Generalized Meshing Environment for Computational Mechanics

Lawrence W. Spradley*
ResearchSouth, Inc., Huntsville, Alabama 35816
Rainald Löhner†
George Mason University, Fairfax, Virginia 22101
and
T. J. Chung‡
University of Alabama in Huntsville, Huntsville, Alabama 35816

Introduction

ONE of the most time-consuming tasks in performing any discrete analysis is the construction of a suitable mesh to represent the computational domain. Mesh generation today remains one of the pacing items in computational mechanics and consumes too many work hours and too many computer resources. This critical need for more progress in three-dimensional mesh generation certainly requires efficient geometry definition systems and advances in meshing methods. However, another major factor toward this progress is efficient methodology and computer software for automation of the meshing process. It is precisely this aspect of meshing that is the focus of this Note.

We will not discuss details of mesh-generation techniques and codes but will refer to recent reviews[1,2] and will present only a brief synopsis of the methods used in our computer software. Algebraic methods for structured mesh generation is now on firm ground and is generally based on transfinite blending function interpolation as typified by the early work of Spradley and Anderson.[3] Elliptic structured-mesh generation is also now commonly used in industry and is typified by works of Thompson.[4] A recent survey of unstructured methods is given by Thompson and Weatherill[1] and by Löhner and Parikh.[2] Unstructured-mesh generation methods and codes are now available for both tetrahedral- and hexahedral-type elements. These methods are primarily based on three common techniques: Delaunay triangulation, advancing front methods, and octree-based techniques. Work has recently been done on combining the cited methods.[5,6] Some limited techniques are also available for the generation of quadrilateral or hexahedral meshes[7]; however, tetrahedral-generation methods are more widely used and, hence, have received more exposure. Mesh quality measures are especially important in unstructured techniques, and some type of smoothing or mesh improvement is usually required. Laplacian methods and a variational method[8] for the optimization of unstructured meshes are used in our meshing technology and software.

A graphical user interface has been developed by Spradley et al.[9] for operation on Unix workstations and Windows personal computers that pulls together a wide variety of tools and incorporates them into a generalized meshing environment for computational mechanics. The software, termed OptiMesh, communicates with the user through input geometry files from a variety of CAD packages and/or inputs existing meshes generated by a more than a dozen industry codes. An algebraic method, the general interpolants method, is used to obtain multiblock structured hexahedral meshes from the geometry files. Likewise, the advancing front method is used to generate unstructured tetrahedral meshes from the same CAD files. Existing meshes, generated previously and provided as input, can be smoothed and otherwise improved and then output onto data files. The structured or unstructured meshes can be smoothed using either the Laplacian or variational methods. The generated meshes or the existing meshes can be locally remeshed, h refined, or the element types changed, all under the interactive control of the graphical interface. A package of graphics tools is included in the software for viewing of the meshes during the interactive session.

Unstructured-Mesh Generation

The unstructured-mesh generation used is based on the advancing front method as described by Löhner and Parikh.[2] Background meshes represent the most primitive way of specifying, in a general way, the desired element size and shape in space. The background mesh is a mesh of tetrahedra that completely covers the domain to be meshed. At the nodes of this background mesh, the user defines the desired element size, element stretchings, and stretching directions. Although cumbersome for general three-dimensional problems with large local variations of element size and stretchings, the background mesh is very useful for adaptive remeshing applications and is, therefore, always kept in the mesh-generation module.

A flexible way of specifying mesh parameters and resolution, which combines the smoothness of functions with the generality of boxes or other discrete elements, is to define sources. The element size for an arbitrary location in space is given as a function of the closest distance to the source. Sources offer a convenient and general way to define the desired element size in space. They can be introduced rapidly in the software in an interactive mode with a mouse-driven menu. Experimentation with different source strengths can produce quite good graded tetrahedral meshes.

