TWELVE WAYS TO FOOL THE MASSES WHEN DESCRIBING MESH GENERATION PERFORMANCE

Timothy J. Tautges[1]

David R. White[2]

Robert W. Leland[3]

[§]*Sandia National Laboratories, Albuquerque, NM,*

University of Wisconsin-Madison, Madison, WI

[1]*tjtautg@sandia.gov*

[2]*drwhite@sandia.gov*

[3]*leland@sandia.gov*

ABSTRACT

Mesh generation for finite element analysis is far from a solved problem. Although several automatic meshing algorithms exist, other difficulties of setting up a problem for finite element analysis still make this an interactive process. Undaunted, we continue to perform research on, and therefore publish, papers describing their work to overcome these problems. In our efforts to describe our work "in the best possible light", we often obscure the real technical issues in these publications, rather than honestly assessing both the pros and cons of the described approach. This is an especially fruitful endeavor when reporting on mesh generation, given the 3D nature of the problem and the natural tendency of layman to avoid understanding the details of this problem. This paper describes twelve tried-and-true methods for obscuring rather than elucidating the performance of mesh generation technology.

Keywords: mesh generation, hexahedral, quadrilateral, geometry decomposition

1 INTRODUCTION

Mesh generation for finite element analysis is far from a solved problem. Even when fully automatic algorithms exist, as in tetrahedral meshing, the other parts of the process are sufficiently difficult that a great deal of user interaction is still necessary. Undaunted, we continue to work on, and therefore publish, new approaches to the mesh generation problem. In our efforts to describe our meshing technology "in the best possible light" compared to other approaches, we often make an algorithm look better than it really is. Given the 3D nature of the meshing problem, it can be difficult to tell whether a method will really be as effective as is implied

[§] SANDIA IS A MULTIPROGRAM LABORATORY OPERATED BY SANDIA CORPORATION, A LOCKHEED MARTIN COMPANY, FOR THE UNITED STATES DEPARTMENT OF ENERGY UNDER CONTRACT DE-AC04-94AL85000.

in a publication. Since users tend to avoid understanding the details of the underlying methods, they often can't tell from the publication either.

This illustrates some techniques used to describe, or not describe, mesh generation performance. While its primary purpose is to focus attention on the ways in which performance can be overstated, it also gives tips for fully describing both benefits and shortcomings of meshing algorithms. I hope that this will improve the information exchange in the meshing community, and move us toward better solutions more quickly.

2 TWELVE WAYS

Twelve items are listed below which often serve to obscure the real performance of meshing algorithms or techniques. An attempt is made to describe both ways in which each item is used to obscure performance, and then ways in which the item can be used to further the understanding of the given technology. Some of these methods apply only to specific types of mesh generation, e.g. all-hexahedral meshing, while others are more generic. The more generic and widely applicable items are listed first.

2.1 Generate a large mesh where a small one will do

There are few things more effective at distracting from the real technical issues behind a meshing method than generating a really, really big mesh. Small details in the geometric model should be removed so they don't make generating this mesh too difficult. If questions arise about which analysis application the mesh is being prepared for, or why fine geometric details are not resolved by such a fine mesh, make a vague reference to Moore's Law and answer condescendingly that applications always need larger and larger models. This technique is especially helpful for reporting on parallel mesh generation, since it distracts from issues like parallel performance, parallel IO, and parallel partitioning.

Parallel computing applied to analysis codes does introduce a need for ever-larger meshes. Beyond a certain point, though, the reduction in discretization error due to finer mesh resolution is wasted because of the lack of fidelity to the original geometric model.

2.2 Simplify meshing at the cost of analysis

There are two ways to reduce the relative time to mesh compared to the overall analysis time: reduce the actual meshing time, or increase the overall analysis time. Since people pay attention to interactive meshing time, and because computers are always getting faster anyway (see previous item), one need only concentrate on minimizing the actual meshing time. Even if this comes at the cost of increased overall time to analysis, at least it's no longer meshing's fault.

