

FULLY AUTOMATIC ADAPTIVE MESH REFINEMENT INTEGRATED INTO THE SOLUTION PROCESS

Joseph R. Tristano, Zhijan Chen, D. Alfred Hancq, Wa Kwok

ANSYS Incorporated, Canonsburg, PA U.S.A.

{joe.tristano,james.chen,al.hancq,wa.kwok } @ansys.com

ABSTRACT

Finite element analysts and designers need to feel confident in the results of their analyses before sending a product to prototype or production. Mesh discretization can greatly influence the desired results. In this paper we present framework for adaptive mesh refinement to obtain FEA results with a desired accuracy. The process involves adaptively refining the mesh based on solution error norms until the result desired converges to certain accuracy. The adaptive refinement/meshing process must be fully automatic and very robust. We present an exhaustive method to create a fully automatic and integrated process that takes advantage of many of the mesh refinement and mesh optimization algorithms found in literature. The results of the process provide the user with the desired accuracy **in the smallest number of iterations possible.**

Keywords: h-adaptivity, adaptive mesh refinement, adaptive finite element

1. INTRODUCTION

Finite element analysts and designers need to feel confident in the results of their analyses before sending a product to prototyping or production in order to save their company's time and money. This is especially true for users who are not experts in FEA. The discretization of the model intended for analysis can greatly influence the outcome of their desired results. In order to feel confident with the desired results we have developed a fully automatic adaptive mesh refinement process that is integrated into the solution process. The mesh is refined based upon error norms and other information output by the solver. In order to make the process reliable and robust, multiple methods for refining the mesh have been implemented. The process also needs to be efficient, i.e., the process needs to mark enough nodes for refinement such that the iterations are not wasted in the solver. The importance of this work lies in the method to integrate the tools and procedures found in literature into a fully embedded automatic and robust adaptive solution process.

The remainder of this paper will first discuss the overall framework of the process. We will then discuss the previous work in the area on which the building blocks for the process were built. We will then go into detail of the sub processes within the process such as methodology for selecting nodes and elements for refinement and the refinement process itself. We will finally show some examples and discuss areas for future work. This paper will focus on refinement of tetrahedral meshes. However, the method can easily be extended to other element types.

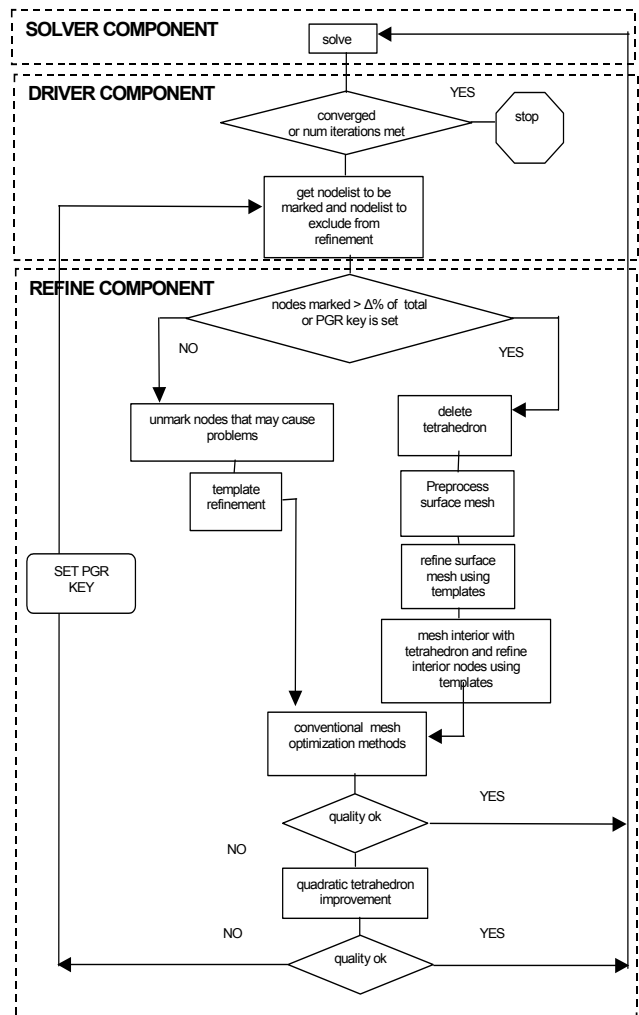


Figure 1 Overview of Adaptive Solution Process

1.1 Process Overview

The adaptive solution process is an iterative process illustrated Figure 1.

