Adaptive Refinement of Quadrilateral Finite Element Meshes Based on MSC.Nastran Error Measures

Mark E. Botkin
GM R&D Center
Warren, MI 48090-9055

Rolf Wentorf and B. Kaan Karamete
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

ABSTRACT

This paper describes and demonstrates a process for adaptive refinement of quadrilateral curved shell meshes using error estimates from MSC.Nastran®. The meshes have been generated originally using Unigraphics'(UG) Scenario application. Although refinement procedures for finite element meshes have been in use for many years, automated procedures have most generally been developed for triangular meshes. Furthermore, the new procedure uses the UG/Parasolid model so that new points created during the process are on the curved part surface. The software demonstrated in this report was developed to the specifications required for automotive applications by the Scientific Computational Research Center of Rensselaer Polytechnic Institute (RPI) as part of General Motors' membership in their Simulation-Based Engineering Program. The software was demonstrated on two realistic automotive body components. Refined quadrilateral meshes were produced which exhibited smooth grading from the refined elements to the coarser, unrefined areas of the mesh.

INTRODUCTION

Mesh Refinement Process

Error analysis and mesh refinement have been the subject of much research [1], although the more automated procedures have been employed only for triangulation techniques. Although much work has taken place on error estimation for quadrilateral elements [2], it is much more difficult to implement refinement procedures for such meshes. Most quadrilateral meshes are created using *mapping* procedures, which do not lend themselves to adaptive refinement. In many methods incompatible grading is used which incorporates midside nodes and multi-point constraints for transitioning. Although several papers [3, 4, 5, 6, 7, 8] have dealt with refinement of quadrilateral elements, these have all dealt exclusively with two-dimensional regions with no consideration for the problem of placing newly created points on a three-dimensional surface. In more recent years paving techniques [9,10] have been developed which have revolutionized the finite element meshing process. Such methods—members of a general class of methods called *advancing front*—also do not lend themselves to refinement. The primary purpose of this project is to improve the refinement process for paved meshes which have been generated automatically on three-dimensional curved surfaces defined by Unigraphics solid models. An overview of the entire process can be seen in Figure 1 and will be described in detail in a later section.

Error Analysis

An extensive amount of research has taken place over the past 20 years [2,11] in error analysis of finite element problems. The goal of such computations is to *estimate* the errors in a

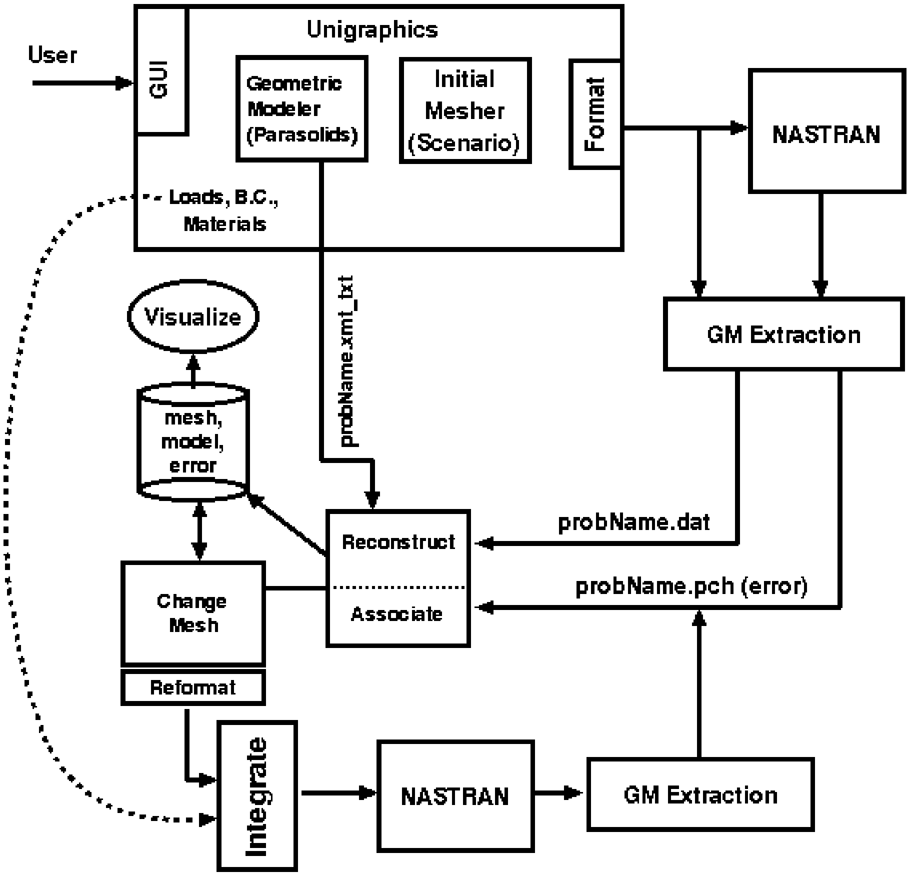


Figure 1 Refinement Process

solution given a certain descretization. Then, the mesh can be changed to reduce the errors. Obviously, the more accurate the error estimate the more efficient will be the refinement process. The number of cycles necessary to obtain the desired accuracy will need to be as small as possible in order to be efficient. Although much work has been accomplished in this area, very few procedures have been implemented in commercial finite element software. One such procedure has been implemented in MSC.Nastran [12] which will be used in this study. For completeness, the MSC.Nastran error estimation procedure will be explained in the next section.

ERROR ESTIMATION IN MSC.NASTRAN [12]

Error Estimates for Grid Point Stress Data

The error estimate to be developed here will be based upon grid point stresses. First, an averaging procedure must be used to obtain continuous stresses at grid points since stresses are normally computed *within* an element using the element's shape function. Therefore, a grid point which is shared by several elements can legitimately have several stress quantities associated with it. The averaging equation used in MSC.Nastran is

$$s_g = \sum_{i=1}^{N_e} \left(W_i s_{ei} \right) \qquad (1)$$

in which σ_g is the weighted mean value of the stress component at the grid point, σ_{ei} the value of the stress component in the i^{th} element in the neighborhood of the grid point, and W_i a weighting factor assigned to the i^{th} element. Equal weighting, i.e., $W_i = 1/N_e$, is assumed in MSC.Nastran where N_e is the number of elements connected to the grid point.

