Hexahedral Mesh Refinement Using an Error Sizing Function

Gaurab Paudel,[1] Steven J. Owen,[2] and Steven E. Benzley[1]

[1] Brigham Young University gaurab.paudel@gmail.com,seb@byu.edu
[2] Sandia National Laboratories sjowen@sandia.gov

Abstract: The ability to effectively adapt a mesh is a very important feature of high fidelity finite element modeling. In a finite element analysis, a relatively high node density is desired in areas of the model where there are high error estimates from an initial analysis. Providing a higher node density in such areas improves the accuracy of the model and reduces the computational time compared to having a high node density over the entire model. Node densities can be determined for any model using the sizing functions based on the geometry of the model or the error estimates from a finite element analysis. Robust methods for mesh adaptation using sizing functions are available for refining triangular, tetrahedral, and quadrilateral elements. However, little work has been published for adaptively refining all hexahedral meshes using sizing functions. This paper describes a new approach to drive hexahedral refinement based upon an error sizing function and a mechanism to compare the sizes of the node after refinement.

Keywords: hexahedral, meshing, adaptation, refinement, sizing function, error estimates

1 Introduction

Mesh adaptation based on a sizing function is not a new topic. Procedures that incorporate quadrilateral, triangular, and tetrahedral mesh adaptation that rely on error-based sizing functions are available in the literature[1, 2]. In addition, there are a few techniques that generate an initial hexahedral mesh using the geometry features of the model[3]. However, conformal hexahedral mesh refinement, relying on an error-based sizing function, is a topic of current interest.

Traditional hexahedral meshing methods have not effectively used a sizing function because of connectivity restrictions imposed by traditional generation techniques[4, 5, 6]. This paper presents a method for a conformal hexahedral mesh refinement procedure based upon a sizing function. We incorporate the hexahedral mesh refinement techniques developed by Parrish[7] with a sizing function to drive refinement. The Parrish technique uses a set of seven transition templates and incorporates a special treatment for concave regions to ensure conformal and local mesh refinement. The sizing function used is developed from computed error estimates. However, criteria such as feature size or user specifications could also be included in the sizing function. To validate the method, comparisons between the actual refined node size and the ideal target node size are presented.

2 Background

The accuracy of finite element solutions can be improved by adapting the mesh. For example, meshes can be smoothed - known as r-adaptation - to improve quality. In addition, p-adaptation, which involves increasing the degree of the basis functions of the elements in the mesh, and h-adaptation, which involves increasing the number of elements, are traditional adaptation approaches. In addition, coarsening[8] can be used to reduce the number of elements. Although coarsening, r adaptation, and p adaptation are valid methods, this paper focuses specifically on h-adaptation i.e. refining by increasing the number of elements locally, to increase accuracy.

2.1 Refinement

As shown in Figure 1, 3-refinement[7] splits and exsiting hex three times along an edge, and 2-refinement splits an exisitng hex two times along and edge. 3-refinement is simple to implement but often can over refine a region of interest. 2- refinement[9] has more constraints on it's implementation but can often provide more gradual and controlled refined regions. The work reported here is based on 3- refinement, but can be easily adapted for 2- refinement.

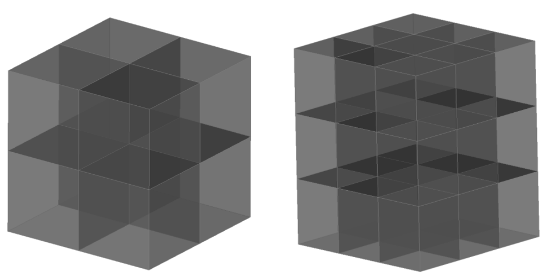

Fig. 1. 2 refinement and 3 refinement

Another common approach for adapting hexahedral meshes involves introduction of hanging nodes at edge centers as shown in Figure 2. This approach is straightforward to implement, and no transition elements to surrounding hexahedra are required. However, for this work we only consider conformal refinement techniques and do not introduce hanging nodes.

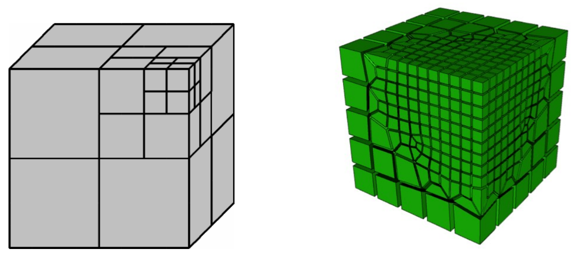

Fig. 2. Refinement with hanging nodes compared to conformal refinement

2.2 Current Methods

We note that a mesh can be adapted before any analysis is run if an a priori knowledge of the physics and geometry of problem is known. However, a fully automatic sizing function based all-hexahedral mesh adaptation procedure, including both refinement and coarsening, based on computed error is the central motivation of this research.