It is clearly seen that this adaptive solution process does not lend itself to failure. Various techniques are used to protect the refine process from producing poorly shaped elements. It is imperative that the mesh generated from the refinement process is of adequate quality.

The process is as follows. The initial solve takes place. The solver driver then determines which nodes must be refined and which nodes must be excluded from refinement (to prevent refinement of singularities). Depending on the amount of nodes marked, two paths may be taken. The first path is the template based tetrahedron refinement based on Staten's work. The second path uses a Pseudo Global Refinement (PGR) method that: deletes the tetrahedron, refines the surface mesh and performs some optimization of the surface mesh, re-meshes with tetrahedron and refines any marked interior nodes using templates. Extensive tetrahedron optimization is performed using conventional methods of B.Joe [1] for each method of refinement. If the conventional optimization methods fail, more specialized quadratic tetrahedron optimization methods are called (Kwok et al. [2]). This process iterates until the solution has converged or the number of adaptive iterations has been met.

If either method fails, various styles of the PGR algorithm work to refine the model to create a high quality tetrahedron mesh.

The processes communicate via data passed through COM Interfaces. These interfaces allow us to maintain each procedure of the process independently. The interfaces are implemented in several components (DLL's). The solver component solves the FE model. The driver component determines what elements should be refined based on analysis type specific criterion. This component communicates between the solver and mesh refiner and determines when the adaptive loop should stop. This component also charts the progress of adaptivity for the user, which enables the user to track the convergence of the result per refinement iteration along with displaying the number of nodes and elements for each iteration. The mesh refiner asks the driver for a mesh object and the nodes and elements to be refined and excluded from refinement. This component is self-encapsulated and determines whether templates or PGR refinement should be executed. If the quality of the output mesh is not adequate, more refinement and mesh optimization are performed to increase the quality to acceptable levels. This componentization also enables quick prototyping and implementation of new refinement algorithms, solvers, and error estimators.

1.2 Previous Work

The literature contains many works regarding error estimates [3],[4] and refinement methods[5],[6], however, the methods used to fully automate and integrate these tools into a robust failsafe algorithm have been trivialized. When dealing with

complex real world geometry and quadratic tetrahedron, this process is never trivial.

1.2.1 Adaptive Refinement

Diaz-Morcillo [7] presented an adaptive mesh refinement process for electromagnetic problems. Since the elements are usually linear and the geometry is usually simple for this class of problems, the framework for the algorithm is allowed to be quite simple: solve, estimate error, refine. Jones [8] also describes a method very similar to Diaz's. His work also concentrates more on the refinement algorithm's than creating a fail-safe process that can run on real world geometry.

1.2.2 Template Based Refinement

We have chosen to use Staten's [6] (Figure 2) template based refinement over existing methods of refinement. This method was chosen over Delaunay[5] refinement because it has shown the ability to maintain overall mesh quality after successive refinements.

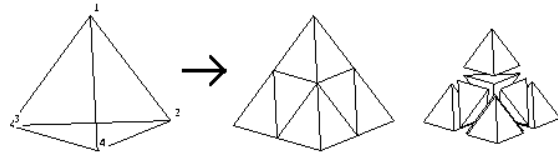


Figure 2 Staten's Template Based Refinement

2. ADAPTIVE REFINEMENT SUBPROCESSES

2.1 Methodology for Choosing Candidates for Adaptive Refinement

Adaptive H refinement is supported for structural, thermal, and modal analysis. The result quantity ϕ , the expected accuracy E (expressed as a percentage), and the region R on the geometry that is being subjected (scoped) to adaptive analysis may be selected. The user-specified accuracy is achieved when convergence is satisfied as follows:

$$100 \left(\frac{\phi_{i+1} - \phi_i}{\frac{1}{2}(\phi_{i+1} + \phi_i)} \right) < E, \quad i = 1, 2, 3, \dots, n \text{ (in } R) \quad (1)$$

Elements will be flagged for refinement on some basis measure. The criterion for which elements and nodes are selected for adaptive refinement depends on analysis type and on what results quantities are requested. In addition, depending on analysis type and the requested result type of ϕ , nodes may be flagged for refinement to augment the element list. This is done to help ensure that true convergence is achieved on the result ϕ .

