MESQUITE Design: Issues in the Development of a Mesh Quality Improvement Toolkit

Lori Freitag and Thomas Leurent
Argonne National Laboratory
Argonne, IL 60490 USA
freitag@mcs.anl.gov

Patrick Knupp and Darryl Melander
The Sandia National Laboratories
Albuquerque, NM 87185 USA
pknupp@sandia.gov

Abstract

Poor mesh quality is known to adversely affect both solution efficiency and accuracy. There has been considerable research on a wide variety of mesh improvement algorithms, but the impact of these algorithms on applications has been limited because they are typically embedded in particular meshing software packages. To rectify this situation, we are developing a stand-alone mesh quality improvement toolkit called MESQUITE. In this paper, we describe the motivation, goals, and design of MESQUITE and give some computational results using the underlying algorithms that show the benefit of such a package.

Introduction

Mesh quality is potentially impacted in many stages of the mesh generation process from defeaturing CAD models to postprocessing via a smoothing scheme to mesh adaptivity. At any stage after a mesh is created, its quality can be improved by techniques such as node point movement and topology modification. The simplest algorithm is Laplacian smoothing [4], which moves each vertex to the geometric center of its neighboring vertices. Although this method is often effective, it does not guarantee mesh improvement and can create inverted elements. Thus, in many situations, more sophisticated smoothing algorithms are needed. In structured meshing, for example, the Winslow smoother is widely used because it guarantees invertibility in two dimensions. Using a local approach to Winslow smoothing is equivalent to using SOR to solve a quasi-linear system of equations. For small problems, this is sufficient, but for large problems, one must turn to iterative solvers involving multigrid, Krylov subspace, or other techniques.

In unstructured meshing, effective smoothers with invertibility guarantees often require the use of numerical optimization techniques, and topology modification routines are also highly specialized.

Applications scientists doing computational field simulations generally do not have the time or expertise to write their own sophisticated mesh quality improvement algorithms; *they rightly want to concentrate on the results of their field simulations.* This situation creates the need for stand-alone, comprehensive, and widely available mesh quality improvement software similar in spirit to the "solver" packages that encapsulate solver expertise and remove this burden from the analyst. In this paper, we present our plan for creating such a toolkit, called MESQUITE (Mesh Quality Improvement Toolkit), under the DOE SciDAC Terascale Simulation Tools and Technology (TSTT) project [14].

Mesh Improvement Algorithms

In this section we briefly summarize some of the mesh quality improvement algorithms that have been developed. A detailed and complete survey is prevented by lack of space.

Structured Meshes

Nearly all structured mesh smoothing is performed by using some variation of the Winslow elliptic smoother [15]. Many of the variations center on finding appropriate choices for weighting functions or source terms to control the mesh. All of these methods involve solving a quasi-linear system of second-order elliptic partial differential equations. Alternative methods for smoothing structured grids are often based on variational methods in which some measure of mesh quality is optimized. Some researchers have preferred to improve their meshes via direct optimization of a discrete function of grid edge-lengths, areas, and angles [2]. These approaches are similar in spirit to the mesh quality optimization techniques used for unstructured mesh generation except that they optimize the entire mesh at once instead of performing a series of local optimizations. For additional information see [11].

Unstructured Meshes

Unstructured mesh quality can be measured in different ways ranging from à priori geometric and algebraic metrics [12] to solution-based measures that can often only be determined à posteriori [1]. The mesh can be improved according to these metrics by using one of two broad classes of algorithms: local reconnection or face swapping to change mesh topology for a given set of vertices (see, e.g., [9]), or mesh smoothing to relocate grid points to improve mesh quality without changing mesh topology (see, e.g., [13, 10, 8]). These techniques can be used individually but are most effective when used together [7].

