

FORMATION OF PYRAMID ELEMENTS FOR HEXAHEDRA TO TETRAHEDRA TRANSITIONS

Steven J. Owen

*Sandia National Laboratories
Albuquerque, NM, U.S.A.*

and

Sunil Saigal

*Carnegie Mellon University;
Pittsburgh, PA, U.S.A.*

RECEIVED
MAR 01 2000
OSTI

ABSTRACT

New algorithms are proposed for the modification of a mixed hexahedra-tetrahedra element mesh to maintain compatibility by the insertion of pyramid elements. Several methods for generation of the pyramids are presented involving local tetrahedral transformations and/or node insertion near the hex/tet interface. Local smoothing and topological operations improve the quality of the transition region. Results show superior performance of the resulting elements in a commercial finite element code over non-conforming interface conditions.

Keywords: pyramid, transition element, mesh generation, multi-point constraint

1. INTRODUCTION

Many automated tetrahedral mesh generation methods have become commonplace in the CAE industry. In contrast, all-hexahedra *free meshing* methods have proven to be more challenging^{1,2,3,4}. Some engineering disciplines have shown a preference for hexahedral elements over tetrahedra^{5,6}. While all-hexahedral element meshes may be preferred, some tetrahedral elements may be acceptable within the same model. For example, blocky or easily mapped or swept regions of the solid may be first filled with hexahedra. Regions that may be more critical to the analysis, such as boundary layers or regions of high stress may also be better

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

served by hexahedra. Tetrahedral elements may fill the remaining, more geometrically complex, or less critical regions of the solid. Other applications may include Hex-Dominant^{7,8} mesh generation, where the volume is filled with hexahedral elements using an advancing front procedure, leaving more complex regions on the interior of the volume to be filled with tetrahedra. Geometrically, since two tetrahedra faces are required to interface with a single hexahedron, discontinuities will arise at the boundary between the two element types. Traditional uses of the finite element method require that elements conform. In two-dimensions, this principle implies that no single element edge will have more than two elements adjacent. In three-dimensions, no single face will have more than two adjacent elements. Edge a-b and Face c-d-e-f in Figure 1 have more than the maximum two adjacent elements, thus rendering the mesh mathematically deficient for finite element analysis. This paper presents several solutions to the problem of non-conforming meshes that contain tetrahedra and hexahedra elements.

Dewhirst^{9,10} explores several possible alternatives for interfacing hexahedral elements directly with tetrahedra. While recognizing that displacement discontinuities will arise at the diagonal edge (Edge c-e in Figure 1), an attempt is made to minimize the error. When ten-node tetrahedra are used, the mid-diagonal node (g) can be constrained to displace as a function of the four surrounding corner nodes by means of multi-point constraints (MPC's). To constrain the mid-node, the solution of four non-linear simultaneous equations is required. Bretl¹¹ discusses the formulation of the MPC equations. While an improvement in results over unconstrained mid-nodes is noted, the computational expense in regions where MPC's are defined can be severe.

Interfacing directly with four-node tetrahedra was also proposed. While acceptable error was reported¹⁰ in a limited number of test cases, it is well documented^{5,6} that four-node tetrahedra tend to be overly stiff, performing poorly in structural applications.

An additional alternative is to simply drop the mid-diagonal node on the ten-node tetrahedron. While some commercial finite element applications will allow dropped mid-side nodes, significant error can be introduced at the discontinuity.

The ANSYS¹² finite element code also provides a general purpose alternative similar to MPCs. *Interface constraint equations* can be defined on the model where nodes do not match with nodes of adjacent elements. This procedure is typically used for interfacing coarse element

meshes with finer meshes, but can be used to constrain the mid-diagonal node between a tetrahedron and hexahedron. While some reduction in error across the interface is achieved, conformability is nevertheless violated. The error can be reduced as the mesh is refined.

Although these methods for interfacing hexahedra to tetrahedra are widely used, it should be noted that most require modification of the finite element code to handle these special cases. Even when options are available in an existing finite element code to handle discontinuities, it is not unusual to require significant user interaction to specify appropriate conditions at the interface nodes. An automatic method is required for interfacing the two element shapes without the need for complex constraint equations or modifying an existing finite element code. Automatic formation of pyramid elements at the interface between hexahedra and tetrahedra appears to be a promising solution and is addressed in this study.

Various pyramid finite element formulations, including five-node and thirteen-node formulations are available in commercial FE codes. Pyramid element formulations in the form of a degenerate hexahedron are also available in the literature where one face of the hexahedron collapses to a single node. Figure 2 shows the basic 20-node hexahedron, along with its degenerate 13-node instance of a pyramid. The eight nodes on the top face have been collapsed into a single node. The pyramid element, including in its degenerate hexahedral form, has been used for many years in some commercial finite element codes.

To assess the accuracy of the pyramid element, results from an all-pyramid mesh were compared with all-tetrahedra and all-hexahedra meshes under uniaxial tension and bending in a simple cubical volume. To generate an all-pyramid mesh, a grid of hexahedra was first generated and each hexahedron was subdivided into six pyramids by placing a node at the hexahedron centroid. Analysis was run on the commercial FEA code, ANSYS, and results from a simple set of patch tests are shown in Table 1. For the uniaxial tension case, similar to tetrahedra and hexahedra, pyramids predict the analytical results exactly. For bending studies, the results showed that quadratic, 13-node pyramids performed reasonably well, equivalent to the 10-node tetrahedra. Although poor results were exhibited by linear pyramids under bending conditions, these results were consistent with those of linear tetrahedra under similar loading conditions.

2. PYRAMID MESHING

Three principle methods are presented for insertion of pyramid shaped elements in an existing mixed element mesh at the interface between hexahedra and tetrahedra. The first method involves modifying the hexahedra, while maintaining the tetrahedral elements unchanged. The other methods propose transformations to the tetrahedra while maintaining the hexahedral elements. These methods and their performance are described in the following section.

2.1 Hexahedral Node Insertion

Compatible interfaces between hexahedra and tetrahedra may be defined by simply inserting an additional node at the centroid of all hexahedra immediately adjacent to a tetrahedron. This process involves splitting the hexahedron into seven elements; two tetrahedra and five pyramids. Figure 3 shows the results of this process where, for example, hexahedral elements A,B,C,D are immediately adjacent tetrahedral elements, as in Figure 3(a). The dashed diagonal lines at the hexahedral faces represent the non-conforming condition at the interface between the two element shapes. Figure 3(b) shows each hexahedral element at the interface, split into two tetrahedra and five pyramids where an additional node has been inserted at the centroid of each hexahedra. If tetrahedra are adjacent at additional faces of a hexahedron, additional pyramids can be split to generate more tetrahedra.

While this method is relatively simple, and tends to produce reasonable quality elements, some drawbacks exist. It is usually the hexahedra for which the user has spent additional effort to place within the model. Unless the elements are sufficiently fine that removal of some of the interface hexahedral elements will cause little difference, most users would tend to object to the modification of this one layer. In addition, this method tends to generate too many elements, increasing the number of elements near the hex/tet interface by a factor of seven or more. For this reason, this method was not pursued further.

As it is preferable that the pyramids be formed in the region otherwise occupied by tetrahedra, this can be accomplished either prior to the tetrahedral meshing operation, or as a post-process. As an *a-priori* operation, after placement of the hexahedral elements, pyramids can be constructed from each of the quadrilateral faces of the interface hexahedra. Although not implemented as part of this study, the *a-priori* procedure would require tedious intersection

checks to ensure the pyramids generated did not interfere with opposing boundaries or elements. It would also involve logic to resolve compatibility where intersections would occur. These special procedures would be similar to those needed for an advancing front-type scheme¹³ for tetrahedral mesh generation. Once the pyramids had successfully been formed, a tetrahedral mesh generator could then be used to fill the remaining volume.

In contrast, a procedure for forming pyramids after the initial formation of tetrahedra would not involve advancing front-type intersection checks. Instead, local transformations are performed on the tetrahedra nearby the interface hexahedral elements. Provided the volume of all tetrahedra remains positive during these operations, overlapping elements will not be an issue. This paper presents two such methods for forming pyramid elements as a post-process to the tetrahedral meshing operation.