There are several methods that use sizing functions to refine the nodes or to adjust the node densities at the time of the initial mesh generation. Quadros, et al.[10], and Zhang and Zhao[11] have introduced mesh refinement using a sizing function based on geometric features of the model, however, they do not discuss hexahedral mesh refinement based on the error estimates. Anderson et al.[1] developed a refining and coarsening technique that uses error estimates as the sizing function, however this method is limited to all quadrilateral elements and does not consider the hexahedral mesh. Zhang and Bajaj[12] introduce hexahedral mesh refinement using volumetric data, but do not consider converting the error estimate from the finite element analysis into a mesh size for refinement. Wada et. al[13] discuss adaptation of hexahedral meshes using local refinement and error estimates, however their method does not compare the refined size of the mesh to the target size from the error estimates.

As mentioned by Anderson[1], most of the adaptation techniques are limited to triangular and tetrahedral elements. In 2010 Kamenski[14] presented mesh adaptation using the error estimates but his method is limited to triangular elements. De Cougny and Shephard[15], discuss tetrahedral mesh adaptation but they do not consider a hexahedral technique. Kallinderis and Vijayan[2] also discuss tetrahedral and triangular mesh refinement and coarsening but do not consider hexahedral elements. Babuska et. al[16] have presented a refinement technique based on a sizing function derived from error estimates, however their method is limited to non conformal rectangular elements with hanging nodes.

A hexahedral, when compared to a tetrahedral, mesh can provide more accurate results and, as mentioned in the introduction, is often the choice of an analyst. However, hexahedral adaptation techniques are not common. Hexahedral adaptation is a time consuming process, and requires knowledge of physics of the problem so that the generated mesh produces an acceptable error estimate from the finite element analysis. This paper presents unique and simple criteria to refine a hexahedral mesh using a sizing function and compares the refined size of the mesh with the target mesh size.

2.3 Sierra Mechanics Refinement Technique

In practice, rather than using a sizing function to drive the refinement, the error measures themselves are often utilized. For example, Sierra[17], a potential application for the work of this paper, is an advanced suite of analysis tools and provides three main approaches for driving refinement based upon an error measure. Although these techniques are currently used for tetrahedral and hanging node refinement, they could also be applicable for driving conformal hexahedral refinement in an adaptive analysis. For each of the three approaches, an error metric is computed for each element in the mesh, and the elements are ordered $M_{i=1..N}$ from minimum to maximum error. The three approaches are:

1. Percent of Elements: A threshold, a, which represents a percentage of the total number of elements N in the mesh that will be refined is specified. Starting from the element in M with the highest error and working towards the smallest error, a percent of the elements in the list are identified for refinement.
2. Percent of Max: A percentage threshold, b that represents the percentage of maximum error in the mesh that will be identified for refinement is specified. For example, if the maximum error of all elements in $M_{i=1..N}$ was 50 percent with $b = 90$ percent, then all elements with error > 5 percent would be identified for refinement.
3. Percent of Total Error: A percentage threshold, g, which represents a percent of the total error in the mesh that will be identified for refinement is specified. For example if we represent the total error of all elements in $M_{i=1..N}$ as:

$$||e||_{total} = \sum_{i=0}^{N} ||e||_i$$

Starting from the element in M with the highest error and working to the smallest error, those elements that contribute to a total error of $g. ||e||_{total}$ would be identified for refinement.

Sierra refinement is performed based upon one of the above approaches, followed by subsequent analysis iteration. After each iteration, elements can be again identified for refinement. This procedure continues until a convergence or error threshold has been achieved.

For the work presented in this paper, rather than using the error measure directly, a sizing function is developed from calculated error estimates. This provides the opportunity to utilize the sizing function as a general field to drive meshing or refinement. It also provides a field for which we can validate the resulting refinement operations to determine the effectiveness of the refinement algorithms at reaching the desired size.

3 Hexahedral Mesh Refinement

It is desired to have the size of a mesh be as close as possible to the sizes provided by the sizing function in order to obtain high computational accuracy in the results without significantly increasing the computation time.

3.1 Sizing Functions

Sizing functions are used as the mechanism for refining a mesh. There are several ways to generate sizing functions. Error estimates can be used to define a sizing function. Geometric characteristics, curvature, and sharp features in the model can also be used to define a sizing function [10]. Other bases for sizing functions include: the stress or strain gradients, change in the material properties, points of application of loading, and the location of boundary conditions.

