

# A novel energy-based approach for merging finite elements

Or Yogev, Andrew A. Shapiro and Erik K. Antonsson\*,<sup>†</sup>

*California Institute of Technology, Pasadena, CA 91125, U.S.A.*

## SUMMARY

A novel approach for merging two intersecting finite elements is presented and demonstrated. The solution mimics concepts from biology and uses principles rooted in continuum mechanics.

The problem of attaching (or merging) two coincident finite elements is common when using the plastering technique as part of the advancing front method. This problem is particularly challenging for 3-D meshes of non-convex shapes. Some automatic meshing methods require portions of the partially formed mesh to coincide and merge. This problem is generally solved with heuristic rules, which lack generality, and may have difficulties with unforeseen situations.

The problem of merging two overlapping polyhedra may also appear in other applications such as computer graphics and CAD software.

A new approach to address the problem of merging is presented here. This solution does not utilize heuristic rules, but rather uses an approach based on minimization of strain energy. A fully automatic merging routine has been created that can address, in an optimum way, any situation of two nearby or overlapping elements that are to be merged. This approach, with minor adjustments, is suitable for most types of 3-D elements. Copyright © 2010 John Wiley & Sons, Ltd.

Received 9 June 2009; Revised 18 May 2010; Accepted 25 May 2010

KEY WORDS: meshing; element merging; morphogen; strain energy; energy minimization; 3-D

## 1. INTRODUCTION

The problem of attaching (or merging) two coincident finite elements is common when using the plastering technique as part of the advancing front method [1, 2]. Generation of a mesh for non-convex shapes utilizing the advancing front method usually starts at multiple locations along the boundary and proceeds towards the center. At some point during the process, each of the individual portions of the mesh must coincide and merge. This problem, from a geometric viewpoint, is complex because the adjacent element orientations are generally not able to be predicted, and it is difficult to determine *a priori* the best configuration for merging adjacent elements. As a result, this type of problem is generally solved with heuristic rules or a lookup table approach [3]. These

---

\*Correspondence to: Erik K. Antonsson, California Institute of Technology, Mail Stop: 104-44, 1200 East California Blvd., Pasadena, CA 91125, U.S.A.

<sup>†</sup>E-mail: erik.antonsson@caltech.edu

rules handle different situations in different ways and are primarily determined by the skill and experience of the programmer. The disadvantages of using heuristics include lack of generality, and difficulties with unforeseen situations.

A new approach to address the problem of merging is introduced here. This solution does not utilize any heuristic rules. Instead it uses a natural approach based on minimization of strain energy that can automatically merge two cells into an optimum merged configuration. Other processes that are usually integrated with the advancing front method to generate good quality meshes (such as healing techniques [4] and mesh quality control [5, 6]) are integrated into the new energy minimization merging approach presented here.

Meshed regions can consist of a large number of elements, so a robust and computationally efficient method for merging elements is critical. The method introduced here produces valid meshes for all cases, and utilizes an analytical expression for the conjugate gradient method to minimize computational cost.

### 1.1. Prior related work

Despite considerable work, the problem of generalizing meshes for arbitrary non-convex three-dimensional bodies remains an open area of research. Excluding mesh-free methods, 3-D meshes generally comprise either tetrahedral or hexahedral elements. In the tetrahedral case, three popular methods have been developed for automated mesh generation for three-dimensional bodies: *octree*, *Delaunay triangulation*, and *advancing front*.

The *octree* method was developed in the 1980's by Mark Shephard [7, 8]. The idea behind the octree method is that a given geometrical domain is subdivided into cubes that are successively subdivided until a desired resolution has been achieved. New methods have been recently developed to accelerate an octree-based mesh generation method using parallel algorithms and clusters of many computer processors [9, 10]. The main shortcoming of the octree method is that a pre-defined surface or volume is weakly matched to its representative mesh: the surface of the mesh is not conformal. Instead, more elements are formed near the boundary in order to refine the mesh to match the boundary [11].

On the contrary, the *Delaunay triangulation* method matches the body surface more accurately [12–14]. This method is based on the *Delaunay criterion*, which tests a triangulated set of vertices such that any circumsphere that predefines the mesh does not contain any other vertices [15]. This criterion is used in generating tetrahedral meshes for 3-D shapes [16–19]. Although the *Delaunay triangulation* method can generate a mesh for any given convex body defined as a convex hull, it usually fails to generate meshes for non-convex domains [20], although some interesting attempts to deal with this impediment have demonstrated partially successful results [17, 21].