2.2 Pyramid Open Method

The first method, implemented as part of this study, consists of *opening* a pyramid from the quad face of a hexahedron. The base of the pyramid is formed from the quad face, and the triangle faces of the pyramid are formed from the tetrahedra. As shown in Figure 4, initially, a node (C) is introduced at the midpoint of the diagonal edge A-B. All tetrahedra sharing an edge with A-B are then split. This is accomplished by traversing around edge A-B and splitting each adjacent face. For example, in Figure 4, the tetrahedron initially defined by nodes A-B-N₁-N₂ becomes two tetrahedra defined by C-B-N₁-N₂ and A-C-N₁-N₂. A new pyramid element can then be formed from the nodes A-N_n-B-N₁-C. The pyramid can be *opened*, by performing a constrained Laplacian smooth on node C, as shown in Figure 4(b). The new location of C is defined as:

$$C = \frac{A + B + \sum_{i=1}^n N_i}{n + 2} \quad (1)$$

where n is the initial number of faces adjacent to edge A-B, and N_i are nodes at opposite ends of edges from node C. Node C is also constrained to ensure that the tetrahedra will not become inverted. All adjacent tetrahedra are checked to ensure a minimum element shape metric (described later). If any tetrahedron fails, a new location for the node is computed by taking a weighted average of its initial location and the location last computed. This process

continues until valid tetrahedra and pyramid are defined. In the event a valid location cannot be defined using the above procedure, a method employing optimization techniques¹⁴ may be used for locating the node. The node location is optimized based on the objective of maximizing the minimum shape metric for any element adjacent to node C.

An additional case may arise if, as a result of previous insertion of pyramid elements, edge A-B is adjacent to an existing pyramid element, as shown in Figure 5(a). In this case, a pyramid (N₁-N₂-N₃-A-B) has previously been inserted in order to maintain transitions to an adjacent hexahedron. This case will arise only if the tetrahedral transformation method, described later, is employed in conjunction with the pyramid open method. Consequently, it is necessary to split the pyramid, as shown in Figure 5(b) into three elements: two tetrahedra (C-N₁-B-N₂, C-N₂-B-N₃) and one pyramid (N₁-N₂-N₃-A-C). Doing so maintains the quad base of the existing pyramid, while providing the topology enabling the insertion of a new pyramid.

One additional nuance to the pyramid open method occurs when two of the triangular faces of a tetrahedron form the quadrilateral face of a hexahedron as shown in Figure 6. In this example, tetrahedra A-B-E-D is the only tetrahedra adjacent the hexahedron. In this case, splitting edge A-B to form node C would be unsuccessful. Instead, edge E-D would be split. This in turn requires that all tetrahedra adjacent edge E-D be split. The resulting pyramid element is comprised of Nodes A-B-D-E-C. Node C can then be repositioned using the node relocation procedure described above. Although occurring infrequently, this case can arise when the interface between hexahedra and tetrahedra is non-planar.

The pyramid open method is robust and relatively simple to implement. It is rare that local geometry or topology will prohibit the algorithm from succeeding. In spite of its robustness, element qualities for tetrahedra adjacent to the pyramids tend to be lower. While clean up and smoothing can improve qualities dramatically, the tetrahedral transformation method, described next, tends to give higher quality and fewer numbers of elements.

2.3 Tetrahedral Transformation

The tetrahedral transformation method involves merging two adjacent tetrahedra to form a single pyramid element. The two tetrahedra whose faces form the triangles of the quad face of the adjacent hexahedron are the candidates for merging. Since it is likely that these two tetrahedra will not share a common face, it is necessary to perform the *swap23* operation.

The *swap23* is a basic transformation that defines three tetrahedra from what was initially two adjacent tetrahedra. The process is illustrated in Figure 7. A set of five vertices *abcde*, initially defining two tetrahedra *abcd* and *acbe* are transformed to define the three tetrahedra *aebd*, *becd* and *cead*. Any number of *swap23* operations may be performed until the topology is sufficient for the merging of two tetrahedra to form a pyramid. The necessary condition for this to occur is for the diagonal edge (edge A-B in Figure 1) to have exactly two adjacent tetrahedra or exactly three adjacent triangle faces.

The order in which swaps are performed may also affect the final quality of the pyramid and its neighboring tetrahedra. In order to locally improve the quality of the elements, the order of swapping is dictated by the maximum potential element quality of the resulting elements. An example of this procedure is illustrated in Figure 8.

To simplify visualization, Figure 8(a) shows a “top” view of the quadrilateral diagonal edge A-B (side view of the quadrilateral). The quadrilateral face in this case is defined by nodes A-N₁-B-N₅. There are initially four tetrahedra adjacent to edge A-B. The objective is to reduce this number to two. The dotted lines in Figure 8(b) show the three possible two-three swaps which can be performed. To select potential two-three swap transformation pairs, each pair of tetrahedra adjacent to a face sharing edge A-B is tested. A shape metric for each of the three potential tetrahedra to be formed as a result of a *swap23* operation is computed and their metrics summed. If a potential tetrahedron has a negative shape metric (the tetrahedron is inverted), the potential swap is immediately discarded as an alternative. The five-node set defined by nodes A-B-N₂-N₃-N₄ in Figure 8(b) shows a case where a negative shape metric would result if a two-three swap was performed. The maximum value for the sum of the metrics for the potential tetrahedra is used to select the transformation to be performed. In this case, the tetrahedra A-B-N₁-N₂ and A-N₂-B-N₃ were transformed into A-N₃-B-N₂, B-N₃-N₁-N₂ and N₁-N₃-A-N₂. As a result of the swap, only one of the three resulting tetrahedra is now adjacent to edge A-B, reducing the total number of tetrahedra adjacent to edge A-B by one as shown in Figure 8(c). This process is performed once again. Figure 8 (c) shows two potential transformations which could be performed. The *swap23* operation resulting in the highest quality elements is effected resulting in the configuration shown in Figure 8(d). At this point, the necessary conditions for the formation of a pyramid are satisfied. The two tetrahedra, A-B-N₁-N₃ and A-N₅-B-N₃, are merged to form the pyramid A-N₁-B-N₅-N₃.

It should be noted that geometric configurations of tetrahedra adjacent to edge A-B may not allow any two-three swaps to succeed. This occurs when all potential transformations result in at least one inverted tetrahedron. To avoid such a case, while infrequent, the pyramid open method is reverted to when geometric conditions do not allow local transformations.

The shape metric, Q , used to determine the order of transformations, is the ratio of respective radii of the inscribed to circumscribed spheres of the tetrahedron, and is given as

$$Q = 3 \frac{r}{R} \quad (2)$$

where r is the inscribed sphere radius and R is the circumscribed sphere radius. Joe¹⁵ presents a closed form expression of the metric in terms of the tetrahedron's node locations. This measure provides a maximum value of 1.0 for a regular tetrahedron. If the volume of the tetrahedron is less than zero, the element is inverted and the metric is modified to return a negative value.

3. MESH QUALITY IMPROVEMENT

While the methods presented above do not allow the formation of inverted elements, the quality of the pyramids and tetrahedra may be lacking. In order to improve local quality of elements in the transition region after the transformations have been performed, both smoothing and topological improvement are performed. Smoothing involves the repositioning of nodes to improve element shape, without changing element configurations, while topological improvement increases the quality of the mesh by altering the local connectivity of the nodes to form new elements.

3.1 Smoothing

The constrained Laplacian smoothing¹⁶ operation is performed locally within the transition region. To accomplish this, a local search is done beginning with nodes on the interface between the hexahedra and tetrahedra. Nodes within the volume that are topologically within three edges are marked as those needing to be smoothed. Laplacian smoothing involves repositioning nodes to reside at the centroid of its surrounding nodes that are connected by an edge. Since this technique may sometimes decrease the element quality and even invert otherwise reasonable elements, the decision to move the node is based on whether the minimum

shape metric of the node's surrounding elements is improved. If no improvement can be made, the node remains at its original location.

