Mesh Optimization Using a Genetic Algorithm to Control Mesh Creation Parameters

Jeremy P. Dittmer[1], C. Greg Jensen[1], Michael Gottschalk[3] and Thomas Almy[4]

[1]Stanford University, jdittmer@stanford.edu
[2]Brigham Young University, cjencsen@byu.edu
[3]Pratt & Whitney, michael.gottschalk@pw.utc.com
[4]Pratt & Whitney, thomas.almy@pw.utc.com

ABSTRACT

High-quality FEA mesh generation is often a time intensive and costly process within the current iterative parametric product design cycle. A mesh optimization scheme is proposed and developed in which mesh creation parameters are optimized using an evolutionary algorithm. A mesh quality metric is proposed and used to study the effectiveness of the various mesh creation parameters and, in turn, to determine which should be used as design variables of the optimization problem. A genetic algorithm is employed to iteratively remesh 3-D surfaces and tend toward an optimal mesh based on the quality metric. The optimization approach was implemented on two components of Pratt & Whitney aircraft engine hardware. Results indicate that this approach is effective in generating high-quality meshes without altering the mesher itself, developing new meshing algorithms or post-processing meshes to improve element quality.

Keywords: mesh optimization, genetic algorithm, ANSYS

1. INTRODUCTION

Many modern computer-aided modeling, analysis, and manufacturing systems (CAx) provide both interactive and automatic finite element mesh generation of surface and solid entities that describe the parts or products being virtually engineered as new designs. Unfortunately for complex products, the interactive approach is too time consuming to factor into the design process and the quality of automatically created meshes often does not meet engineers' criteria for element shape and density. This is especially true in the case of hexahedral meshes that are useful in simplifying and improving the accuracy of finite element models. Huang et al. states that existing mesh generators available in commercial software are incapable of automatically producing high-quality hexahedral meshes [1]. Though commercial finite element analysis (FEA) packages have some ability to control and direct the automatic mesh generation process, determining a correlation between these user controlled mesh parameters and acceptable quality of the generated mesh is difficult if not impossible. Since the validity of analysis results is heavily dependent upon mesh quality, obtaining better meshes in the shortest amount of time is essential for the integration of FEA into the automated design process.

Several approaches are presently employed to improve the quality of automatically generated surface meshes. For well over two decades researchers have focused their efforts on the development of more reliable and robust algorithms for mesh generation. These algorithms can be grouped into three general categories of interest: rapid and effective open-loop techniques (e.g., the paving and Q-morph algorithms [2]), intelligent mesh generators which use objective functions and small-scale optimization techniques [3], and mesh generators that use optimization as the basis for mesh creation [4-5]. The latter utilize genetic algorithms successfully for optimal placement of mesh nodes. When utilizing new (research) mesh creation algorithms is not an option, another technique that is frequently employed in industry is that of modifying existing nodes and elements. Mesh smoothing routines have likewise long been an effective method of improving mesh quality in a pre-existing mesh. Many techniques are available for performing mesh smoothing. Some of the more advanced ones use gradient-based optimization techniques to quickly determine the optimal distribution of existing nodes [6-9]. Others iterate using brute-force methods, such as Laplacian smoothing, to improve mesh distribution and corresponding element quality [10-11]. Beyond this geometrical optimization of element shape, some schemes have been developed to modify and optimize the topology of the mesh by editing the node-adjacency structure of the mesh. Routines and optimizers include methods and operators such as edge swapping, vertex removing, edge collapsing, etc. to edit and improve the mesh topology [8-9],[11-13]. Special operators are required for maintaining a valid mesh in the case of quadrilateral and hexahedral meshes [14]. Still other mesh improvement methods involve generating a new mesh based on information learned from previous attempts. Several algorithms use *a posteriori* techniques to improve the mesh once regions of inaccuracy are located [10],[12],[15].

Although the above techniques have undoubtedly improved the quality of meshes available to the mesh researcher, the accessibility to such techniques within commercial FEA is still limited. It is a well accepted fact that it takes software companies years to adopt and dispense new methods and techniques that have been discovered, developed and tested first by researchers. Often the effectiveness of their implementation is called into question: smoothing algorithms, for example, are restricted by the node/element configuration of the starting mesh and may not be able to improve a mesh to meet the desired criteria. This paper proposes a new method for automatically generating an optimal mesh within the framework of existing FEA software, using some of the principles explained above. Rather than optimizing initial node placement or operations to be performed on existing elements, the mesh control parameters available in a commercial FEA package (like ANSYS, HyperWorks, Patran, etc.) can be optimized to yield a high-quality mesh. Mesh quality evaluation metrics are derived on a case by case basis to ensure that the optimizer produces results that satisfy the designer's criteria. The method is designed specifically for optimizing quadrilateral meshes on area geometry. This has application to 2D problems or 3D problems in which hexahedral elements are generated by extruding a quadrilateral mesh.

