MESHING OF HEXAGONS INTO CONVEX QUADRILATERALS

Nemanja Petrović & Branko Kolundžija

Beckman Institute &
Department of Electrical Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801
nemanja@ifp.uiuc.edu

Department of Electrical Engineering
University of Belgrade
Belgrade 11120, Yugoslavia
kol@kiklop.etf.bg.ac.yu

ABSTRACT

Efficient electromagnetic analysis of the composite metallic and dielectric structures in the frequency domain based on the Method of Moments applied to the Surface Integral Equations is provided, if building blocks have the form of bilinear surfaces (in particular, flat quads), and if analysis is supported by the higher order basis functions. Heterogenous surfaces of many 3D structures can be easily represented as a combination of connected non-overlapping polygons. Subdivision of the polygons into the minimal number of mutually connected flat quads of good shape is based on subdivision of the hexagons into the convex quads, with possible addition of new nodes only in the interior of the hexagons. We classified hexagons into 46 classes and for each class we found the subdivision scheme. We demonstrated that all subdivision schemes can be unified into four “cut and try” algorithms. The effectiveness of the approach is illustrated on a typical example. This method is implemented in the software tool for antenna design [6].

Keywords: hexagon, quadrilateral, mesh, combinatorial counting

1. INTRODUCTION

Problems in radar scattering, antennas and microwave fields involve objects made of conductors, insulators and their combinations. When these problems are treated in the frequency domain, numerical solutions are often based on the Method of Moments (MoM) [1] applied to the Surface Integral Equation (SIE) [2, 3]. In that case, the unknown quantities that are to be determined are electric currents induced over metallic surfaces and equivalent electric and magnetic currents placed over dielectric surfaces. Hence, 3D structures are described by their surfaces, i.e. the building blocks are in the form of surface patches (triangles, quads, etc.). Notice the difference between such geometrical modeling of 3D structures and that applied in the Finite Element Method (FEM) solution of differential equations [4]. Namely, in the later case the unknown quantity is the volume distribution of the field. Hence, 3D structures are described by their volumes, i.e., the building blocks are in the form of volume bricks (tetrahedrons, hexahedrons, etc.).

Application of the MoM to the SIEs results in the full matrices (full system of linear equations). If N is the number of unknowns, the memory re-
requirements and matrix fill time are proportional to $N^2$, and matrix inversion time is proportional to $N^3$. Thus, at PC computers the maximal size of the problem in number of unknowns is about $N = 15000$. For given type of basis functions the number of unknowns is proportional to the complexity of the problem and the area of boundary surfaces in $\lambda^2$, where $\lambda$ is the wavelength of electromagnetic waves. Hence, it is desirable that surfaces patches are geometrically flexible and that the number of corresponding basis functions per wavelength squared is minimal.

1.1 Bilinear surfaces (flat quads) and higher order basis functions

Basically, we can use the lowest order basis functions over each patch, which are not related with neighboring basis functions through the continuity equation. Maximal size of such patches is $0.1\lambda$, resulting in minimum 200 unknowns per wavelength squared for metallic surfaces. For dielectric surfaces the number of unknowns is doubled. Efficiency of analysis is significantly improved if basis functions are adopted in the form which automatically satisfy the continuity equation (e.g. rooftop basis functions). Such basis functions can be easily realized over triangles and quadrilaterals. Maximal size of such patches is about $0.15\lambda$, resulting in minimum 100 unknowns per wavelength squared. However, application of such sophisticated basis functions introduces the following restriction in the geometrical modeling: two neighboring patches must have a common edge between two common nodes, i.e., they must be electrically connected.

The most often used surface patches are flat triangles, flat quadrilaterals, and, recently, bilinear surfaces. (The bilinear surface is the simplest curvilinear quadrilateral defined by four arbitrary positioned vertices.) It is found that for the same accuracy of analysis, triangular mesh requires at least twice the number of unknowns compared to quadrilateral mesh [5]. Disadvantage of the flat quadrilaterals is that they are defined with four nodes placed in a plane, and hence they are not suitable for modeling of curved surfaces. This disadvantage is overcome by the use of more general bilinear surfaces, which in particular case degenerate into flat quadrilaterals. Regarding the efficiency, bilinear surfaces are preferable over the triangles. However, the modeling performed by bilinear surfaces is more restricted than that by triangles. For example, all triangle shapes are allowed, while concave quadrilaterals should be avoided.