There are several examples for which this approach has been particularly effective. One example is so-called "THEX" meshes, where all-hexahedral meshes are generated by splitting tetrahedra into hexahedra using mid-point subdivision. Although it may be necessary to generate your own analysis results (because few analysts choose in practice to use meshes like these), this is balanced by being able to show pictures of very complicated all-hexahedral meshes generated with this technique. Inside-out hexahedral meshing schemes have also used this approach effectively. For these methods, fine Cartesian mesh is used on the interior and the exterior is fitted to the boundary, or resolves the boundary in a stair-step fashion. Of course, the finer the Cartesian mesh, the better the resolution of the boundary (and the more impressive the pictures). Those concerned about analysis time for these problems should reread item 1.

Probably the most effective application of this item is the use of quadratic tetrahedral meshes. Attention has been focused almost exclusively on time to mesh, eliminating any questions about the analysis time of these meshes compared to that of hexahedral meshes with similar analysis accuracy. This argument has also benefited from the widespread belief that automatic all-hexahedral meshing is impossible, while

all-tetrahedral meshing has been a solved problem since the 1980's.

2.3 Quote time to mesh without accounting for related work

When comparing meshing approaches, it is best to compare the weaknesses of other approaches with the strengths of yours. Since the focus is on reducing the time to mesh measured using the other approach, one need only compare times for tasks included in the other approach. For example, using STL files as input to your mesh generator eliminates the problem of CAD translation; since the original approach has no issue with assigning boundary conditions, one need not include that time in the comparison. Another good example is the use of overset grids, where clearly it is easier to resolve specific details of the geometry. The interactive effort to choose overset regions and insert them into the background mesh should be de-emphasized; users of other approaches will be unfamiliar with this process anyway, so they probably won't notice. This strategy has also been effective at describing multi block-structured meshing, grafting, and mesh cutting.

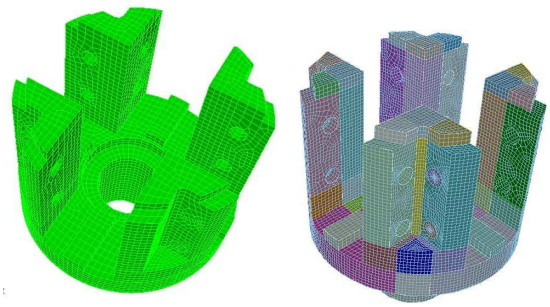

Figure 1: Model shown with all regions the same color (left) or not (right).

2.4 Use examples which appear more complicated than they are

Because quantifying the effort required to hex-mesh a given model a-priori is extremely difficult[4], we rely on qualitative visual assessment of meshing complexity, both before and after the generation of a mesh. It has been said that a picture is worth a thousand words; however, people forget to say that pictures are worth *plus or minus* 1000 words. That is, a well-chosen picture can eliminate discussion which would otherwise occur about the details of a given meshing approach. To this end, a part should be made to appear more complicated than it really is. Two proven methods to accomplish this are the addition of extraneous features which are actually handled automatically, and showing models meshed with decompositions which are not highlighted. For example, in Figure 1, drawing the model with all regions the same color (left) makes the model appear more complicated than when meshed regions are highlighted separately (right). Another favorite technique is to describe an algorithm using the most complex models handled successfully by the algorithm, implying that they represent typical performance. Since there are no standard test suites for meshing difficulty, there are no options for benchmarking performance anyway.

2.5 Obscure important details about the model

Simplifying assumptions are often necessary to complete difficult meshing problems. However, because users tend to prefer not to know the details about a given meshing approach, one need not volunteer all the nitty gritty details when offering new meshing approaches as solutions. Some common details that are sometimes mentioned which should not be include:

Non-conforming interfaces: Although many analysis codes and users would like to have mesh which is compatible across adjoining regions, achieving this compatibility often makes mesh generation much more difficult. Furthermore, it is difficult to notice incompatibility in 3D views of meshes anyway. Of course, if any questions arise about this issue, one could think of it as an analysis code's problem anyway.

Non-boundary-fitted mesh: Some of the greatest difficulties in meshing are caused by fine details on part boundaries. Generating meshes for these parts is greatly simplified if these details are not resolved. Since by definition these details are difficult to see, it will be hard to notice their absence in published pictures of meshes resulting from this approach. If analysts complain about not being able to resolve small details on the boundary, one can always offer to use h-refinement to improve the resolution locally. Whether the targeted analysis code supports h-refined meshes or not is an analysis issue, and is therefore not relevant to a discussion about mesh generation.