In the area of node point movement, Laplacian smoothing and its variants that use weighting schemes or constraints to improve its effectiveness are commonly used. In addition, there has been recent work to develop optimization-based node movement techniques that can be inexpensive, can guarantee mesh validity, and are effective for a wide variety of mesh quality measures. The objective functions for these techniques can be patch-based or local (smoothing one or a few vertices at a time) or non-patch based (smoothing all the vertices at once). Examples of these techniques are numerous and range from simple line search procedures (e.g., [13]) to methods that maximize the minimum quality metric of a local submesh based on steepest descent algorithms for nonsmooth functions [8] and methods that optimize the average quality of an element patch using nonlinear conjugate gradient methods [10]. Effective methods that combine optimization-based smoothing techniques with Laplacian smoothing and its variants or with each other have been shown to obtain high-quality elements at a low computational cost [13, 6].

The MESQUITE Vision

Many successful improvement algorithms have been developed in the context of particular data structures or software packages and are thus not widely available to mesh generation developers or application scientists. Currently, no stand-alone, portable package for mesh improvement exists that encompasses many of the latest, most sophisticated algorithms for robust performance, that can be used for any element type and order, that can be referenced to both isotropic and anisotropic ideal elements, that provides access to both smoothing and flipping algorithms, and that provides both patch-based and nonpatch-based objective functions. The package that comes closest to these goals is Opt-MS [5]. It is stand-alone and portable and uses some optimization algorithms for homogeneous simplicial meshes. However, it provides smoothing algorithms only for isotropic elements on local patches and no topological changes are supported.

The MESQUITE software package will address all of the issues mentioned above and will provide a freely available mesh quality improvement toolkit that meets the following design goals.

Versatile and Comprehensive. MESQUITE will work on structured, unstructured, and hybrid meshes in both two and three dimensions. The design will permit improvements to meshes composed of tetrahedral, hexahedral, prismatic, pyramidal, and even polyhedral elements. MESQUITE will incorporate the best quality improvement methods from both continuum variational methods and from discrete objective functions composed of mesh quality metrics. It will incorporate methods for node movement, topology modification, and hybrid improvement. Node movement strategies will include both local patch based iteration schemes and nonpatch based global objective functions. MESQUITE will be applicable to both adaptive and nonadaptive meshing and to both low- and high-order discretization schemes. MESQUITE will, however, strive to include only the minimal number of mature algorithms needed to cover as many contexts as possible. In addition, it will incorporate the research algorithms enabled by its development.

Effective. MESQUITE will use state-of-the-art algorithms and metrics to guarantee improvement in mesh quality. Because the definition of mesh quality is application specific, we will provide referenced quality metrics that allow the user to untangle meshes; improve mesh smoothness, element size, shape, and orientation; and control element anisotropy. The software will be customizable, enabling users to insert their own quality metric and/or objective functions, and mechanisms for easily creating an combined approach that uses one or more improvement algorithms.

Interoperable. To ensure that MESQUITE will be interoperable with a large number of mesh generation packages, we will use the common interfaces for mesh query currently under development by the TSTT center. These interfaces will provide uniform access to mesh geometry and topology and will be implemented by all TSTT center software. We are working with the interface design team to ensure efficient information access through strategies such as information caching and agglomeration. We will also help create the interfaces necessary to accommodate topological changes generated by mesh swapping and flipping algorithms and to work with meshes constrained to a geometrical model.

Robust. MESQUITE will guarantee that there will be no degradation to mesh quality as measured by the objective function and metric used for improvement. We will use sound software engineering principles and robust numerical algorithms. A comprehensive suite of test problems will be developed to verify the correct execution of the code. Extensive error checking will be employed, and mechanisms for reporting failure problems will be provided.

Efficient. We will make effective use of high-performance architectures and languages. In particular, we will design the outer layers of MESQUITE using object-oriented design in C++ but will implement the inner kernels using easily optimizable coding styles (arrays, inlined functions, etc.). To ensure efficient use of computationally intensive optimization algorithms, we will employ a strategic use of inexpensive smoothers, such as Laplacian smoothing, as "preconditioners" for the more expensive optimization techniques and loose stopping tolerances that attain sufficient improvement in mesh quality.