An estimate of the error in a particular component of stress at a point can be computed by assuming that the error of the corresponding stress component, computed for the elements in the neighborhood of the grid point, is related to the difference between the average stress, σ_g, and each element stress, σ_{ei}. An estimate of the *error*, δ_g,in the stress component at the grid point is:

$$d_g = \sqrt{\sum_{i=1}^{N_e} \left(W_i d_{ei} \right)^2}$$

$$= \frac{1}{\sqrt{N_e}} \sqrt{\frac{\sum_{i=1}^{N_e} \left(d_{ei} \right)^2}{N_e}} \qquad (2)$$

where δ_g is the *probable error* and $\delta_{ei} = \sigma_{ei} - \sigma_g$.

While the probable error provides an estimate for each of the three stress components at each of the grid points considered, it is more useful and desirable to combine these three estimates into a single representative measure at each point. The root mean square (RMS) value of the three estimated errors for each of the three stress components is as follows:

$$ErrorEstimate = \sqrt{\frac{\sum_{i=1}^{N_c} \left(d_{gi}\right)^2}{N_c}} \qquad (3)$$

where N_c is the number of stress components (in the case of a shell element $N_c = 3$, i.e., σ_x, σ_y, and τ_{xy}).

It should be noted that the RMS error is a reasonable measure of precision in many practical cases, but examples have been shown in which the RMS error is a poor error measure. Eq. 3 is assumed to provide an approximate error estimate for the grid point stress data.

Element Stress Discontinuities

Other error estimates may be generated by associating the error with the elements rather than with the grid points. For example, RMS errors for each stress component of an element may be computed from values of δ_{ei} that are computed for each of the N_g connected vertex grid points where $\delta_{ei} = (\sigma_e - \sigma_{gi})$. This computation is done for each stress component for all of the elements of interest.

$$d_e = \sqrt{\frac{\sum_{i=1}^{N_g} \left(d_{ei}\right)^2}{N_g}} \qquad (4)$$

As explained above, a more useful measure involves combining all three stress components as:

$$ErrorEstimate = \sqrt{\frac{\sum_{i=1}^{N_c} \left(d_{ei}\right)^2}{N_c}} \qquad (5)$$

Even though the above equations may not represent rigorous error measures, in a practical sense they are very useful since very few commercial finite element software codes provide error estimation of any kind.

Error Analysis Example

A simple, well-known example [1] will be used here to demonstrate computing error measures in MSC.Nastran. The well-known plate-with-a-hole will be used as shown in Figure 2 (which is actually one-quarter of a plate) for which dimensions are shown in Figure 3.

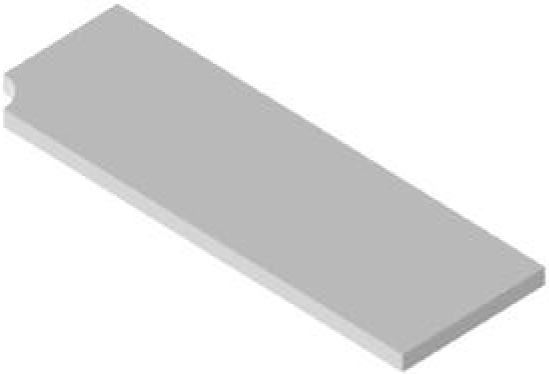


Figure 2 Plate Solid Model

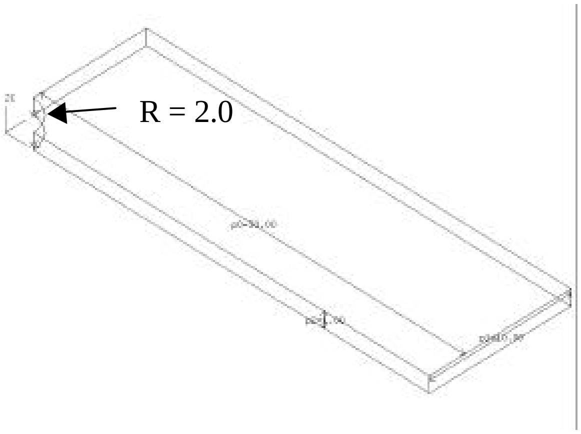


Figure 3 Dimensions of Plate

The plate has an applied uniform load of 1.0 N/mm at the ends. The resulting errors are shown in Figure 4.

Although there is no exact error result for comparison purposes, the values in Figure 4 look reasonable based upon knowledge of stress concentrations at small holes in plates. These results will be used for example purposes in the refinement process.

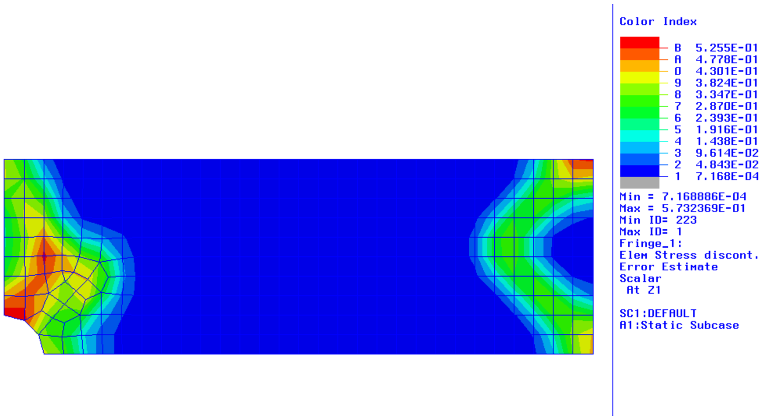


Figure 4 Error Prediction Results

THE REFINEMENT PROCESS

Manual Refinement Example

As an example of reductions that *can* be achieved in the errors shown in Figure 4, the distribution of nodes around the hole was changed *manually* using a finite element preprocessing program. The node distribution was changed from the uniform spacing of 1.0 mm to a geometric distribution ranging from 1.0 mm at the plate edge to 0.25 mm at the hole. This new mesh was then solved in MSC.Nastran and the error results are shown in Figure 5.

As can be seen in Figure 5 the error results in the refined region have been significantly reduced. As noted earlier, however, this example does not use the theoretically exact solution as the basis of comparison, but it qualitatively shows that the error measure generally indicates the area in which refinement is necessary and that refinements in those areas can reduce the error.