The constrained Laplacian smoothing operation depends on the definition of a shape metric. It was therefore necessary to define an appropriate shape metric for the pyramid element. While the inscribed to circumscribed sphere radius ratio can be used for tetrahedra, it can not be directly used for pyramids. For the pyramid, shape metrics of the four tetrahedra that can be derived from the pyramid, as shown in Figure 9, are first computed. The minimum radius ratio of the four tetrahedra is then used as the required pyramid shape metric.

The metric is further normalized by a constant, α , where α was experimentally determined to be approximately 0.86034. The constant α , was defined by computing the maximum shape metric for one of the four tetrahedra where the apex node is constrained to lie on a line normal to the base passing through its centroid. This metric defines an ideal pyramid with a base defined as a perfect square of length b and with apex node at height $h = 0.98154b$. As a result, an ideal pyramid will have a shape metric of 1.0. In this way, metrics for pyramids can be directly compared and summed with those of tetrahedra that are to be used in the smoothing algorithm.

Experimental results show that the proposed pyramid shape metric works well when utilized with constrained Laplacian smoothing. The pyramid metric can be directly compared with the tetrahedral shape metric from equation 2, with reasonable outcome.

Topological Improvement

The ability to swap faces to improve tetrahedra shape metrics is also used as a means of improving the element quality in the transition region. Joe¹⁵ provides the basis for tetrahedra improvement used in this study. Essentially, local permutations of connectivities are examined in order to find improvement in overall shape metric. Local transformations are performed on the existing tetrahedra by *flipping* faces to achieve this improvement. Of the transformations described by Joe, the T23, T32 and T44 flips were used. No changes were made to pyramids or to the boundary of the mesh.

A cubical domain was considered to evaluate the metrics for the elements generated using the proposed techniques. The results of forming pyramids from tetrahedral

transformations are shown in Figure 10. In this figure, of the 25 pyramids formed, two conditions existed in which local transformations could not be used. As a result, the pyramid open method was automatically invoked. Table 2 displays minimum and average shape metrics for pyramids and tetrahedra for the model shown in Figure 10.

The improvements in metrics for this domain due to smoothing and topological improvement were next evaluated. From trial and error, it was found that an initial smoothing iteration, followed by topological improvement, followed next by additional iterations of smoothing appeared to provide significant improvement, yet remain fairly cost effective. Table 2 lists the element shape metrics for both tetrahedra and pyramids after each step of the improvement operation for the model shown in Figure 10. The number of elements used for each shape as well as the minimum and average shape metric for the element shape are shown. After pyramids have been inserted, and without any topological improvement, tetrahedra are of unacceptable quality. A minimum tetrahedra shape metric of approximately 0.1 is generally considered acceptable. After smoothing and topological improvement, the element shape metrics are easily satisfactory for finite element analysis.

4. TEST CASES

Several tests were run, in order to determine the effect of various interface conditions on the analysis results. A recent version of the commercial finite element code, ANSYS, was used for analysis. A simple block divided by an arbitrary cutting plane, as shown in Figure 11(a), was used for testing. The cutting plane served as the interface between two volumes in which hexahedra or tetrahedra elements were generated. Figure 11(b) shows an example of the pyramid elements formed at the hex/tet interface. The evaluation involved defining various interface conditions at the interface plane and comparing their performance under uniaxial tension and bending.

Eleven different element configurations, as shown in Table 3, were tested. For the tension case, a uniform uniaxial surface pressure was applied on the opposite ends of the model. Minimal constraints were applied to inhibit rotations and translations. For the bending case, uniformly varying loads were placed on either end of the model to simulate equal and opposite moments. The maximum percent error with respect to the exact analytical solution for stress and displacement for the tension tests and stress only for the bending tests are shown in Table 3.

For tension, the percent error is computed from the nodal solutions on the interface plane. For bending, the maximum percent error is computed from nodes at the top and bottom surfaces farthest from the neutral axis. A result is listed as *exact*, if the percent error is on the order of 10^{-9} or less.

For each test shown in Table 3, either linear or quadratic hexahedra were generated in volume 1 with linear or quadratic tetrahedra generated in volume 2. Interface conditions included either placing two tetrahedra directly adjacent to a single hexahedron, or using pyramids formed by tetrahedral transformations or pyramid opens. The results corresponding to the use of constraint equations (CE) when incompatible nodes arise at the interface are also shown in Table 3.

In some cases it was necessary to drop midside nodes in order to maintain compatibility between linear and quadratic elements. For example, in cases 2 and 4, the Hex-8 and Tet-10 configurations required interfacing pyramids, (Pyr-9) with dropped midnodes. In this case, a full 13-node pyramid has the four nodes of its quad base removed. The ANSYS analysis code recognizes these edges as linear and handles them appropriately. For comparison, the final two cases illustrate the Hex-8 and Tet-10 with the midnodes not dropped. In cases 10 and 11, respectively, Hex-12, a linear hexahedron with four quadratic edges at the interface, is directly adjacent a full Pyr-13 or Tet-10 element.

The results shown in Table 3 clearly indicate the advantage of using pyramid transition elements over other hexahedra to tetrahedra transition methods. In the pure tension case, the pyramid elements are able to reproduce exact results for both stress and displacement. This is in direct contrast to non-conforming interface conditions where errors at the interface can be in the range of 30 percent for this model. While constraint equations appear to perform well in tension, poor performance for these equations was indicated for bending.

The results for bending also indicate favorable performance of pyramid transition elements. While linear pyramids with tetrahedra perform poorly under bending conditions, the results for quadratic transition pyramid are equivalent to results from all-hex and all-tet meshes with error in the range of 2 percent.

5. EXAMPLES

Two applications where this technology has been applied are illustrated here. A thermal finite element model of a heat sink is first considered and is shown in Figure 12. The coils, ideally represented by hexahedral elements must interface with the surrounding material, which cannot be easily subdivided into hexahedra. Both the entire model and a close-up of the exploded mixed element mesh are shown in Figure 12.

An electro-magnetic (EMAG) device shown in Figure 13 is next modeled using mixed element shapes. This device is a common benchmark problem used in the EMAG industry¹⁷. Both the device and the surrounding air must be filled with finite elements. Figure 13(a) shows the electro-magnetic device with the air modeled as a cylindrical shape surrounding it. Figure 13(b) and Figure 13(c) show the mixed element mesh of the device and the surrounding air, respectively. In this case, very high aspect ratio hexahedral elements were required to fill one of the volumes of the device. Figure 13(d) is a close-up showing the high aspect ratio pyramids required to interface between hexahedra and tetrahedra. Use of pyramid elements in this case reduced the number of elements needed to accurately model the device by a full order of magnitude¹⁸. In both cases, the hex elements were first meshed, and the remaining volumes were meshed with tetrahedra. The tetrahedra at the interface of the hex elements were then automatically converted to pyramids to maintain compatibility.

6. CONCLUSION

Several methods for interfacing tetrahedra directly with hexahedra have been presented. The formation of pyramid shaped elements appear to be a promising alternative for resolving incompatibilities between hexahedral and tetrahedral elements. Pyramids formed as a post-process to the formation of tetrahedra using tetrahedral transformations, in practice, provided the best quality elements. Where local topological conditions did not permit successful tetrahedral transformations, the pyramid open method was employed. Experimental results showed that the pyramid open method was used in about 10 percent of the cases. Also required was an additional step of smoothing and topological improvement operations to ensure elements were of acceptable quality for finite element analysis. Uniaxial tension and bending tests indicated that pyramid transition elements were in all cases superior to non-conforming

interface conditions. Pyramid transitions were also shown to be superior to multi-point constraint methods.

ACKNOWLEDGMENT

The authors wish to acknowledge the generous support of Ansys Inc. for the software computer resources and funding enabling the development of this research.

LIST OF FIGURES

Figure 1. Two and three-dimensional non-conforming meshes

Figure 2. 20-node hex and its 13-node degenerate instance of a pyramid

Figure 3. Exploded view of pyramid transitions defined within first layer of hexahedra. Tetrahedra (not shown) meet at triangle faces

Figure 4. The pyramid open method.

