 CMES, vol.7, no.3, pp.283-291, 2005

Eliminating Slivers in Three-Dimensional Finite Element Models

R.H. Moore[1] **and S. Saigal**[2]

Abstract: An efficient method for treating slivers and other poorly shaped elements in finite element solutions is presented. A major difficulty for finite element analyses arises from the creation of slivers in automated mesh generation. Sliver shaped elements can degrade the accuracy of a solution and are difficult to remove from a mesh. The proposed method treats slivers by first merging them with neighboring elements to form polyhedra and next subdividing the polyhedra into well-shaped tetrahedral elements. The method does not require the cumbersome and expensive operations of addition or rearrangement of nodes. The validity and accuracy of the present method are demonstrated through the evaluation of a performance metric for a set of example problems.

keyword: slivers, finite elements, Delaunay tessellations, element quality.

1 Introduction

It is commonly known that mesh quality influences the accuracy of finite element solutions. There is presently no proven method for generating a quality mesh for complex three dimensional (3D) domains. For tetrahedral mesh generation, Delaunay tessellation (Delaunay, 1934) is a common first step for domain partitioning. Refinement steps (Shewchuk, 1998) follow to improve the element shapes and to remove poorly shaped elements. One challenge in mesh generation has been the removal of slivers, which are elements whose volumes are nearly zero, but whose edge lengths are nearly equal to each other. Slivers are especially common in tessellations that start on rectangular grids. These sliver elements raise the condition number of the assembled matrix, thereby reducing the accuracy of direct solvers and increasing the solution times for iterative solvers. Quality three-dimensional mesh generation remains an open problem, although several breakthroughs have recently occurred (Shewchuk, (1998); Li (2000); Edelsbrunner, Li, Miller, Stathopoulos, Talmor, Teng, Ungor, and Walkington, (2000); Edelsbrunner and Guoy (2002); Cheng, Dey, Edelsbrunner, Facello, and Teng (2000); Cheng and Poon (2003); Cheng, Dey, Ramos, and Ray (2004)). Alternate approaches include bypassing poor quality meshes through the use of meshless methods, such as the methods presented by Han and Atluri (2004) and Li, Shen, Han, and Atluri (2003). In the absence of body forces, these meshless methods can bypass volume integrations within a domain and instead use boundary integrations. Thus, many of the numerical challenged caused by slivers are avoided.

One approach to treating slivers is to remove the slivers from the mesh and has been addressed in the works by Li (2000); Edelsbrunner, Li, Miller, Stathopoulos, Talmor, Teng, Ungor, and Walkington, (2000); Edelsbrunner and Guoy (2002); and Cheng, Dey, Edelsbrunner, Facello, and Teng (2000). However, sliver removal methods are computationally expensive and involve the addition and rearrangement of nodes. An alternative approach to treating slivers is due to Calvo, Idelsohn, and O˜nate (2003) and Idelsohn, O˜nate, Calvo, and Del Pin (2003) and uses *extended Delaunay tessellation*, where elements that share nearly the same circumsphere are joined into one new polyhedron "element." Since finite element shape functions do not exist for arbitrary polyhedrons, the Natural Element Method (NEM) shape functions from Sukumar, Moran, Semenov, and Belikov (2001) were used to integrate the stiffness matrices for these polyhedron elements. This approach does not involve the addition or rearrangement of nodes, yet it effectively treats slivers in a 3D mesh. Its use of natural element method shape functions is, however, computationally expensive. Additionally, the NEM do not yet rest upon the same solid foundations as the finite element methods in terms of theoretical proofs for uniqueness, error bounds, and convergence rates. A method that allows the use of standard finite element shape functions would be more desirable.

[1] CMU, Pittsburgh, PA, U.S.A.
[2] USF, Tampa, FL, U.S.A

The present paper demonstrates the use of standard finite elements in conjunction with extended Delaunay tessellation, where nearby elements may be joined together into polyhedra. A modified joining approach to merge slivers into nearby polyhedra is proposed. The advantages over the NEM based approach include faster computing times, possibly higher accuracy, less complex coding, and the retention of most of the theoretical developments and proofs of the finite element method. Unlike the sliver removal approach of Li (2000); Edelsbrunner, Li, Miller, Stathopoulos, Talmor, Teng, Ungor, and Walkington, (2000); Edelsbrunner and Guoy (2002); and Cheng, Dey, Edelsbrunner, Facello, and Teng (2000), the proposed approach does not require the addition or rearrangement of nodes. Thus, the technique presented here can be directly employed within existing FEM codes.

2 Current Approach

In this section, a brief overview of the approach is first presented followed by more detailed descriptions.