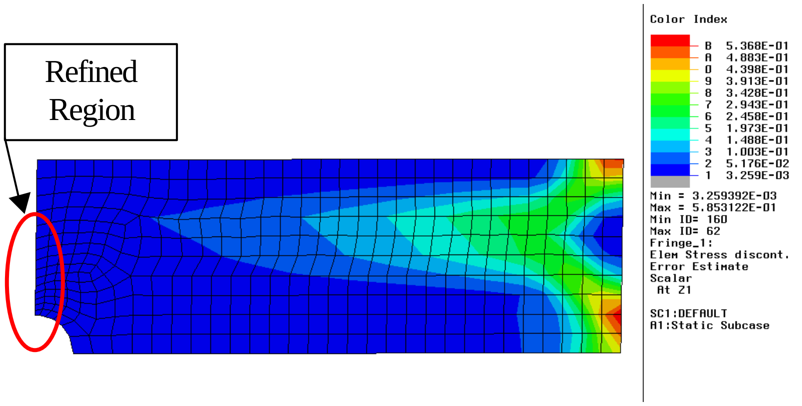


Figure 5 Errors for *Manually* Refined Mesh

Automatic Refinement

Although the above example shows the potential effectiveness of mesh refinement, in order to have an efficient adaptive modeling process it will be necessary to develop an automated refinement process. While methods based upon mapped meshing have been looked at in the past, only fully-automatic methods will be addressed here since these will be the only practical methods for speeding up the overall modeling process.

Figure 6 shows a region in which *paving* was used to create quadrilateral elements (for the purpose of this example, not all elements are shown). The paving process begins adding elements along the boundary automatically placing nodes in the interior of the region. The method places nodes so that the resulting element is as regular as possible, i.e., the skew and aspect ratios are within the default ranges. If it is desired to incorporate additional nodes, as shown in the figure, the paving method does not possess such a capability as triangulation methods, such as *tesselation*, do. Normally, *hard points* are added to paved methods by *moving* existing points to the location of the desired hard point. Obviously, though, as was shown in the example of Figure 5 refined boundaries will result in refined paved meshes, which, however, is a limited capability of what is needed for a refinement process to be developed in this project. The desired capability needs to be able to add one or more nodes within any element within a region and obtain a suitably graded mesh.

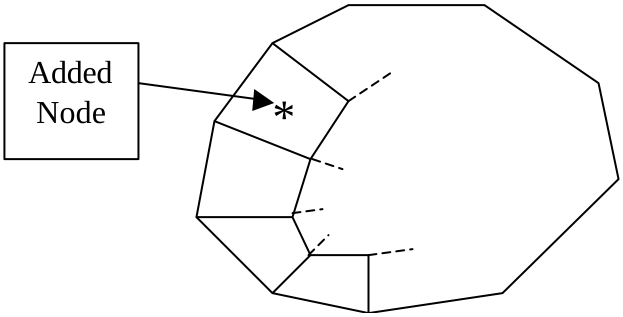


Figure 6 Paved Mesh with Added Node

RPI Quadrilateral Refinement Software

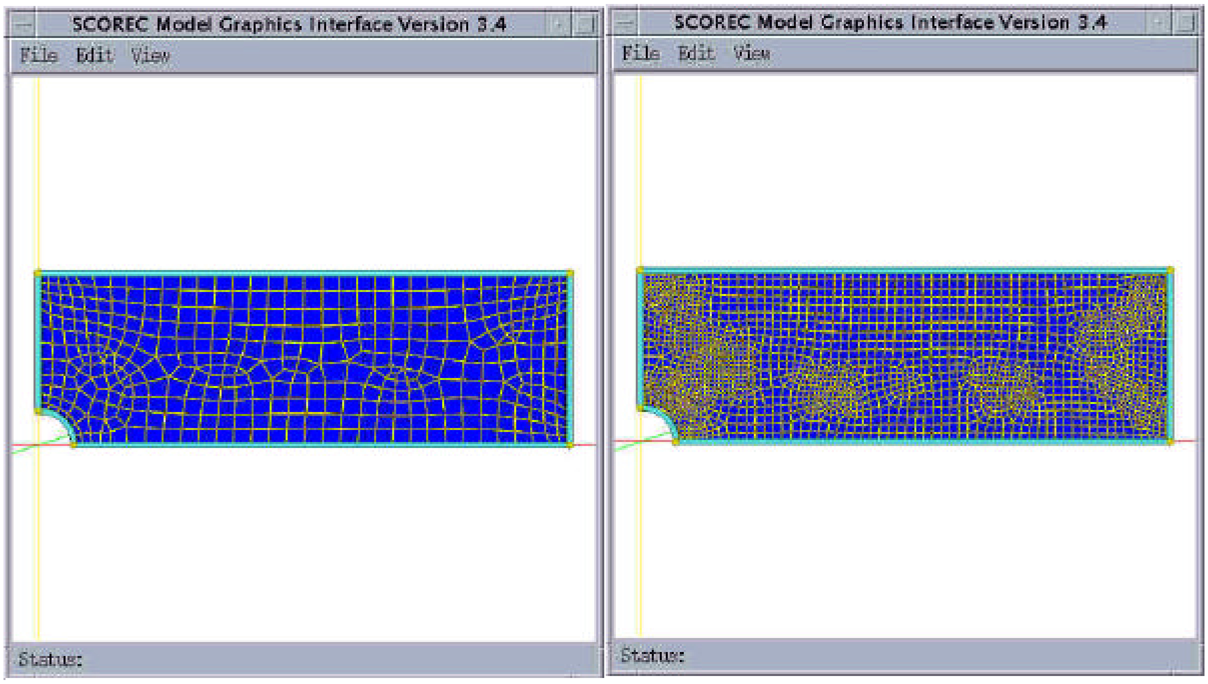
A refinement module has been developed (based upon the procedure shown in Figure 1) which refines quadrilaterals and creates a mostly-quad mesh using procedures described in References 13, 14, 15, and 16 using the MSC.Nastran error measures shown above in Eqs. 1-5. Input to the refinement scheme is as follows:

- Input data for mesh refinement:
  - The finite element surface mesh (quadrilateral or mixed) represented by file *ProbName.dat* in Figure 1.
  - Error values for each grid and element as computed by MSC.Nastran and represented by file *ProbName.pch* in Figure 1.
- The user of the refinement software must select:
  - The error *cutoff* values (see Figure 4) above which an element gets refined.
  - The number of levels of refinement.