Figure 5. Splitting of existing pyramid to enable pyramid open.

Figure 6. Pyramid open used when a single tetrahedra is adjacent a hexahedra.

Figure 7. The *swap23* operation, where two tetrahedra are transformed into three.

Figure 8. Local tetrahedral transformations to define pyramid.

Figure 9. Division of a pyramid into four tetrahedra for pyramid shape metric.

Figure 10. Exploded view of pyramids formed using tetrahedral transformations. Hexahedra (not shown) adjacent to quad faces.

Figure 11. (a) Finite element model used for tests. (b) Exploded view of pyramids at the interface. Tetrahedra have been removed.

Figure 12. Heat Sink finite element model using tet, hex and pyramid elements.

Figure 13. EMAG benchmark problem modeled with pyramid interface elements.

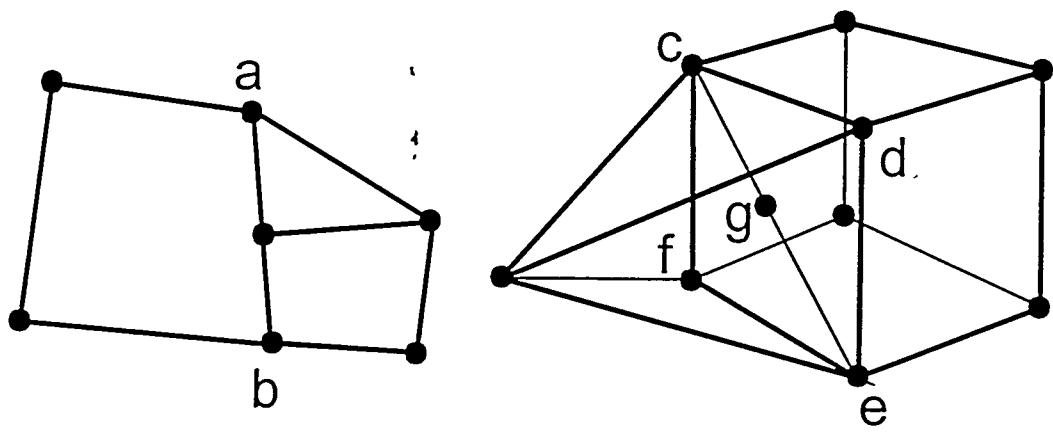


Figure 1. Two and three-dimensional non-conforming meshes

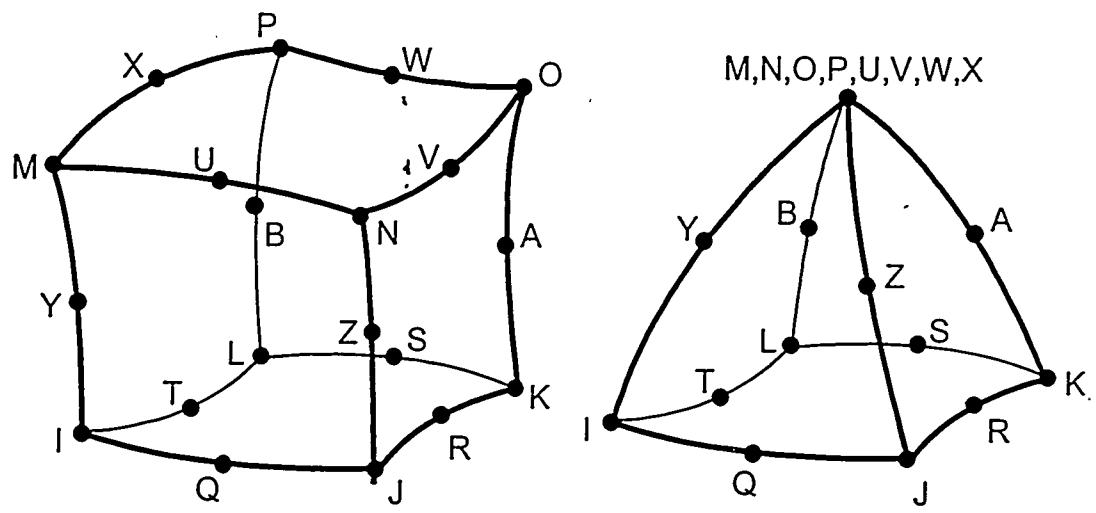
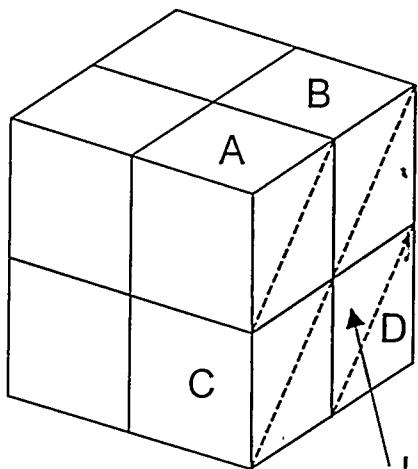
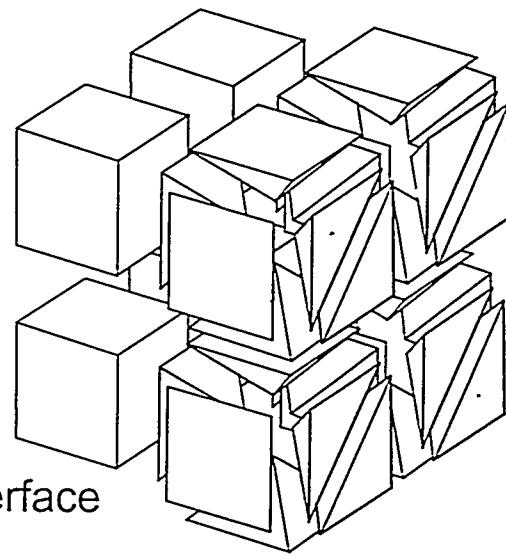


Figure 2. 20-node hex and its 13-node degenerate instance of a pyramid



(a)



(b)

Figure 3. Exploded view of pyramid transitions defined within first layer of hexahedra. Tetrahedra (not shown) meet at triangle faces