Mesh Improvement Methods

Practical implementations of either the advancing front method or Delaunay mesh generators indicate that, in certain regions of the mesh, abrupt variations in element shape or size may be present. These variations appear even when trying to generate perfectly uniform meshes. The usual way to circumvent this problem is to improve the uniformity of the mesh by smoothing. One of the smoothers used in this work is the so-called Laplacian smoother, referred to as the spring-analogy smoother. The sides of the element are assumed to represent springs, which are then relaxed in time using explicit time stepping until an equilibrium of spring forces has been established. Each side of the element can be visualized to represent a spring. The spring constant is set in the software based on tests of the method; however, the user can adjust it as desired. At the boundary of the subdomain, the boundary points are allowed to move along the boundary curve but cannot leave the boundary itself.

The variational mesh optimizer that is employed follows the methodology by Cabello et al.[8] The key point of the method is the proper definition of the reference element. Because of their inherent regularity and their nondirectional preference, equilateral triangles (in two dimensions) and regular tetrahedra (in three dimensions) are taken as reference elements. To construct a proper measure of the deformation between a reference or ideal element and the corresponding physical element, it is required that the measure of the deformation should not be affected by preserving shape transformations, such as translation or rotation. The measure of the deformation is represented by a functional, is restricted to its dependence up to the first-order derivative of the transformation, and is to be dependent only on its gradient. The functional is then required to be invariant to rigid motions of both the reference and physical element. The optimal location of the physical nodes is computed by an iterative conjugate gradient algorithm. For boundary points, two types of boundary conditions may be prescribed: Dirichlet boundary condition (fixed point) or tangential boundary condition (free to move along the boundary curve).

Graphical User Interface

A complete documentation and user's guide for the meshing software is given in Ref. 9. Some routines are written in the C language to drive the code and to call the Fortran routines to perform the technical jobs. The software also uses the OpenGL graphics rendering system for portability among computer platforms. A graphical user

Received Sept. 18, 1997; revision received May 18, 1998; accepted for publication May 21, 1998.

*Technical Director.
†Professor, Institute for Computational Sciences and Informatics, and Consultant to ResearchSouth. Senior Member AIAA.
‡Distinguished Professor, Mechanical and Aerospace Engineering.

interface (GUI) has been developed for controlling the operation of the software. The Motif widget sets are used for development of the interface for Unix platforms, and the Win32 system is used for Pentium personal computer platforms running Windows NT. The GUI is termed OptiMesh and has a complete memory management system in operation. The interface is executed by the user, and the software dynamically allocates the storage for any size mesh.

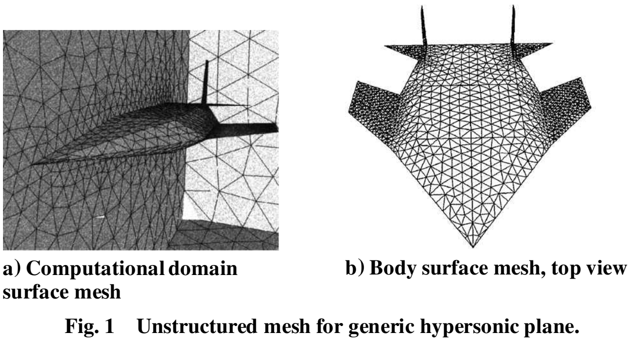
A CAD menu is set up to construct geometric definitions or to interface with commercial CAD packages (not discussed here). Next is the import menu for reading a data file or finite element model of a mesh. The third menu is optimize, which allows the actual meshing tasks of smoothing, remeshing, and h refining and the mesh tools options. A graphics display menu gives the user a variety of graphics plotting options for viewing the mesh. The run info menu option gives the user statistical information about the mesh, the save menu writes output files in many user-selectable formats, the red quit menu will terminate a run normally, and the help menu system provides on-line interactive guides to operation of the software.

The current file interface choices include the following: the ANSYS code[10]; the I/FEM code from Intergraph Corporation[11]; the structures codes PATRAN,[12] NASTRAN,[13] and DYNA3D[14]; and both ASCII and binary PLOT3D formats. Others are now being added, and virtually any code's mesh can be accommodated if the format is given. The mesh optimization tools submenu gives the user eight options: spring smoothing, variational optimizer, remesh, h refine, linear to quadratic element type, quadratic to linear element type, join triangles to form quadrilateral elements, and split quadrilaterals to form triangular elements. The output file menu gives the user the opportunity to write the new mesh onto files in virtually any standard code format or a customized format for specific users.