For this paper, developing a sizing function based on the error estimate of an initial calculation will be used. The error estimate should be robust enough to ensure the increase in accuracy of the results, and also steer the adaptation only in the desired area of the model. The generation of the error estimate is an important area of study. For this work, error estimates are obtained from the existing finite element code. Physical phenomenon in engineering and sciences can be modeled using partial differential equations. However, complex mathematical models using the partial differential equations might not have an analytical solution. Fortunately, finite element analysis can provide an approximate solution to these complex models[18]. As these solutions are an approximation to the analytical solution, there are several sources of error.

As cited by Grastch and Bathe[18], the computation of error estimates and using it as criteria for subsequently refining the region where error estimates are high should be computationally cheaper than refining the entire model. The error estimates should be accurate enough to closely represent the unknown actual error. The goal of the error estimates is to steer the mesh adaptation. For this work, the built in error estimates produced by the simulation code, ADINA[19] are used.

3.2 Tools and Requirements

The error estimates from the finite element analysis approximate the expected error produced from the numerical model. Our refinement process is driven by a sizing function generated from error estimates. The error estimates from the finite element solvers are converted into an Exodus II[20] file, a random access, machine independent, binary file, that is used as a sizing function by the mesh generating toolkit, CUBIT[21]. The Exodus file format stores all the information about the initial mesh. It can be used for input and output of results and can also be used for post-processing of results. The Application Programming Interface (API) to create the exodus file is available in the public domain and a manual to create such file is also available. An example to create an Exodus file to drive the refinement process can be found in Reference[22].

3.3 Algorithm

Our proposed algorithm refines a hexahedral mesh locally based on the generated sizing function. The main goal of this algorithm is to complete the refinement process without the need for a user intervention. The error estimate, used to define the sizing function, determines whether a node should be considered for refinement or not. Often, after the mesh has been refined, the quality of the elements degrade. The degradation is usually a result of insertion of templates in the transition zones between the refined and non-refined regions. Hence, smoothing is performed on the elements within and near the refinement region to improve the element quality. A flowchart of the algorithm is given in Figure 3.

3.4 Algorithm Example

This section outlines the input, refinement criteria, and comparison of target and current sizes, of the algorithm. An example is used to explain the steps outlined in the algorithm. For simplicity, the algorithm example section is further divided into three sub-sections: input, refinement criteria and comparison of current size and target size.

3.5 Input

For this example, a quarter piston modeled with a load on its base plate is used. Initially, a coarse mesh, as shown in Figure 4, is generated using an all

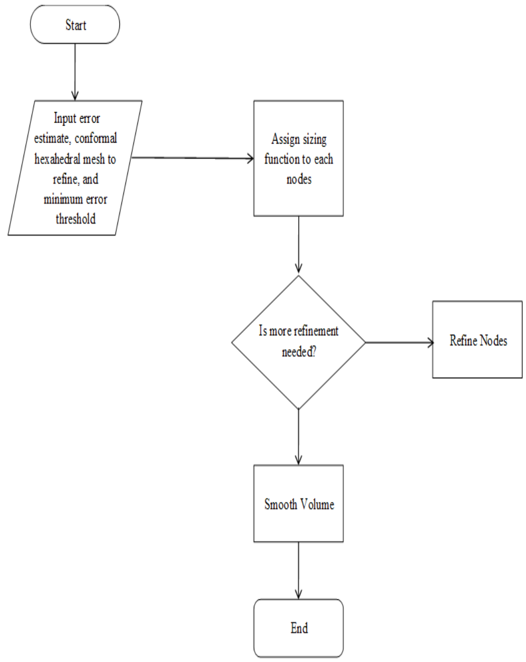

Fig. 3. Algorithm Flowchart

hexahedral meshing technique[21]. Next, the appropriate boundary conditions and loading, are applied. The initial mesh and boundary conditions are then exported to ADINA[19] to perform the finite element analysis. Error estimates are generated as a part of the ADINA analysis and the band plot of the error estimate is shown in Figure 4. The error estimate from the finite element analysis is then written in Exodus II, a binary file format, and is used to compute the sizing function to drive the refinement process.