Efficiency of analysis based on quadrilateral meshing is significantly improved by adding higher order basis functions to zero order basis functions. The order of approximation in each direction of the quadrilateral depends on electrical length of the quadrilateral in that direction. By increasing the electrical size of the patch the number of unknowns per $\lambda^2$ required for good analysis decreases. For quadratic patch of the size 1 by 1 $\lambda$ the number of unknowns required is about 30. Maximal order of basis functions that still give stable solution is 7 to 9. The patches that are suitable for these basis functions are two to three wavelengths in size, requiring about 20 to 25 unknowns per wavelength squared [6]. Therefore, we prefer bilinear surface and higher order basis functions that automatically satisfy continuity equation for currents at patch ends and junctions to the other types of surfaces patches and basis functions.

1.2 Goals of polygonal meshing

Maximal size of patches of two to three wavelengths enables users to completely describe some real structures (even electrically large), using relatively small number of patches. However, manual modeling of complicated shapes by bilinear surfaces can be very tedious. Hence, there is a necessity to provide users with different tools for easy modeling of complicated structures. Many complicated 3D structures can be easily represented as a combination of connected non-overlapping polygons. Note that from condition that polygons are non-overlapping follows that each side of a polygon is shared by at least one neighboring polygon, i.e., if we add a new node at the edge, neighboring polygon will be affected. Note also that polygons can be arbitrarily positioned in the space. Because of that, such meshing is more difficult than that applied in 2D problems solved by the FEM [4]. For example, in the case of 2D problems the meshing can be performed by putting simple uniform mesh over the plane in which 2D structure is placed. This can not be applied to the surface of a body.

The basic goal of meshing is to subdivide polygons into electrically connected convex quadrilaterals. Additional goals are that number of unknowns required for accurate analysis is minimal, and that meshing is performed quickly. Having in mind that better efficiency is achieved with larger patches, the goal of meshing is to introduce minimal number of new nodes (possibly zero). However, it is not possible to “pave” polygon with odd number of nodes by non-overlapping quadrilaterals. Moreover, it is not possible in the general case to subdivide polygon with even number of nodes into convex quadrilaterals without creating new
nodes either inside the polygon, or along the polygon boundary. Obviously, segmentation of polygons, which close surface of 3D structure, could be facilitated if all polygons are subdivided independently, one by one, i.e. without adding new nodes into their boundaries. Finally, it should be noted that number of unknowns can be decreased by using quadrilaterals of better shape.

One possible way of independent subdivision of polygons can be obtained by proper application of the advancing front technique ("paving") [7]. The technique is rather complicated and difficult to implement, but enables control of the size of elements, which is important in obtaining good shapes of patches when the FEM is applied. Additional problems in application of this technique is how to avoid complicated polygons that can remain by the end of the paving process [7, 8]. In order to obtain good shapes, usually very dense mesh is necessary. That may be expensive, because good shape and uniform size of the patches and is not so critical when the MoM solution of SIEs is combined with the bilinear surfaces and the higher order basis functions.

1.3 Shape quality factor for bilinear surfaces

In the case when SIEs are solved by the MoM the mesh quality factor (Q-factor) of flat quad is defined in the following way. Let us consider a quad shown in Figure 1. With three vertices fixed, \( \bar{r}_{uv} \)

\[ \bar{r}_{uv} = \alpha \bar{r}_u + \beta \bar{r}_v. \] (1)

Roof top basis function, which is spanned over two interconnected parallelograms, results in constant charge distributions over these parallelograms. When quads deviate from the parallelograms, the charge distributions deviate from the constants. This deviation decreases convergence of the results. Hence, the basis Q-factor is defined as ratio of the minimum and maximum charge distribution. After some manipulations it can be shown that Q-factor is expressed in terms of coefficients \( \alpha \) and \( \beta \) as [5]

\[ Q' = \frac{2 - |\alpha| - |\beta|}{2 + |\alpha| + |\beta|}. \] (2)

This definition assigns unit quality factor to parallelograms, positive values to convex quads, zero values to quads which degenerate into triangles (one angle is equal to 180\(^\circ\)), and negative values to concave quads. Since the order of approximation along one quad side can be from 1 to 10, a rectangle whose neighboring edges relate as 1 to 10 or even more is as much desirable as square. However, thin quads can cause stability problems if ratio of neighboring sides is greater than few hundreds. In order to avoid very thin quads we introduce an auxiliary shape quality factor as

\[ Q'' = \frac{\text{area of the quadrilateral}}{\text{length of the longest edge}^2}. \] (3)

Finally, an integral (composite) shape quality factor is defined as

\[ Q = \begin{cases} Q', & Q'' > Q_0 \\ Q'', & Q'' \leq Q_0 \end{cases}, \] (4)

where \( Q_0 \) is critical value of \( Q'' \) at which thin quads can cause numerical instability. Usually, \( Q_0 \) is adopted between 0.01 and 0.1.