Computational Results

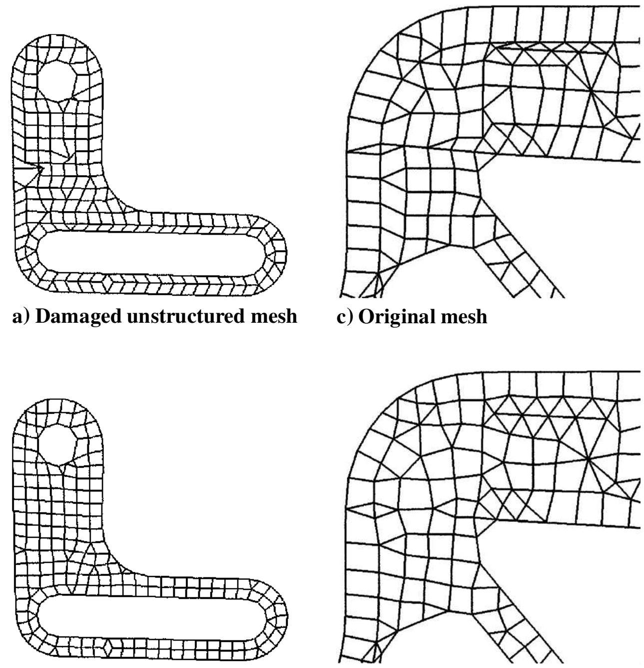
A generic hypersonic airplane has been meshed with a tetrahedral mesh to illustrate the applications of OptiMesh to complete tip-to-tail configurations. Figure 1a shows the symmetry plane mesh on the body and walls of the computational domain, and Fig. 1b shows a view of the surface mesh from the top of the airplane. Note that a relatively coarse mesh is shown here for illustration only; the computational mesh contains about 5×10^6 elements.

Two examples are presented here of use of the software to improve an existing mesh. The existing meshes were generated externally by other codes in the industry, read into OptiMesh, repaired or improved, and then rewritten out to files. Figure 2a shows a very simple unstructured mesh generated by an industry code that has an obvious (and intentional) error or damage to an element. Also note that certain of the triangles are small relative to most of the others. The spring smoother very quickly repairs the bad element and smoothes the overall mesh for marked improvement (Fig. 2b). Models with many damaged elements can likewise be processed. Figure 2c is an example of the use of OptiMesh to improve the quality of an existing mesh. The model is a portion of a wheelchair design that was generated by the Intergraph Corporation finite element code system,[11] and the variational smoother was employed to improve the mesh. The original mesh is shown in Fig. 2c, and the smoothed mesh is shown in Fig. 2d. It is seen that the element sizes are more uniform and the angles between the element sides are less skewed. From a basic viewpoint, this can be termed a better mesh, although some would argue that a better solution is needed before calling it a better mesh.



a) Computational domain surface mesh b) Body surface mesh, top view

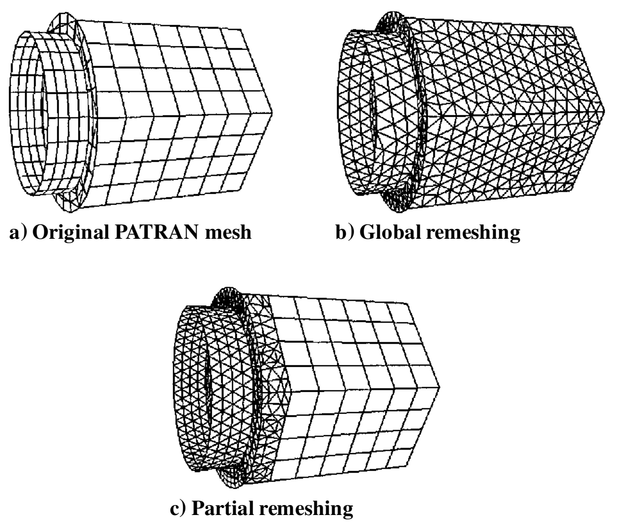
Fig. 1 Unstructured mesh for generic hypersonic plane.



a) Damaged unstructured mesh c) Original mesh

b) Repaired unstructured mesh d) Smoothed mesh (variational)

Fig. 2 Mesh improvement methods for computational mechanics.