3.6 Refinement Criteria

After each node is assigned a scalar error estimate, it is compared with the minimum specified threshold allowable error, g, for the particular problem. Normally the value for g and the error estimates are scalar values between 0 and 100, representing a percent error. Nodes with error estimates greater than the allowable user defined error are identified for further refinement. Nodes with error estimates lower than g are not refined but may be subsequently smoothed to improve the mesh quality. In this section the specific criteria is presented, based on the sizing function computed from error estimate, to

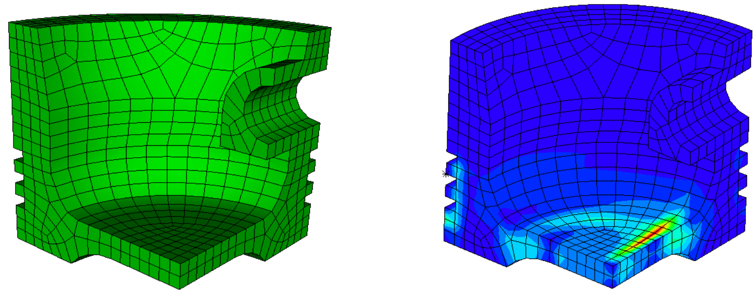

Fig. 4. Initial coarse mesh and Band plot of error estimate

identify the nodes needed for refinement. The terms basic to this technique are defined below.

The size of a node, h_a, is computed using Equation 1. h_a is the average of the lengths of edges attached to the node.

$$h_a = \frac{1}{n}\sum_{i=1}^{n} l_i \tag{1}$$

where:

h_a = size of a node

l_i = length of i^{th} edge

n = number of edges attached to the node

The relationship between error and mesh size can be approximated, for elasticity and heat problems, from the Poisson heat equation[23] as:

$$|e| = Ch^2 \tag{2}$$

where:

C = a constant

h = the element edge length

If the error and the element edge length for a node in the mesh are known, then C_n for that node can be computed as:

$$C_n = \frac{e_n}{h_a^2} \tag{3}$$

where:

C_n a constant for the node

e_n error estimate at the node

h_a = size of a node as defined in Equation (1)

The user provides a threshold error measure for the entire mesh, $|e| = g$ at which below g is acceptable. For this work, the target size is computed from the error measure but it can also be computed from the geometric feature or size provided by the user. Then, a target size from the above equation at the node is computed as:

$$t_s = \sqrt{\frac{g}{C_n}} \tag{4}$$

where

t_s = target node size

g = minimum allowable error provided by user

C_n = a constant computed from Equation 3

The size ratio, S_r, is defined in Equation 5 as the ratio of the target size of the node, t_s, determined using the error estimates, to the actual size of the node, h_a.

$$S_r = \frac{h_a}{t_s} \tag{5}$$

where:

S_r = node size ratio

h_a = actual node size as defined in Equation (1)

t_s = desired node size determined from the error measure

The terms defined in Equations 1 through 5 are used to identify the nodes that require refinement. It is assumed that the nodes with a size ratio of less than or equal to one are acceptable and need no refinement. Similarly, if the size ratio at the node approaches 3, then it indicates that at least one refinement operation should be performed. The value of 3 is assigned because 3-refinement is used as the mechanism for refinement. It is assumed that when the refinement is performed the size of a node will decrease by a factor of three and hence a size ratio of one is obtained after the refinement is performed. Since each split operation will reduce the local element size by a factor of 3, as the size ratio approaches 9, the node will be marked for two split operations.

Likewise, a size ratio that approaches 27 will be marked for three splits. For this work, using an average of powers of three for the thresholds seemed to provide acceptable results. Table 1 summarizes the approach.

Table 1. Size ratio range and number split operation

Size Ratio Range	Number of refinement split operations
0 - 2	0
2 - 6	1
6 - 18	2
> 18	3

If the size ratio is below the allowable threshold, g, then no refinement is performed on the node. It is assumed that it is perfect size or it is over refined. The nodes with the size ratio between 2.0 and 6 are identified for the first level of refinement. These nodes are identified for the first level of refinement because they have size ratio near 3 and less than 9. Hence, after the refinement their size ratio should come close to 1. Similarly, the nodes with size ratio between 6 and 18 are identified for the second level of refinement based on the fact that these nodes have size ratio near 9 and less than 27. Hence, when they are refined the new size ratio should be close to 1. The nodes with size ratios greater than 18 are identified for three levels of refinement. This criteria for refinement is continued until less than 10 percent of total nodes meet the size ratio criteria. This 10 percent is chosen to ensure that computation time is not wasted performing a refinement that will not gain a significant level of accuracy in the finite element solution. This criteria serves as the exit criteria for the refinement process. Figure 5 shows the refined mesh of the quarter piston with load on its base.