2.6 Describe 2D algorithm assuming easy extension to 3D

Working in two dimensions is always easier than working in three dimensions, and is also far simpler to describe to others. If working on surface-based meshing, it is also useful to assume that a surface parameterization will always be available, obviating the need to ever consider the 3D aspects of a problem. Since referencing other work is an important part of research publications, it can also be helpful to reference numerical analysis work, where the dimension often appears as a parameter in equation formulations. Working in two dimensions also makes implementation far easier, since there is no need for handling complicated geometry. If one can get by with planar surfaces always bounded by linear or quadratic curves, all the better. Of course, when using this approach, one should always mention plans to extend the work to 3D, and where possible even show pictures of 3D models for which the technique *should* work.

2.7 Describe "automatic" algorithm which uses arbitrary tunable parameters

The use of tunable parameters is a common technique used across computational simulation. It is amazing the level of agreement one can get with experiment by adjusting a few parameters[8]. When applied to physical models, e.g. material properties, it is sometimes easy to tell when parameter values do not reflect reality. Parameters used in meshing algorithms often have no physical interpretation, or an interpretation which requires knowledge which most readers have no desire to obtain. Also, since there are no standard test suites for mesh generation, one need not worry about benchmarking with a common set of parameters. To be most effective, users of this technique should make an effort to use parameters which *look* like they have a physical interpretation, but one for which it is difficult to tell whether a particular value is plausible or not. As with item #4, it helps to find models as complicated as possible to demonstrate algorithms which use this technique.

2.8 Quote time to mesh, leaving out important details

For lack of an a-priori quantitative measure of difficulty, time-to-mesh is often used when assessing mesh generation capability. However, even such a simple metric as time can be used to obscure real results. For example, there is no better way to decrease the time to mesh a difficult problem than by assigning your best person to the job, while implying that they represent "average" performance. Another effective technique is to report results for problems which are remarkably similar to ones solved in the past. However, by far the most ingenious technique we have observed is to quote the time to mesh an important model in days, failing to mention that deadlines forced those doing the meshing to work twelve hours per day. For really difficult models, one can use months, without mentioning all the Saturdays and Sundays spent in the office.

2.9 Mesh single-region models

Most meshing approaches are developed using single-part examples, since working with assemblies is, in general, more difficult. The added details introduced by assemblies are difficult to show in publications anyway, given the inherent 3D nature of these problems, and therefore aren't worth the effort. One could even argue that assembly models differ only in coupling on the boundary, and therefore are more a boundary condition (and therefore an analysis problem) anyway. Instead of using assembly models to demonstrate

the superiority of a given algorithm, it is far easier to add extraneous details to make the results look better (see Item #4). Note that one should not make reference to item #1 when using this technique, since that may imply that added computational power would otherwise enable analysis of assembly models.

2.10 Use an approach which works by itself but breaks other parts of the process

As discussed in item #2, people focus mostly on time to mesh, without regard to how a given technique affects other parts of the analysis process. Therefore, we should not feel constrained only to methods which fit into current analysis practice. In some cases, novel approaches are far easier to make work, can be made to appear to solve the problem, even though they are not supported by current analysis codes. The best example of this in practice is the generation of hex-dominant meshes (with the remainder of the mesh composed of pyramids and tetrahedra). Because of the robustness of tetrahedral meshing in general, it is relatively easy to "close" such meshes. We have observed that including tables reporting the relative number of each element gives the impression that these data are important, and does not indicate very clearly the relative volume of the part filled by each element (which might not look so favorable). When available, pictures showing analysis results on these meshes are quire useful; unfortunately, since these meshes are not generally supported by analysis codes, these pictures may not be available. In their stead, showing shaded pictures of "shrunken" elements, with each element type shown in a different color, can be impressive.

Other meshing techniques which can use this approach include Chimera/overset grids and non-conforming h-adapted meshes. In these cases, it is often easier to find analysis codes to compute on these meshes, since those codes are often developed by the same groups promoting those meshing approaches. In fact, this may be a good lesson for those developing mixed-element meshing techniques.