This paper should serve as a roadmap for applying our methods to other mesh structures. Two practical examples of how our methods provide high-quality hexahedral meshing of jet engine components, i.e. a single vane blade and a much more complex hollow fan blade, are given. The method presented is described in terms of ANSYS commands and procedures, however, the reader should realize that any full function FEA preprocessor will have similar commands and procedures buried deep in their documentation. There are two important contribution made by this paper. First, a clear demonstration to the FEA research community that commercial codes are presently capable of high quality meshes, if used properly, and second the automated approach to mesh refinement within commercial FEA codes. Limitations of our methods are discussed followed by our recommendations and conclusions.

2. MESH QUALITY METRIC

The first step in developing an optimization tool for mesh generation was to determine a metric for measuring mesh quality. This mesh scoring system was developed based on the following objectives:

(1) Maximize element quality throughout the mesh
(2) Maximize adherence to a desirable mesh pattern
(3) Create a mesh without any triangles
(4) Minimize element count
(5) Other necessary mesh evaluation criteria

The importance of these objectives and methods used for quantitative evaluation of each are described in the following paragraphs.

Maximize element quality (throughout the mesh): Individual element quality directly determines the overall mesh quality. Element quality across a given mesh is evaluated by extracting results from shape testing that ANSYS performs while generating the mesh. Three of these tests were selected for use in quantifying the shape quality of a given element [16-17]:

(1) Aspect ratio (ideal is 1)
(2) Maximum corner angle (ideal is 90°)
(3) Maximum parallel deviation (ideal is 0°).

Macros were written to calculate the mean of each of these statistics for all elements in a given mesh as well as generating an average deviation score. This was done to ensure that mesh quality scores reflected both high quality elements and consistency from element to element across the mesh. The deviation score is calculated as follows:

$$\frac{1}{N}\sum_{i=1}^{N}\left|x_i-\overline{x}\right| \tag{1}$$

where N is the number of elements, x_i is the value of a statistic for an individual element, and $\overline{x}$ is the mean for that statistic. This deviation was used instead of the standard deviation to reduce the number of multiplication and square root operations required (ANSYS macros run slowly – the macro language, APDL, is interpreted – and any speed enhancement is valuable). The six values (one average and one deviation for each of the three shape tests) are then scaled so that they will have comparable magnitudes. To correctly weight each value, observations were made regarding typical results for each shape test score. The macro then transforms each so that a perfect result will yield zero and an average result will be on the order of one. This is accomplished using the following calculation:

$$\frac{score_i - perfect\ \ score}{average\ \ score - perfect\ \ score} \quad (2)$$

Finally, the six values are summed to yield one value which represents the element quality score for the mesh. It should be noted that the metric is designed so that smaller values represent better meshes and will only produce positive values.

Maximize adherence to a desirable mesh pattern: Characteristics of a desirable mesh pattern vary with varying area topology. For example, a frequently sought-after attribute of a high quality mesh is for elements of the mesh to be arranged in rows and columns as illustrated in Fig. 1 below. Further examples will be provided in the results section but in general, attributes of a desirable mesh pattern are noted, and methods are developed to quantitatively evaluate these attributes. The methods for the quantitative evaluation of some mesh pattern attributes will be explained as they pertain to the examples provided.

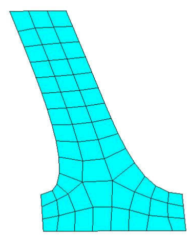


Fig. 1. Desirable mesh pattern with most elements contained in rows and columns.

Create a mesh without any triangles: Triangles are excessively stiff and degrade the quality of the mesh. The triangle score is calculated simply on the basis of whether or not there are triangles present (there will either be one or zero triangles). No triangles yields a score of zero, the presence of a triangle produces a score of one.