3.7 Comparison of Current and Target Size

One of the means to determine if the algorithm is performing adequately to achieve the desired accuracy in the results is to compare the refined size of node to the target size of the node. The node size ratio, S_r, is used as the criteria to compare the efficiency of the algorithm. It is assumed that the algorithm should perform such that the size ratio of all the nodes should be less than or equal to 1.0. If a coarsening algorithm were to be implemented, size ratios less than 1 would be minimized and all the size ratios would be close to 1.0. It is recognized that an exact match everywhere where $S_r = 1$ is impossible, however the approach presented here can be validated by statistically examining how close final mesh matches the intended sizing function.

Based on Equation 5, a size ratio for the node before the refinement and after the refinement is computed. Theoretically, there should not be any change

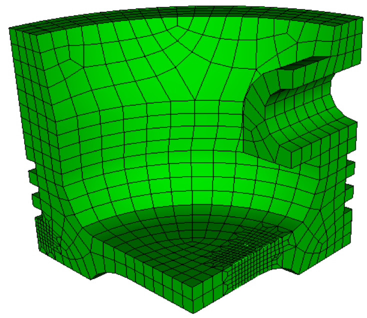

Fig. 5. Refined mesh of quarter piston

in the volume of elements with nodes having size ratios below 1.0 and the volume of elements with nodes having size ratios greater 3 should be reduced significantly after the refinement. Also, there should be less than 10 percent of the volume that fall in the size ratio greater than 2.0. The plot in Figure 6, shows that for this problem, there is not much change in the volume for size ratios below 1.0 and in additon, there was not much initial volume with size ratio greater than 3. Although there is some volume greater than 2.0, this condition prevails because the refinement is deemed complete if there are less than 10 percent of nodes that require refinement. In Figure 7, the change in the percentage of volume falling in the particular size ratio before and after refinement is shown. Note that that most of the change is around size ratio 1.0 and there is negative change in volume for size ratio greater than 2.0. This shows that most of the nodes have size ratio 1.0 after the refinement and refinement is taking place in the node with size ratio greater than 2.0.

4 Example

This section gives a complete example of all hexahedral sizing based refinement. Shown are the initial coarse mesh, the band plot of the stress error provided by the finite element analysis, the refined mesh, a histogram showing the results of refinement, and the analysis using the refined mesh. All initial meshes were generated with CUBIT[21] and the finite element analysis was performed using ADINA[19].

Since, when a node is refined and three nodes are created, it is difficult to get size ratio exactly 1.0. Also, addition of templates in the transition zones tends to over refine the mesh. The goal is therefore is to get most of the nodes with size ratio close to 1.0 and 2.0.

In this example, a gear rotating about its axis is modeled. A torque is applied at the center of the gear and three teeth are constrained. Figure 8

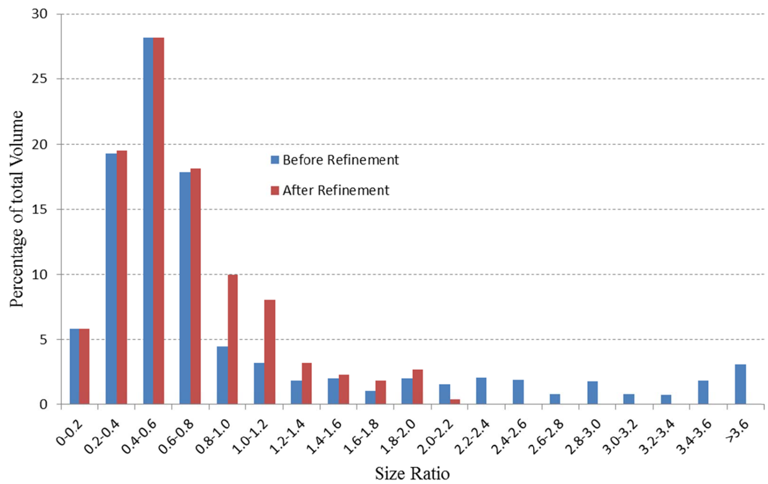

Fig. 6. Plot of size ratio and percentage of total volume before and after refinement