The *advancing front* (AF) method deals with the issues of non-convexity successfully and today is becoming a common method for generating meshes for both convex and non-convex bodies. In the AF method, the tetrahedral or hexahedral elements are generated toward the interior part of a domain from its pre-meshed surface [2, 22]. This method is analogous to a wave transmitted in the body. In the case of tetrahedral elements, the algorithm first generates a triangular base on each facet of the front, and then optimizes the location of a fourth node. The fourth node might be a new or an existing node of an adjacent element. The algorithm keeps running a collision detection sub-algorithm to ensure that no elements are overlapping. The *advancing front* method has also been used with meshes of hexahedral elements [1, 3, 23, 24].

In a similar way to the tetrahedral case, the plastering method begins with predefined meshed boundaries that advance the fronts inwardly into the domain [25–28]. As the fronts collide, multiple geometric operations are performed to smooth and seam the elements in order to minimize the remaining void of the un-meshed area. By the end of this process a new element is inserted into the void, which completes the meshing procedure. This algorithm suffers from many deficiencies when the fronts collide because of the complexity of dealing with the remaining void. One of the solutions to this problem is to use a tetrahedral mesh for the remaining void and then to transform the tetrahedral elements into hexahedral ones [29, 30]. The drawback of this method is the poor quality of the hexahedral elements generated during the transformation step.

Acknowledging the complexity of hexahedral elements, the Q-Morph method starts with a triangulated surface mesh, which is transformed into hexahedral elements as the front progressively moves in the solid body [31]. The final shape of the quadrilateral mesh will be on the same scale as the initial surface mesh. The Q-Morph method also avoids expensive intersection calculations which are common to the advancing front method.

Similarly, the H-Morph method starts with a tetrahedral mesh, which is transformed into hexahedral elements as the front (initially a set of prescribed quadrilateral surface facets) moves progressively into the volume [3]. The H-Morph method is able to generate a mesh without the need to identify or process special element geometries.

The whisker weaving method is a purely topological mesh approach, defined as the geometrical dual of a mesh as a collection of surfaces, called the Spatial Twist Continuum (STC) [32, 33]. This method starts with a pre-quadrilateral surface mesh and then constructs a hexahedral connectivity mesh by advancing into the solid. Though improved, the whisker weaving approach has only been successfully used to generate meshes of relatively simple shapes [34].

Utilizing the simplicity and reliability of the plastering method, while avoiding the need to mesh complex voids, the unconstrained plastering method shows promising results in generating 3-D hexahedral meshes [24]. The unconstrained plastering begins with un-meshed surfaces where new hexahedral elements are formed by the intersection of the projected fronts of the surfaces, as they inwardly progress into the body. The unconstrained plastering is not a finished concept yet, but rather a work in progress [35].

### *1.2. New method*

This paper provides a novel method to deal with the challenge of the plastering method to create hexahedral element connectivity when several fronts intersect. Our approach is motivated by the challenge of attaching (or merging) two coincident finite elements in an optimum way as part of the plastering approach. The problem of merging two overlapping polyhedra may also appear in other applications such as computer graphics and CAD software.

The common solution for addressing merging of finite elements is to create a set of heuristic rules [3]. These rules test and categorize states of elements according to their relative position and orientation. Each state corresponds to a process that merges elements in a predefined manner according to a lookup table. The use of heuristic rules creates many problems. First, it requires a long set of conditional statements that introduce complexity into the code. Additionally, the rules must address and capture all situations regarding the relative position and orientation of the elements, and decide which kind of merging process to execute. Since the number of possible orientations is large, it is difficult to ensure that all possible situations are addressed. This is the motivation for developing a new method that automates the merging process between two finite

elements regardless of their orientations. In this paper a novel approach is presented that mimics merging phenomena observed in nature, such as the merging of biological cells and bubbles. By using the basic principles used in nature, a fully automatic merging routine has been created that can address any situation of two nearby or overlapping elements that are to be merged.

### 1.3. Overview

In Section 2 the idea of formulating each finite element as an artificial cell containing some particular properties is introduced. The principle of minimum potential energy, as observed in bubble merging processes, is also established and explained. Section 3 describes the governing equations for applying this method to an eight-node hexahedral linear brick finite element. Section 4 presents the derivation of the conjugate gradient method used here to minimize strain energy in the configuration of the merged elements. An analytical expression for the strain energy gradient has been derived, which significantly increases the speed of convergence. Section 5 discusses the computational cost of merging elements using this approach and compares it with other methods. Section 6 shows the results of the merging of two elements using this method. Several examples illustrate the capabilities of this approach in automating the merging process without using heuristic rules in Section 7. Conclusions are presented in Section 8.

