. As its name implies, one essential step of the FEM involves the discretization of the problem domain into smaller elements. This discretization is called the *mesh* of the domain. *Unstructred meshes*, topology of which is not regular, proved themselves to capture the geometry of the complex domains better than the structured meshes. A wide range of engineering problems, such as compressible inviscid and viscous flows, where the underlying function is directional, it is desirable to use a mesh that is adaptive both in size and shape. *Unstructured anisotropic meshes* that have varying local topology, spacing and directionality reduce the problem size and result accurate solutions for problems with complex geometry and with rapidly changing solutions. A good unstructured anisotropic meshing algorithm uses elements of properly chosen size, shape and direction that adapt to the geometry and flow field of the physical problem. Efficient generation of a high quality anisotropic mesh is a non-trivial problem.

If the flow field for the physical problem is known *a priori*, one can use special point placement methods [3, 13, 11], to generate a set of mesh points that are especially suitable to construct an anisotropic mesh. These methods constructs a packing of ellipses in the domain, where the anisotropy at each point is defined by the size and alignment of the ellipse at that point. The centers of the ellipses are used as the mesh points and special rules are introduced to determine the connectivity between the points. However, it is often the case that the flow field for the physical problem is not known *a priori*. Moreover, flow field, i.e. required anisotropy, changes as the physical problem is simulated. Under such circumstances, a mesh adaptation techniques should be coupled with the numerical solution procedure. After computing an approximate solution on an initial mesh constructed based on an estimated flow field, the adaptation methods are used to modify the mesh to capture the anisotropic properties of the problem better. Various mesh adaptation techniques such as refinement [12], smoothing [14] are given in the literature. Another one called *alignment* proposed by Iliescu considers using face swaps. However, this method assumes the given flow field is uniform. This paper gives a new mesh alignment algorithm to generate anisotropic meshes with non-uniform flow fields. This mesh alignment algorithm is simple and efficient; it does not reconstruct the mesh from scratch and it does not modify the point set of the original mesh. The algorithm can be coupled with other mesh generation and adaptation algorithms.

Another area that motivates the anisotropic meshing is the parametric surface meshing. Our algorithm can be used to improve the quality of the surface meshes.

Outline The following section reviews the anisotropic mesh generation and adaptation techniques and introduces the tools to represent anisotropy. Then the problem is formally defined. We provide the algorithm followed by the proof of its termination with an optimally aligned mesh. Finally, results and discussions concludes the paper.

Background

Anisitropic Meshing

Expreiments with anisotropic meshes proves that the simulation accuracy is improved problems like CFD. The last decade witnessed new attempts for meshing techniques that considers the anisotropy driven by the problem. Some of these are proven to terminate generating the desired output, whereas others stand as practical heuristics. Mostly, ideas used for isotropic meshing (e.g. sphere packing, Delunay triangulation, advancing front and smoothing) is extended. However, the extensions are not trivial. For instance, an anisotropic equivalent of the Delaunay triangulation that optimizes a global criteria (maximizes the minimum angle) through a local property (empty circumcircle property) is not known. None of the existing Delaunay based anisotropic meshing methods are as powerful as their isotrpic counterpart. This section gives a review of anisotropic mesh generation and adaptation methods.

Ellipse packing Many isotropic meshing algorithms [] considered sphere packing to guarantee high quality discretization. Similarly, various ellipse packing based techniques [2, 3, 11, 13] are proposed for anisotropic meshing.

Both Bossen [2] and Shimada *et al.* [11] used particle simulation to construct an ellipse packing. A metric tensor is used to quantify the anisotropy. A set of particles interacting each other via repulsive and attractive forces are simulated. A modified version of the van der Waals forces are used to define the repuslion and attraction.

To improve the construction complexity and quality of the ellipse packing Li em et al. [13] considered using an advancing front approach to generate an ellipse packing.

Particles are packed in order of the dimension, i.e., vertices, edges, faces, and volumes. A quick initial guess at the final particle configuration is obtained by using hierarchical spatial subdivision [**?**]. The particle size and direction are adjusted individually by the control spacing function. This makes precise control of the triangle size possible throughout the mesh. A population control mechanism is used during the relaxation to remove any superfluous particle that is largely overlapped by its neighbors, and to subdivide any lone particle missing some neighbors, so that a given domain is filled with an appropriate number of particles. They [**?**] define the force for two particles centered at $\boldsymbol{x}_i$, and $\boldsymbol{x}_j$ based on the current distance $l = ||\boldsymbol{x}_i - \boldsymbol{x}_j||$, the target stable distance l_0, and the corresponding linear spring constant k_0 at the target distance. The target distance l_0 is calculated as the sum of the two lengths l_{ij} and l_{ji}, measured along the line segment that connects the centers $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, from the center to the boundary of each particles.