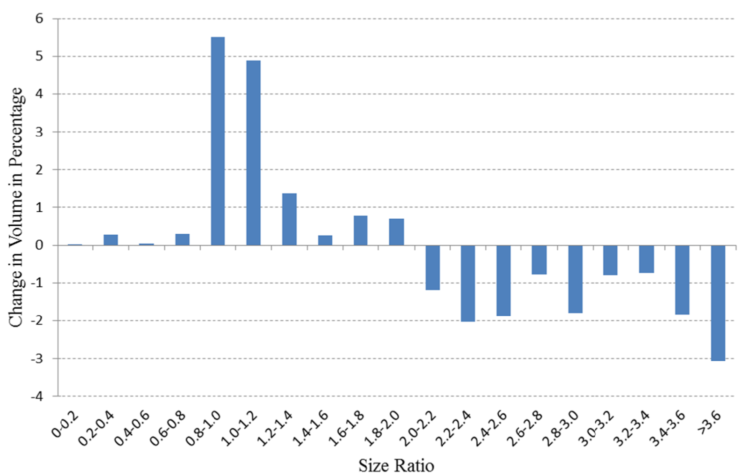

Fig. 7. Plot of size ratio and percentage of total volume before and after refinement

shows the initial mesh and the band plot of error estimates. For this example the minimum threshold error used is 8 percent.

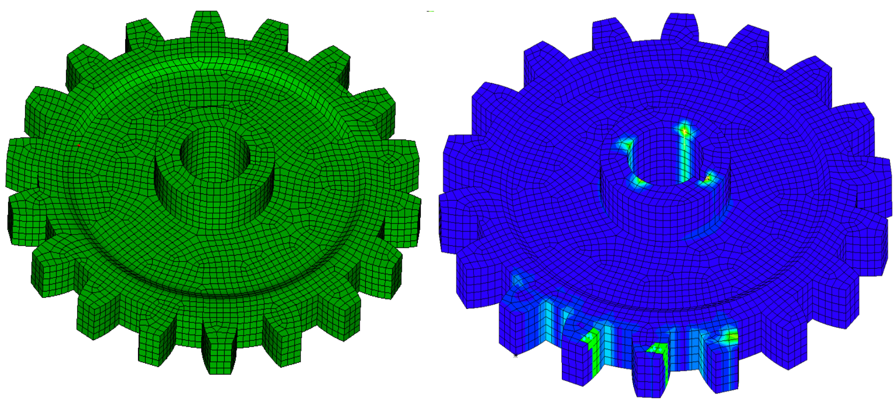

Fig. 8. Initial Mesh and Error band plot of gear example

Figure 9 shows the refined mesh. Note that refinement is implemented around the constrained teeth where there is a high error estimate. Figure 10 is a plot of the volume before and after the refinement vs. size ratio. Notice that most of the volume is below a size ratio of 2.0. Also note that there is little change in volume for size ratios less than 1.0. In Figure 11, the change in the percentage of volume falling in the particular size ratio before and after refinement is shown. Notice that most of the change is around a size ratio of 1.0 and there is negative change in volume for size ratio greater than 2.0. This shows that most of the nodes have a size ratio of 1.0 after the refinement and that refinement is taking place in the volume where the size ratio greater than 2.0. Also, it should be considered that when the coarsening algorithm is implemented the volumes should come close to the size ratio 1.0, the ideal element size ratio. Figure 12 shows the error estimate of an analysis on the refined mesh.

5 Conclusions and Future Work

This paper presents a sizing function algorithm that selects the nodes in all hexahedral meshes for refinement and then generates a refined all conformal hexahedral mesh for subsequent finite element analysis. The sizing function is developed from error estimates from an intial analysis of the problem. Only elements in the volumes indicated size changes are modified. As a result, the locally refined mesh is able to capture the physics of the problem more