Minimize element count: Reducing the element count reduces the time required to solve the finite element model. This is favorable, especially when the model itself is being optimized and needs to be meshed and solved repeatedly. To minimize element count in the area whose mesh is being optimized, a score for element count is calculated. The method for calculating this score may be exclusive to specific geometry when enough information is available about the area to know how many elements should be present, but we have also developed a generic method:

$$Element\ \ count\ \ score = \frac{N}{A/S^2} \quad (3)$$

where N is the actual number of elements present in the mesh, A is the surface area of the area being meshed and S is the target element size for the meshing operation. The latter is a user defined value in ANSYS that is the desired edge size for elements created by the mesher. Thus, the denominator of the above equation represents the number of square elements of the desired size that would be required to cover the entire area (the desired number of elements), and the entire quotient is the ratio of actual element count to desired element count. Again, a smaller value represents a better score. Though this measure has limitations in its objectivity when applied to meshes with a small number of elements or to areas with irregular shapes, the denominator should give some degree of normalization to the quantity and most importantly, will still accurately reflect the relative quality of successive meshes in terms of element count.

Other necessary mesh evaluation criteria: Specific mesh optimization problems may require additional mesh quality metrics. One example is that of generating meshes that have a fixed number of elements across critical regions. For instance, the narrow vertical region of the mesh in Fig. 1 consistently has three elements across its width where the mesh in Fig. 2 below has three elements across at the top, but quickly reduces to one element representing the width. Whether the desired element count across such a region is one, two, three, or more, providing a consistent element thickness is critical. In this case a metric would be required to quantify how well a mesh maintained element thickness in critical regions (such criteria were in fact developed and successfully employed in obtaining optimal meshes with constant element thicknesses in the specified regions). The hollow fan blade example included below provides an example requiring even more custom metrics.

All of these objective scores are weighted and combined to form one score for a given mesh. The weighting constants are tuned on a case by case basis so that this overall mesh score accurately reflects quality of mesh for the given problem. In general, mesh pattern is the most important, followed by element quality, followed by element count. As indicated above, lower score for each metric represent better meshes. Therefore, the mesh optimization problem becomes the numerical optimization problem of minimizing the mesh score.

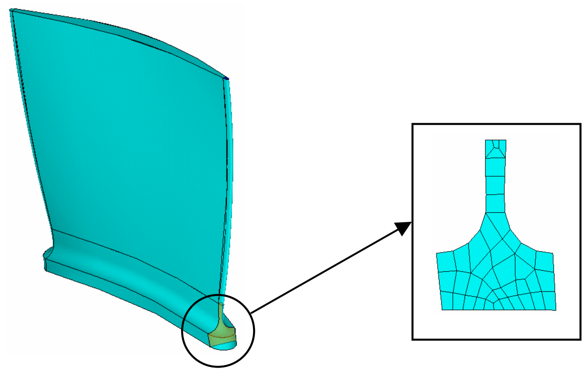


Fig. 2. Test area geometry for design variable study.

3. DESIGN VARIABLES

Once a mesh quality metric had been developed, it was possible to study the set of mesh control parameters to determine which had the greatest effect on controlling and driving the mesh towards an optimum. In ANSYS, it was determined experimentally that the SmartSize meshing algorithm provided the greatest flexibility in generating meshes. SmartSizing provides the user with a wide range of settings and is designed to “give the mesher a better chance of creating reasonably shaped elements during automatic mesh generation.” (from ANSYS help). In mesh optimization terms, this corresponded to a large design space and greater probability of locating an optimal mesh. The effect and interaction of six potentially significant mesh parameters were investigated. A list of these parameters along with their descriptions from ANSYS help follows.

FAC – Scaling factor applied to the computed default mesh sizing.

EXPND – Mesh expansion (or contraction) factor. EXPND is used to size internal elements in an area based on the size of the elements on the area's boundaries. For example, applying a value of 2 to this setting will allow a mesh with elements that are approximately twice as large in the interior of an area as they are on the boundary.

TRANS –Mesh transition factor. TRANS is used to control how rapidly elements are permitted to change in size from the boundary to the interior of an area. A TRANS value of 2.0 would permit elements to approximately double in size from one element to the next as they approach the interior of the area.

ANGL – Maximum spanned angle per lower-order element for curved lines.

GRATIO – Allowable growth ratio used for proximity checking.

ESIZE – Default element edge length on surface boundaries.

An exhaustive search of the design space for every combination of two parameters was performed on test geometry provided by Pratt & Whitney to determine which parameters interacted with each other and which were most significant in generating high quality meshes. An area similar to the one shown in the Fig. 2 above was used for the tests. This area was selected because it has historically proven difficult to mesh well and accordingly was a likely candidate for mesh optimization.