It is shown that meshing, whose medium value of integral Q-factor is about 0.1, is more than satisfactory in the most cases. Moreover, increasing slightly the order of approximation can neutralize negative influence of quads with poor quality.
1.4 Why hexagons?

Having all this in mind and assuming that initial polygons are not self-intersecting, the following meshing procedure was suggested in [9]: (1) minimal number of nodes is added to polygons’ boundaries so that all polygons have even number of nodes; (2) initially two nodes of each polygon are connected, subdividing the polygon into two polygons with even number of nodes; the procedure is repeated until the initial polygon is subdivided into convex quadrilaterals, concave quadrilaterals, and hexagons; (3) hexagons are subdivided into two to five convex quadrilaterals introducing zero to three new nodes inside the hexagon, as will be described in this paper; (4) remaining concave quadrilaterals are merged mutually or with convex quadrilaterals into hexagons and these hexagons are subdivided according to step (3); (5) once we obtain a mesh of convex quadrilaterals we can apply different refinement techniques to improve the shape of quadrilaterals (e.g., we can apply step (4) to each pair of convex quadrilaterals that belong to the same initial polygon, or we can use experience from FEM meshing [10]).

Each of steps, (1) to (5), can be performed in different ways. In this paper we focus on step (3), i.e., how the hexagon can be subdivided into convex quadrilaterals introducing (if necessary) new nodes inside the hexagon. In [12] the authors have classified a few hexagons, and proposed obvious meshing rule. The classification of the hexagons was attempted in terms of the number of angles close to or greater than 180° (reflex angles).

The idea to merge two neighboring quadrilaterals into the hexagon, and perform hexagon meshing is not new. “CleanUp” [11] process has as its core routine “combine with neighbor” action. However, once the hexagon meshing is formalized, that and other “CleanUp” action routines like shape and size cleanup follow as a bonus of the hexagon meshing process.

All the above mentioned methods also assume that mesh was of the decent quality initially, so that hexagon obtained by merging of two quads is easy to resolve.

The paper is organized as follows. Deterministic method for hexagon subdivision is given based on detailed classification of hexagons in Section 2. More flexible subdivision can be obtained with four unified “cut and try” methods, without determining to which class the hexagon belongs, as shown in Section 3. The meshing of a real example consisted of few mutually connected irregular polygons is shown in Section 4. We finish with the discussion of applicability of our approach in Section 5.

2. DETERMINISTIC METHOD

It seemed natural to classify hexagons by the number of reflex angles (angles greater than 180°), and their mutual position. To resolve remaining ambiguities, we introduced visibility pattern for the hexagons. A point is “visible” from a vertex if and only if it is in the interior of the corresponding angle and if a line connecting the point and the vertex strictly lies in the interior of a hexagon. We refer to visibility as the number of diagonals that could be drawn from each vertex of a hexagon. Visibility is defined for each vertex and varies between zero and three. The intuition behind this is that two hexagons with the same topology and visibility pattern are partitioned in the same manner. We built the deterministic classifier, that takes as input the number of reflex angles, their mutual position, and visibility pattern. These three features are enough to uniquely determine the type of hexagon.

In the following tables we classified all relevant hexagons. We omitted those that are obtained using mirror reflection. The numbers next to vertices are visibilities of each vertex. The proof that these are all relevant hexagons is by construction. First, we have chosen the number of reflex angles and their position within a hexagon. Then we have varied the positions of the remaining vertices to obtain all possible visibility patterns. We mean...
by “solving a hexagon” to find a convex quadrilateral subdivision using zero, one, two or three points.

Table 1 presents the only hexagon with zero reflex angles, and eight hexagons with one reflex angle. The numbers in the vertices show the visibility. Examples (i-v) are partitioned using zero new nodes in the best case. Examples (vi-vii) require one additional node, and example (viii) two additional nodes.