2.11 Report a single quality metric (or none), where several should be used

Some researchers have complained about there being too many different metrics for reporting mesh quality. However, I say if life hands you lemons, you should make lemonade; the more quality metrics one has to choose from, the easier it is to find one which reports favorable results for your meshes. Also, in this situation, it is easy to justify the addition of yet another metric, which can be designed with that goal in mind. Because of the diverse background of researchers in this field, it is also sometimes helpful to describe the new quality metric using lots of complicated mathematical equations; most mesh generation researchers don't or won't take the time to understand these formulations anyway.

A related issue is how to show mesh quality. One useful technique is to use shaded images of the mesh, with the color determined by quality metric. This is useful for two reasons: first, it is likely the smaller elements which will have poor quality, but these will also be difficult to see because of their size; and second, when working with 3D models, poor-quality elements often occur on the interior anyway, or at least can be hidden by judicious choice of viewing angle for the picture.

2.12 Show surface mesh but not interior

Everybody knows that a surface mesh is likely to have better quality than the interior (see item #6). It is also extremely difficult to assess interior mesh quality visually. Therefore, pictures of 3D models should always be shown in shaded mode, preferably from a camera position that shows the most complicated model having the best-quality mesh. In certain cases, it can also be helpful to show carefully-chosen interior features of the 3D mesh, for example of vehicle occupants in automotive interior CFD analysis.

3 BUT SEROUSLY…

If you've gotten to this point in the paper, you may realize that the last section was written facetiously; that is, I'm not *really* recommending you use the practices described there. However, in our zeal to promote the latest and greatest

meshing technology, we sometimes fail to clearly state the limitations in as complete detail as the capabilities of our work. In other cases, it may be very application-dependent which approach will work better, or other issues arise which make it less clear whether there is a "right" answer. At any rate, the following sections discuss some of the issues which arise for each of the items.

3.1 Generate a large mesh where a small one will do

Parallel computing applied to analysis codes only increases the demand for ever-larger meshes. Beyond a certain point, though, the reduction in discretization error due to finer mesh resolution is wasted because of the lack of fidelity to the original geometric model. There are very few cases I have observed where very large, unstructured meshes have been constructed for solid geometry-based domains. Examples where large discretized models are used for simple geometric domains include Direct Numerical Simulation for CFD and whole-earth models for weather modeling. In these cases, though, the geometric domain is trivial and not represented in a CAD system.

Removal of detail from a geometric model is necessary before a coarse mesh (i.e. a mesh with characteristic size much greater than the smallest model feature) can be generated with most meshing systems. However, as the mesh resolution increases, the geometric detail should also be re-introduced, since its resolution will be possible with the finer mesh. Obviously, this may be problematic without automatic meshing algorithms, e.g. for hexahedral meshes. However, even for tetrahedral meshes, it may be argued that this approach may perform better than post-meshing detail removal, as in this case it would not be necessary to always resolve the smallest details in the original model. Indeed, some proponents of post-meshing detail removal are starting to deliver a-priori detail removal capabilities as well [13]. I speculate that such "lazy detail resolution" will become more important as adaptive mesh refinement becomes more commonplace in large FEA applications.

3.2 Simplify meshing at the cost of analysis

There have been times when both THEX-type meshes and stair-step meshes have been demonstrated as a viable alternative to all-hexahedral meshes, when they actually were not viable. On the other hand, I know of no theoretical or empirical published results showing whether or why THEX meshes perform poorly. Indeed, there is at least one case where these meshes have been used as part of an award-winning analysis effort on Teraflop-class computers [14]. We often become so polarized on a given issue that we fail to even consider the possible domains of applicability in a rigorous fashion.

There is no better example of this issue than the still-unresolved hex versus tet debate. There has been some work on the theoretical aspects of this problem [16], and some indication that analysts are beginning to consider using tetrahedral meshes where in the past this may not have been considered for cultural reasons. However, this issue is far from resolved, and will require both theoretical and empirical work to answer remaining questions.

3.3 Quote time to mesh without accounting for related work

Continued delivery of vertically-integrated commercial CAE applications has helped raise the level of awareness on the issue of overall time to analysis, and progress is being made to address this issue in the national lab context as well. Also, there have been good examples of constrained design-analysis systems which are very effective within their application domain [15]. Therefore, progress is being made at addressing the overall time to solution as well as making use of special-purpose algorithms to solve niche (but important) applications.

3.4 Use examples which appear more complicated than they are

This issue will never go away entirely, and one could argue that there is merit in showing the upper bound of capability when reporting on a given meshing algorithm or technique.