A Delaunay tessellation of the three-dimensional domain from any source, for example QHull (2004), is first obtained. Although it is not necessary for all cases, surface slivers are first removed from the initial mesh. Using a modification of the method of Calvo, Idelsohn, and O˜nate (2003), all tetrahedra that have nearly the same circumsphere are joined together into polyhedra. Any sliver that has all four of its nodes contained in an adjacent polyhedron is merged into this adjacent polyhedron. A temporary node is next added to the centroids of the non-tetrahedral polyhedra generated by this process. Each polyhedron is then subdivided into a collection of local tetrahedra by connecting the temporary centroidal node to all of the external triangular facets of the polyhedron. The polyhedron stiffness matrix is generated by computing and assembling the stiffness matrices of the local tetrahedra that share the temporary centroidal node. The temporary centroidal node then is eliminated using static condensation. The resulting stiffness matrix is assembled into the global stiffness matrix. The standard FEM approach is used from there on, although it is possible to return to the polyhedron approach for the evaluation of the displacements and strains within a polyhedron if desired. These steps are discussed in more details in the following sections.

2.1 Detecting slivers

An important and first step in handling poorly shaped elements is detecting them. Shewchuk (2002) recently proposed a scale-invariant quality metric that applies best for elements used to solve the Poisson equation and is given as

$$Q = V\left(\sum_{m=1}^{4} A_m \Big/ \sum_{1\le i\le 4} A_i A_j l_{ij}^2 + 6|V| \max_i \sum_{j\neq i} A_j l_{ij}\right)^{\frac{3}{4}} \tag{1}$$

where A_i and l_{ij} are the face areas and edge lengths, respectively, of the tetrahedron; and V is the volume. This quality metric, Q, is based on the error between the respective gradients of the true and interpolated solutions $\|\nabla u_{exact} - \nabla u_{interpolated}\|_\infty$. It has a high magnitude for high quality elements, is zero for poor elements, and is negative for incorrectly numbered elements or elements with negative volume. For reference, consider a regular tetrahedron whose faces are all equilateral triangles. The quality metric, Q, for the regular tetrahedron is 0.104103. Also, a flat square tetrahedron sliver of unit edge length with one corner raised by 0.021 units has a quality metric, Q, equal to 0.002. The volume of this element for this case is also nearly 0.002. Such a sliver is of an undesirable shape for use in finite element analyses. For convenience, we will consider elements with a quality metric, Q, less than 0.002 to be considered as poor quality elements. This value of quality metric, Q, will be used as a basis to flag poorly shaped elements and to compare alternative shapes.

2.2 Element Joining

The extended Delaunay tessellation by Calvo, Idelsohn, and O˜nate (2003) and Idelsohn, O˜nate, Calvo, and Del Pin (2003) partitions a domain into polyhedron regions by first partitioning the domain into a standard Delaunay tessellation of tetrahedra. In the next step, the tetrahedra are joined together when they share nearly the same circumsphere. Calvo, Idelsohn, and O˜nate (2003) join tetrahedra when the centers of the neighboring tetrahedra are close to each other and within a small fraction of their radii as

$$\|c_1 - c_2\| < \delta r_{rms} \tag{2}$$

where c_1 and c_2 are the circumcenters of two adjacent tetrahedra; r_{rms} is the root mean square radius; and δ is a small tolerance factor, which may be set as 0.1 or 0.01.

Consider a cube, as shown in Fig. 1, where each of the eight nodes falls on the surface of a single sphere. One possible partition of this cube comprises seven tetrahedra, as shown in the exploded view in Fig. 2. Since all nodes of this cube fall on the surface of the same sphere, the circumsphere for each of the 7 tetrahedra is one and the same. If each of the corner nodes of the cube is slightly perturbed, it may result in 7 different circumspheres, each corresponding to one of the seven tetrahedra. These circumspheres will, however, lie very close to each other and satisfy the criterion shown in Equation (2). For both these cases of the cube example, the approach of Calvo, Idelsohn, and O˜nate (2003) results in joining all 7 tetrahedra together and creating a polyhedron with a cubic shape. Thus, the sliver, such as the top element shown in Fig. 2, is successfully joined with other polyhedra in this approach. While this formulation works for a large class of slivers, it is not effective for all cases leading to significant numerical instabilities for these cases.

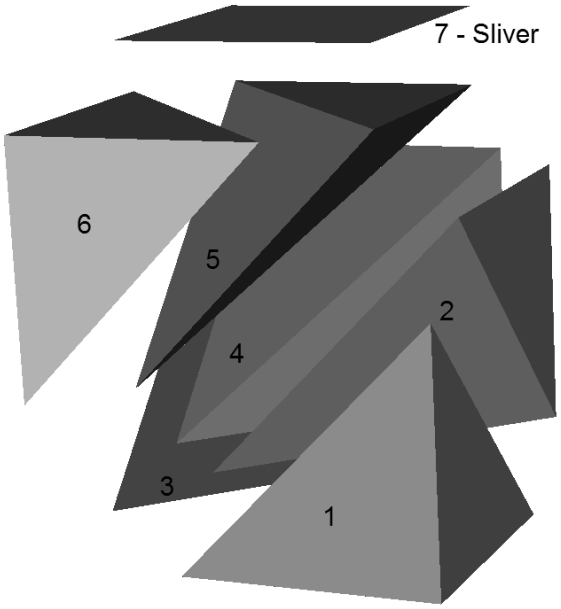

Figure 2 : A cube partitioned into seven tetrahedra. Tetrahedron 7 is a sliver.