Automatic and User Friendly. We will provide a number of mechanisms that lower the "expertise" barrier for using MESQUITE. In the simplest usage scenario, a TSTT mesh pointer is passed to MESQUITE and improved using a default set of metrics and strategies determined by the mesh type. If desired, the user can guide the mesh improvement process by setting a few parameter values indicating preferred metrics and strategies from a list of provided functionalities. Once these parameters are defined, MESQUITE will proceed to optimize the mesh without further input from the user. Additional interfaces will be provided for those who desire advanced functionalities, user defined metrics or objective functions, or more control over the optimization process. Mesh quality assessment and statistics will be provided to help the user determine whether the mesh needs to be improved and to evaluate the effectiveness of MESQUITE.

Preliminary Design

The current architecture of MESQUITE is shown in Figure 1 and illustrates the basic design concepts. At the user interface level, the `MesquiteController` singleton is instantiated the first time a client calls an interface function such as `set_mesh`, which indicates the mesh to be improved, or `add_pass`, which permits combining various methods and quality metrics into hybrid quality improvement schemes. The `GlobalDataSet` and `MeshData` classes provide access to the mesh data through the TSTT interface; lists of elements or nodes to be improved are created from this data and may be culled or modified during optimization. A common quality assessment class, `QualityAssessor`, will permit evaluation of mesh quality before, during, and after optimization.

For the mesh improvement algorithms, we use the strategy pattern so that additional optimization or swapping techniques can be easily added to MESQUITE through the implementation of the `NodeMover` or `TopologyModifier` abstract classes, respectively. For example, in the `NodeMover` class there will be two basic approaches covered: direct optimization (numerical and variational) and indirect methods (partial differential equations derived from Euler-Lagrange equations or from setting the gradient of a discrete objec-

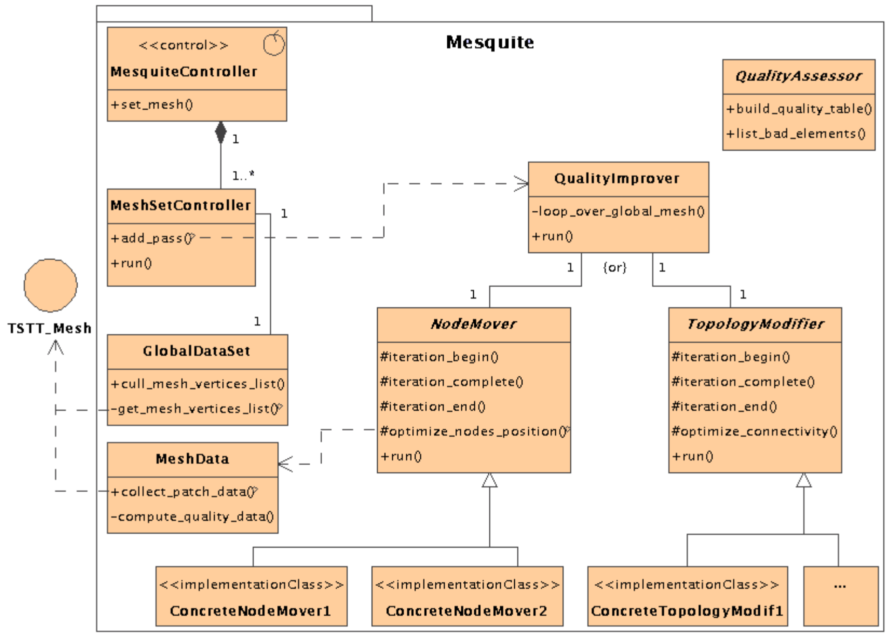

Figure 1. *MESQUITE UML Diagram.* The usual UML standards are applied. A class name in italic denotes an abstract class. A +,−, or # symbol precedes member functions to indicate public, private, or protected visibility. Dashed arrows indicate dependencies; plain lines indicate association; and plain lines with a diamond at the origin indicate composition. Plain arrowheads indicate a generalization (inheritance) relation.

tive function to zero). Both the direct and indirect approaches will be able to operate on local node or element patches, as well as on all of the nodes or elements at once. Thus, a particular objective function may be implemented in any of six ways, permitting an analysis of the relative merits of each approach. Every approach may select from common lists of stopping criteria, quality metrics, and objective function templates. A select number of specific optimization algorithms will be implemented in the prototype to test the design. In later years, additional algorithms will be implemented and existing ones improved.