Figure 7 shows two different levels of refinement that are possible by changing the input parameters. The refinement process is as follows:

The quadrilateral surface mesh is first converted into all triangles to apply triangular refinement templates. The conversion is done by simply splitting the quads at the diagonals. The triangular templates are preferred since the termination of the refinement is possible with triangles and it is not always possible to stop the propagation of quadrilateral refinement without creating invalid (negative Jacobian or area) elements. The information of the original quad or mixed faces that are to be refined is carried over to the all-triangular mesh. The triangular templates are then applied to these to-be refined faces. The refined triangular mesh is then converted to an almost-all quadrilateral mesh by a paving algorithm similar to the one proposed by Owen et al. [14]. It is different than the conventional paving algorithm of Blacker and Stephenson [9], in the sense that the paving is achieved on an existing triangular mesh where quadrilaterals are formed by using existing edges, by inserting additional nodes or by performing local transformations to the triangles. It is advantageous to use such an algorithm to avoid intersection computations commonly associated with advancing front procedures which are also used in the original paving algorithm of Blacker and Stephenson [9].

As another option to quadrilateral conversion, a diagonal merging and splitting algorithm of Lee [15] is also implemented. The best combinations of paired triangles are found by computing the internal angles and choosing the pairs that are close to the ideal square case. The paired triangles are merged from their common diagonals. The remaining triangles are split into quadrilaterals and the split is propagated to every direction that necessitates the further splits of


SCOREC Model Graphics Interface Version 3.4
File Edit View
Status:
SCOREC Model Graphics Interface Version 3.4
File Edit View
Status:

(a) Refinement Level 1 **(b) Refinement Level 2**

Figure 7 Adaptive Refinement of Plate

quads and triangles. One of the disadvantages of this approach, though it is robust, is that the resulting mesh may be finer than the requested size by a factor of two.

Finally, the resulting mesh is optimized by applying various topological cleaning procedures proposed by Kinney [16].

An extremely important aspect of the process shown in Figure 1 is the use of the geometric model (in this case UG/Parasolid that is shown as the data file *ProbName.xmt_txt*). As the new, refined points are created according to the above discussion, they are placed on the surface of the geometric model. Otherwise, if only the original coarse mesh was used to place the new points, they would lie on the *faceted* surfaces of the original elements. The more realistic examples that follow demonstrate this extremely important capability.

AUTOMOTIVE ROOF EXAMPLE

In order to demonstrate this approach on a more realistic problem, the UG automotive roof model [17] shown in Figure 8 will be used. A static loading condition representative of global static bending will also be used.

The model is composed of eleven sheet bodies sewn together and meshed using the UG/Scenario default quad mesher. For purposes of demonstration, the roof panel from Figure 8 will be used, shown in Figure 9. The first version of the RPI refinement software can only handle single surfaces. The unrefined mesh of a nominal element size of 50 mm and generated using UG/Scenario/Version 16 can be seen in Figure 10. This model is curved and demonstrates the ability to create refined points on a curved, Parasolids surface. Figures 11 and 12 show the face errors and the vertex errors generated from MSC.Nastran version 70.5. Figures 13 and 14 show two refined meshes at two levels of refinement.

AUTOMOTIVE FRONT STRUCTURE EXAMPLE

Figure 15 shows a second realistic example taken from Ref. 18. The UG parametric model is composed of five solid bodies whose faces were meshed using the UG/Scenario default quad mesher. As in the previous example, only one face--the cap of the shock tower shown in Figure 16--is used for the refinement example. Also, the shock tower cap is a curved shell in order to demonstrate the ability to create new grids on the surface of Unigraphics/Parsolid models. For purposes of demonstration, a vertical, static load applied at the shock tower will be used. Figure 17 shows the unrefined mesh with a nominal element size of 25 mm produced by UG/Scenario Version 16. Figure 18 shows the errors as obtained from MSC.Nastran. Figure 19 shows the resulting mesh refinement.

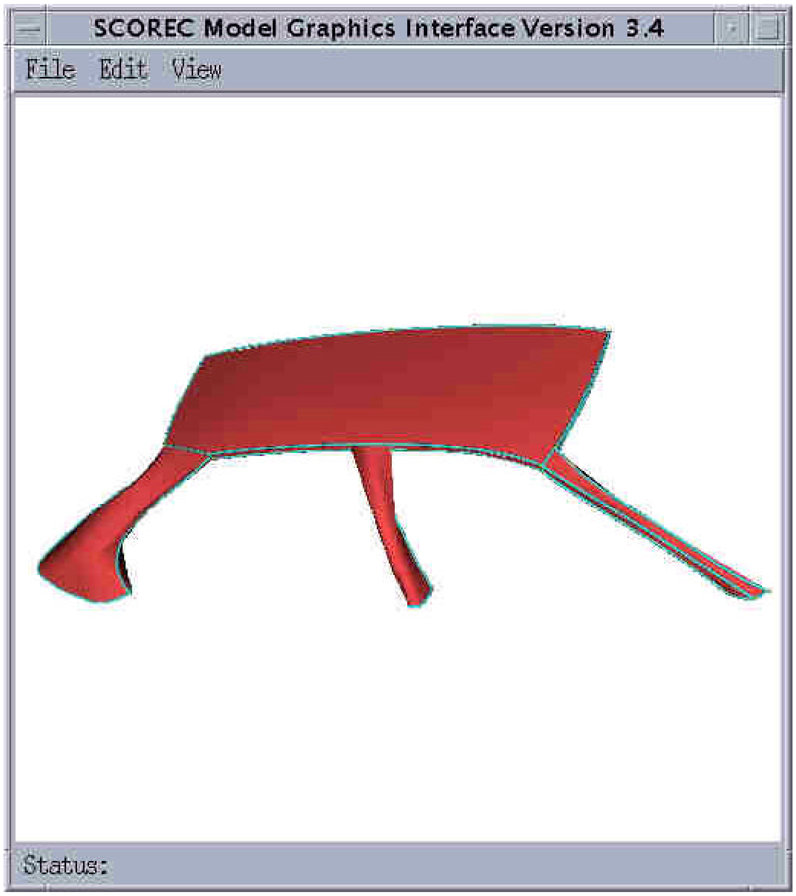


Figure 8 Roof Model

SUMMARY AND CONCLUSIONS

This paper describes a process for adaptive refinement of quadrilateral finite element meshes based upon error measures computed in MSC.Nastran. MSC.Nastran computes assumed errors due to inadequately large element size or improper element size distribution. The error measures are based upon the discontinuity of stresses in the element or at the grid points. The original, unrefined meshes were created from parametric solid models using the UG Scenario structural analysis program. The meshes were composed of nearly-all quadrilaterals using an advancing front method. Several automotive problems were used to demonstrate the software. The resulting refined meshes were composed of well shaped elements which were smoothly graded from coarse to fine. In a future project the software described in this paper will be used to carry out refinement convergence studies to assess the quality of the refined meshes.