## 2. APPROACH

Consider two finite elements (such as those shown in Figure 1(a)) that under certain conditions need to go through a merging process. Our approach is *independent* of the pre-condition setting, which can be the distance between center–center, face–face, node–face or intersection [36] of the paired elements. In this work we use an intersection condition to illustrate the technique as we believe it is the most challenging condition. The merging process takes one face from each element and merges their corresponding nodes.

The method introduced here comprises three steps: first, determine the corresponding faces to be merged; second, determine the pairs of nodes (one from each element) to be merged; and third, optimize the positions of the merged nodes.

Cells in nature, prior to merging, communicate by way of a chemical signaling mechanism [37] which activates the operation of other genes, regulated by the level of the signal exceeding a threshold [38]. This process is mimicked here by considering each finite element to be the analog of a cell. Each cell generates a communication signal at its center, and this signal propagates through the faces of the cell and into the adjacent cells, schematically illustrated in Figure 1(a). Each signal is characterized by an exponential function (1) that decays with distance from the center of the cell. Equation (1) approximates the behavior of a steady-state diffusion equation (as occurs in cell signaling). Although it is not a rigorous solution, solving the diffusion equation more accurately will not have any effect on the performance of this method (but will slow it down). Equation (1) is used to describe this process:

$$\vec{S} = \exp(-\alpha \vec{x}_c) \quad (1)$$

where  $\alpha$  is a constant,  $\vec{x}_c$  is the vector from the coordinates of the center of the cell, and  $\vec{S}$  is the gradient vector of the concentration (or strength) of the signal.

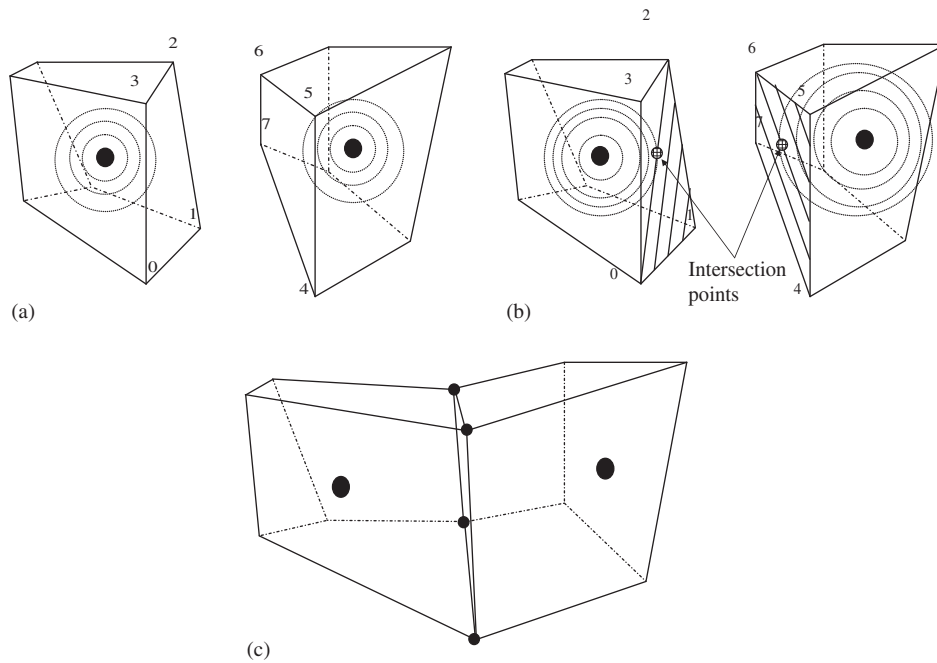


Figure 1. (a) Two cells prior to merging with radiating signals from the center of each cell; (b) the faces of the two cells to be merged; and (c) the cells after merging.

The face of each cell that contains the maximum intensity of the signal from the adjacent cell is chosen for merging, as illustrated schematically in Figure 1(b). Inverse isoparametric mapping [39] is then applied, which maps each cell to its local configuration. By inverse mapping the signal as well, the detection of the point with the maximum signal gradient on the adjacent face becomes straightforward.