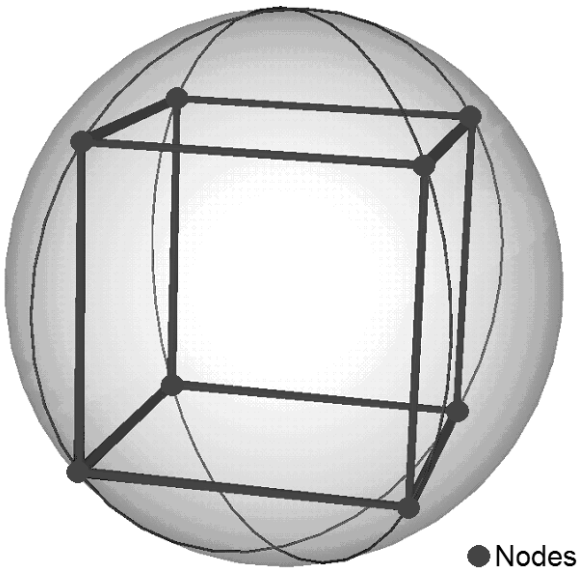

Figure 1 : A cube with a single circumsphere

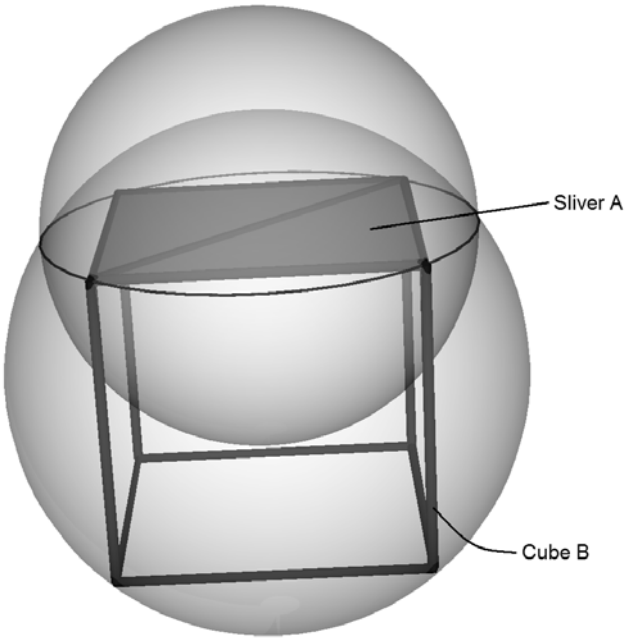

Figure 3 : A sliver with a small circumsphere

Consider for example, the sliver A lying at the top of the cube B as shown in Fig. 3. A desirable outcome of the joining criterion would be that the sliver A is merged with the cube B. However, the circumspheres for sliver A and cube B are significantly different and their circumcenters are relatively far apart. The circumcenter for sliver A is at the top of the cube where the equator of its circumsphere is shown passing through the top four nodes of the cube.