2.1.1 Element Selection Methodology, Structural and Thermal Analysis

A basis measure must be employed to identify poorly shaped elements that are causing analysis inaccuracies. The procedure to select the elements for refinement, aside from the error calculation itself, is the same for both structural and thermal analysis and will be discussed first. For structural analysis, a displacement based error measure developed by Zienkiewicz-Zhu[3] is used as this basis. Essentially an elemental stress error is calculated from the difference between the average nodal stress and the element nodal stress. This stress error is then integrated over the element volume to calculate an energy error that is then used as the structural error basis. This calculation is performed over the element set and can be seen mathematically by [9]:

$$\{\Delta\sigma_n^i\} = \{\sigma_n^a\} - \{\sigma_n^i\}$$

where:

$$\{\Delta\sigma_n^i\} = \text{stress error vector at noden of elementi}$$

$$\{\sigma_n^a\} = \text{averaged stress vector at noden} = \frac{\sum_{e=1}^{N_e^n} \{\sigma_n^i\}}{N_e^n}$$

$$N_e^n = \text{number of elements connected to noden}$$

$$\{\sigma_n^i\} = \text{stress vector of noden of elementi}$$

Then for each element:

$$e_i = \frac{1}{2} \int_{vol} \{\Delta\sigma\}^T [D]^{-1} \{\Delta\sigma\} d(vol)$$

where:

$$e_i = \text{energy error for elementi}$$

$$vol = \text{element volume}$$

$$[D] = \text{stress-strain matrix of the element} \quad (2)$$

$$\{\Delta\sigma\} = \text{stress error vector}$$

A similar calculation for temperature-based problems using elemental heat flux was given by Huang and Lewis[4] and is used as a basis for thermal energy error.

After all the element errors are collected, a cutoff range technique is employed where the elements with the largest error are flagged for refinement. A critical value is calculated and elements with an error above this value will be flagged for refinement according to the equation:

$$e_{crit} = e_{max} - \beta(e_{max} - e_{min}) \quad (3)$$

where :

$$e_{crit} = \text{minimum element error required to be flagged for refinement}$$

$$e_{max} = \text{maximum element error in element set}$$

$$e_{min} = \text{minimum element error in element set}$$

$$\beta = \text{cutoff factor, } \beta \in \mathcal{R}_0^1 \text{ default of .75}$$

As can be seen from the equation above, a cutoff factor of zero would cause only elements with the highest error to be flagged while a factor of one would flag all elements for refinement. At first glance, the default β of .75 would seem to allow a disproportionately large number of elements to be flagged but experience has shown this value to be valid. This is due to a typically large gradient in the distribution of

element error throughout the element set. Although highly dependent on the loading and geometry, typical percentages of elements flagged for refinement range from 5 to 20 percent.

2.1.2 Element Selection Methodology, Modal Analysis

In selecting elements for modal refinement, a different basis mechanism as well as different cutoff technique is employed. The basis function is simply largest element volume. The element set is sorted on that basis and then a cutoff element technique is employed so that a set percentage of the largest elements are flagged for refinement. Typically, 25 percent of the elements will be flagged for refinement but this value is controllable by the user.