The mesh quality scoring routines described above were used to perform this study. The mesh scores generated were scaled to gray levels from white to black (see Fig. 3 below) which corresponded to the minimum and maximum mesh scores in a given design space. The most desirable meshes are those with the lowest scores and are indicated by lighter shades of gray.

The results of the exploration of the various design spaces is contained in the matrix below (Fig. 4). The ranges investigated for each parameter are indicated in the matrix. As noted, the ESIZE value used is a percent above or below a nominal value calculated from the physical geometry of the area being meshed (a smallest edge of the area, for example). Parameters that were not varied were set to values that were previously known to generate meshes of acceptable quality. Plots are placed in the chart so that the plot axes correspond to chart location. For example, the interaction plot of EXPND and FAC is placed in the EXPND column and the FAC row – accordingly, on the plot, EXPND varies in the x-direction while FAC varies in the y-direction. Plotted values increase in the up and right directions. Other data included in each cell are the minimum and maximum mesh scores found in the given design space and comments about parameter interaction and effectiveness.

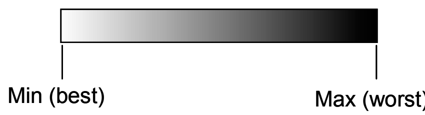


Fig. 3. Intensity-based scaling of mesh scores.

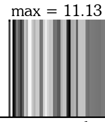
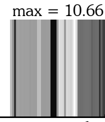
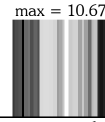
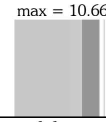
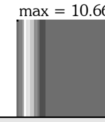
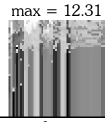
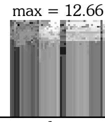
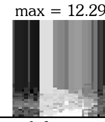
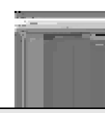
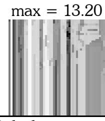
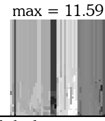
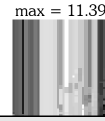
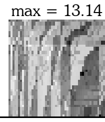
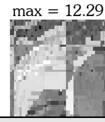
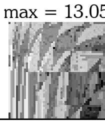
	ESIZE 67-150%	**GRATIO 1.3-2.0**	**ANGL 20-50**	**TRANS 1.3-2.0**	**EXPND 0.6-3.9**
FAC 0.2-2.0	**esize dominates** min = 5.71 max = 11.13 	**gratio dominates** min = 6.22 max = 10.66 	**angl dominates** min = 6.05 max = 10.67 	**trans dominates** min = 6.12 max = 10.66 	**expnd dominates** min = 6.08 max = 10.66 
EXPND 0.6-3.9	**late interaction by expnd** **esize dominates early** min = 5.66 max = 12.31 	**late interaction by expnd** **gratio dominates early** min = 5.91 max = 12.66 	**early interaction by expnd** **angl dominates late** min = 5.85 max = 12.29 	**expnd dominates** min = 6.05 max = 10.66 	
TRANS 1.3-2.0	**esize dominates** min = 5.71 max = 13.20 	**gratio dominates** min = 6.12 max = 11.59 	**angl dominates** min = 6.02 max = 11.39 		
ANGL 20-50	**global interaction** min = 5.71 max = 13.14 	**global interaction** min = 6.00 max = 12.29 			
GRATIO 1.3-2.0	**global interaction** min = 5.63 max = 13.05 				

Fig. 4. Parameter interaction results matrix.

The results clearly show that several of the parameters have very little effect on the mesh quality. This is indicated by plots dominated by vertical or horizontal lines. For example, when FAC and GRATIO are explored together, the horizontal bars indicate that the mesh is virtually unchanged by changing values in FAC. As GRATIO varies, however, significant changes in mesh quality are observed. Three parameters have significant interaction and have been chosen as the three most useful ones to control the mesh. These are ANGL, GRATIO, and ESIZE. The interactive plots of these three parameters, though noisy, indicate distinct patterns suggesting that these are the parameters that should be controlled by an optimization algorithm to identify an optimal mesh.

4. OPTIMIZATION APPROACH

Based on the noise and discontinuities in the interaction plots shown in Fig. 4 above, it was determined that a gradient based optimization algorithm would not be suitable for this problem. Instead, a genetic algorithm was chosen to deal with the non-smooth design space, handle the multiple-objective nature of this optimization problem and explore the extremities of the design space. C code and an ANSYS macro were written that interacted with each other via data files to automate the optimization of the mesh. A diagram of this interaction is shown below.