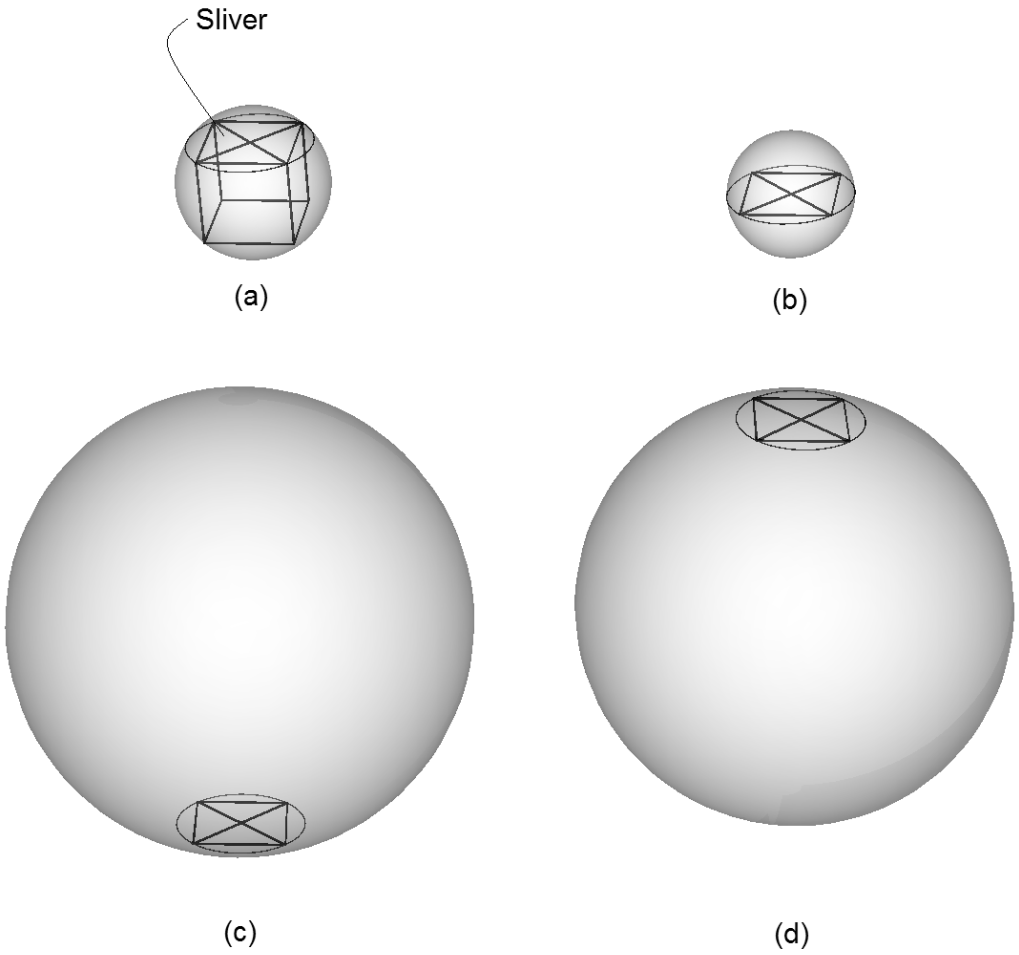

Figure 4 : Boundary slivers with four different sized circumspheres (a) The sliver shares the circumsphere of the cube (b) The sliver has a small circumsphere whose circumcenter is co-planar with the sliver. (c) The sliver has a large circumsphere whose circumcenter is far above the sliver. (d) The sliver on the bottom of a domain has a large circumsphere whose circumcenter is far below the sliver.

The circumcenter for the cube B, on the other hand, is at the centroid of the cube. For this case, the locations of the centers are too far apart and the joining criterion of Equation (2) will not be satisfied. The sliver A, thus will not be joined with the cube B. This failure to join slivers affects slivers embedded within a domain as well as slivers on the outside boundaries of a domain.

Four common examples of boundary slivers are shown in Fig. 4. The desired circumsphere for a cube with a top surface sliver is shown in Fig. 4a. If the sliver shares the circumsphere of the cube, it will join properly. In Fig. 4b, the top surface sliver is shown with a circumsphere whose circumcenter is on the same plane as the sliver. This case is the same as the one shown in Fig. 3 and has the same undesirable result that the sliver is not joined with the cube. For clarity, the cube and its circumsphere are omitted from Figs. 4b-d. Surface slivers that are nearly co-planar may also have nearly infinite circumspheres, as shown in Figs. 4c and 4d, where the circumcenters are far above and far below the cube center, respectively. The circumcenter for the sliver shown in Fig. 4c is far above the circumcenter of the cube, so that this sliver will not join with the cube. Similarly, the sliver shown in Fig. 4d may be located at the bottom boundary of the domain and its circumcenter may be far away from the domain, so that this sliver will also not join. The present study is aimed at addressing these limitations of the existing methodologies available for the treatment of slivers.

To address these limitations, a modified joining criterion has been developed as follows:

1. Flag all slivers or tetrahedra that have a poor quality metric, for example a quality metric, Q $<$ 0.002. These elements are termed poor elements or slivers, although they may have shapes other than a sliver.

2. Join all adjacent tetrahedra using the criterion stated in equation (2). Two tetrahedra or polyhedra are adjacent if they share at least one surface face. Consider tetrahedra separated by poor elements as adjacent tetrahedra.

3. Test each remaining sliver against all polyhedra that share at least one face with this sliver. If all four nodes of a sliver are shared with a polyhedron, then the sliver is merged with the polyhedron. The sliver A at the top of Fig. 3, for example, shares two faces with the cube B below it. Additionally, all four nodes of sliver A are also part of cube B. Sliver A, therefore, will be merged with cube B when this test is applied.

The slivers shown in Figs. 3 and 4b-d will all be joined with their adjacent elements by the test described in step 3 above. This is because all four nodes of each of these slivers are contained in the adjacent polyhedron, which is a cube for these examples. Step 3 is effective both for slivers internal to the domain and for slivers on the boundary. This modified joining process was able to join all internal slivers, including the ones that are missed with the simple center proximity test stated in equation (2). The joining process is a local process and should have a computational cost proportional to the number of elements.

2.3 Element Sub-Partitions

After employing the step of Element Joining, the domain is partitioned into a set of tetrahedra and polyhedra. The stiffness matrices generated for the tetrahedra are assembled into the global FEM equations the same way as is done in standard finite elements. However, the polyhedra cannot be assembled using standard finite element procedures, since there are no general shape function definitions for general polyhedra. To address this limitation, these polyhedra are subdivided into better shaped tetrahedra for corresponding finite element treatment. To subdivide a polyhedron, we first compute its centroid and then form a set of tetrahedra by connecting the outside faces to this centroid as follows.