Table 2 presents the hexagons with two reflex angles that are in directly opposite vertices. Examples (i-iii) are divided using zero new nodes, and others using two new nodes.

Table 3 presents the hexagons with two reflex angles that are in separated with one non-reflex angle. Examples (i-xii), are solved using one node, example (xiv) using two nodes, and example (xiii) using three nodes.

Table 4 presents the hexagons with two reflex angles that are in adjacent vertices. All examples are solved using two nodes.

Table 5 presents the hexagons with three reflex angles. Example (i) is the hexagon with three adjacent reflex angles. In examples (ii-vi) any two vertices with reflex angles are separated by a vertex with non-reflex angles. Examples (iii-v) are solved using one node, examples (i,vi) are solved using three nodes, and example (ii) can be solved using one or three nodes. Interestingly, ‘star’ example can be subdivided by one node if there is a point simultaneously visible from the vertices of the three acute angles.

Table 6 presents the hexagons with three reflex angles in such configuration that two are adjacent, and third is non-adjacent to previous two. Examples (i-v) are solved using two nodes, and example (vi) using three nodes.

We have examined each and every one of these examples and found the rule that gives the minimal number of convex quadrilaterals that subdivide the hexagon according to scheme pictured at Figure 2. Most of the partitions are trivial. Those that are more complicated will be covered in Section 3. It is not difficult to see that all partitions from the topological point of view are performed in one of four ways shown in Figure 2, i.e. by inserting 0, 1, 2, or 3 nodes inside the hexagon. Observing
Even though the deterministic procedure is fast and robust we face two challenges. First, there is quite a bit of redundancy in the implementation because many shapes divide using the same or slightly modified procedure. Second, it is obvi-

3. Unified Cut and Try Method

these similarities we came out with four unified “cut and try” procedures. We will therefore omit the descriptions of 46 resolutions and focus on “cut and try” procedures.

Table 3. Hexagons with two reflex angles that are separated by one non-reflex angle.

Table 4. Hexagons with two adjacent reflex angles.

Table 5. Hexagons with three reflex angles.
ous that in some cases adding an extra node could boost the quality factor and we are looking for a mode to subdivide a hexagon using more nodes than minimum. This leads to a generalized “cut and try” approach. It uses no a priori information about the structure of the hexagon, and does not know what would be the minimal number of nodes to divide hexagon. It will consecutively try to mesh the hexagon using zero, one, two, and three nodes. The procedure will try to “cut” quadrilateral from the hexagon by introducing one new node. The same procedure is repeated for the remnant of the hexagon. Obviously, there might be more than one resolution for a single hexagon. We can then devise the trade-off between the number of new nodes and the quality factor.

This set of algorithms is implemented recursively: the method checks if a hexagon can be divided using \( i \) nodes, adds the first node, and calls the procedure for division using \( i + 1 \) nodes.

In all cases a simple rule is followed: a quadrilateral is convex if the diagonals meet. By adding new nodes in the interior of the hexagon, we try not to ruin the existing visibility that is crucial for meshing, and we choose the position of new nodes within the ‘feasible space’ that will guarantee visibility. For example, in Figure 4(i), node \( T_1 \) is chosen to be visible from \( B, D, \) and \( F \), and not to obstruct the view from \( A \) to \( C \).

The notation of the vertices becomes important in the “cut and try” method. Because we have used no knowledge about the shape of the hexagon, for the current notation of hexagons the method can
deduce that division is not possible, even if we know that it actually is. Also there might be multiple solutions. That is why all six possible notations of the vertices have to be checked out, as well as six counterclockwise notations that correspond to mirror image cases. Although this may seem excessive, there are a few fast checks at the beginning of each of the following methods, that will, in case they are not satisfied, eliminate the current hexagon notation from further consideration.

We now show the “cut and try” procedures for division using zero, one, two, and three nodes.

**Zero node division.** Check if subdivision is possible and subdivision are trivial - a diagonal between directly opposite vertices is drawn. If there is more than one solution, maximum being three, the one with the highest quality factor will be chosen.