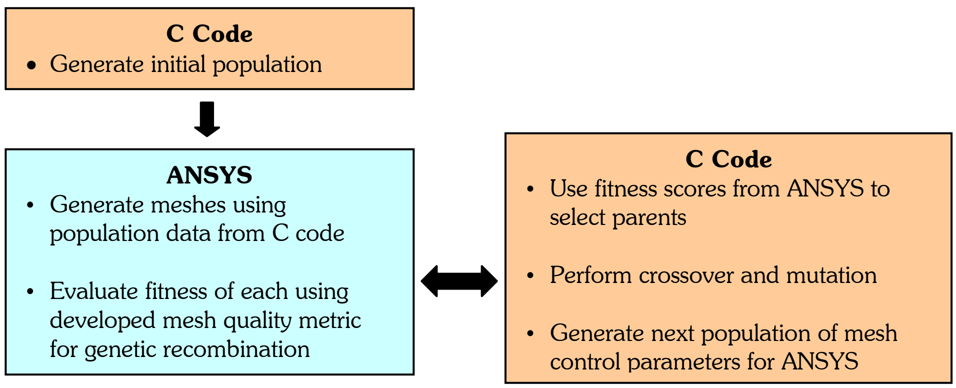


Fig. 5. Diagram of ANSYS macro and C code interaction.

In implementing the genetic algorithm, a number of design decisions were made to control the speed to solution and quality of optimum found:

Bit depth: 8 bits were used for each design variable to code the chromosomes in binary. The code is built so that this can be changed if desired, but 8 bits per variable proved sufficient to consistently generate good results (3x8 bits corresponds to a design space of 16,777,216 parameter sets).

Elitism: Generational elitism was applied so that good genetic material from previous generations is not discarded inadvertently. Each generation of children was compared to the generation of parents and the best of the group is selected to be parents for the generation to follow. This method of elitism was used to promote quick convergence to an optimum – mesh optimization may be only a part of a larger optimization problem so finding an optimized mesh quickly is important. Additionally, the best mesh so far is saved as the optimization progresses so it can be restored at any time if necessary.

Roulette wheel selection: Roulette wheel selection or tournament selection can be used in choosing parents for the next generation by passing an argument to the C code. Experimentally, it was determined that roulette wheel selection converged more quickly.

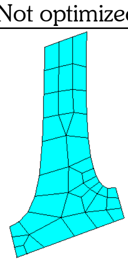
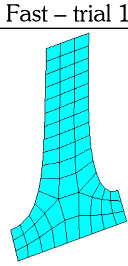
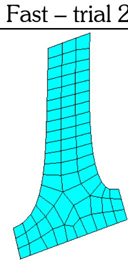
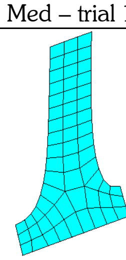
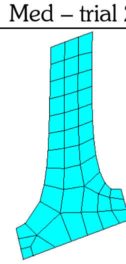
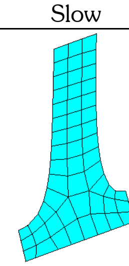
Reverse lookup: The C code maintains a history of all parameter sets used and compares each new child to this list. If the parameter set is a repeat (a clone), the set is tagged so that ANSYS doesn't have to regenerate the mesh for these parameters – it just uses the fitness value from the history. The meshing in ANSYS and evaluation of the mesh quality metric are the most time-consuming parts of the mesh optimization process so this approach can potentially save a lot of time. The actual amount of time saved depends on the mutation probability – a higher mutation probability reduces the chances that a clone will be generated.

To create a flexible tool that can accommodate varying user needs, the remaining genetic algorithm control values are allowed to vary based on user input. Rather than requiring the user to control each of these, groups of values were chosen and assigned to slow, medium, and fast settings. The fast setting will find an optimal mesh, but this mesh may or may not be the global optimum. Though not guaranteed to locate the global optimum, probability of doing so is greatest using the slow setting. Tab. 1 contains the descriptions of the remaining parameters and values.