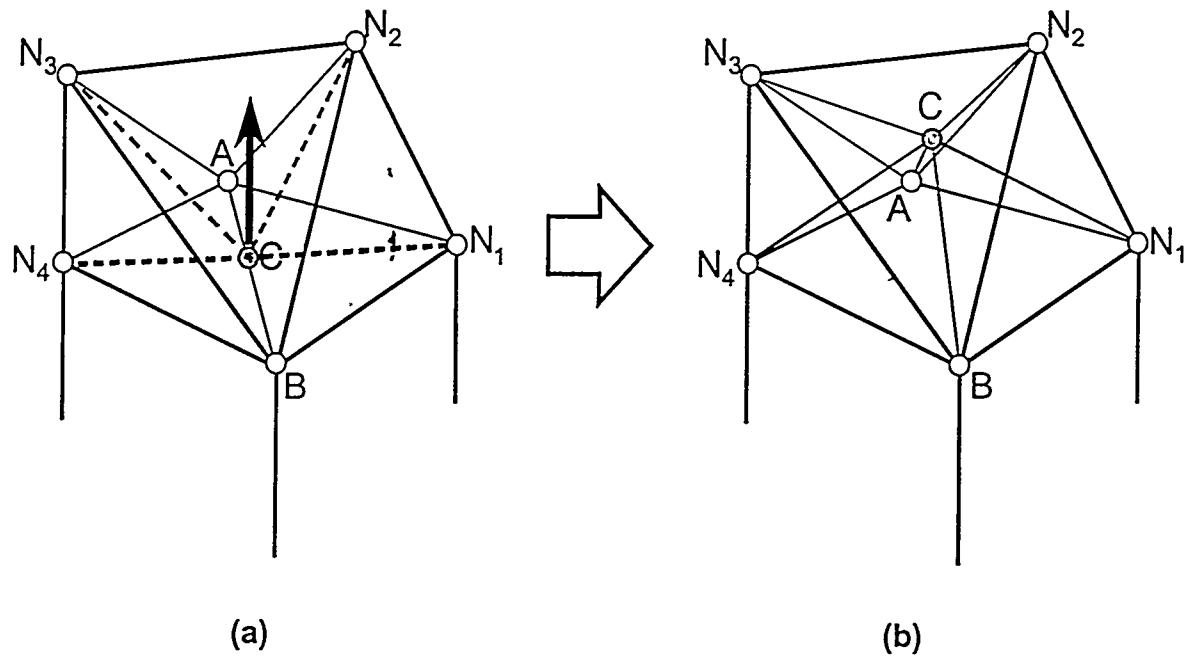


Figure 4. The pyramid open method

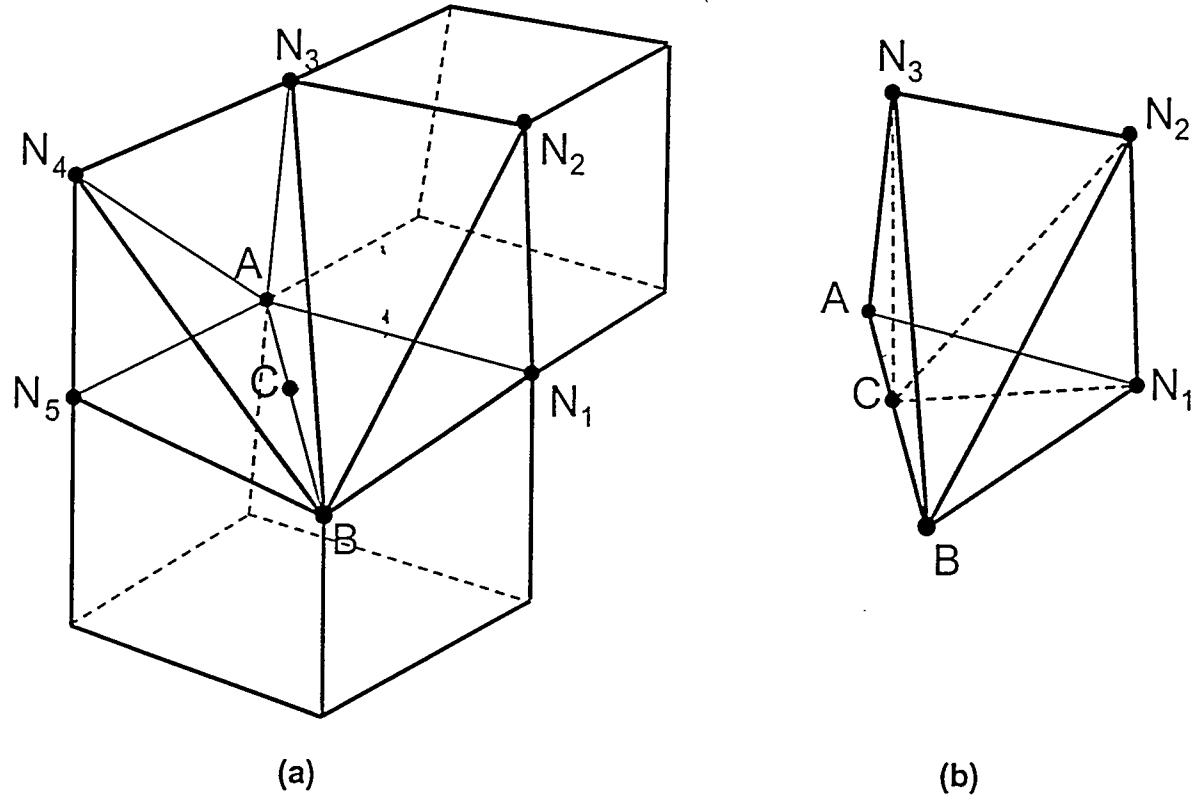


Figure 5. Splitting of existing pyramid to enable pyramid open.

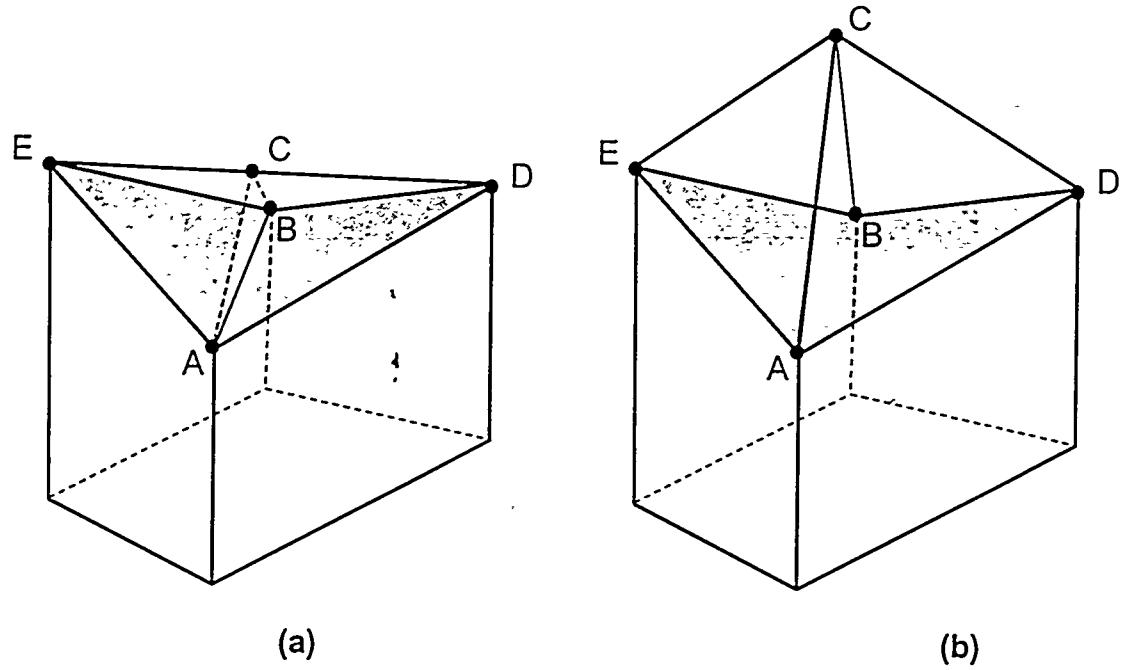


Figure 6. Pyramid open used when a single tetrahedra is adjacent a hexahedra.

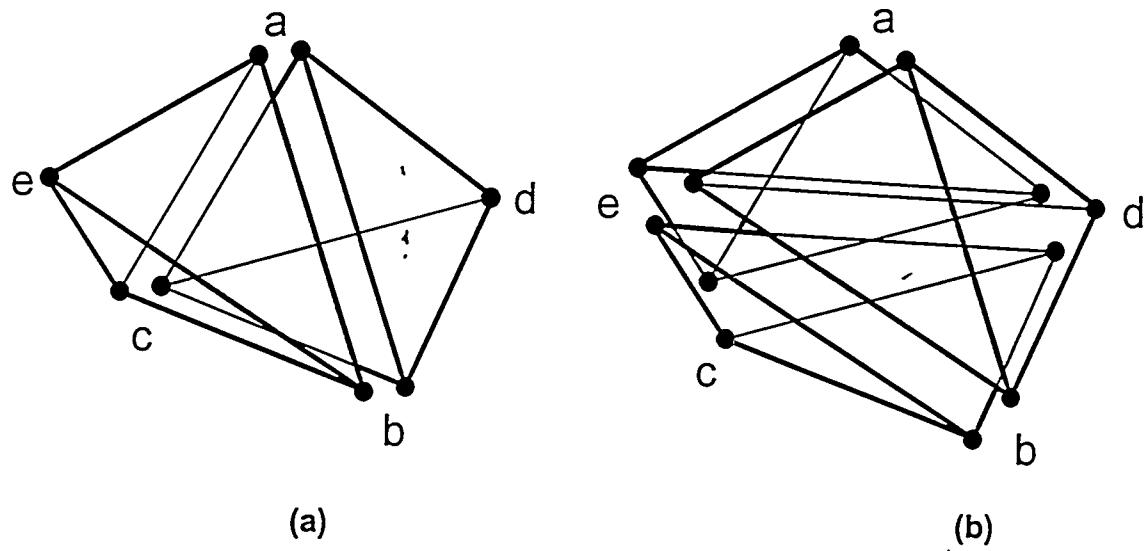


Figure 7. The *swap23* operation, where two tetrahedra are transformed into three.