**One node division.** There are two necessary conditions for a hexagon to be divided into three quadrilaterals. First, for a given vertex notation all reflex angles must be either among vertices \( A, C, \) and \( E \), or \( B, D, \) and \( F \). Second, there must be direct visibility from \( A \) to \( C \), \( C \) to \( E \), and \( E \) to \( A \) (Figure 3(i)). The first condition follows from the fact that three angles in non-adjacent vertices (\( B, D, E \)) must not be reflex, because this angles will not be divided, meaning they will remain reflex. This is why a hexagon with two adjacent reflex angles can not be subdivided using one node. The second condition follows directly from Figure 3(i). One node is added following the rule: locate the \( \triangle ACE \) (shaded area at Figure 3(ii)); reflex angles, if any, must be in vertices \( A, C \) or \( E \); newly added node must be within this triangle, and visible from \( B, D, \) and \( F \). The rays of angles in \( B, D, \) and \( F \) effectively reduce the feasible space to the dark shaded quadrilateral within

![Figure 3. “Cut and try” procedure for division using one node. Difficult example.](image-url)
\( \triangle ACE \) (Figure 3(ii)). The centroid of a quadrilateral is a good initial guess for the newly added node.

Difficult example, Figure 3(ii), shows how the initial feasible area for \( T_1 \) (light shaded \( \triangle ACE \)) was reduced by rays \( BC, DC, \) and \( FE \) to the dark shaded area.

If there is more than one solution, maximum being two, the one with the highest quality factor will be chosen.

Two node division. Figure 4(i) shows the “cut and try” method for division with two nodes.

There are two easy to check necessary conditions for division using two nodes. First, there must be a pair of directly opposite vertices with non-reflex angles (\( B, E \)). Second, there must be direct visibility from \( A \) to \( C \), and from \( D \) to \( F \), meaning that quadrilateral \( ACDF \) is the feasible space positioning \( T_1 \) and \( T_2 \). If quadrilateral \( ACDF \) happens to be concave the feasible space reduces to the part that is visible from all four vertices. Point \( T_1 \) is chosen to be visible from \( B, D \) and \( F \), but it must not be within \( \triangle ABC \), or \( \triangle DEF \). Once \( T_1 \) is fixed, then \( T_2 \) is chosen within hexagon \( AT_1CDF \), but in such a fashion not to obstruct the view from point \( T_1 \) to points \( D \) and \( F \) (must be within the \( \angle FT_1D \)).

If we use a requirement that all new quadrilaterals must be convex, then point \( T_1 \) must be simultaneously visible from \( D \) and \( F \), and point \( T_2 \) must be simultaneously visible from \( A \) and \( C \). We start with area of quadrilateral \( ACDF \) restricted to the part visible from all four vertices (all shaded area in Figure 4(ii)). Feasible area for \( T_1 \) is further restricted to the area visible from \( B \) (right darker shaded area), and feasible area for \( T_2 \) is further restricted to the area visible from \( E \) (left darker shaded area). If we choose \( T_1 \) first, then \( T_2 \) must be in \( \angle FT_1D \), and if we choose \( T_2 \) first, then \( T_1 \) must be in \( \angle AT_2F \). In other words, \( T_1 \) shall not ‘shadow’ \( T_2 \) and vice versa.

What if, after fixing the point \( T_1 \), the remaining hexagon cannot be divided using one node? It is not the position of \( T_1 \) to ‘blame’, but the fact that the remaining part is too complicated to be divided by one node. We used no \textit{a priori} knowledge about the hexagon, so we do not know in advance if there are reflex angles or their mutual position (e.g., there might be two reflex angles in

\textbf{Figure 4.} “Cut and try” procedure for division using two nodes. Difficult example.
adjacent vertices of the remaining part).

**Three node division.** The problem seems much more difficult to deal with than two nodes division. There are too many interdependencies between three points. It is not quite clear what should be the feasible area for the positions of three new nodes. We restrict to an easier problem: we require \( T_1 \) to be within the \( \triangle FBC \) (Figure 5(i)). It is a reasonable restriction, because for good quality we may look for three nodes symmetrically distributed as in Figure 5(i). Of course, it may happen that \( T_1 \) is on the other side of the diagonal \( FC \), but that case will be surely discovered for other notation of vertices.

So, we restrict the position of \( T_1 \) to \( \triangle FBC \) at Figure 5(ii,iii). We require therefore \( \angle FBC \) to be non-reflex, and we require visibility from \( B \) to \( F \).

In all five cases that could only be divided using three nodes, the following three rules led to a solution.