Name and description	Fast	Medium	Slow
Initial population size – To encourage exploration of the entire design space, the initial population size can be controlled independently. This allows the user to specify a larger first population to cover as much of the design space as possible (start with a large amount of genetic material), then reduce to a smaller population size for subsequent generations (increase optimization speed).	20	40	60
Subsequent population size	10	20	30
Number of generations – The algorithm will terminate if the maximum number of generations have been generated and evaluated.	20	20	20
Crossover probability	1.00	1.00	0.80
Mutation probability	0.001	0.001	0.05
Convergence Criteria – To see if the optimization run has converged, the C code sorts the entire history and checks the best *n* values to see if they are within a numerical tolerance. Clones are not counted.	5 values within 0.1	5 values within 0.1	7 values within 0.1
Maximum iterations – To ensure that the optimization process executes in a timely fashion, a maximum number of iterations is assigned. Once the actual number of meshes generated exceeds this value, the macro restores the best value it has found so far whether or not the convergence criteria have been satisfied.	50	90	400

Tab. 1. Variable genetic algorithm control settings.

As can be seen, convergence is accelerated in the fast setting by reducing mutations, maximizing crossover, and reducing generation sizes. The opposite settings slow the optimization, but allow the algorithm to more fully explore the design space. This is illustrated in Tab. 2 below which contains the results of tests performed on geometry like that depicted in Fig. 2 above. The number of iterations required to achieve the result and best score are included along with pictures of the meshes.

Not optimized	Fast – trial 1	Fast – trial 2	Med – trial 1	Med – trial 2	Slow
 8.875	 50 iterations 1.259	 50 iterations 2.277	 96 iterations 0.940	 80 iterations 1.882	 280 iterations 0.813

Tab. 2. Effect of speed control settings on mesh results.

Some features of the data presented in this table deserve further comment. Most importantly, there is not a clear improvement in mesh quality from the quick solution to the slow and robust one, at least not from a visual inspection. Furthermore, these dubiously improved meshes require many more iterations to generate. One explanation for these irregularities lies in the mesh scoring system – on close inspection, it is clear that the mesh generated using the slow setting has fewer elements and consequently, a better mesh score, even though the mesh pattern is very similar.

Remaining irregularities can be explained by the nature of the genetic algorithm. The genetic algorithm has a tendency to lock on to a local optimum, rather than seeking out a global one. The local optimum that is found depends on multiple stochastic processes – the initial generation created by the algorithm and genetic recombination to produce subsequent generations. A better local optimum may be found when, through random mutation, a better solution is identified. For this reason, the fast settings are designed with a small mutation probability so they will quickly drive the solution to a local optimum. The slow setting has several features that increase its likelihood of identifying more of a global optimum. Its initial population is larger and so it has a better chance of randomly selecting a better design on the first round. Furthermore, it has a significantly higher mutation probability and is therefore more likely to locate better local optima in successive generations (of which more are allowed). Choosing between settings is therefore left to the user – those who wish to minimize the time required to arrive at an optimal mesh can use the fast setting knowing that the algorithm will quickly settle on a local optimum. It may not be the very best mesh, but often will be good enough. For the user that would like to have as high fidelity as possible and is not concerned about the process taking a little longer, the slow option provides an ideal approach that will more thoroughly explore the design space.

5. RESULTS AND DISCUSSION

The optimization method described above has proved effective in solving a number of meshing problems for Pratt & Whitney, the organization for which the process was developed. Two such problems, a cross section of an airfoil from a stator and a plate model of a hollow fan blade will be described in detail.

5.1 Airfoil Fillet

Pratt & Whitney has developed automated meshing tools which produce meshes dominated by hexahedral (brick) elements producing better quality with fewer elements compared to tetrahedral elements. A technique often used for generating bricks in a mesh is to seed a two-dimensional cross section of a volume with constant cross sectional topology and sweep the mesh pattern through the volume. In these cases, the quality of the three dimensional mesh depends directly on the quality of the seed mesh. Of particular interest, is the fillet region where critical load stresses are often found and accurate results are essential. The fillet cross section and an example of a typically poor cross sectional mesh are shown in Fig. 2 above. Empirical studies proved that no fixed settings could be applied to the ANSYS mesher to consistently generate good meshes in these areas making it an ideal candidate for mesh optimization.