MESQUITE Impact

The successful completion of MESQUITE will provide a robust and effective mesh improvement toolkit to the broader scientific community. This will allow both mesh generation researchers and application scientists to benefit from the latest developments in mesh quality control and improvement.

Benefits to Mesh Generation

MESQUITE will aid mesh generation researchers by providing a tool that improves the quality of the meshes generated by their algorithms. For example, some hexahedral mesh generation methods create topologically valid but poor quality elements; such methods will benefit from mesh quality improvement techniques that smooth the nodes enough to achieve acceptable geometric mesh quality. Furthermore, MESQUITE will provide a research platform for the development of new mesh improvement algorithms. By providing modular software with well-defined interfaces, MESQUITE will enable other researchers to easily experiment with new algorithms, new combinations of algorithms, and new quality metrics. These new algorithms can be directly compared (both in terms of relative quality improvement and in terms of efficiency) with the methods already in place. Currently such comparisons are tedious to perform and are therefore seldom done.

Benefits to Applications

Any PDE-based simulation community will potentially benefit from the development of MESQUITE. In general terms, mesh improvement techniques will reduce the total time to solution by reducing the time needed to create a suitable mesh (see the first example below), reducing the time to solution (see the first and second examples below), and improving the accuracy of the approximate solutions. To illustrate these benefits, we now give two particular examples from the areas of biomedical computing and compressible fluid dynamics simulations.

Biomedical Computing. Fischer, et. al. [3] have applied spectral techniques to the problem of simulating turbulent flow in an arteriovenous graft to predict wall shear stress for medical diagnosis. A series of relatively coarse unstructured hexahedral meshes (Figure 2) were created for the simulations. Some of the elements were found to possess poor shape quality and Laplacian smoothing failed to improve the mesh. The hexahedral mesh smoother described in [10] was a good candidate to try but was embedded within the CUBIT meshing code. By a slow process, the smoothing code was extracted and adapted to the problem. Using the smoothed and improved mesh, we obtained a 17 percent reduction in the number of flow-solver iterations, effectively trading four hours of application run-time for twenty minutes of mesh smoothing run-time. The project took six months, however, because the smoothing software was not stand alone.

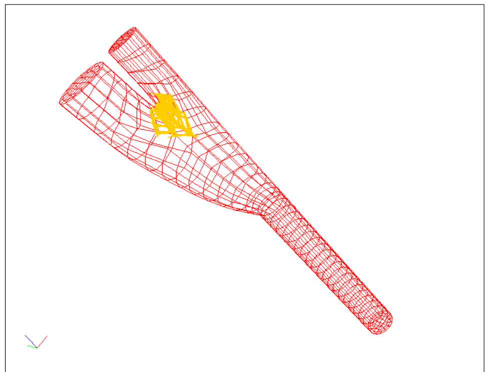

Figure 2. *Arteriovenous Graft Mesh*

Compressible Fluid Dynamics. Our second case study shows the effect of mesh quality on convergence behavior for weakly compressible flow over a cylinder (this work was reported earlier in [7]). A series of three meshes with increasing quality was used. A baseline, poor quality mesh (the left mesh in Figure 3) was generated by inserting random points into a mesh and swapping using the Delaunay criterion. The smallest angle in this mesh is 0.56°, the largest 178.86°. The middle mesh in Figure 3 was obtained by performing five passes of optimization-based smoothing on the vertices of the first mesh, which improves the extremal angles to 12.3° and 145.6°. The rightmost mesh in Figure 3 was obtained from the first mesh by performing five passes of smoothing alternating with passes of swapping. This mesh has extremal angles of 23.2° and 131.9° and shows the advantage of combining swapping and smoothing techniques. Flow around the cylinder was computed by using a second-order, edge-based, vertex-centered finite volume solver. For this example, the random mesh fails to converge. In contrast, the improved meshes both converge, with the smoothed and swapped mesh converging 25% faster. In each case, the mesh optimization procedures required less time than a single cycle of the multigrid solver.