The centroid of a polyhedron, which is formed by joining nT number of tetrahedra including poorly shaped slivers, may be computed using the formula for composite bodies as

$$Centroid_P = \sum_{T=1}^{nT} Volume_T \, Centroid_T \Big/ \sum_{T=1}^{nT} Volume_T \tag{3}$$

where Volume$_T$ and Centroid$_T$ are the volume and centroid, respectively, of the tetrahedron T. A new set of sub partitions is now constructed by connecting each outer face triangle of the polyhedron to this centroid. The outer face triangles are quickly found by identifying the faces of the nT tetrahedra that are not shared by another tetrahedra. All matched faces are internal and are ignored in this process of subdivision. The outside face could be considered as the base of a tetrahedron, and the centroid as its tip.

2.4 Local Stiffness Matrix and Force Vector

The polyhedron has a local stiffness matrix and a local force vector that are assembled the same way as corresponding quantities in standard finite elements. For example, let the polyhedron comprise nSubTets number of tetrahedral sub partitions. Further, let each tetrahedral sub partition have a stiffness matrix [K]$_e$. The local stiffness matrix for the polyhedron is then given as

$$[K]_{Local} = \mathop{A}_{e=1}^{nSubTets} [K]_e \tag{4}$$

where $\boldsymbol{A}$ is the assembly operator. Since the original mesh did not contain the temporary centroidal node, it is desirable in many instances to remove it. Let the local assembled stiffness and force relationship be given as

$$\begin{bmatrix} K_{rr} & K_{re} \\ K_{er} & K_{ee} \end{bmatrix} \begin{Bmatrix} u_r \\ u_e \end{Bmatrix} = \begin{Bmatrix} F_r \\ F_e \end{Bmatrix} \tag{5}$$

where u$_e$ represents the degrees of freedom associated with the centroidal node, and u$_r$ the degrees of freedom associated with the rest of the nodes. Following the static condensation procedure (Cook, 1981), the condensed finite element equations may be written as

$$[K_r^c] \{u_r\} = \{F_r^c\} \tag{6}$$

where $[K_r^c] = \Big[[K_{rr}] - [K_{re}]\,[K_{ee}]^{-1}\,[K_{er}]\Big]$; and $\{F_r^c\} = \{F_r\} - [K_{re}]\,[K_{ee}]^{-1}\,\{F_e\}$. The local stiffness matrix and force vector of equation (6) are assembled into the global stiffness matrix and force vector using standard finite element procedures.

In summary, the approach proposed above modifies a Delaunay tessellation by merging poorly shaped elements with adjacent elements to generate polyhedra that are then locally subdivided into better shaped tetrahedra. No nodes are rearranged in this process. Static condensation is used so that no new degrees of freedom are added to the global system of equations.

3 Results

The underlying goal of mesh quality improvement is improved accuracy of the desired analysis. It is shown through a series of examples considered here that not only does the proposed method improve the quality of the element shapes, but more importantly, it also improves the assembled matrix condition number and reduces the level of error in subsequent analyses.

3.1 Shape Quality

Element shape quality is an important topic for the meshing community, where meshes are refined to improve the quality factors associated with the worst elements in a given mesh. There are several quality factor definitions available in the literature, and most of these are targeted to a specific class of problems. As mentioned earlier, Shewchuk (2002) proposed a scale-invariant quality metric, Q, that is targeted towards the solution of the Poisson equation, $-\nabla^2 u(x,y,z) = b(x,y,z)$. This metric was listed in Equation 1 and relates to the interpolation error of the solution gradients as $\|\nabla u_{exact} - \nabla u_{interpolated}\|_\infty$.

The quality metric, Q, for a sliver approaches zero, which indicates that the solutions obtained by using slivers in a mesh will have errors with large gradients. The joined tetrahedra approach proposed above yielded elements with quality numbers of 0.084 for all the analyses performed on a set of test problems and reported below in this paper. Recall that a regular tetrahedron has a quality measure of 0.104103. For domains with nodes arranged along a regular cubic grid, the joining method will yield cubic polyhedra that are subdivided into similarly shaped tetrahedra, as shown in Fig. 5. Each tetrahedron shown in Fig. 5 is expected to have a quality metric of 0.084.

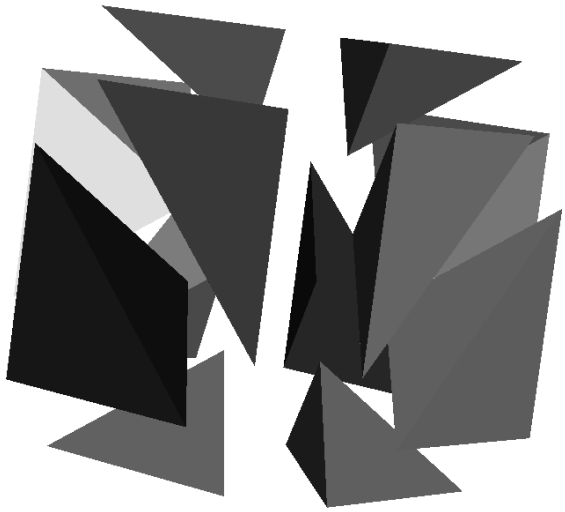

Figure 5 : A cube partitioned into 12 similarly-shaped tetrahedra

In the course of analyzing the set of examples, we detected that between 0.3% and 16% of the total tetrahedral elements generated by an initial Delaunay tessellation have a quality metric less than 0.002. The analysis results were significantly affected by the poor quality elements, even for the case with 0.3% poor quality elements. The present joining approach never had a quality factor less than near the expected value of 0.084. These results indicate that the present joining method improves the quality of the worst shape elements.