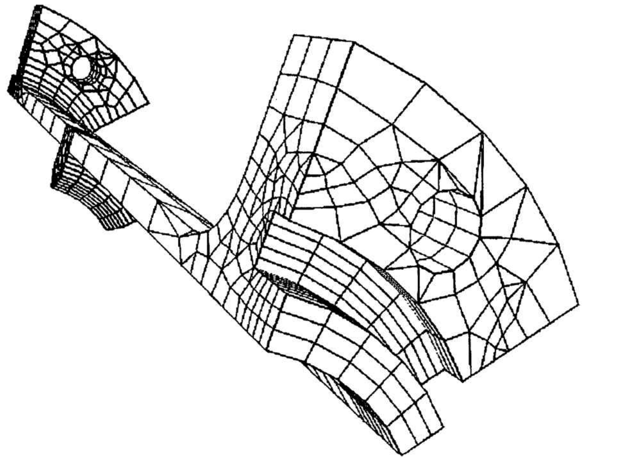
a) Original PATRAN mesh b) Global remeshing

c) Partial remeshing

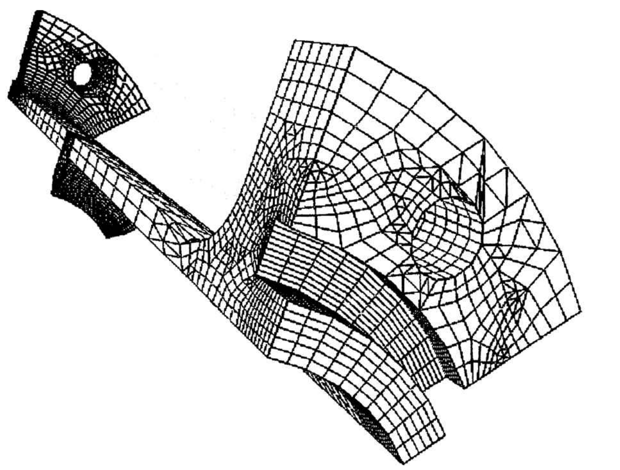
Fig. 3 Remeshing of existing finite element model.

Figure 3 shows a before-and-after-remeshing view of a finite element model. The model, shown in Fig. 3a, is a spacecraft bus used for NASA applications.[9] Figure 3a is an all-quadrilateral surface mesh generated by the PATRAN code.[12] Figure 3b gives the results of a full remeshing and shows the surface-only, all-triangular mesh. Figure 3c shows the application of the software to a partial remeshing of the same model. This application results in a mixed-element-type mesh, which is an inherent capability of the software.

Figure 4 shows a finite element model of a portion of a nozzle chamber for a rocket-powered vehicle.[9] Figure 4a is the original mixed triangle/quadrilateral surface mesh as output from an ANSYS model.[10] The interior is a mixed tetrahedral/hexahedral mesh. Figure 4b is the mesh after global h refinement with the OptiMesh software. The quadrilaterals and triangles on the surface (and the interior mesh) have been refined and can now be rewritten out in ANSYS (or other) format so that the fluids or stress analysis can be rerun with a higher fidelity.



a) Surface mesh: original



b) Surface mesh: h refined

Fig. 4 Existing finite element mesh h refinement.

Conclusions

Methodology and computer software have been developed and assembled into a generalized meshing environment for computational mechanics. It was shown that the generalized tools can generate unstructured meshes from commercial CAD databases, repair/improve existing meshes, and locally remesh/refine meshes from commercial finite element codes. The conclusion is then reached that this meshing environment is a very general tool with extensive and valuable applications to computational mechanics and is a step forward in advancing the critical automation of meshing technology.

Acknowledgment

This work was sponsored in part through Small Business Innovation Research contracts with NASA: NASA Marshall Space Flight Center Contract NAS8-40164.

References

[1]Thompson, J. F., and Weatherill, N. P., "Aspects of Numerical Mesh Generation: Current Science and Art," AIAA Paper 93-3539, June 1993.