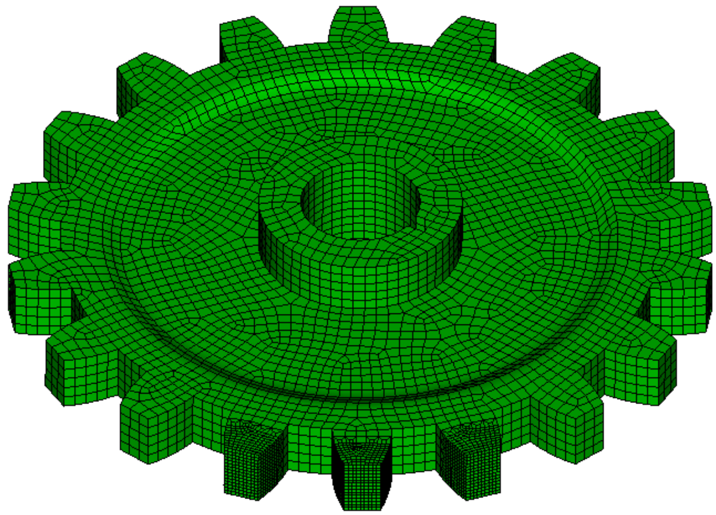

Fig. 9. Refined Mesh of gear example

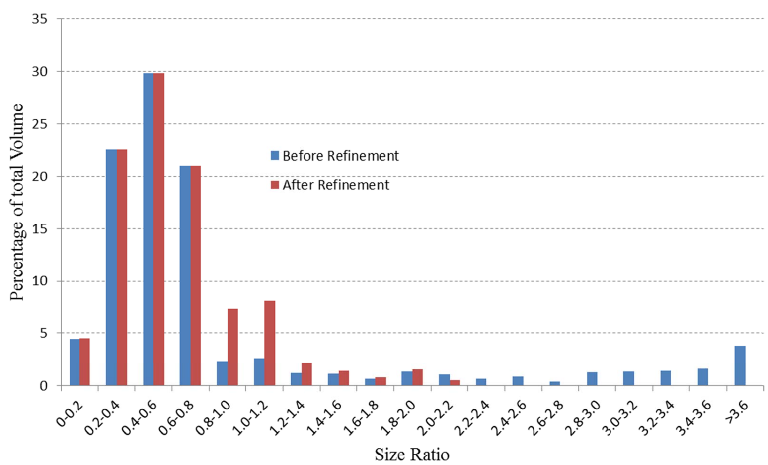

Fig. 10. Plot of percentage of total volume and size ratio before and after refinement for gear model

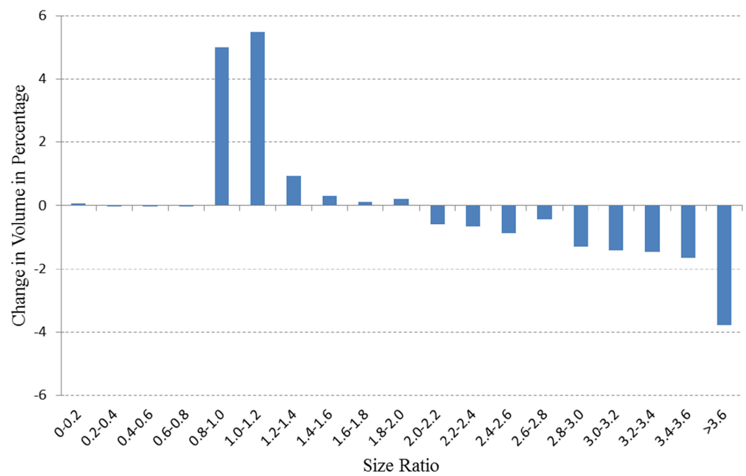

Fig. 11. Plot of size ratio and change in volume in percentage

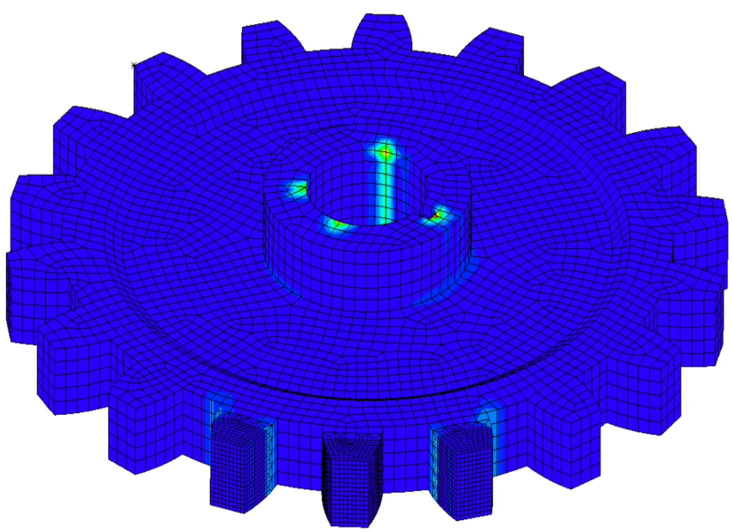

Fig. 12. Band plot of error on the refined gear mesh

accurately, with a minimum increase in the computation time thus providing high efficiency and accruacy in the finite element solution. The results from the examples shown in this paper are promising but improvements can be done on this technique to work more efficiently.

This work provides a platform for the total adaptation of hexahedral elements. Only refinement was considered here, but coarsening[8] could be included. Including coarsening would provide a method to fully adaptive hexahedral meshes.

Currently, this method uses 3-refinement which often over refines the mesh and does not provide good gradation between refined and coarsened volumns. When this technique is used with the 2-refinement technique developed by Edgel et al. [9] it should provide more gradation in the refinement.