Once the faces to be merged have been identified, the best four pairs of nodes to be merged must be selected. Each pair of nodes will be merged into a single node initially positioned at the *mean* of the coordinates of both original nodes. At the end of the process, the two cells merge to each other without any overlap, as illustrated in Figure 1(c).

For two hexahedral elements, there are a total of 24 different configurations of merged node pairs from which to choose. The selection of the optimum configuration mimics the process of bubble merging [40]. When bubbles merge, they adopt a shape that reduces their total merging surface energy. This principle minimizes the distortion of each bubble from a spherical shape.

Using this energy minimization principle for all 24 possible combinations of nodes to merge, the pairing of nodes that results in the minimum distortion to both cells, and thus introduces the minimum change in strain energy (when each pair of nodes is merged into a single node positioned at the *mean* of the original coordinates of the pair) is selected. Any contribution to the change in strain energy will originate only from the distortion of the cells. This idea is applied rigorously here by considering the constitutive properties of the materials of each cell to be quasi neo-Hookean, but only for the purpose of merging the cells.

Prior to the merging process, both cells are in a *reference configuration*. While the merging process is being applied, a finite displacement constrains each pair of nodes. The strain energy density relationship for neo-Hookean materials is given by  $w$  in (2). Note that Equation (2) is not the general neo-Hookean relation but rather a representation of incompressible material, nevertheless this relation is sufficient for the technique.

$$w = C_1(I_1 - 3) \quad (2)$$

where  $C_1$  is a constant, and:

$$\begin{aligned} I_1 &= \vec{C}' \cdot \vec{I} \\ &= (\vec{F}'^T \vec{F}') \cdot \vec{I} \end{aligned} \quad (3)$$

$$\vec{F}' = \det(\vec{F})^{-1/3} \vec{F} \quad (4)$$

$$\begin{aligned} \vec{F} &= \frac{\partial \vec{x}}{\partial \vec{X}} = \frac{\partial(\vec{X} + \vec{u})}{\partial \vec{X}} \\ &= \vec{I} + \frac{\partial \vec{u}}{\partial \vec{X}} \end{aligned} \quad (5)$$

where  $\vec{u}$  is the displacement,  $\vec{x}$  is the new coordinate,  $\vec{X}$  is the initial coordinates of the cell,  $\vec{C}$  is the Cauchy stress tensor,  $\vec{F}'$  is the distortional part of the deformation gradient, and  $\vec{F}$  is defined in (5) such that  $\det(\vec{F}') = 1$  [41]. This is the pure measure of distortion.

The deformation gradient  $\vec{F}$  is composed of two parts, the identity tensor,  $\vec{I}$ , and the derivatives of the displacements with respect to the reference configuration. The displacements of the nodes are known, and the derivatives can be computed using the isoparametric mapping [23, 42]. Once the strain energy density  $w$  is determined, the total strain energy can be computed by integrating  $w$  over the volume of the cell as shown in (6):

$$U = \int_{\Omega} w dV \quad (6)$$

Out of the 24 possible combinations, the one that produces the minimum strain energy  $U$  is chosen. At the end of this process, the two cells are merged with no overlap, and the strain energy required for the operation is the minimum of all 24 cases.

Out of all 24 combinations, some may result in non-convex cells. This represents an ill-conditioned case where the one-to-one isoparametric mapping breaks down. A way to overcome this problem is to condition the determinant of the Jacobian matrix, evaluated at the Gauss quadrature points, as part of the energy calculation process. An energy estimator will be assigned into particular state if, and only if, the condition presented in Equation (7) is satisfied. In case the latter condition fails, the resultant energy will receive the value *MAX ENERGY* and thus will prevent this state from being selected.

$$\det(\vec{J}_i) > 0 \quad \forall i = 1, 2, \dots, 8 \quad (7)$$

Once a state is been selected it can still be improved, in terms of energy minimization. The initial assumption of merging a pair of nodes by taking the average of their corresponding coordinates will not generally result in a global energy minimization for the two merged cells. In order to

improve the initial assumption, the conjugate gradient method is applied to the displacements to find the global minimum of the strain energy, as shown in (8), where  $\vec{u}_k^1$  are the displacements of the nodes corresponding to the selected face of the first cell, and  $\vec{u}_k^2$  is defined in the same way for the selected face of the second cell. The iteration of  $\vec{u}_{k_j}^m$ ,  $j \geq 0$  in the conjugate gradient method satisfies the recursion shown in (9) [43, 44].