3.2 Condition Number

As a sliver approaches zero volume, the derivatives of the shape functions employed for the slivers approach infinity. Compared with a well-shaped element, the slivers exhibit much stiffer characteristics. When slivers are assembled into a global stiffness matrix, there may be several orders of magnitude difference between the respective diagonal terms. Past experience has shown that these differences in diagonal terms can lead to significant solution errors. The condition number is a measure of the sensitivity of a matrix to small errors. Matrices with large condition numbers are termed ill-conditioned. Ill-conditioned matrices reduce the accuracy for direct solvers and often require longer solution times for iterative solvers (Golub and Van Loan, 1996). Preconditioning may accelerate the solution times, but it is preferable to have a well-conditioned matrix to start with.

A common measure of the condition number, κ, for a stiffness matrix, [K], is $\kappa = \|K\| \left\|K^{-1}\right\|$. For symmetrical matrices that are often encountered in the finite element method, the matrix norm, $\|K\|$, is related to the spectral radius of the matrix, ρ(K), which is the largest eigenvalue (Golub and Ortega, 1992) of the stiffness matrix. The condition number is then given as $\kappa = \rho(K)\rho(K^{-1})$. Note that this measure for the condition number is computationally expensive, since a matrix inverse and two eigenvalue solutions are required. A less expensive approach can be expressed in terms of the ratio of the largest to smallest eigenvalues of the original stiffness matrix as

$$\kappa_R = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{7}$$

The stiffness matrix [K] is often pre-scaled using a diagonal matrix to increase its usefulness (Cook, 1982) as

$$[K_{scaled}] = [S]\,[K]\,[S] \tag{8}$$

where $S_{ii} = \frac{1}{\sqrt{K_{ii}}}$.

As a simple example, consider a unit cube partitioned equally into 8 sub-cubes with its nodes initially spaced uniformly on a 3x3x3 grid. The nodes are then perturbed randomly in all three spatial coordinates up to an amount of +/-0.5x10^{-7} so that a unique Delaunay tessellation can be generated. One internal sliver element was generated by this procedure and had the nodal coordinates of (0, 0.5, 0.5), (0, 0.5, 0), (0.5, 0.5, 0), and (0.5, 0.5 + ε, 0.5) where the value of ε was varied from 10^{-16} to 10^{-8}, as illustrated by the inset sketch in Fig. 6. Consider the case where the fourth sliver node is perturbed by a small amount, ε, in the y-direction to provide a small positive volume. As shown in Fig. 6, both measures for the condition number are similarly reduced as the point is perturbed towards creating greater volumes for the sub-cube. The double precision calculations presented here are not able to compute volumes for perturbations smaller than 10e-16. If the sliver truly had a zero volume, then the derivatives of the finite element shape functions for the sliver will be infinite and the system will not be solvable. The present approach for joining elements produced a matrix condition number less than 6 (six) in all cases, including cases when slivers with truly zero volume were merged into nearby elements. Thus, merging slivers into nearby elements allows solutions that could not be obtained otherwise, reduces the condition number, may speed iterative solutions, and improves the accuracy of direct solvers.

3.3 Error and Accuracy

While the previous two sections show that merging poorly shaped elements into polyhedra improves the mesh quality and reduces the condition number of the assembled stiffness matrix, this section demonstrates that the present merging procedure significantly improves the accuracy of the desired solution.

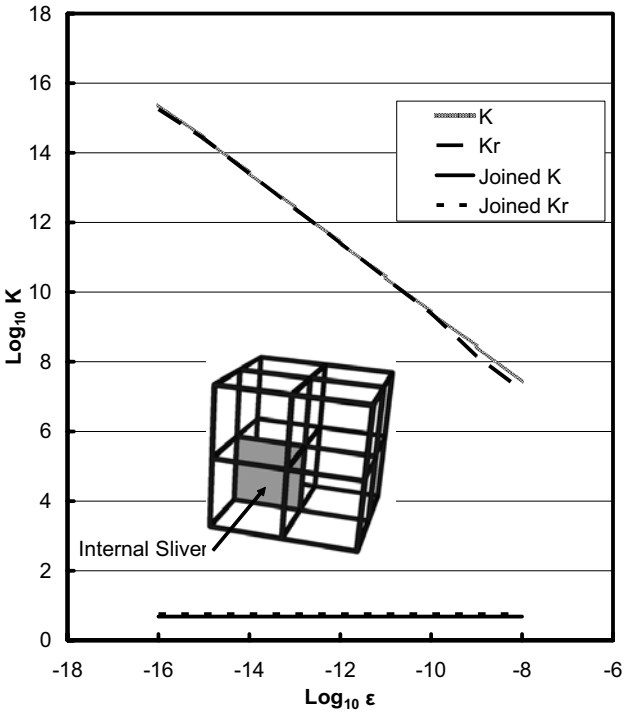

Figure 6 : Effect of sliver height on matrix condition Number. ε is the perturbation magnitude and K is the condition number.