In this work, sizing function is developed using computed error estimates. However, other criteria such as feature size or user specified field function could be included in the sizing function.

References

1. Anderson, B.D., Benzley, S.E., Owen, S.J: Automated All-Quadrilateral Mesh Adaptation: In Proceedings of 18th International Meshing Roundtable pp. 557-574 (2009)
2. Kallinderis, Y., Vijayan, P.: Adaptive Refinement-Coarsening Scheme for Three-Dimensional Unstructured Meshes. American Institute of Aeronautics and Astronautics Journal 31, 1440-1447 (1993)
3. Zhang, H., Zhao G.: Adaptive hexahedral mesh generation based on local domain curvature and thickness using a modified grid-based method. Finite Elements in Analysis and Design, 43(9), 691-704 (2007)
4. Knupp, P.M.: Next Genration Sweep Tool: A Method For Generating All-Hex Meshes on Two-And-One-Half Dimensional Geometries: In Proceedings of 7th International Meshing Roundtable, pp. 505-513 (1998)
5. Scott, M.A., Earp, M., Benzley, S.E.:Adaptive Sweeping Techniques. In Proceedings of 14th International Meshing Roundtable, pp 417-432, (2005)
6. Cook, W.A., Oaks, W.R.: Mapping Methods for Generating Three-Dimensional Meshes: Computers in Mechanical Engineering, 1, 67-72 (1983)
7. Parrish, M., Borden,M., Staten, M., Benzley, S.E.: A Selective Approach to Conformal Refinement of Unstructured Hexahedral Finite Element Meshes.In Proceedings of16th International Meshing Roundtable, pp. 251-268, (2007)
8. Woodbury, A.C. Shepherd, J.F., Staten, M.L., and Benzley, S.E.: Localized Coarsening of Conforming All-Hexahedral Meshes. In Proceedings of 17th International Meshing Roundtable, pp. 603-619, (2008)
9. Edgel, J.: An Adaptive Grid-Based All Hexahedral Meshing based on 2- Refinement. M.S. Thesis, Brigham Young University, (2010)
10. Quadros, W.R., Vyas,V., Brewer, M., Owen, S., Shimada, K,: A computational framework for generating sizing function in assembly meshing. In Proceedings of 14th International Meshing Roundtable, pp 55-72, (2005)

11. Zhang, H., Zhao, G.: Adaptive hexahedral mesh generation based on local domain curvature and thickness using a modified grid-based method. Finite Elements in Analysis and Design, 43(9), 691-704, (2007)
12. Zhang, Y., Bajaj, C.: Adaptive and Quality Quadrilateral/Hexahedral Meshing From Volumetric Data. In Proceedings of 13th International Meshing Roundtable, pp. 365-376 (2004)
13. Wada, Y., Okuda, H.: Effective adaptation technique for hexahedral mesh: Concurrency and Computation: Practice and Experience,14,(6-7), 451-463 (2002)
14. Kamenski, L.: A Study on Using Hierarchical Basis Error: Iin Proceedings of 19th International Meshing Rountable, pp. 297-314, (2010)
15. Cougny, H.L.D., Shephard, M.L.: Parallel Refinement and Coarsening of Tetrahedral Meshes. International Journal for Numerical Methods in Engineering, 99(46), 1101-1125 (1999)
16. Babuska, I., Hugger, J., Strouboulis, T., Koops, K., Gangara, S.K.: The asymptotically optimal meshsize function for bi-p degree interpolation over rectangular element: Journal of Computational and Applied Mathematics, 90, 185-221 (1998)
17. Stewart, J.R. and H.C. Edwards, A framework approach for developing parallel adaptive multiphysics applications. Finite Elements in Analysis and Design, July 2004. 40(12): p. 1599-1617.
18. Grtsch, T., Bathe, K.J.: A posteriori error estimation techniques in practical finite element analysis: Computer and Structures, 83, 235-265, (2005)
19. ADINA AUI 8.5.2. 2008, ADINA R and D, Inc., www.adina.com.
20. Schoof, L.A. Yarberry, V.R.: Exodus II :Finite Element Data Model. Sandia National Laboratory, Albuquerque, NM, SAND92-2137, (1995)
21. Cubit 12.1 user documentation. Research report, Sandia National Laboratories, Albuquerque, N.M., (2010)
22. Paudel, G., Hexahedral Mesh Refinement using an Error Sizing Function, MS Thesis, Brigham Young University, Provo, UT, (2011)
23. Malvern, L.E., Introduction to the Mechanics of a Continuous Medium1969, New Jersey: Prentice-Hall (1969)