5.1.1 Mesh Pattern Metric

As mentioned above, specific methods were developed to quantify adherence to a desirable mesh pattern. In the case of the airfoil fillet, the most important characteristic of a desirable mesh pattern is that it maintains a constant number of elements through the narrow region of the area, which is really part of the blade rather than the fillet (in Fig. 5 above, the element count through the thickness quickly reduces from three to one). Less important, though still present in an ideal mesh are rows and columns of elements persisting inward from the short line segments on the sides of the area (See Fig. 1 for an ideal mesh pattern). These two factors are combined together according to the following equation:

$$\left(1-\frac{E_3}{E_2}\right)+P \tag{4}$$

where E_1 is the subset of elements which are in rows and columns in the blade region of the area, E_2 is all elements of the region which are not part of E_1 and E_3 is a subset E_2 consisting of elements that are in rows and columns starting at the short, lateral line segments. P is a penalty value of 2 applied when E_1 elements extend less than 75% of the way through the blade region.

5.1.2 Results

Table 2 above shows the effectiveness of using this mesh optimization technique for the airfoil fillet problem. The optimization tests were performed on a Pentium 4 class PC and required less than one minute for the fast trials to several minutes for the slow trials, compared to several hours to over a day to achieve comparable results manually. These significant time savings and hands-off operation make the technique exceptionally valuable – especially in optimizations of assemblies which might require hundreds of repeated meshes.

5.2 Hollow Fan Blade

To reduce weight and associated stresses on engine components, Pratt & Whitney has developed hollow fan blades for its engines. These blades consist of a system of hollow channels bounded by a solid region and enclosed in thin metal skins as shown the cutaway view in Fig. 6 below. This model is a demonstration model provided by Jason Elliott of the hollow fan blade group at Pratt & Whitney.

Given the many thin surfaces of the hollow fan blade design, plate elements with thicknesses are used rather than 3D elements to reduce the element count. Since these models are required for very long transient analyses such as bird impact tests, low element count is of the utmost importance. To date, similar hollow fan blade geometry has been meshed painstakingly by hand requiring an immense investment of man-hours. If mesh optimization proved applicable to the individual areas of the model – specifically for the unpredictably shaped blade surface areas (also called skins and defined by the cavity walls) and solidity – it would yield an automated, high-fidelity mesh of the entire hollow fan blade geometry.

5.2.1 Mesh Pattern Metric

Since the topology of each skin area is different, the only mesh pattern score used was one that calculated the percentage of elements in rows and columns. This was done according to the following calculation:

$$1-\frac{I+E}{N} \quad (5)$$

where I represents the number of interior nodes that are vertices for four elements, E is the number of edge nodes that are vertices for two elements, and N is the total number of nodes in the mesh.

5.2.2 Additional Mesh Quality Score

Another metric was calculated to ensure that element edge sizes did not exceed the design specification of 0.3. A macro was developed to incur a penalty for each oversized element. This proved to be an important addition to our method and forced the genetic algorithm to be predisposed to selecting chromosomes with edges shorter than 0.3. Fig. 7 below illustrates the success of the genetic mesh optimization approach for the hollow fan blade model.

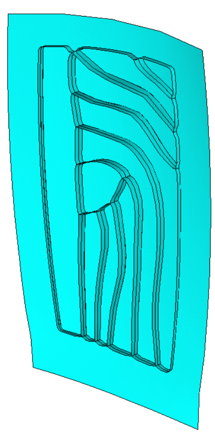


Fig. 6. Hollow fan blade cutaway view.

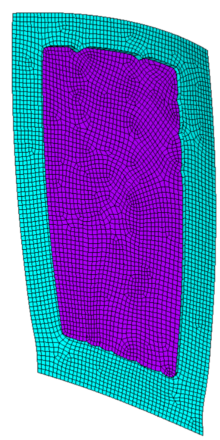


Fig. 7. Fully meshed hollow fan blade shell model.

Though the optimization process can be fairly time-consuming (almost one hour for this blade and 6 hours for actual proprietary hollow fan blade geometry), the process required no user interaction and generated a high-fidelity mesh of very consistent element size containing only 7,500 quadrilateral elements (the mesh of the actual Pratt proprietary geometry contained about 30,000 quadrilateral elements).

6. CONCLUSIONS

The authors believe that automation, optimization, and standard work are three of the most important tools for improved quality and efficiency in the engineering environment. While this paper has used ANSYS as the meshing tool the authors have investigated other popular commercial meshers (i.e., HyperWorks, Partran, etc.) and have found that all of them provide similar meshing control parameters. Each meshing package that was investigated had the necessary toolkit and macro/programming interface to allow our optimization algorithm to run. Commercial CAD and preprocessors/CAE programs provide the controls and programmability necessary to accomplish automation and standard work in a generic environment.