Conclusions

MESQUITE will provide a stand-alone, interoperable software toolkit for the improvement of mesh quality in a wide variety of contexts. The toolkit will be versatile, comprehensive, effective, robust, automatic, efficient, and user friendly. The software requirements have been articulated in detail and the design phase is well underway. We expect to deliver a prototype version to the TSTT by fall 2002. The availability of this toolkit will benefit both mesh generation researchers and scientific simulation applications.

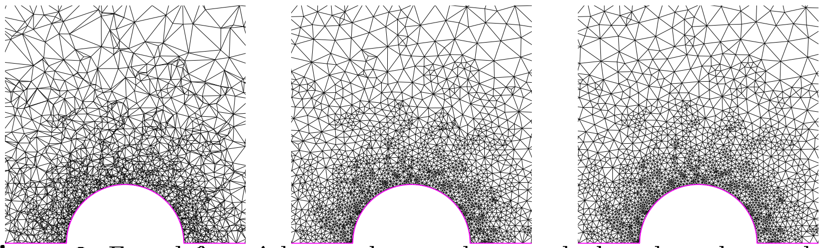

Figure 3. From left to right: random mesh, smoothed mesh, and smoothed and swapped mesh

Acknowledgments

The work of the first two authors was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38. The work of the second two authors was supported by the Department of Energy's Mathematics, Information and Computational Sciences Program (SC-31) and was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-ACO4-94AL85000.

References

[1] R. E. Bank and R. K. Smith. Mesh smoothing using a posteriori error estimates. *SIAM Journal on Numerical Analysis*, 34(3):979–997, June 1997.

[2] J.E. Castillo. A discrete variational grid generation method. *SIAM Journal of Scientific and Statistical Computing*, 12(2):454–468, 1991.

[3] P. F. Fischer et. al. Comparison of experimental and numerical simulations of flow within an arterio-venous graft. In *Proceedings of the International Mechanical Engineering Congress and Exposition*. ASME, 2000.

[4] David A. Field. Laplacian smoothing and Delaunay triangulations. *Communications and Applied Numerical Methods*, 4:709–712, 1988.

[5] Lori Freitag. Users manual for Opt-MS: Local methods for simplicial mesh smoothing and untangling. Technical Report ANL/MCS-TM-239, Argonne National Laboratory, Argonne, Ill., 1999.

[6] Lori Freitag and Patrick Knupp. Tetrahedral element shape optimization via the jacobian determinant and condition number. In *Proceedings of the Eighth International Meshing Roundtable*, pages 247–258. Sandia National Laboratories, 1999.

[7] Lori Freitag and Carl Ollivier-Gooch. A cost/benefit analysis of simplicial mesh improvement as measured by solution efficiency. *International Journal of Computational Geometry and Applications*, 10(4):361–382, 2000.

[8] Lori Freitag and Paul Plassmann. Local optimization-based simplicial mesh untangling and improvement. *International Journal of Numerical Methods in Engineering*, 49:109–125, 2000.

[9] Barry Joe. Construction of three-dimensional improved quality triangulations using local transformations. *SIAM Journal on Scientific Computing*, 16:1292–1307, 1995.

[10] Patrick Knupp. Hexahedral and tetrahedral mesh shape optimization. International Journal of Numerical Methods in Engineering, to appear.

[11] Patrick Knupp. *The Fundamentals of Grid Generation.* CRC Press, Boca Raton, FL, 1993.

[12] Patrick Knupp. Algebraic mesh quality metrics. *SIAM Journal of Scientific Computing*, 23(1):193–218, 2001.

[13] Mark Shephard and Marcel Georges. Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering*, 32:709–749, 1991.

[14] Terascale Simulation Tools and Technologies (TSTT) Center. see `http://www.tstt-scidac.org`.

[15] A. M. Winslow. Numerical solution of the quasilinear poisson equations in a nonuniform triangle mesh. *Journal of Computational Physics*, 2:149–172, 1967.