Note that here the particle can be the spheres or cells, if the isotropic mesh is preferred, and it can be the ellipse or rectangular if the anisotropic mesh is preferred.

Mesh Adaptation The other approach to generate the anisotropic mesh which conforms to the control spacing is to refine or smooth the previous mesh according to some quality measures of the element of the previous mesh. There are various quality measures for the triangle element.

Buscaglia and Dari [4] focused on optimizing a quality criteria to solve the anisotropic mesh adaptation problem. Quality of each element is defined as the product of a shape factor and a size factor. Shape factor is defined as $20.78V_a/P_a^2$, where V_a is the area of the element a and P_a is the perimeter. Size factor is defined as a function of the desired mesh size and the diameter of the element. The measure is normalized so that the equilateral triangle have a quality value 1. They used a fixed background mesh to provide the distance metric. To optimize the quality of each element locally, node repositioning, reconnection, insertion and deletion operations.

Castro Diaz and Hecht *et al* [5] used control space to capture anisotropy and similar node operations. They also used edge based operations, such as edge splitting, edge suppression. These local mesh improvement processes are iterated several times.

A generalization of the Laplacian smoothing is suggested by Zhang and Trépanier [14]. They construct a function defining a non-linear spring system function that represents the total potential energy of the triangulation. Desired directionality is formulated within this function. A gradient method based on the steepest descent is used to minimize the function and hence create a directionally stretched triangulation.

Skalický and Roos [12] studied mesh refinement technique to generate anisotropic hybrid (quad and triangle) meshes.

Measuring Anisotropy

Conventional metric to define the distance in Euclidean geometry becomes obsolote for anisotropic meshing. Metric tensor, also called Riemannian metric, is used to quantify the element size and stretching [2, 11, 13]. A

Metric tensor, $M(x)$, at a point x is a symmetric 2×2 (3×3 in 3D) matrix and all of its eigenvalues are positive. For 2D,

$$M(x) = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} \begin{pmatrix} \lambda_2 & 0 \\ 0 & \lambda_1 \end{pmatrix} \begin{pmatrix} cos(\alpha) & sin(\alpha) \\ -sin(\alpha) & cos(\alpha) \end{pmatrix} \quad (1)$$

Inverse square roots of the eigenvalues corresponds the radii of the ellipse describing the anisotropy, i.e. $r_1 = \frac{1}{\sqrt{\lambda_1}}$ and $r_2 = \frac{1}{\sqrt{\lambda_2}}$ (see Figure 1).

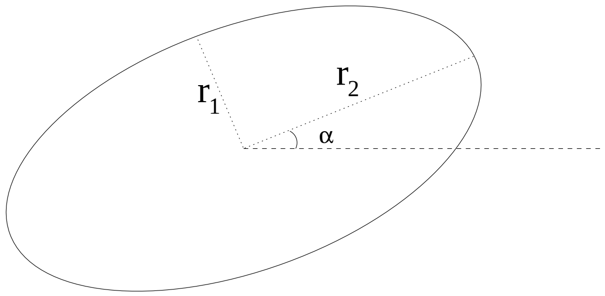

Figure 1: **Ellispe is used to represent anisotropy.**

For the purposes of the problem considered we use a simpler anisotropy representation. For each point x V(x) denotes a vector whose direction corresponds to direction of the anisotropy (angle α) and size correponds to the stretching factor (the ratio of the two radii r_2/r_1). A vector and the size of the anisotropy. Since our problem formulation considers neither the relocation of mesh points nor refinement/coarsening of the mesh, this measure serves our purposes.

Problem Formulation

Following is a formal definition of the anisotropic mesh alignment problem that is used to solve convection dominated problems.

Definition 1 (Anisotropic Mesh Alignment) *Given a mesh $\mathcal{M}$ on a domain $\mathcal{D}$ and a flow field $\mathcal{F}$ defined over $\mathcal{D}$, construct an anisotropic mesh $\mathcal{M}'$ that has the same point set as $\mathcal{M}$ but better alignment with respect to $\mathcal{F}$.*

A solution for the special case of this problem where the flow field is assumed to be uniform is proposed by Iliescu [9]. His algorithm extends to 3D as well [10]. The assumption is again that there is a single vector that defines the direction of the flow throughout the domain. However, when the flow field is not uniform, his algorithm is not guaranteed to terminate at finite number of steps. Furthermore, the method only considers the direction of the flow and ignores the size of the flow vector. However, most simulations define anisotropy as a metric tensor, which can be seen as a combination of direction and size. Hence we would like to consider constructing meshes that could align the mesh with respect to the desired non-uniform anisotropy, i.e., the the size and the direction of the flow field vector can vary throughout the domain.