Another way to address this issue is by developing suites of test problems on which algorithms can be compared. One such effort is described later in this paper.

A corollary to this issue is models which are much more difficult to mesh than they appear. One such model is shown in Figure 2; part of the difficulty with this model is the proximity of surface boundaries when surfaces through the sweep are projected to a common plane. Other examples include models with large features in close proximity to each other (which show small characteristic size in the absence of features anywhere near that size) and models with details (often translation artifacts) too small to see visually.

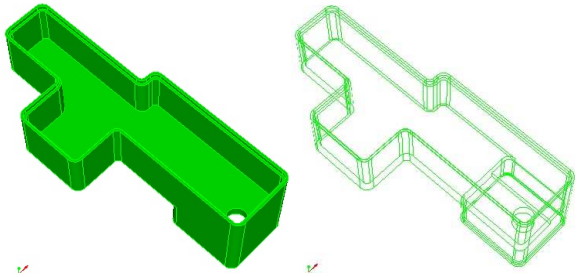

Figure 2: Model that's surprisingly difficult to hex-mesh, due to parallel lines in projected cross-sections.

3.5 Obscure important details about the model

Modeling non-conforming interfaces has received attention relatively recently, and is sometimes done in practice using "tied contact" surfaces. Unfortunately, results have not been encouraging, and the tied contact technique is not often used in practice. One bright spot in the area of non-conforming interfaces is the use of embedded boundary techniques in CFD analysis [20]; in certain cases, it may be worth exploring these techniques for application to FEA.

3.6 Describe 2D algorithm assuming easy extension to 3D

Working in two dimensions can be an important part of developing robust 3D meshing algorithms. However, the literature is full of cases where 3D turns out to be far more difficult than the analogous 2D algorithm. For example, pure Delaunay-based triangulation has provable quality bounds in two dimensions which do not apply in three dimensions; provable-quality tetrahedral meshing which also works in practice has only recently been found [13]. Likewise, there are many algorithms which work rather well in the plane (i.e. two topological dimensions) but which perform poorly in 3D space. Examples include intersection detection in advancing-front surface meshing [6], and winding number-based inside-outside checks[7].

A more current example of the difficulty of extending algorithms to three dimensions is the lack of a robust 3D all-hexahedral meshing algorithm, almost fifteen years after a robust all-quadrilateral algorithm was reported [17]. In particular, direct extension of the algorithm to three dimensions was quite unsuccessful [18]. In this case, it is the complexity added by the third topological dimension which makes the problem much more difficult.

In all these cases, adding a third dimension, either geometric or topological, can render inapplicable the theoretical or heuristic foundations of a 2D algorithm. In my experience, extending to that third dimension is rarely trivial.

3.7 Describe "automatic" algorithm which uses arbitrary tunable parameters

A key measure of success is whether an algorithm can succeed the first time it is applied to a given model, with no manual adjustment of parameters. While the use of adjustable parameters gives much-needed degrees of freedom for solving difficult problems, their use should be clearly described when reporting those solutions. This is another case where having a standard set of test cases for mesh generation would be helpful.

3.8 Quote time to mesh, leaving out important details

The proper conditions under which a benchmark is done will vary with the purposes of the benchmark. If testing is being done to see how fast an expert can use a given tool, then it makes sense to perform the test using the most experienced person with that tool. Great time savings can be obtained by constraining model variations to a small range of design parameters, and many good simulation tools have been developed under these conditions. However, in all cases, the conditions imposed by these tests should be clearly stated in benchmark reports.

3.9 Mesh single-region models

Simulation models are being constructed with increasing fidelity to the as-designed systems, which are often multi-part models. Furthermore, current hexahedral meshing approaches require decomposition to reduce general parts to more primitive shapes. These models have the same characteristics as multi-part models coming from design systems. This is not likely to change even after a robust all-hexahedral meshing algorithm is found, due to the inherent structure in a hexahedral mesh and the desire to match that to the structure of the domain.

Meshing multi-region models can be significantly more difficult, because of the coupling introduced by mesh matching on the interfaces. Although there are certain analyses where single-region models are more common (e.g. CFD, structural optimization of single parts), mesh generation systems are evolving from niche products into packages applicable to a wider range of physics. To be effective across a spectrum of analysis types (structural, thermal, etc.), a meshing algorithm should be able to handle multi-region models, and regions with pre-determined surface meshes.