2.1.3 Node Augmentation Selection methodology

Although the element selection methods described above perform well in obtaining an accurate global solution, node augmentation based on the result may be required to ensure that a local converged result is achieved. This is applicable to results where local refined meshes are required to achieve an accurate solution. Consider the case where ϕ is based on equivalent stress and a stress concentration exists. Consider also that in that region the element error is low enough compared to another region such that the other region in the model is flagged for refinement. Thus, as a result, the change in ϕ will be likely small enough to satisfy the requested accuracy E, although true convergence has not been reached. To aid in this, a list of nodes on the region R will be flagged for refinement that have values near ϕ and thus ensures that a false convergence will not occur. This is node augmentation is only applicable to certain type of results where this false convergence can occur and includes:

- Structural stresses and strains
- Structural post tools based on stress and/or strain
- Structural contact results
- Thermal heat fluxes

Nodes are selected for refinement based on a cutoff range technique. However, since convergence may be applied to either the minimum or maximum of ϕ , and in addition, ϕ may be positive or negative, logic is required to handle each permutation. Each node in the set will test the following logic statement and will be flagged for refinement if the statement is true:

$$\begin{array}{l} \text{Converging on } \phi_{max} \\ \phi < 0 \wedge (\alpha\phi_n > \phi) \\ \phi \geq 0 \wedge (\phi_n > \alpha\phi) \end{array} \quad \begin{array}{l} \text{Converging on } \phi_{min} \\ \phi < 0 \wedge (\phi_n < \alpha\phi) \\ \phi \geq 0 \wedge (\alpha\phi_n < \phi) \end{array} \quad (4)$$

where:

$$\phi = \text{result quantity to be converged upon}$$

$$\phi_n = \text{result quantity at node n}$$

$$\alpha = \text{cutoff factor, } \alpha \in \mathcal{R}_0^1, \text{ default of .8}$$

Care must be taken when compiling this node list. Artificial singularities may exist in the model due to the applied loads that could cause a divergent refinement series. A list of boundary conditions known to cause these singularities is compiled and any nodes in the region R that exist on these boundaries are excluded as candidates for node augmentation. Note that geometric singularities may exist in model that the program cannot detect and thus care must be taken by the user to ensure that a proper region R is selected for the result quantity where convergence is of interest.

2.2 Refinement Process

2.2.1 Refinement Driver

After the nodes and elements are marked for refinement, the refinement driver loads the refine component. Data, such as the previous mesh, marked and excluded nodes and elements, and the BRep are passed into the refine component via a COM Interface (Figure3).

```
interface IDGAnsAdaptive : IUnknown
{
    [helpstring("method RefineMesh")]
    HRESULT RefineMesh( IUnknown *piBRep,
        IUnknown *piOriginalMesh,
        BOOL *a_brefineelems,
        BOOL *a_brefinenodes,
        BOOL *a_bElementExclusionList,
        BOOL *a_bNodeExclusionList,
        int iRefineLevel,
        int iRefineDepth,
        IUnknown *piNewMesh,
        IUnknown *piProgressIndicator,
        int iMinimizeMarked,
        BOOL *pbShapeFailure,
        BOOL bDoPseudoGlobalRefinement );
};
```

Figure 3 COM Interface for Refinement Component

2.2.2 Node Marking and Element Improvement

The mesh is converted into a local data structure and the appropriate nodes are marked. During the node marking leg, the algorithm may determine that some quadratic tetrahedron may invert during the refinement process. At this time a few things may happen: Mid-side nodes on element edges interior to the topological body are perturbed, poor boundary elements may be deleted, nodes may be unmarked.

When bending interior edges, we examine the diagonal of the tetrahedron that is chosen to be refined that will invert the children tetrahedron. The mid-side node is perturbed until the children elements will have an acceptable shape.

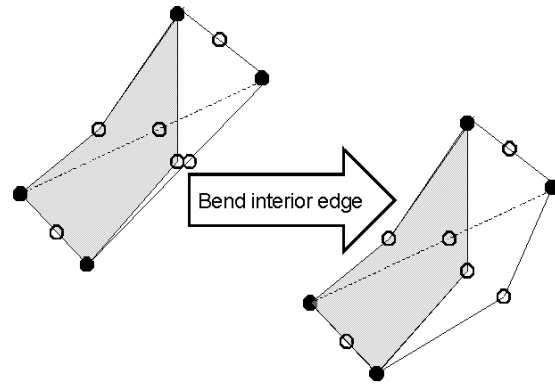


Figure 4 Bending Interior Edges

There may be cases where there is not enough room on the interior of the model to bend an interior element edge to create an acceptable quality element (Figure 4). In this case, we determine if the tetrahedron is a poor element that lies on the boundary of the topological body. If this poor element lies on the boundary, a few operations may be performed to either modify the tetrahedron or simply delete it. Some of these operators include:

- Splitting the longest edge to create better-shaped elements
- Swapping element edges to create a better configuration
- Moving interior corner nodes to improve resulting element shapes
- Deleting the element if it is a kite or cap on the boundary
- Modifying a mid-side node of the elements on the boundary.