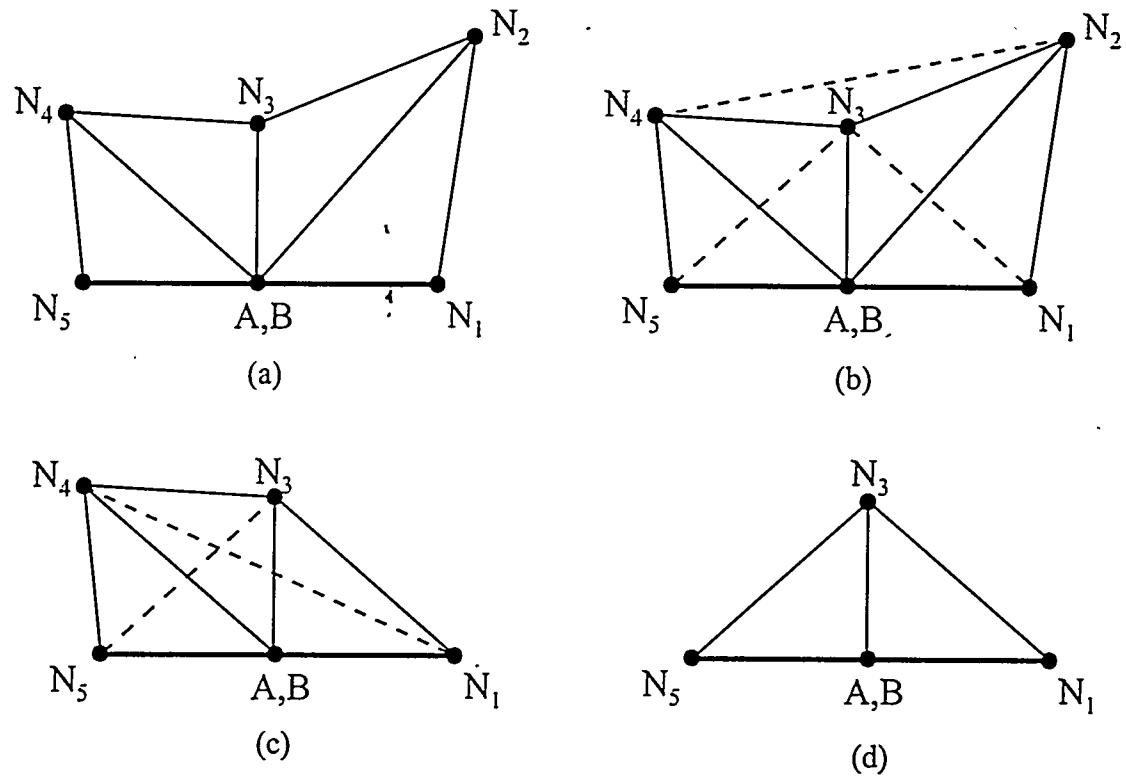


Figure 8. Local tetrahedral transformations to define pyramid.

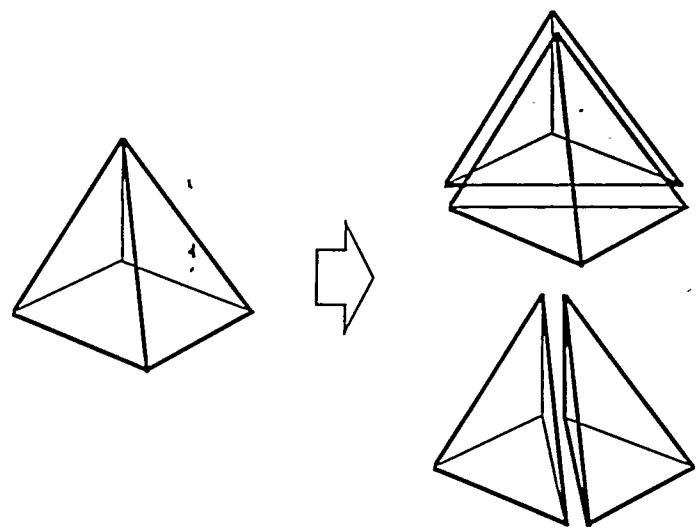


Figure 9. Division of a pyramid into four tetrahedra for pyramid shape metric.

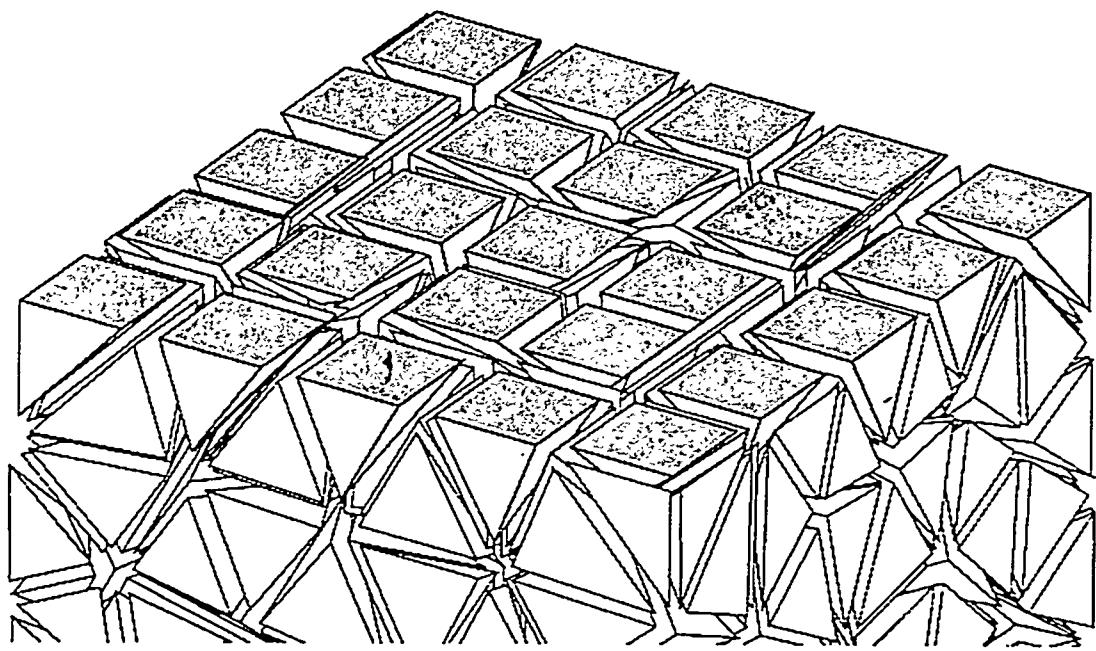
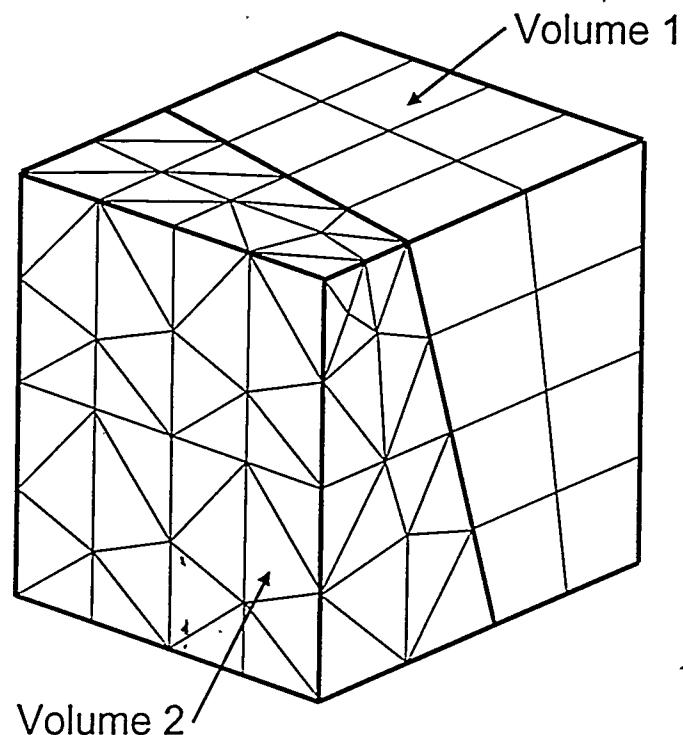
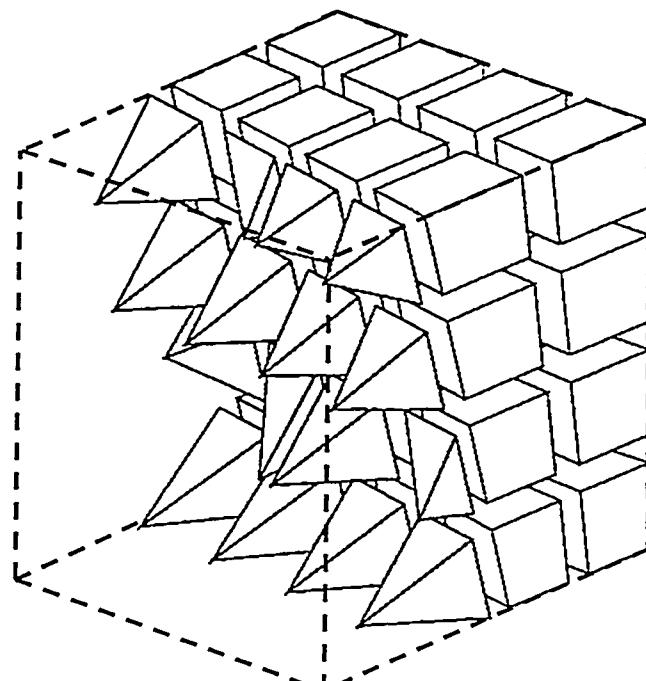