The mesh optimization algorithms were implemented successfully and have provided a drastic reduction in the time required to produce acceptable airfoil meshes. These methods are now being applied to other aircraft engine hardware. A Pratt & Whitney these meshing algorithms fit into a larger automation tool which is used for multidisciplinary optimization. These algorithms have made it possible to minimize the time required to generate the mesh while producing very high fidelity results. A benefit of applying these meshing algorithms in a design process is consistent meshing which meets all standard work criteria for analysis. For industry, any amount of time reduction in an optimization run comprised of hundreds or thousands of iterations results is a huge time savings especially when there is no trade-off of fidelity or accuracy.

6. REFERENCES

[1] Huang, S. H., Shi, J., Maloo, S. and Steinthorsson, E., Automatic generation of quadrilateral multi-block topology for FEA/CFD applications, *IEEE ICIT'02,* Bangkok, THAILAND, 2002, pp 1300-1305.

[2] Owen, S. J., Staten, M. L., Canann, S. A. and Saigal, S., Quad-morphing: advancing front quad meshing using triangle transformations, *Proceedings 7th International Meshing Roundtable*, Dearborn, MI, 1998.

[3] Sarrate, J. and Huerta, A., Efficient unstructured quadrilateral mesh generation, *International Journal for Numerical Methods and Engineering,* Vol. 49, 2000, pp 1327-1350.

[4] Huang, H.-L. and Ho, S.-Y., Mesh optimization for surface approximation using an efficient coarse-to-fine evolutionary algorithm, *IEEE,* 2001, pp 584-591.

[5] Fujiwara, Y. and Sawai, H., Evolutionary computation applied to mesh optimization for a 3-D facial image, *IEEE Transactions on Evolutionary Computation*, Vol. 3 No. 2, 1999.

[6] Knupp, P. M., Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I – a framework for surface mesh optimization, *International Journal for Numerical Methods and Engineering,* Vol. 48, 2000, pp 401-420.

[7] Knupp, P. M., Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II – A framework for volume mesh optimization and the condition number of the Jacoian matrix, *International Journal for Numerical Methods and Engineering,* Vol. 48, 2000 pp 1165-1185.

[8] Zavattieri, P. D., Dari, E. A. and Buscaglia, G. C., Optimization strategies in unstructured mesh generation, *International Journal for Numerical Methods and Engineering,* Vol. 39, 1996, pp 2055-2071.

[9] Buscaglia, G. C. and Dari, E. A., Anisotropic mesh optimization and its application in adaptivity, *International Journal for Numerical Methods and Engineering,* Vol. 40, 1997, pp 4119-4136.

[10] Petersen, S. B. and Martins, P. A. F., Finite element remeshing: A metal forming approach for quadrilateral mesh generation and refinement, *International Journal for Numerical Methods and Engineering,* Vol. 40, 1997, pp 1449-1464.

[11] Dari, E. A. and Buscaglia, G. C., Mesh optimization: how to obtain good unstructured 3D element meshes with not-so-good mesh generators, *Structural Optimization*, Vol. 8, 1994, pp 181-188.

[12] Rassineux, A., Villon, P., Savignat, J.-M. and Stab, O., Surface remeshing by local hermite diffuse interpolation, *International Journal for Numerical Methods and Engineering,* Vol. 49, 2000, pp 31-39.

[13] Bottasso, C. L., Anisotropic mesh adaptation by metric-driven optimization, *International Journal for Numerical Methods and Engineering,* Vol. 60, 2004, pp 597-639.

[14] Tchon, K.-F., Dompierre, J. and Camarero, R., Automated refinement of conformal quadrilateral and hexahedral meshes, *International Journal for Numerical Methods and Engineering,* Vol. 59, 2004, pp 1539-1562.

[15] Ilin, A. V., Bagheri, B., Metcalfe, R. W. and Scott, L. R., Error control and mesh optimization for high-order finite element approximation of incompressible viscous flow, *Computer Methods In Applied Mechanics And Engineering,* Vol. 150, 1997, pp 313-325.

[16] ANSYS 7.1 Software program, ©Ansys, Inc., Canonsburg, PA USA, 2003.

[17] Oddy, A., Goldak, J., McDill, M. and Bibby, M., A distortion metric for isoparametric finite elements, *Transactions of the CSME*, Vol. 12, No. 4, 1988.