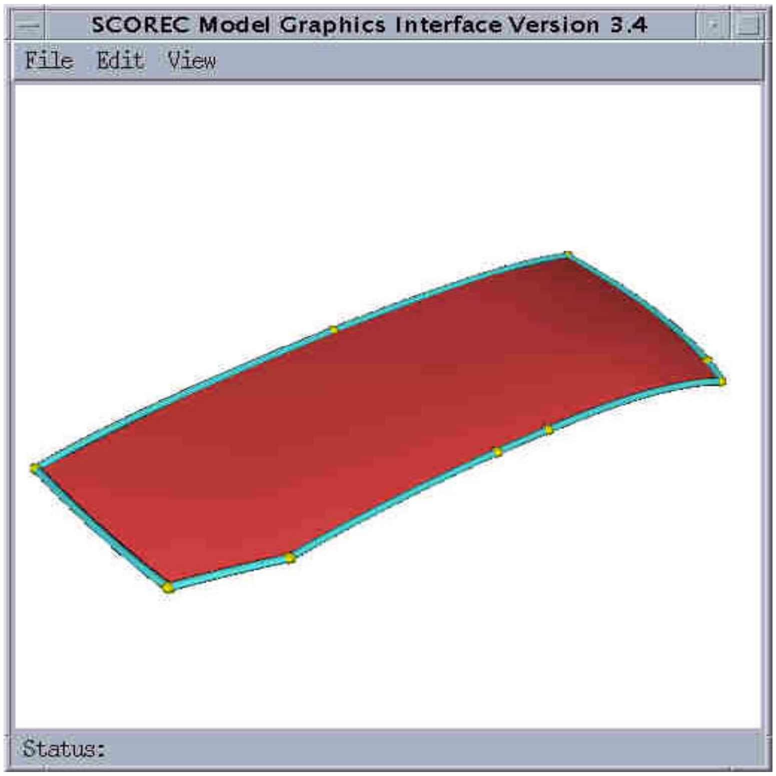


Figure 9 Roof Panel Model

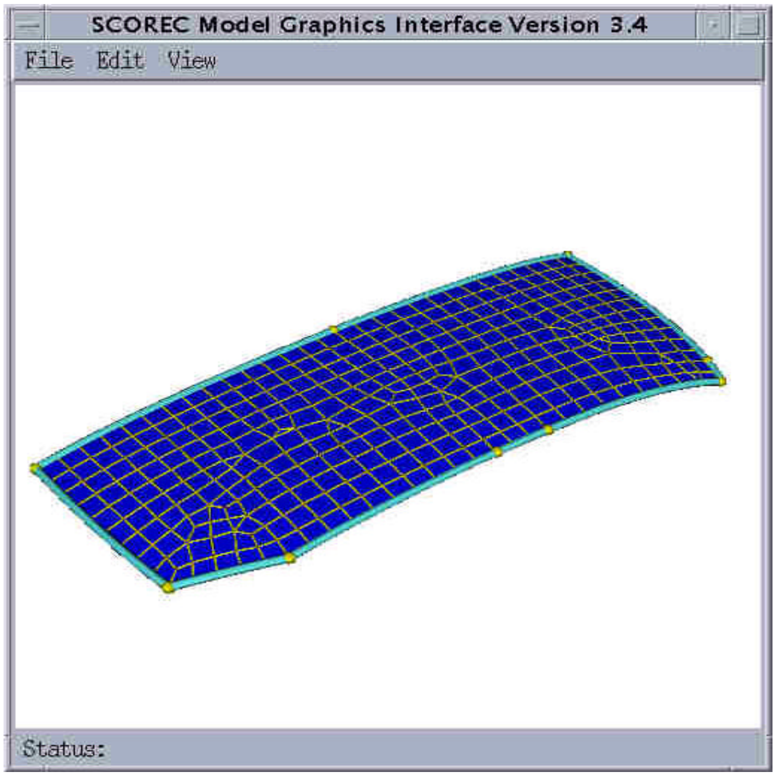


Figure 10 Unrefined Mesh

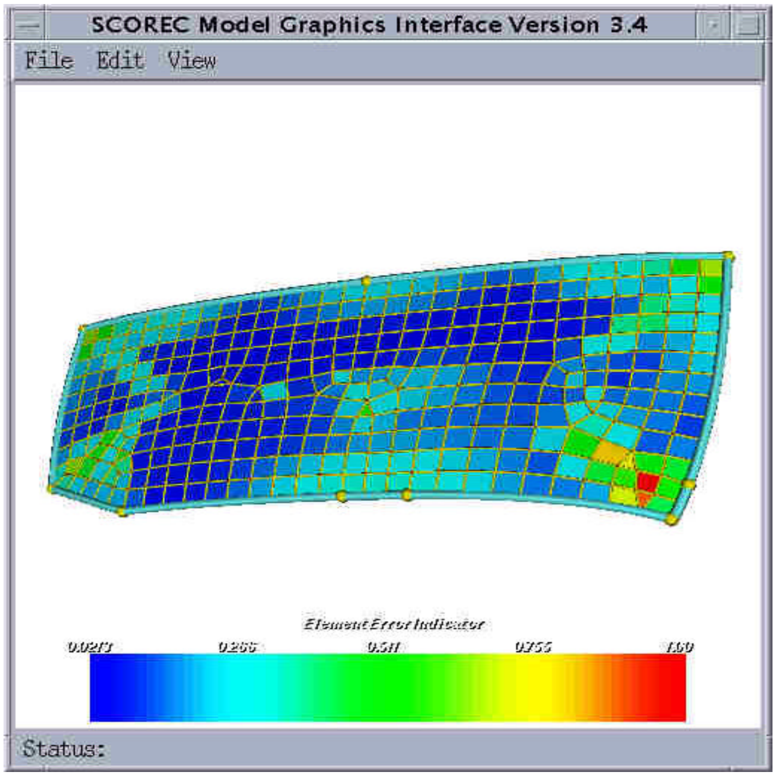


Figure 11 Face Error Measures

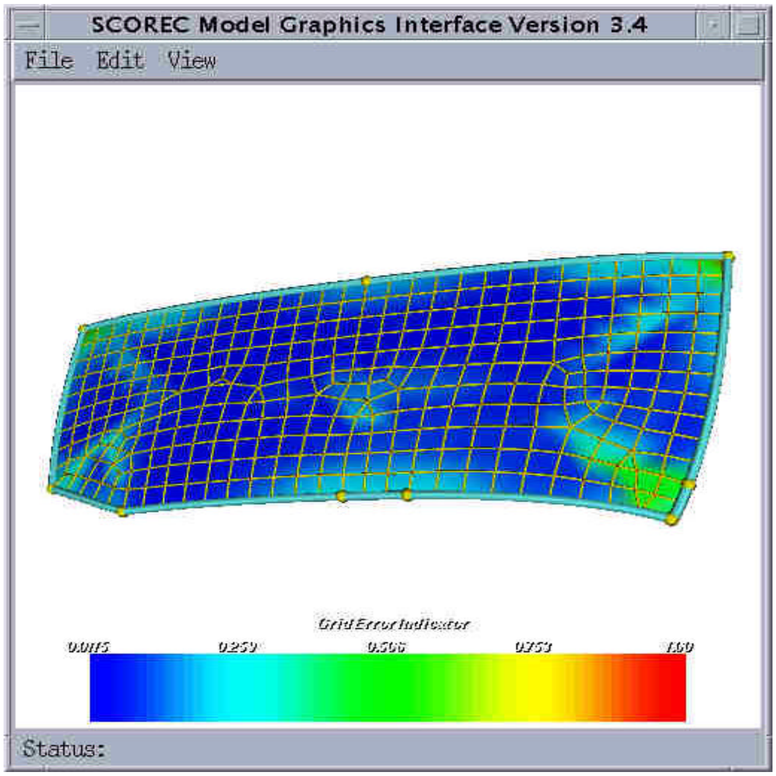


Figure 12 Vertex Error Measures

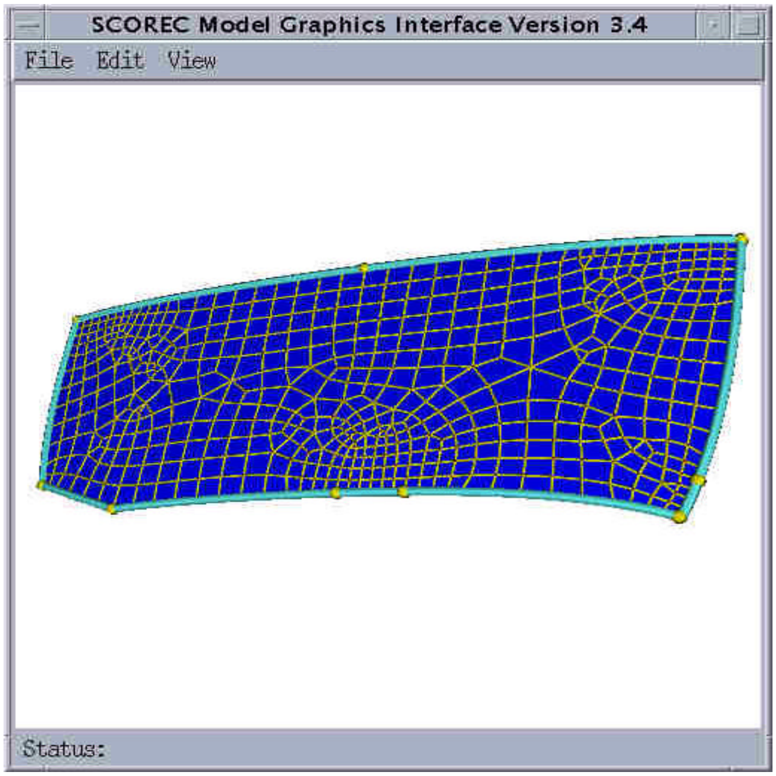


Figure 13 Refined Mesh One

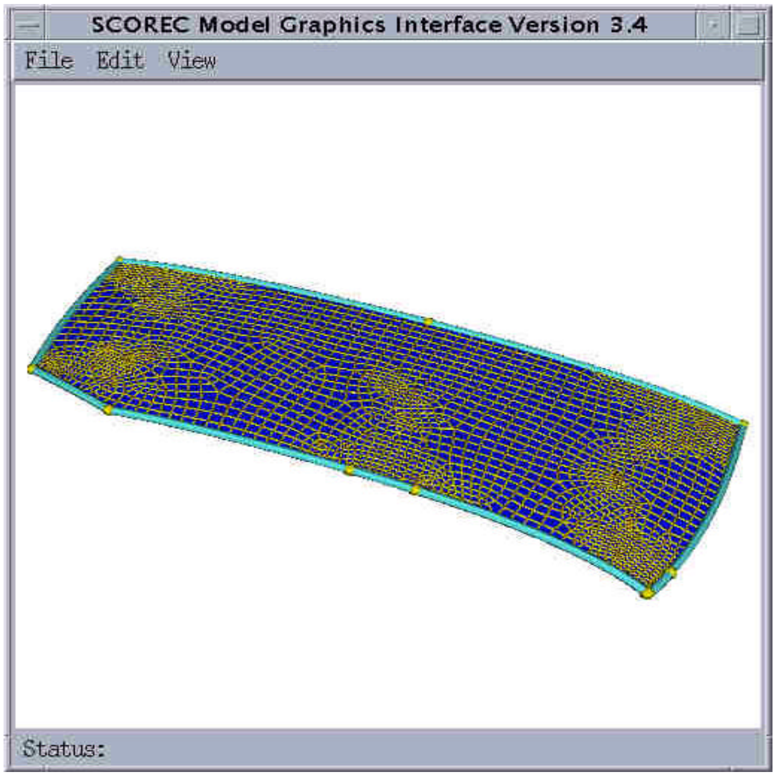


Figure 14 Very Refined Mesh

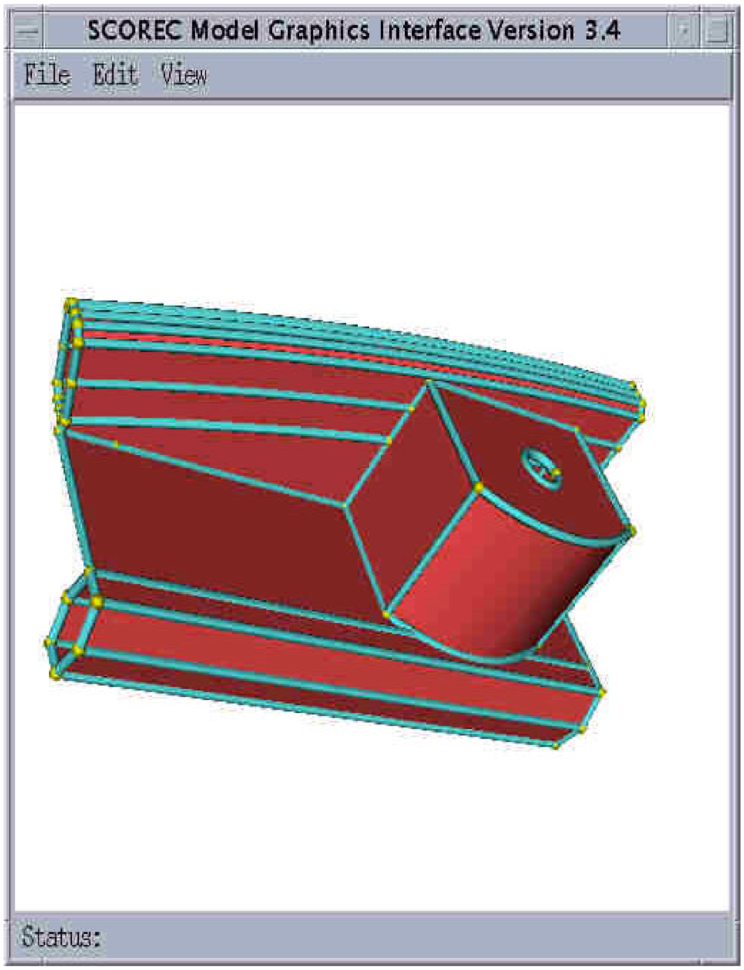


Figure 15 Front End Model

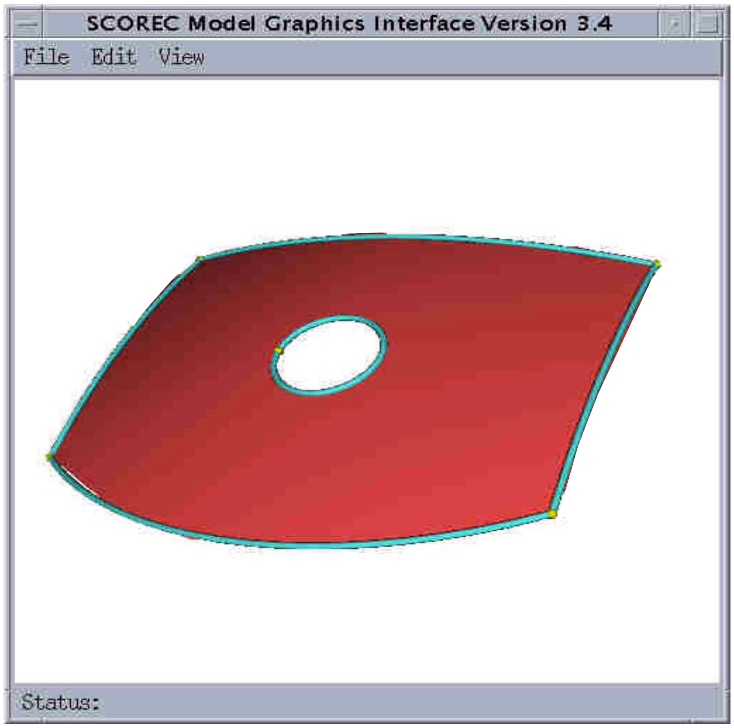


Figure 16 Shock Tower Cap Model

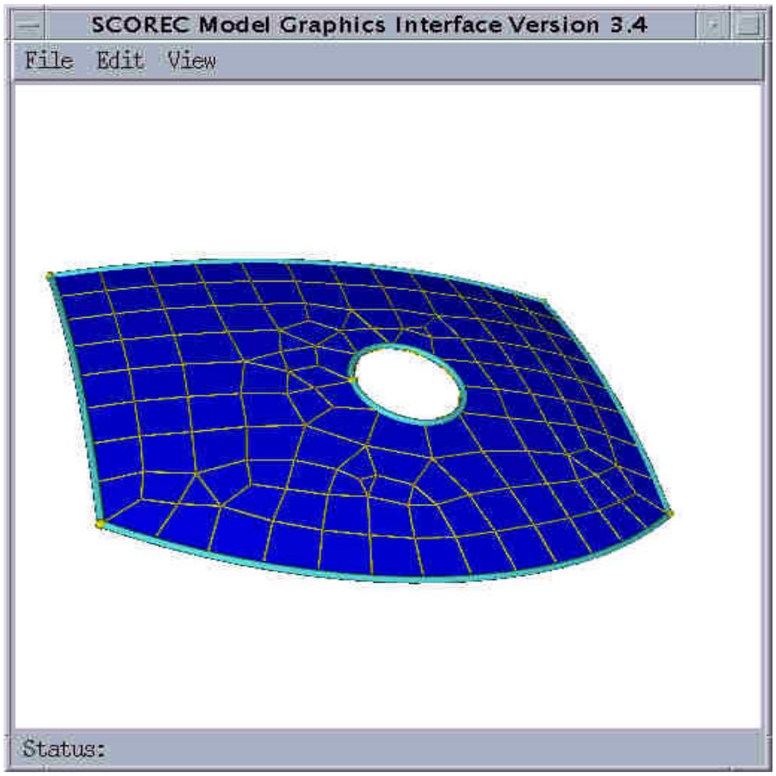


Figure 17 Unrefined Mesh

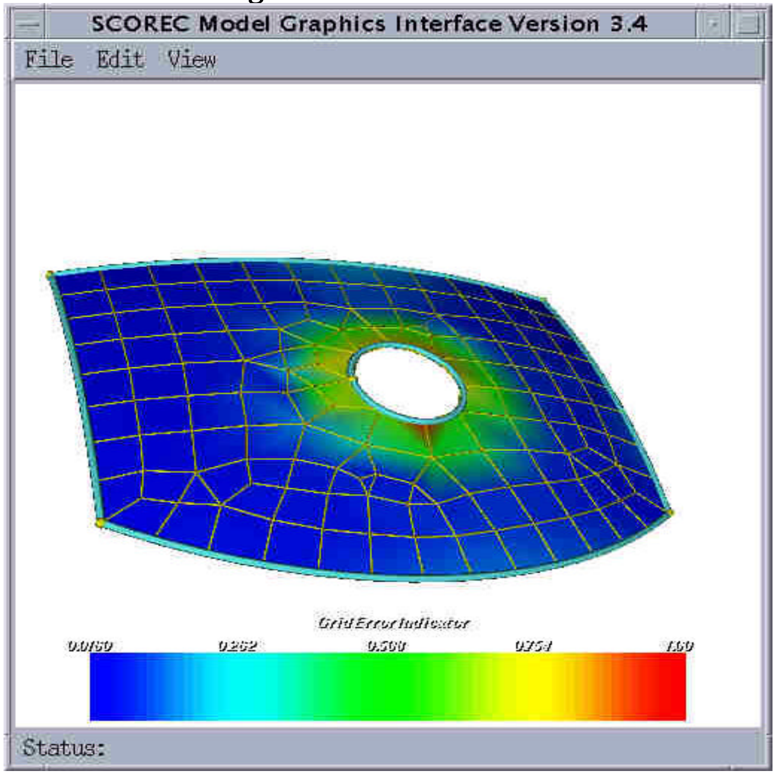


Figure 18 Vertex Error Estimate

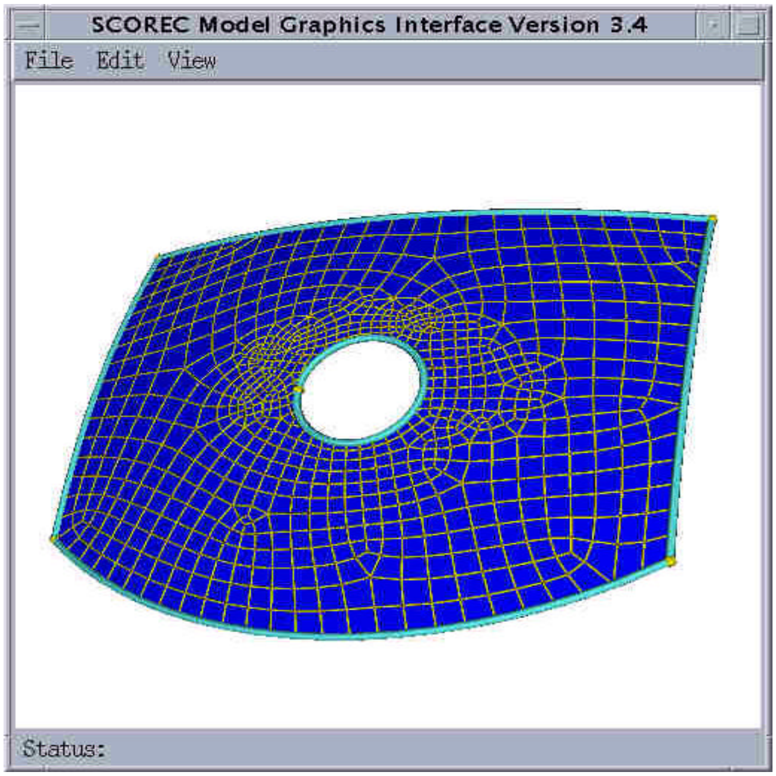


Figure 19 Refined Mesh

REFERENCES

1. Botkin, M.E., “An Adaptive Finite Element Technique for Plate Structures,” AIAA Journal, Vol. 21, No. 5, May 1985, Page 812.