If all of these methods fail to produce a valid configuration, the nodes marked on the poor element will be unmarked for refinement. Although the driver component wanted the nodes to be refined, the unmarking of nodes has been proven to be an acceptable practice via empirical studies.

2.2.2 Template Based Refinement

The template base refinement follows Staten's work with slight modifications. There are cases where after some refinement that the diagonal to refine the tetrahedron upon will create invalid elements. In this case, we again try to bend interior mid-side nodes to create an acceptable quality element.

2.2.3 Pseudo Global Refinement (PGR)

In the cases where template based refinement fails, we try a new method called Pseudo Global Refinement (PGR). The PGR method is an innovative technique that utilizes template based surface mesh refinement and local remeshing techniques to perform mesh refinement during an adaptive solution process.

The PGR method is comprised of the following steps.

1. Remember marked interior node locations (“memory nodes”) and delete tetrahedron
2. Split boundary edges that have very high transitions as to improve the quality of the resultant mesh
3. Improve quality of surface mesh via triangle quality swapping
4. Refine marked nodes on surface using template refinement and clean surface mesh
5. Fill body with tetrahedron
6. Refine Interior Nodes closest to “memory nodes”
7. Optimize Mesh

2.2.3.1 PGR Surface Mesh Aspect Ratio Adjustment

To obtain a successful refinement using the PGR method, the surface mesh is massaged to prevent poor element quality later in the process (Figure 5). This preventative method reduces high transitions in the mesh by looping through marked nodes and determining if any edge, $E_i @ \text{node } M$, with length, $L_i @ \text{node } M$, exceeds the ratio β when compared to the smallest edge length at the node, $L_{\min} @ \text{node } M$. If this edge ratio exceeds β , the edge is split to reduce the high transition. In practice, we have found that $\beta=10.0$ is a suitable number.

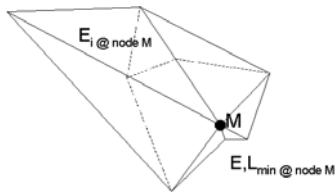


Figure 5 Splitting High Transition Edges

2.2.3.2 PGR Surface Mesh Optimization

We perform topological optimization of the mesh along with quality optimization of the mesh. The quality optimization of the mesh is required for quadratic elements. An element may have acceptable quality when it is linear but adding mid-side nodes to the element may invalidate it (Figure 6). Therefore, the quality swap is extremely important. A quality swap is one in which a swap is performed if the quality of the elements will improve in the new configuration. This step provides a high quality surface mesh to send to the template based surface mesh refinement.

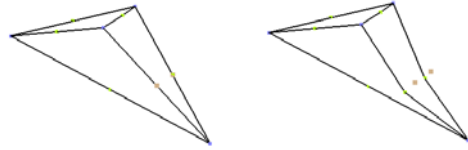


Figure 6 Acceptable Linear Element Becomes Invalid with the Addition of Mid-Side Nodes

2.2.3.3 PGR Template Surface Mesh Refinement

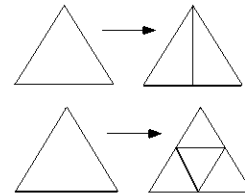


Figure 7 Template Triangle Refinement

PGR Template triangle refinement uses Staten’s method [6] of refinement to refine the marked nodes (Figure 7). This refinement is then followed by topological optimization of the triangles [10] and mixed Laplacian/Optimization smoothing[11] of the refined surface mesh.