$$\min(U(\vec{u}_k^1 + \vec{u}_k^2) : \vec{u} \in R^3) \quad k = 1, \dots, 4 \quad (8)$$

$$\vec{u}_{k_{j+1}}^m = \vec{u}_{k_j}^m + \alpha_j \vec{d}_j \quad (9)$$

The directions  $\vec{d}_j$  are generated by the rule in (10):

$$\vec{d}_{j+1} = -\vec{g}_{j+1} + \beta_j \vec{d}_j \quad (10)$$

where

$$\begin{aligned} \vec{d}_0 &= -\vec{g}_0 \\ \vec{g}_j &= \nabla U(\vec{u}_{k_j}^m) \end{aligned} \quad (11)$$

The step size  $\alpha_j$  is determined by a global line search developed by Fletcher–Reeves [45], shown in (12):

$$\alpha_j = -\frac{\vec{g}_j^T \cdot \vec{d}_j}{\vec{d}_j^T \cdot \vec{g}'_j} \quad (12)$$

An additional constraint was developed by Wolf [46], shown in (14), with the inequality (13):

$$0 < \delta < \sigma < 1 \quad (13)$$

where  $\sigma$  and  $\delta$  are two scalars, and  $\beta$  is determined according to Fletcher–Reeves algorithm (15):

$$\begin{aligned} U(\vec{u}_{k_j}^m) - U(\vec{u}_{k_j}^m + \alpha_j \vec{d}_j) &\geq -\delta \alpha_j (\vec{g}_j^T \cdot \vec{d}_j) \\ |U'(\vec{u}_{k_j}^m)| &\leq -\sigma (\vec{g}_j^T \cdot \vec{d}_j) \end{aligned} \quad (14)$$

$$\beta = \frac{(\vec{g}_{j+1}^T \cdot \vec{g}_{j+1}^T)}{(\vec{g}_j^T \cdot \vec{g}_j^T)} \quad (15)$$

The primary difficulty when using the conjugate gradient method is determining  $\vec{g}_j$  in a closed analytical form. In this paper a closed, compact, expression for  $\vec{g}_j$  is presented that can easily be modified for different types of finite elements. The next sections describe the steps for applying this method using a hexahedral linear ‘brick’ finite element.

### 3. HEXAHEDRAL LINEAR BRICK ELEMENT

Utilizing the procedure outlined above, the two faces to be merged, and their corresponding pairs of nodes, need to be identified. This procedure is described in detail here for a hexahedral linear brick element.

Assuming that every cell is convex, a one-to-one isoparametric mapping can be defined (16), where  $(r, s, t)$  represents a point in the local configuration of the cell, and  $(X, Y, Z)$  represents the point in the reference configuration of the cell.

$$\begin{aligned} X &= \sum_{i=1}^8 N_i(r, s, t)X_i \\ Y &= \sum_{i=1}^8 N_i(r, s, t)Y_i \\ Z &= \sum_{i=1}^8 N_i(r, s, t)Z_i \end{aligned} \tag{16}$$

The inverse isoparametric mapping of points in the reference configuration to the local configuration is determined using the Newton-Raphson method. For a given  $\vec{X} = \{X, Y, Z\}$ , the corresponding  $\vec{r} = \{r, s, t\}$  is found, using (17):

$$\begin{aligned} f(\vec{X} - \vec{r}) &= 0 \\ f(\vec{X} - \vec{r}) &\approx f(\vec{X} - \vec{r}_0) + \frac{\partial f(\vec{X} - \vec{r})}{\partial \vec{r}} \delta \vec{r} = 0 \end{aligned} \tag{17}$$

where  $\delta \vec{r}$  is a small increment from the initial guess  $\vec{r}_0$ , which is determined by (18):

$$\delta \vec{r} = - \frac{\partial f(\vec{X} - \vec{r})}{\partial \vec{r}}^{-1} f(\vec{X} - \vec{r}_0) = \vec{M} f(\vec{X} - \vec{r}_0) \tag{18}$$