3.10 Use an approach which works by itself but breaks other parts of the process

If mesh generation was the end of the process, we would have many more meshing algorithms to choose from, including all-hexahedral algorithms. There are many approaches to all-hexahedral meshing or hex-dominant meshing which are sufficiently robust in the mesh generation stage. The trouble is, the meshes they generate cannot be used by downstream analyses, or the analysis codes must be modified to handle these types of meshes.

There are many examples of meshes like this. Hex-dominant mesh generation is one approach, where hex elements are generated first in an advancing-front layer and tetrahedra used to close the mesh. Although there are many meshing tools able to generate meshes like this[9][10], relatively few analysis codes can handle the resulting meshes, and therefore they have had little impact.

Another example of this item is the use of tetrahedral meshes in cases where the analyst would prefer to use hexahedra. Although the mesh generation step is far easier with tetrahedra, there are cases where the analysis takes more time to obtain a given accuracy of results. Likewise, some advocates of hexahedral meshing often assume the need for hexahedra without ever seriously questioning why they are desired. In both these cases, a well-informed statement about why one element type was preferable over another might raise the level of dialog on this important issue.

3.11 Report a single quality metric (or none), where several should be used

There are almost as many mesh quality metrics as there are methods for generating the meshes, and some well-known metrics (e.g. positive scaled jacobian) which indicate necessary but not sufficient quality. However, there are often other factors which are necessary for a mesh to be of practical use. For example, the shortest edge length in a mesh can severely limit the time step in explicit codes due to CFL conditions.

Because of this, there is no single metric for describing a “good” mesh. Rather, the most useful rule is to clearly state what the formulation of any quality metric used is (explicitly or by reference), as well as describing the actual quality data. Recent work has been done to unify quality metrics under a common mathematical framework which can also account for variations in the definition of an “ideal” element[11]. This

research has already had an impact in how mesh quality is reported (the same author popularized the use of the scaled jacobian metric), and I encourage its use.

3.12 Show surface mesh but not interior

Showing only surface mesh, sometimes for a very carefully-chosen region of a model, can make a mesh look much better than it is. A scientist once related to me his experience evaluating a meshing tool in the following manner:

> *"The mesh looked fine on the surface, but when I saw the interior I wanted to call 911"*

The only way to overcome this problem is to pay attention to quality metrics used to describe a mesh.

4 GOT MESH?

One recent effort to address some of the difficulties in assessing various meshing approaches is the construction of a web site for storing and accessing geometry and mesh models. This web site, http://www.gotmesh.org, allows the download and upload of geometry and mesh models in various formats, along with numerical and graphical mesh quality data. This web site allows the storage of mesh and geometry models in various formats (ACIS .sat, IGES, etc.), and the indication of relations between models, for example between a model decomposed for hex meshing and the original model it came from. I hope that this web site will serve as a means of evaluating new meshing approaches, and encourage others to upload new models there when possible.

5 SUMMARY

In every technical field, there is a tendency to over-state the capabilities of the latest algorithms in the effort to cast that work in the "best possible light". Mesh generation is no different, and indeed, there are characteristics of this field which make it particularly susceptible to this practice. The purpose of this paper is to highlight some of the methods used (inadvertently or not) to obscure rather than elucidate the true performance of various mesh generation techniques on real applications, while also discussing some of the real difficulties in many of these areas as applied to mesh generation.

ACKNOWLEDGEMENTS

I would like to acknowledge the many researchers (including myself) who have published material which provided inspiration for this paper (as well as a bit of entertainment over the years).

6 REFERENCES

[1] David H. Bailey, "Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers", Supercomputing Review, Aug. 1991, pg. 54-55.

[2] J. Steger, F.C. Dougherty, J. Benek, "A Chimera grid scheme", Advances in Grid Generation, Ghia K.N. and Ghia, U, (Eds), ASME FED, 1983, V 5, pp. 59-69.

[3] Jankovich, Steven R., Steven E. Benzley, Jason F. Shepherd and Scott A. Mitchell, ""The Graft Tool: An All-Hexahedral Transition Algorithm for Creating a Multi-Directional Swept Volume Mesh", Proc. 8th International Meshing Roundtable, SAND99-2288, Sandia National Laboratories, Oct. 1999.