In this study we provide an algorithm to solve the non-uniform flow alignment problem. With this algorithm, numerical solutions of the simulation problems can be improved without changing the mesh points. The algorithm considers all possible pairs of neighbor triangles and applies edge flip (face swap) operations whenever the alignment is to be improved. We prove that the algorithm terminates after computing an optimal triangulation. Details of the 2D algorithm, its extension to 3D and the proof will be be discussed in the full paper. We also plan to investigate possible improvements with other geometric operations, refinement and smoothing of the mesh with respect to the flow field.

Face Swap: A Topological Operation

Face swap operation is used to change the topolgy of a triangulation. It is the main element of many triangulation algorithms [1]. This simple operation is indeed a very powerful tool to compute various type of optimum triangulations. For a given point set in 2D it is possible to compute the triangulation that minimizes the maximum angle (Delaunay Triangulation) through a series of face swaps starting with an arbitrary triangulation of the points. It is also proven that any triangulation can be computed by face swapping. Swap is a reversible operation.

Face swap operation is also used to adapt isotropic meshes together with some smoothing operations [6]. Their results indicate that quality of the mesh is drastically improved by the coupling of swapping and smoothing.

Algorithm

In the following algorithm, $\mathcal{F}(x)$ denotes the flow field vector at point x, $m(AB)$ denotes the midpoint of the line segment AB, $size(v)$ is the size of the vector v and $anisotropySize(e)$ describes the anisotropy size of a mesh

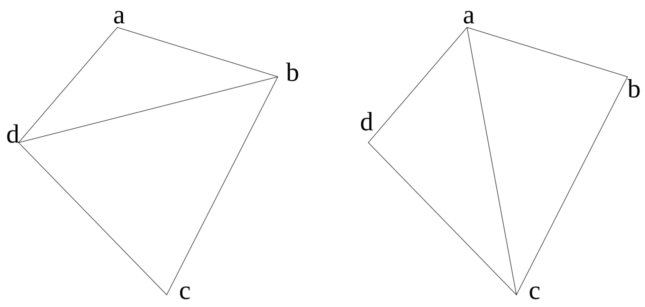

Figure 2: **Face Swap Operation**

element e. A quality of anisotropy is defined for each edge as the angle between the edge and the flow vector defined on the mid-point of it.

ALGORITHM Mesh Alignment
repeat
  for each pair of neighbor triangles ABC and BCD
    in triangulation $\mathcal{M}$
  if $ABCD$ is convex
    and $angle(\mathcal{F}(m(AD)), AD) < angle(\mathcal{F}(m(BC)), BC)$
    and $size(\mathcal{F}(m(AD)) > anisotropySize(ABD)$
    and $size(\mathcal{F}(m(AD)) > anisotropySize(ACD)$
  flip the edge (BC to AD)
  until there are no such pairs to be flipped
    in the triangulation

Figure 3: **Alignment Algorithm**

Algorithm Verification

Theorem 1 *Algorithm Mesh Alignment terminates after computing a local optimum solution for mesh alignment problem.*

Proof There are finite number of nodes in the initial configuration. Since the algorithm does not relocate the nodes there are finite number of triangulations. The termination proof is similar to that of Delaunay triangulation.

Let Q be a vector of quality values for all edges of the current triangulation. An edge swap operation never deteriotes a quality value in this vector. Hence this vector is strictly decreasing. $\square$

Results

The algorithm is implemented. Figure 4 illustrates the results.