The example of Poisson's equation on a unit cube with an internal energy source from Calvo, Idelsohn, and O˜nate (2003) and Idelsohn, O˜nate, Calvo, and Del Pin (2003) is studied here with the energy source b(x,y,z) given as

$$b(x,y,z) = \frac{-e^{(kxyz(1-x)(1-y)(1-z))}}{1-e^{(k/64)}}$$
$$\left(\begin{array}{l} 2kxy\,(1-x)\,(1-y) - [kxy\,(1-x)\,(1-y)\,(1-2z)]^2 \\ +2kxz\,(1-x)\,(1-z) - [kxz\,(1-x)\,(1-z)\,(1-2y)]^2 \\ +2kyz\,(1-y)\,(1-z) - [kyz\,(1-y)\,(1-z)\,(1-2x)]^2 \end{array} \right) \tag{9}$$

The exact solution for this system is obtained as

$$u(x,y,z) = \frac{1 - e^{(k\,x\,y\,z\,(1-x)\,(1-y)\,(1-z))}}{1 - e^{(k/64)}} \tag{10}$$

It is noted that there was a typographical error in Calvo, Idelsohn, and O˜nate (2003) and Idelsohn, O˜nate, Calvo, and Del Pin (2003) for the internal source term b(x,y,z). The error has been duly corrected here.

The energy source problem of equation (9) was considered using a constant k=200 on a unit cube with three different values of nodal spacing. The cube was discretized into a uniform 3D grid with 5, 9, and 17 nodes on an edge. Since a Delaunay tessellation of a uniform cubic grid will produce slivers with zero volumes, the non-corner nodes of the grid were perturbed in all three directions by a random amount with up to three different tolerances of 10^{-4}, 10^{-3}, and 10^{-2}, respectively. The resulting Delaunay tessellation still contains slivers, but the slivers have non-zero volumes. The Delaunay tessellation was generated using the commonly available algorithm QHull (2004). Since the QHull program may further perturb the nodes of an input grid, the perturbed nodes were read back from the QHull output to assure that the correct geometry was used in subsequent analyses. For the Element Joining procedures, a tolerance constant, δ=0.01, was employed for comparing the closeness of the centers of adjacent elements according to equation (2).

The L_2 error employed in the present study is defined as

$$L_2 = \left\{ \int_{\Omega} \left(u_{exact} - u_{approximate}\right)^2 d\Omega \right\}^{\frac{1}{2}} \tag{11}$$

The L_2 error is plotted in Fig. 7 for 3 different sized meshes for the joined and unjoined partitions, respectively. The joined approach had the expected convergence rate (slope of 2 on the log-log plot), whereas the unjoined approach converged more slowly and had higher errors overall. The perturbation magnitude clearly affects the element quality as shown in Fig. 7, where larger values of the perturbation generate better quality elements and have lower error. Note that the unjoined approach failed for perturbations of 10^{-5}, since some elements were of such poor quality, that the stiffness matrix could not be computed using double precision. The joined approach has essentially the same magnitude of error, regardless of the amount of perturbation.

The results from the present approach are shown in Fig. 7 and are seen to be slightly better than the results presented by Calvo, Idelsohn, and O˜nate (2003) and Idelsohn, O˜nate, Calvo, and Del Pin (2003), who employed the shape functions of the NEM. Presumably, the numerical integrations for computing the stiffness matrix integrals are more accurate for the finite element solutions, since the shape functions are polynomials and can be integrated exactly using one point gauss integration. The NEM shape functions are non-polynomials and may require more integration points for the same accuracy.

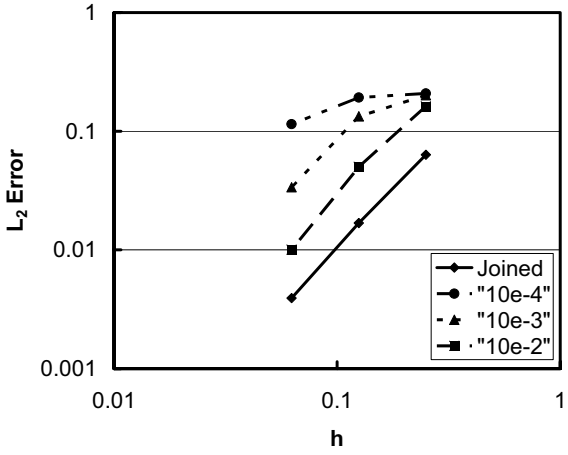

Figure 7 : L_2 convergence rates for joined analyses and analyses with slivers. h is the distance between nodes along an edge.