2. Babushka, I. And Rheinbolt, W.D., “Adaptive Approaches and Reliability Estimations in Finite Element Analysis,” *Computer Methods in Applied Mechanics and Engineering*, No 17/18, 1979, pp. 519-540.
3. Zhu, J.Z.; Hinton, E.; Zienkiewicz, C., “Mesh enrichment against mesh regeneration using quadrilateral elements**,”** Communications in Numerical Methods in Engineering, v 9, n 7, Jul 1993,. p 547-554.
4. Zhu, J.Z., Zienkiewicz, O.C., Hinton, E., and Wu, J., “A New Approach to the Development of Automatic Quadrilateral Mesh Generation, “ IJNME, Vol. 32, 849-866, 1991.
5. Lee, C.K. and Hobbs, R. E., “Automatic Adaptive Finite Element Mesh Generation Over Arbitrary Two-dimensional Domain Using Advancing Front Technique,” Computers and Structures, Vol. 71, pp. 9-34, 1999.
6. Lee, C.K. and Hobbs, R. E., “Automatic Adaptive Refinement for Plate Bending Problems Using Reissner-Mindlin Plate Bending Elements,” International Journal for in Numerical Methods in Engineering, Vol. 41, pp. 1-63, 1998.
7. Kettil, P. and Wiberg, N.-E., “Adaptive FE-analysis of Column Supported Reissner-Mindlin Plate,” Computers and Structures, Vol. 72, pp. 605-626, 1999.