Figure 10. Exploded view of pyramids formed using a combination of tetrahedral transformations and pyramid opens. Hexahedra (not shown) adjacent to quad faces.



(a)



(b)

Figure 11. (a) Finite element model used for tests. (b) Exploded view of pyramids at the interface. Tetrahedra have been removed.

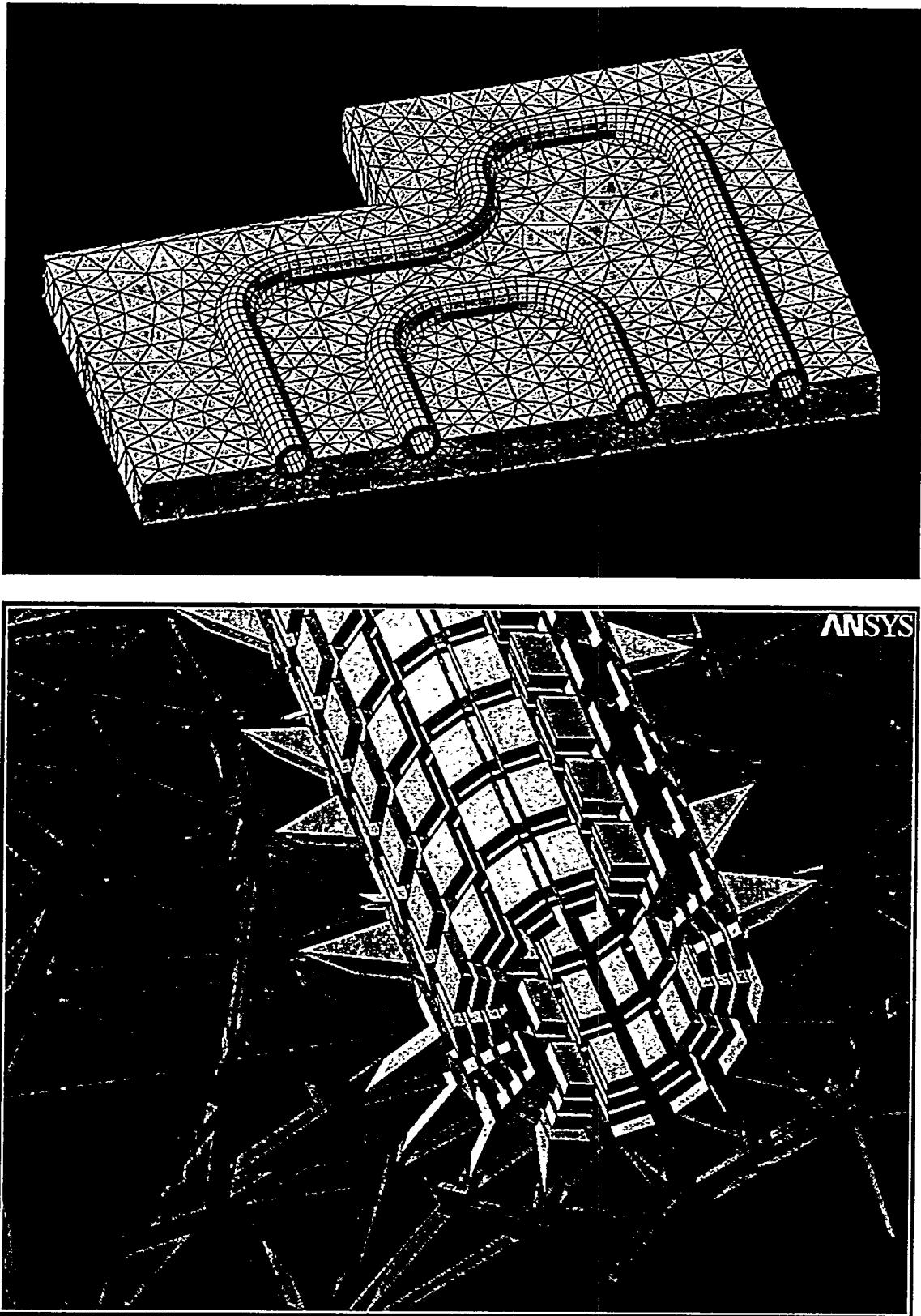
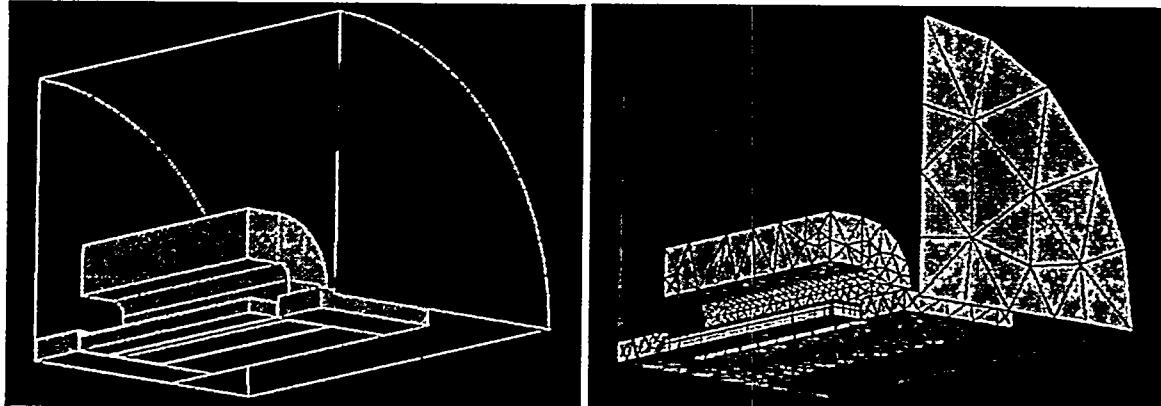
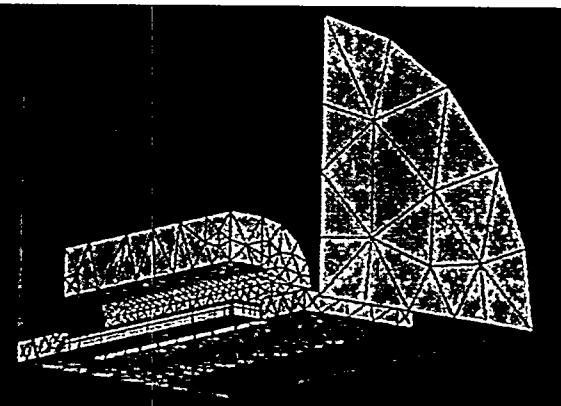


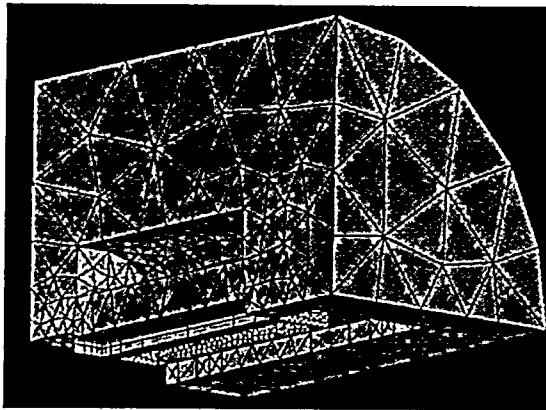
Figure 12. Heat Sink finite element model using tet, hex and pyramid elements.