2.3 Tetrahedron Optimization Methods

After the topological body bound by the refined surface mesh is filled with tetrahedron, mesh optimization is performed on the mesh. B. Joe’s [1] tetrahedron optimization techniques have been implemented for the initial optimization leg of the code. For cases where the mesh cannot be improved by Joe’s techniques, more advanced and specialized tetrahedron optimization operators are used. These methods consist of the same methods used to improve tetrahedron quality during the node marking leg of the process.

Improving poor boundary element tetrahedron by reposition their mid-side nodes is one of the most common specialized tetrahedron operators called in the process (Figure 8). The operator is quite simple. The smallest or inverted angle is found on the boundary face. The unconstrained edge is determined, i.e., the edge that does not lie on a topological edge is found. The direction of the edge, \vec{E} , is crossed with the surface normal, \vec{N}_{surface} , to determine the direction, \vec{V}_{move} , in which the mid-side node of the moving edge should move. This process moves the node a small δ based on the distance between the mid-side nodes of the adjacent edges until the quality of the tetrahedron adjacent to the edge are acceptable.

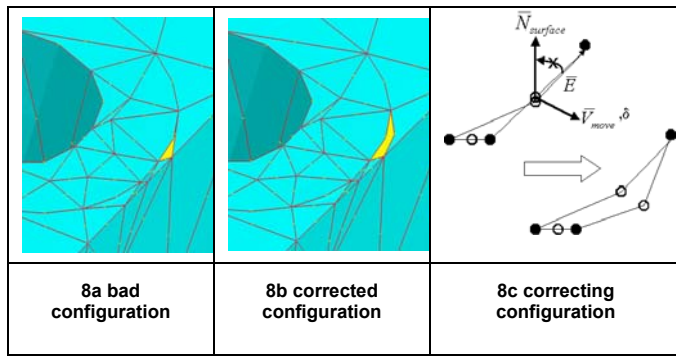


Figure 9 Improving Bad Boundary Element

3. EXAMPLES AND FUTURE WORK

3.1 Example

3.1.1 Pressure Vessel

The following example illustrates our process on a moderate complexity CAD geometry of a pressure vessel. The intersections of the fillets in the model contain many sliver surfaces, which have been defeated via “Mesh Based Defeating” [12]. The loading condition is shown in Figure 8. The initial mesh (Figure 10) was intentionally set to the coarsest setting possible to show the robustness of the refinement process. It is clearly seen that the initial mesh contains many high aspect ratio tetrahedron. The model was set to adapt on the maximum normal stress of the whole model with a convergence criterion of 7% allowable change in the maximum normal stress (Figure 11). After four iterations, the model converged to .5% change in maximum normal stress (Figure 12, Table 1). The large increase in the number of nodes is due to the massaging of the surface mesh during PGR. PGR allows the complex fillets to be refined while maintaining the overall quality of the mesh.