8. Diez, P. and Huerta, A., "A Unified Approach to Remeshing Strategies for Finite Element *h*-Adaptivity," Comput. Methods Appl. Mech. Engrg., vol. 176, (1999), p 215-229.
9. Blacker, T.D. and Stevenson, M. B., *“Paving: A new approach to automatic quadrilateral meshing technique*,” Advances in Engineering Software, 13(5-6):811-847, 1991.
10. Owen, S.J., “A Survey of Unstructured Mesh Generation Technology,” The 7th International Meshing Roundtable, Dearborn, Michigan, October 26 - 28, 1998, p. 239-267.
11. Zienkiewicz, O.C and Zhu, J.Z., "A simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis," Int. J. Numer. Methods Eng., Vol. 24, p. 337-357, 1987.
12. "MSC.Nastran V68 Reference Manual," The MSC.Nastran Encyclopedia On-line Documentation, The MacNeal-Schwendler Corporation, Los Angeles, California..
13. B. Kaan Karamete, Rao V. Garimella and Mark S. Shephard, "Edge Recovery Method on an Existing Surface Mesh for Boundary Layer Meshing", Fifth US National Congress on Computational Mechanics, 2nd Symposium on Trends in Unstructured Mesh Generation August 4-6, 1999, University of Colorado, Boulder, CO, USA.
14. S.J. Owen, M. L. Staten, S. A. Canann and S. Saigal, "Q-Morph: An Indirect Approach to Advancing Front Quad Meshing", Int. J. Numer. Meth. Engng., 44, 1317-1340 (1999).
15. C. K. Lee and S. H. Lo, "A New Scheme for the Generation of a Graded Quadrilateral Mesh", Comput. Struct., 52, 847-857 (1994).
16. P. Kinney, "Clean Up: Improving Quadrilateral Finite Element Meshes", Proc. 6th Int. Meshing Roundtable, 1995, 449-461.
17. Botkin, M.E., “Modeling and Optimal Design of a Carbon Fiber Reinforced Composite Automotive Roof,” International Journal of Engineering with Computers, Vol. 16, pp. 16-23, 2000.
18. Botkin, M. E., " Structural Optimization of Automotive Body Components Based Upon Parametric Solid Modeling," to be presented at the AIAA Multidisciplinary Design Analysis & Optimization Conference, Long Beach, California, Sept. 2000.