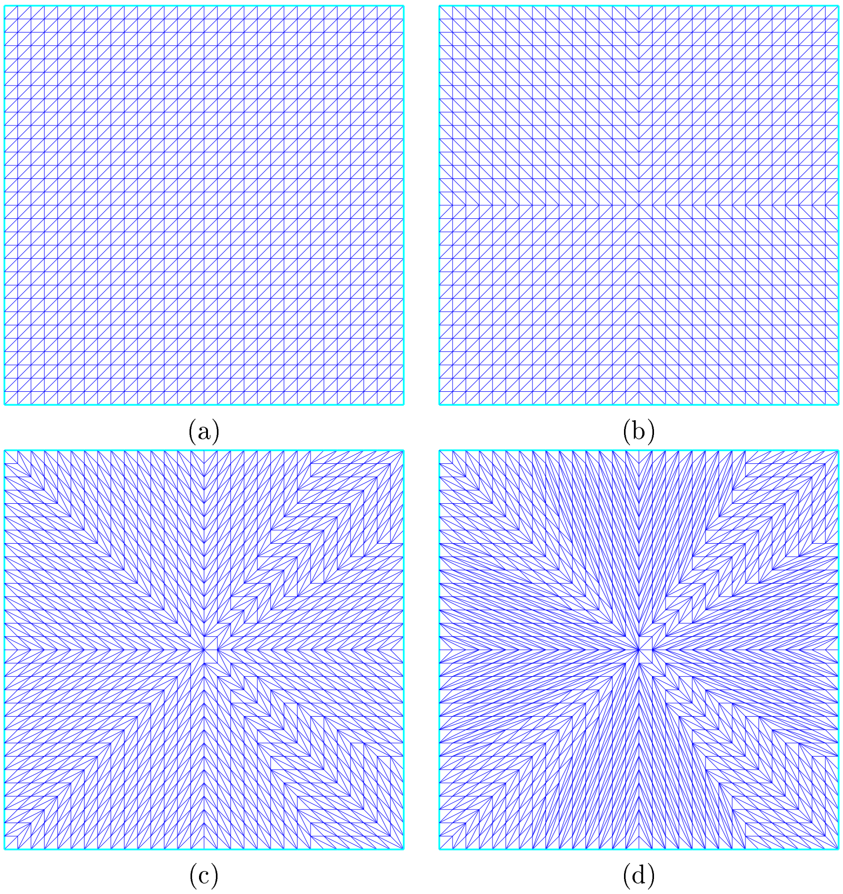

Figure 4: **(a) Isotropic mesh (b) Anisotrpic mesh with little stretching (c) Anisotropic mesh with more stretching.**

Conclusion

In this study we provide an algorithm to solve the non-uniform flow alignment problem. With this algorithm, numerical solutions of the simulation problems can be improved without changing the mesh points. The algorithm considers all possible pairs of neighbor triangles and applies edge flip (face swap) operations whenever the alignment is to be improved. We prove that the algorithm terminates after computing an optimal triangulation. Details of the 2D algorithm, its extension to 3D and the proof will be be discussed in the full paper. We also plan to investigate possible improvements with other geometric operations, refinement and smoothing of the mesh with respect to the flow field.

Acknowledgments

The work reported here was supported, in part, by the Center for Process Simulation and Design (NSF DMS 98-73945).

References

[1] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf. Computational Geometry: Algorithms and Applications, Springer-Verlag, Berlin, 1997.

[2] F. J. Bossen. "Anisotropic Mesh Generation with particles," Technical Report CMU-CS-96-134, Carnegie Mellon University, Pittsburg, PA, May 1996.

[3] F. J. Bossen, P. S. Heckbert. "A Pliant Method for Anisotropic Mesh Generation," Proceedings of 6^{th} International Meshing Roundtable, pp. 63-74, October 1997.

[4] G. C. Buscaglia and E. Dari "Anisotropic Mesh Optimization and its Applications in Adaptivity," International Journal of Numerical Methods in Engineering, **40**, pp. 4119-4136, 1997.

[5] M. J. Castro Diaz, F. Hecht "Anisotropic Mesh Generation," Technical Report 2672, INRIA, October 1995.

[6] L. Freitag and C. Ollivier-Gooch. "Tetrahedral Mesh Improvement Using Swapping and Smoothing," International Journal of Numerical Methods in Engineering, **40**, pp. 3979-4002, 1997.

[7] P.-L. George and F. Hecht Nonisotropic Grids, Handbook of Grid Generation, ed. J. F. Thompson, B. K. Soni, N. P. Weatherill, CRC Press, 1998.

[8] P.-L. George and H. Borouchaki Delaunay Triangulation and Meshing, Editions Hermes, Paris, 1998.

[9] T. Iliescu. "A Flow-Aligning Algorithm for Convection-Dominated Problems," International Journal of Numerical Methods in Engineering, **46**, pp. 993-1000, 1999.

[10] T. Iliescu. "A 3D Flow-Aligning Algorithm for Convection-Diffusion Problems," Applied Mathematics Letters, **12**, pp. 67-70, 1999.

[11] K. Shimada, A. Yamada and T .Itoh. "Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles," Proceedings of 6^{th} International Meshing Roundtable, pp. 375-390, October 1997.

[12] T. Skalický and H.-G. Roos "Anisotropic Mesh Refinement for Problems with Internal and Boundary Layers," International Journal of Numerical Methods in Engineering, **46**, pp. 1933-1953, 1999.

[13] X.-Y. Li, S.-H. Teng, and A. Üngör. "Biting Ellipses to Generate Anisotropic Meshes," in Proceedings of 8^{th} Int. Meshing Roundtable, Lake Tahoe, CA, pp. 97-108, October 1999.

[14] H. Zhang and J.-Y. Trépanier. "An Algorithm for the Optimization of Directionally Stretched Triangulations," International Journal of Numerical Methods in Engineering, **37**, pp. 1481-1497, 1994.