[4] D. White, S. Saigal, S. Owen, "Meshing Complexity of Single Part CAD Models", Proceedings, 12th International Meshing Roundtable, Sandia National Laboratories report SAND 2003-3030P, pp.121-134, Sept. 2003.

[5] J. R. Shewchuk, "What Is a Good Linear Finite Element? Interpolation, Conditioning, Anisotropy, and Quality Measures", unpublished preprint, http://www.cs.berkeley.edu/~jrs/papers/elemj.ps, 2002.

[6] Roger J. Cass, Steven E. Benzley, Ray J. Meyers, Ted D. Blacker, "Generalized 3D Paving: An Automated Quadrilateral Surface Mesh Generation Algorithm", Int. J. Numer. Meth. Engrg., 39:1475-1489 (1996).

[7] J. O'Rourke, "Computational Geometry in C", Cambridge University Press, 1994.

[8] Timothy J. Tautges, "MELCOR 1.8.2 Assessment: The DF-4 BWR Damaged Fuel Experiment", SAND93-1377, Sandia National Laboratories, Albuquerque, NM, October 1993.

[9] Steven J. Owen, Sunil Saigal, “H-Morph: an indirect approach to advancing front hex meshing”, Int. J. Numer. Meth. Engrg, 49:289-312 (2000).

[10] Robert W. Leland, Darryl J. Melander, Ray W. Meyers, Scott A. Mitchell, Timothy J. Tautges,"The Geode Algorithm: Combining Hex/Tet Plastering, Dicing and Transition Elements for Automatic, All-Hex Mesh Generation", Proceedings, 7th International Meshing Roundtable, Sandia National Laboratories, pp.515-521, October 1998

[11] Patrick M. Knupp, "Matrix Norms & the Condition Number; A General Framework to Improve Mesh Quality via Node-Movement", Proc. 8th International Meshing Roundtable, SAND99-2288, Sandia National Laboratories, Oct. 1999.

[12] C. R. Dohrmann, S. W. Key, M. W. Heinstein, “Methods for connecting dissimilar three-dimensional finite element meshes”, Int. J. Numer. Meth. Engrg, 47:1057-1080 (2000).

[13] “Simmetrix Releases Simulation Modeling Suite 5.2”, press release, Simmetrix Inc, April 27, 2004, http://www.tenlinks.com/NEWS/PR/simmetrix/042704_simmodelv52.htm.

[14] Manoj Bhardwaj, Kendall Pierson, Garth Reese, _ Tim Walsh, David Day, Ken Alvin and James Peery _ Charbel Farhat and Michel Lesoinne, “Salinas: A Scalable Software for High-Performance Structural and Solid Mechanics Simulations”, SC 2002, Baltimore, MD, Nov. 16-22, 2002.

[15] Naveen Rastogi, Fatma Kocer and Rodolfo Palma, “A Computer Aided Optimization Tool to Design Electromagnetic Retarders”, SAE Technical Paper Series 2004-01-0382, 2004 SAE World Congress, Detroit, Michigan, March 8-11, 2004.

[16] J. R. Shewchuk, “Constrained Delaunay Tetrahedralizations and Provably Good Boundary Recovery”, Proceedings, 11th International Meshing Roundtable, Sandia National Laboratories report SAND2002-2777P, pp. 193-204, September 15-18 2002.

[17] Ted D. Blacker, Michael B. Stephenson, "Paving: A New Approach To Automated Quadrilateral Mesh Generation", Int. J. Numer. Meth. Engrg., 32:.811-847 (1991).

[18] Michael B. Stephenson, Scott A. Canann, Ted D. Blacker, “Plastering: A New Approach to Automated, 3D Hexahedral Mesh Generation; Progress Report I”, Sandia National Laboratories, SAND89-2192, February 1992.

[19] Naveen Rastogi, Fatma Kocer and Rodolfo Palma, “A Computer Aided Optimization Tool to Design Electromagnetic Retarders”, SAE Technical Paper 2004-01-0382, http://whitepapers/2004_01_0382.pdf.

[20] H. Johansen and P. Colella, “A Cartesian grid embedded boundary method for Poisson's equation on irregular domains”, J. Comput. Phys. 147, 60 (1998).