4 Summary and conclusions

A method for handling slivers and low quality tetrahedron elements has been presented. The method starts with a Delaunay tessellation. Poor quality tetrahedra are then joined together with adjacent tetrahedra to create polyhedra. A temporary node is next added to the centroids of the non-tetrahedral polyhedra generated by this process. Each polyhedron is then subdivided into a collection of local tetrahedra by connecting the temporary centroidal node to all of the external triangular facets of the polyhedron. The polyhedron stiffness matrix is generated by computing and assembling the stiffness matrices of the local tetrahedra that share the temporary centroidal node. The temporary centroidal node is eliminated using static condensation. The resulting stiffness matrix is assembled into the global stiffness matrix.

The method reduces the matrix condition number, increases the element quality for the lowest quality elements, allows solutions for truly poor quality slivers, and provides the theoretically expected L_2 error convergence rates when slivers are present. The method neither adds nor rearranges points. Sliver exudation is also not necessary. The joining process is a local process and has a computational cost proportional to the number of elements. The method retains the power, speed, and convergence rates of the standard finite element procedures.

References

Calvo, N.; Idelsohn, S. R.; O˜nate, E. (2003): The extended Delaunay tessellation, Engineering Computations: *Int J for Computer-Aided Engineering*, 10 September, vol. 20, iss. 5-6, pp. 583-600(18).

Cheng, S.-W.; Dey, T. K.; Edelsbrunner, H.; Facello, M. A.; Teng, S.-H. (2000): Sliver exudation. *J. Assoc. Comput. Mach.* 47, pp. 883-904.

Cheng, S.-W.; Poon, S.-H. (2003): Three-Dimensional Delaunay Mesh Generation. Also see "Graded conforming Delaunay tetrahedralization with bounded radius-edge ratio," *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 295-304.

Cheng, S.-W.; Dey, T. K.; Ramos, E. A.; Ray, T. (2004): Quality Meshing for Polyhedra with Small Angles, *SoCG* '04, June 8–11, Brooklyn, New York, USA. ACM.

Cook, R. D. (1981): Concepts and Applications of Finite Element Analysis, Second Edition John Wiley & Sons, Inc., New York, ISBN 0-471-03050-3.

Delaunay, B. (1934): Sur la sphère vide. Bulletin of the Academy of Sciences of the U.S.S.R. Classe des Sciences Mathématiques et Naturelle, Series 7 (6), pp. 793-800.

Edelsbrunner, H.; Li, X.-Y.; Miller, G.; Stathopoulos, A.; Talmor, D.; Teng, S.-H.; Ungor, A.; Walkington, N. (2000): Smoothing and cleaning up slivers. In "Proc. 32nd ACM Sympos. *Theory Comput.*, pp. 273-277.

Edelsbrunner, H.; Guoy, D. (2002): An experimental study of sliver exudation, *Engin. with Computers* 18, pp. 229-240.

Golub, G. H.; Ortega, J. M. (1992): Scientific Computing and Differential Equations, An Introduction to Numerical Methods, Academic Press, New York.

Golub, G. H.; Van Loan, C. F. (1996): Matrix Computations, 3rd ed., Johns Hopkins University Press, Baltimore, MD.

Han, Z. D.; Atluri, S. N. (2004): Meshless Local Petrov-Galerkin (MLPG) approaches for solving 3D Problems in elasto-statics, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 6, No. 2, pp. 169-188.

Idelsohn, S. R.; O˜nate, E.; Calvo, N.; Del Pin, F. (2003): The meshless finite element method, *Int. J. Numer. Meth. Engng* 58, pp. 893–912.

Li, Q.; Shen, S.; Han, Z. D.; Atluri, S. N. (2003): Application of Meshless Local Petrov-Galerkin (MLPG) to Problems with Singularities, and Material Discontinuities, in 3-D Elasticity, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 4, No. 5, pp. 571-586.

Li, X.-Y. (2000): Spacing Control and Sliver-free Delaunay Mesh, Proceedings, 9th International Meshing Roundtable, Sandia National Laboratories, pp. 295-306, October.

Qhull (2004): www.qhull.org, The National Science and Technology Research Center for Computation and Visualization of Geometric Structures, (The Geometry Center), University of Minnesota.

Shewchuk, J. R. (1998): Tetrahedral Mesh Generation by Delaunay Refinement, Proceedings of the Fourteenth Annual Symposium on Computational Geometry (Minneapolis, Minnesota), pp. 86-95, Association for Computing Machinery, June.

Shewchuk, J. R. (2002): What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures, Eleventh International Meshing Roundtable (Ithaca, New York), pp. 115-126, Sandia National Laboratories, September.

Sukumar, N.; Moran, B.; Yu Semenov, A.; Belikov, V. V. (2001): Natural Neighbour Galerkin methods, *International Journal for Numerical Methods in Engineering*, vol 50, pp. 1-27.