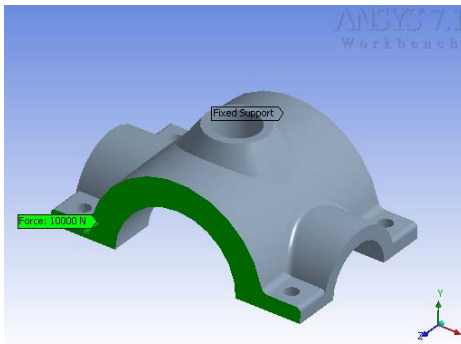


Figure 8 Pressure Vessel Loading Condition

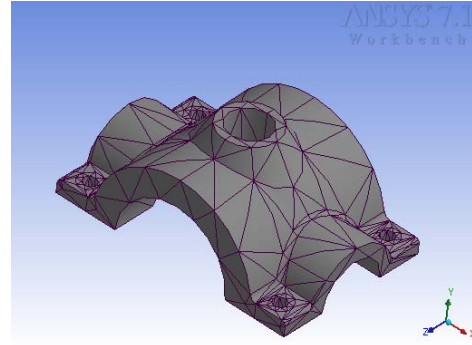


Figure 10 Pressure Vessel Initial Mesh

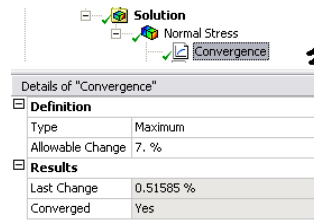


Figure 11 Convergence Control on Solution

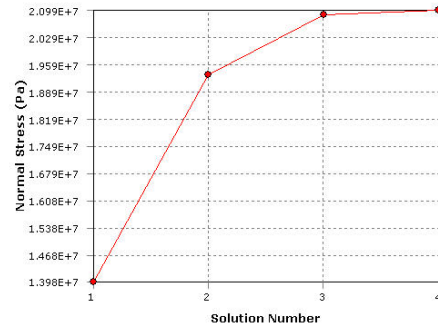


Figure 12 Pressure Vessel Convergence

Table 1 Pressure Vessel Convergence

Solution Number	Normal Stress (Pa)	Change (%)	Nodes	Elements
1	1.3980E+07		2080	1056
2	1.9378E+07	32.062	13099	4643
3	2.0886E+07	7.8	35255	22418
4	2.0994E+07	0.51585	123451	83988

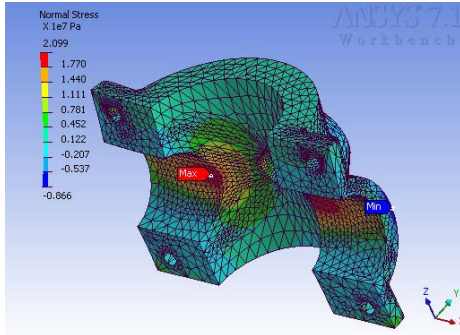


Figure 13 Pressure Vessel Converged Normal Stress

3.1.2 Plate with Hole

The next example illustrates an academic problem of a plate with a hole. The interesting item with this model is the scoping of results to the top surface (region R as discussed above). As stated above, the scoping control limits the marking of nodes to nodes lying on the scoped surface (Figure 17). The model converges to 1.88 % change in maximum equivalent stress in four iterations (Figure 16, Table2).

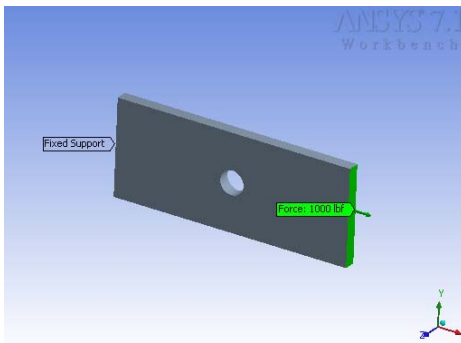


Figure 14 Plate with Hole Loading Condition

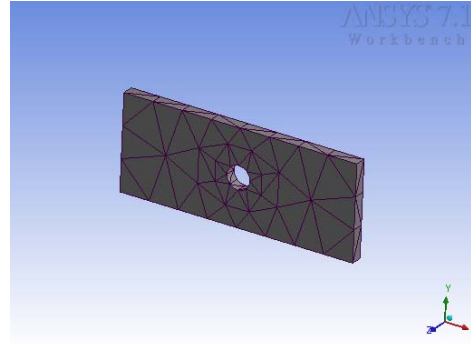


Figure 15 Plate with Hole Initial Mesh

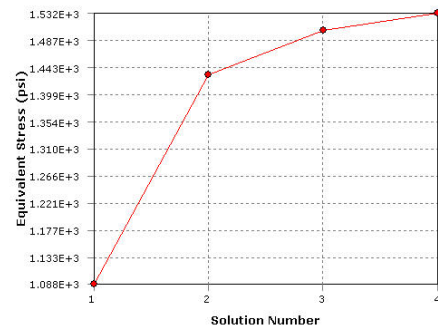


Figure 16 Plate with Hole Convergence

Table 2 Plate with Hole Convergence

Solution Number	Equivalent Stress (psi)	Change (%)	Nodes	Elements
1	1088.4		438	174
2	1430.5	27.161	2388	1183
3	1502.9	4.9363	12494	7543
4	1531.6	1.8895	35341	22925