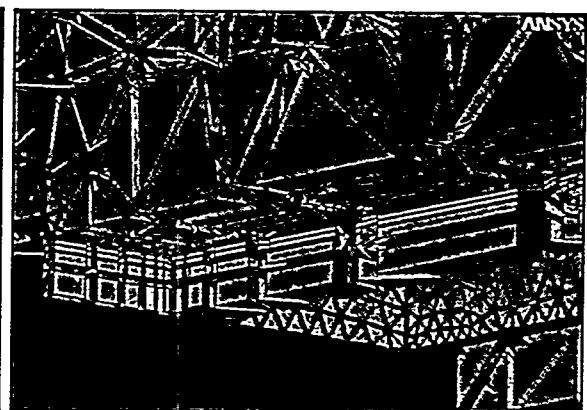
(a)



(b)



(c)



(d)

Figure 13. EMAG benchmark problem modeled with pyramid interface elements

LIST OF TABLES

Table 1. Comparison of FEM results for, all-tetrahedral mesh, all-hexahedral mesh and all-pyramid mesh.

Table 2. Comparison of element shape metrics after improvement operations performed.

Table 3. Comparison of FEM results under various hexahedra to tetrahedra interface conditions

	Elements in Model*	% Error		
		Tension		Bending
		Displacement	Stress	Stress
1	Hex-8	exact	exact	0.91
2	Tet-10	exact	exact	1.47
3	Pyr-13	exact	exact	1.77

*xxx-n refers to element shape xxx with n nodes

Table 1. Comparison of FEM results for, all-hexahedra, all-tetrahedra and all-pyramid mesh.

		Initial Mesh (tets only)	Pyramids inserted	Smooth	Topological Improvement	Final smooth
Tet	Num. elems	660	627	627	623	623
	Min. metric	.586944	.000016	.184845	.407329	.414901
	Avg. metric	.814500	.806665	.825427	.826503	.835560
Pyr	Num. elems	0	25	25	25	25
	Min. metric	N/A	.799915	.757268	.757268	.769436
	Avg. metric	N/A	.835968	.822949	.822949	.833507

Table 2. Comparison of element shape metrics after improvement operations performed.

	Elements in Model*			% Error		
	Vol 1	Vol 2	Interface Condition	Tension		Bending
				Displacement	Stress	Stress
1	Hex-20	Tet-10	Pyr-13	exact	exact	2.30
2			Tet-9	31.3	34.1	28.3
3			Tet-10 + CE	ord 10 ⁻⁴	ord 10 ⁻⁴	44.1
4	Hex-8	Tet-10	Pyr-9	exact	exact	3.88
5			Tet-7	16.5	29.3	26.1
6			Pyr-13 + CE	exact	exact	.04
7			Tet-10 + CE	exact	exact	40.0
8	Hex-8	Tet-4	Pyr-5	exact	exact	33.4
9			Tet-4	10.2	7.02	33.6
10	Hex-8	Tet-10	Hex-12 + Pyr-13	exact	exact	6.41
11			Hex-12 + Tet-9	30.5	34.7	44.7

*xxx-n refers to element shape xxx with n nodes

Table 3. Comparison of FEM results under various hexahedra to tetrahedra interface conditions

REFERENCES

- 1 T. D. Blacker and R. J. Meyers, 'Seams and Wedges in Plastering: A 3-D Hexhedral Mesh Generation Algorithm', *Engineering with Computers*, 9, 83-93 (1993).
- 2 M.A. Price and C.G. Armstrong, 'Hexahedral Mesh Generation by Medial Surface Subdivision: Part I', *Int. J. Numer. Meth. Engrg.*, 38, 3335-3359 (1995)
- 3 T.J. Tautges, T.D. Blacker and S.A. Mitchell, 'The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Meshes' *Int. J. Numer. Meth. Engrg.*, 39, 3327-3349 (1996).
- 4 R. Schneiders, 'A Grid-based Algorithm for Generation of Hexahedral Element Meshes', *Engineering With Computers*, 12, 168-177 (1996).
- 5 A.O. Cifuentes and A. Kalbag, 'A Performance Study of Tetrahedral and Hexahedral Elements in 3-D Finite Element Structural Analysis', *Finite Elements in Analysis and Design*, 12, 313-318 (1992).
- 6 S.E. Benzley, E. Perry, K. Merkley, B. Clark and G. Sjaardema, 'A comparison of All-Hexahedral and All-Tetrahedral Finite Element Meshes for Elastic and Elasto-Plastic Analysis', *Proc. 4th Int. Meshing Roundtable*, 1995, pp.179-191.
- 7 S.J. Owen, 'H-Morph, An Indirect Approach to Advancing Front Hex Meshing'. *Int. J. Numer. Meth. Engrg.* (accepted for publication 1999).
- 8 R.J. Meyers, T.J. Tautges and P.M. Tuchinsky, 'The Hex-Tet Hex-Dominant Meshing Algorithm as Implemented in Cubit', *Proc. 7th Int. Meshing Roundtable*, 1998, pp.151-158.
- 9 D.L. Dewhirst, S. Vangavolu and H. Watwick, 'The Combination of Hexahedral and Tetrahedral Meshing Algorithms', *Proc. 4th Int. Meshing Roundtable*, 1995, pp.291-304.
- 10 D.L. Dewhirst and P.M. Grinsell, 'Joining Tetrahedra to Hexahedra', *Proc. MSC 1993 World Users' Conf.*, 1993.
- 11 J. Bretl, 'Connecting Solid Finite Element Models That Have Dissimilar Meshes on the Mating Surfaces', *Proc. MSC/NASTRAN User Conf.*, 1984.

- 12 Ansys Version 5.6, © Ansys Inc. Canonsburg, PA (1999)
- 13 S.H. Lo, S.H., 'Volume Discretization into Tetrahedra - II. 3D Triangulation by Advancing Front Approach', *Comput. Struct.*, **39**, 501-511 (1991).
- 14 L.A. Freitag, 'On Combining Laplacian and Optimization-Based Mesh Smoothing Techniques', *Trends in Unstructured Mesh Generation, ASME, AMD 220*, 37-43 (1997).
- 15 B. Joe, 'Construction of Three-Dimensional Improved-Quality Triangulations Using Local Transformations', *Siam J. Sci. Comput.*, **16**, 1292-1307 (1995).
- 16 D.A. Field, 'Laplacian smoothing and Delaunay triangulations', *Comm. Appl. Numer. Meth.*, **4**, 709-712 (1988).
- 17 N. Takahashi, T. Nakata, M. Nakano, H. Morishige, K. Matsubara, J. L. Coulomb, J. C. Sabonnadiere, 'Improvement of Measurement of 3D Static Force Problem (TEAM 20)', EMAG Benchmark problem, 1998.
- 18 G. Miklos and D. Ostergaard, 'Mixed Shape Non-Conforming Edge Elements', *Proc. CEFC 98*, 1998.