[2]Löhner, R., and Parikh, P., "Three-Dimensional Mesh Generation by the Advancing Front Method," *International Journal for Numerical Methods in Fluids*, Vol. 8, July 1989, pp. 1135–1149.

[3]Spradley, L.W., and Anderson, P. G., "Finite Difference Mesh Generation by Multivariate Blending Function Interpolation," NASA CP-2166, Oct. 1980.

[4]Thompson, J. F., "Composite Grid Generation Code for General Three-Dimensional Regions, The Eagle Code," *AIAA Journal*, Vol. 26, No. 3, 1988, pp. 271, 272.

[5]Mavriplis, D. J., "An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness," AIAA Paper 93-0671, Jan. 1993.

[6]McMorris, H., and Kallinderis, Y., "Octree-Advancing Front Method for Generation of Unstructured Surface and Volume Meshes," *AIAA Journal*, Vol. 35, No. 6, 1997, pp. 976–984.

[7]Blacker, T. D., and Stephenson, M. B., "Paving: A New Approach to Automatic Quadrilateral Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 32, May 1991, pp. 811–847.

[8]Cabello, J., Löhner, R., and Jacquotte, O. P., "A Variational Method for the Optimization of Two- and Three-Dimensional Unstructured Meshes," AIAA Paper 92-0450, Jan. 1992.

[9]Spradley, L. W., Löhner, R., and Mahaffey, W. A., III, "Mesh Optimization Tools for Complex Structural Models," ResearchSouth, Inc., Final Rept., NAS8-40164, Huntsville, AL, Jan. 1996.

[10]"ANSYS, Version 4.4a, Version 5.0," ANSYS Corp., Houston, PA, March 1994.

[11]"Intergraph Finite Element Modeling Packages, I/FEM and I/FEMp," Intergraph Corp., Huntsville, AL, Nov. 1993.

[12]"PATRAN Software," PDA Corp., Los Angeles, CA, March 1993.

[13]"MSC/NASTRAN User's Manual," Vol. 1, Version 67, McNeal–Schwendler Corp., Los Angeles, CA, Oct. 1991.

[14]Whirley, R. C., and Hallquist, J. O., "DYNA3D—A Nonlinear Explicit, Three Dimensional Finite Element Code for Solid and Structural Mechanics—User's Manual," Lawrence Livermore National Labs., UCRL-MA 107254, rev. 1, Livermore, CA, March 1993.

J. Kallinderis
Associate Editor

Jets in Ground Effect with a Crossflow

Jorge M. M. Barata*
Universidade da Beira Interior, 6200 Covilhã, Portugal

Nomenclature

D	= diameter of the jet
H	= height of crossflow channel
k	= turbulent kinetic energy
p	= pressure
Re	= Reynolds number
U	= horizontal velocity, $\bar{U} + u'$
$\bar{U}, \bar{V}$	= mean velocities
u', v'	= fluctuating velocities
V	= vertical velocity, $\bar{V} + v'$
X	= horizontal coordinate (positive in the direction of crossflow)
Y	= vertical coordinate (positive in the direction of jet flow)
Z	= transverse coordinate (positive on the right side of the crossflow duct looking upstream)
ε	= dissipation rate of k

Subscripts

j	= jet exit
0	= crossflow

Introduction

TURBULENT jets impinging on flat surfaces through a low-velocity crossflow are typical in impingement cooling applications in industry, as well as in the flow beneath a short/vertical takeoff aircraft that is lifting off or landing with zero or small forward momentum. In this latter application the lift jets interact strongly with the ground plane, resulting in lift losses, enhanced entrainment close to the ground (suckdown), engine thrust losses following reingestion of the exhaust gases, and possible aerodynamic instabilities caused by fountain impingement on the aircraft underside.

Received Dec. 9, 1996; presented as Paper 97-0715 at the AIAA 35th Aerospace Sciences Meeting, Reno, NV, Jan. 6–9, 1997; revision received May 28, 1998; accepted for publication June 9, 1998.

*Associate Professor, Aeronautical Sciences Department, Rua Marquês d'Ávila e Bolama. Member AIAA.