1. If \( D \) obstructs the view from \( C \) to \( F \), move \( F \) along \( FB \) to intersection of \( CD \) with \( FB \) (to the point \( F_1 \)).
2. If \( B \) obstructs the view from \( C \) to \( A \) move \( C \) along \( CF_1 \) to intersection of \( CF_1 \) with \( AB \); i.e., move \( C \) to be visible from \( A \).
3. If \( F \) obstructs the view from \( C \) to \( A \), move \( C \) along \( CB \) until visible from \( A \).

This effectively reduces the feasible area for node \( T_1 \) to the shaded area in Figure 5(ii,iii). It is easy to check, that the remaining part of the hexagon can be divided using two nodes.

### 4. Complex Example

Finally, we show how this method can be used for efficient meshing of complex example, made of few polygons. (For the sake of clarity we show planar example, although in the general case the method is intended for 3D structures.) Figure 6(a) is an outline of a printed dipole antenna taken from [9]. After segmentation procedure described in the paper, we obtain mesh shown in Figure 6(b).

From the viewpoint of the FEM this mesh is of a poor quality. But, from the viewpoint of the MoM solution of the SIEs based on the higher order basis functions the solution is satisfactory (mean value of Q-factor is about 0.1). Due to independent meshing of polygons the CPU time needed for meshing is extremely short and negligible when compared with electromagnetic analysis time. Note that only six additional nodes is introduced by hexagonal meshing, which means that number of quadrilaterals used in the analysis is a little bit higher than theoretical minimum.

The refinement of polygonal meshes is not used in this case. It is shown in this case that refinement can improve mean Q-factor, but cannot provide arbitrary high mean Q-factor. The reason is that initial polygons in this example are particularly irregular (ratio of maximal to minimal edge length is about 50). One possible way to significantly improve the Q-factor is to add even number of nodes onto polygons’ boundaries, in such a manner that ratio of maximal to minimal edge length is lower than some predefined value. For example, we can add 22 nodes marked by hollow circles to the initial outline, as shown in Figure 7(a). After segmentation procedure described in the paper, combined with refinement technique, we obtain mesh shown in Figure 7(b). The mean Q-factor of this mesh is about 0.5. The number of nodes introduced by hexagonal meshing is 13. Note that adding 22 nodes to polygons’ boundaries does not affect simplicity of the paving technique based on the fact that all polygons are independently segmented by iterative subdivision into smaller polygons.

Once we have quantified the quality of a mesh as a function of the shape of quadrilateral and the position of new nodes, we can perform gradient optimization technique to find the position...
of new nodes that maximizes the quality factor. We use minimax principle for optimization: we try to maximize the minimal shape quality factor of \( n, n = 3, 4, 5 \), quadrilaterals obtained in the process of hexagon meshing. In each iteration of gradient optimization, we maximize the worst quality factor

\[
\max_{i=1, \ldots, n-2} \min Q_i.
\]

This guarantees that, for sufficiently small initial step in gradient descent, nodes move only in the feasible space (do not ruin the convexity of quadrilaterals), consistently improving the worst quality factor.

5. CONCLUSION

Two methods for subdivision of hexagons into convex quadrilaterals, deterministic and unified cut and try method, are presented. The goal of segmentation is to insert minimal number of nodes (0, 1, 2, 3) inside the hexagon, without adding new nodes onto hexagon boundary. Deterministic method first determines to which of 46 classes the hexagon belongs (regarding the number and positions of the reflex angles, and visibility between non-adjacent nodes), and apply specific procedure for this class. All these procedures are unified into four cut and try procedures, i.e., procedures based on inserting 0, 1, 2, and 3 nodes, respectively.

The usefulness of the methods is illustrated in the case of specific meshing of surfaces of 3D structures analyzed in frequency domain by the MoM applied to the SIEs. The authors hope that proposed techniques could be useful in the meshing of 2D problems solved by the FEM.

The authors believe that hexagon meshing should be used in the synthesize-analyze loop for mesh refinement. Beyond the action of merging two quads into a hexagon, Figure 2 gives an idea that three, four or five quads in the specific configuration could also be merged into hexagon. Subsequent hexagon meshing could significantly improve connectivity, size and shape of the elements in the mesh.

ACKNOWLEDGMENTS

Thanks are due to anonymous reviewers for valuable comments, and for pointing out the references [10, 11]. The authors are grateful to Alla Sheer for the encouragement and help while writing this paper. The authors wish to thank Robert Fossum for his ongoing support.

REFERENCES