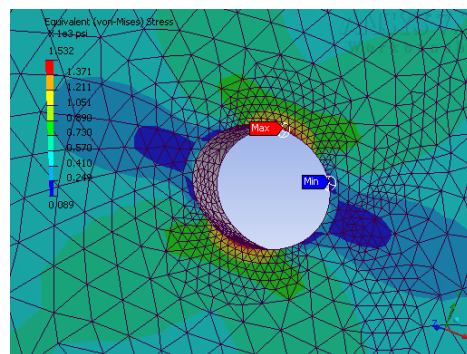


Figure 17 Plate with Hole Scoped Equivalent Stress Result

3.2 Future Work

This paper presented a framework for successfully integrating fully automatic adaptive refinement into the solution process.

This process is quite robust. However, as with any process, there is always room for improvement. Future work in this area may consist of:

- Determining better heuristics for when to choose PGR over Template refinement
- Using an optimization method to place the mid-side nodes when bending interior edges or moving the mid-side node of a poor surface tetrahedron
- Developing new specialized tetrahedron cleanup operators
- Improve refinement for thin models when the aspect ratio of the opposite side of the model is much larger than the side being refined (Figure 18)

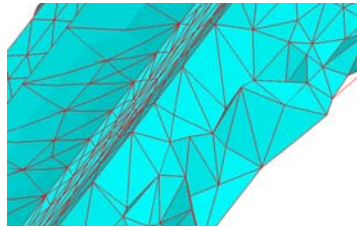


Figure 18 Cross-section of thin model with high aspect ratio tetrahedron

REFERENCES

- [1] Barry Joe, "Construction of Three-Dimensional Improved-Quality Triangulations Using Local Transformations" *Siam J. Sci. Comput.*, Vol 16, pp.1292-1307 (1995)
- [2] Wa Kwok, Z. Chen, J. Tristano, "Quadratic Tetrahedral Mesh Improvements", *Submitted to 12th International Meshing Roundtable*
- [3] O. C. Zienkiewicz,., and J. Z. Zhu, "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis", *International Journal for Numerical Methods in Engineering*, Vol. 24, pp. 337-357 (1987).
- [4] H.C. Huang and R.W. Lewis, "Adaptive Analysis for Heat Flow Problems Using Error Estimation Techniques", *Paper presented at the 6th International Conference on Numerical Methods in Thermal Problems. Also University of Wales, University College of Swansea Internal Report CR/635/89* (April 1989)
- [5] Maria-Cecilia Rivara, "Local Modification of Meshes for Adaptive and/or MultiGrid Finite-Element Methods", *Journal of Computational and Applied Mathematics*, Elsevier, Num 36, pp.79-89, (1991)
- [6] Matthew Staten, "Selective Refinement of Two and Three-Dimensional Finite Element Meshes", *Master's Thesis*, Department of Civil and Environmental Engineering, Brigham Young University, (1996)
- [7] Alejandro Diaz-Morcillo, L. Nuño, J. V. Balbastre and D. Sánchez-Hernández, "Adaptive Mesh Refinement in Electromagnetic Problems", *Proceedings, 9th International Meshing Roundtable*, Sandia National Laboratories, pp.147-155, (2000)
- [8] Mark T. Jones and Paul E. Plassmann, "Adaptive Refinement of Unstructured Finite-Element Meshes", *Finite Elements in Analysis and Design*, Num 25 pp. 41-60, (1997)
- [9] ANSYS Inc, Theory Manual
- [10] Scott A. Canann, S. N. Muthukrishnan and R. K. Phillips, "Topological Refinement Procedures for Triangular Finite Element Meshes", *Engineering with Computers, Springer-Verlag*, Vol 12, pp.243-255, (December 1996)
- [11] Zhijian Chen, J. Tristano, W Kwok, "Combined laplacian and optimization-based smoothing for quadratic mixed surface meshes", *Submitted to 12th International Meshing Roundtable*
- [12] Anton V. Mobley, M. P. Carroll, and S. A. Canann, "An object oriented approach to geometry defeaturing for finite element meshing", *In Proc. 7th Intl Meshing Roundtable '98*, Sandia National Laboratories, (1998)