Since a solution exists, and is unique, this method will always converge. This iterative process terminates when  $\|\delta \vec{r}\| < \epsilon$ . The stiffness matrix  $\vec{M}$  is defined in (19):

$$\vec{M} = \begin{bmatrix} \frac{\partial(\sum_{i=0}^7 N_i(r, s, t)X_i)}{\partial r} & \frac{\partial(\sum_{i=0}^7 N_i(r, s, t)X_i)}{\partial s} & \frac{\partial(\sum_{i=0}^7 N_i(r, s, t)X_i)}{\partial t} \\ \frac{\partial(\sum_{i=0}^7 N_i(r, s, t)Y_i)}{\partial r} & \frac{\partial(\sum_{i=0}^7 N_i(r, s, t)Y_i)}{\partial s} & \frac{\partial(\sum_{i=0}^7 N_i(r, s, t)Y_i)}{\partial t} \\ \frac{\partial(\sum_{i=0}^7 N_i(r, s, t)Z_i)}{\partial r} & \frac{\partial(\sum_{i=0}^7 N_i(r, s, t)Z_i)}{\partial s} & \frac{\partial(\sum_{i=0}^7 N_i(r, s, t)Z_i)}{\partial t} \end{bmatrix} \tag{19}$$

The maximum signal gradient on the face of each cell, generated by the adjacent cell, is simply the intersection point of a line that connects the center of both cells, illustrated in Figure 1(b).

Define  $\vec{A} = \{a_1, a_2, a_3\}$  to be a vector representing the line which connects the center of both cells.  $\vec{O}_i$  represents the coordinates of the  $i$ th cell center point.  $\vec{B}_i = \vec{O}_i + \epsilon \vec{A}$  represents a point that is near the center of the cell along the vector  $\vec{a}$ . Since  $\vec{B}_i$  is an internal point, it can be mapped to the local configuration of the cell using (17), (18), and (19). Defining  $\vec{b}_i$  to be the inverse mapping of  $\vec{B}_i$ , the point  $(0, 0, 0)$  is the inverse mapping of  $\vec{O}_i$ . The inverse mapping of the vector  $\vec{A}$  is simply  $\vec{a}_i = \vec{b}_i - (0, 0, 0) = \vec{b}_i$ . The face that intersects  $\vec{a}_i$  corresponds to the sign of the maximum term in  $\vec{a}_i$ , and also corresponds to the face that intersects  $\vec{A}$  in the reference configuration, i.e.



if  $\vec{a}_7 = (0.14, 0.98, 0.14)$ , then the face above the origin in the  $y$ -direction intersects the vector  $\vec{a}_7$ . Utilizing this method, the two faces to be merged are determined.

Next, the most suitable configuration for merging of the 24 possible combinations of pairs of nodes must be found. For each case, the displacement of every pair of nodes is computed, as shown in (20):

$$\vec{u}_k^1 = \frac{\vec{X}_k^1 + \vec{X}_k^2}{2} - \vec{X}_k^1 \quad \vec{u}_k^2 = \frac{\vec{X}_k^1 + \vec{X}_k^2}{2} - \vec{X}_k^2 \quad k = 1, \dots, 4 \quad (20)$$

where  $\vec{u}_k^1, \vec{u}_k^2$  are the displacements of the pair-merged nodes  $k$ , and  $\vec{X}_k^1, \vec{X}_k^2$  are the initial coordinates of the pair-merged nodes.

The steps for computing the strain energy  $U$ , for every possible combination of node pairs, are explained next.

The deformation gradient tensor  $\vec{F}$ , in terms of displacements  $\vec{u}$  [47], is given by (21):

$$\vec{F} = \frac{\partial \vec{x}}{\partial \vec{X}} = \frac{\partial(\vec{X} + \vec{u})}{\partial \vec{X}} = \vec{I} + \frac{\partial \vec{u}}{\partial \vec{X}} = \begin{bmatrix} 1 + \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} \\ \frac{\partial v}{\partial X} & 1 + \frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z} \\ \frac{\partial w}{\partial X} & \frac{\partial w}{\partial Y} & 1 + \frac{\partial w}{\partial Z} \end{bmatrix} \quad (21)$$

where all the derivatives in (21) are with respect to the reference configuration [23], which is the configuration before the merging process.

Once the displacements at the nodes have been determined, the displacements inside the elements can be approximated using the finite element relation given in (22):

$$\begin{aligned} u &\approx N_i(r, s, t)u_i \\ v &\approx N_i(r, s, t)v_i \quad i = 1, \dots, 8 \\ w &\approx N_i(r, s, t)w_i \end{aligned} \quad (22)$$

where  $N_i(r, s, t)$  are the shape functions with respect to the local coordinates [42] given by the relations shown in (23):

$$\begin{aligned} N_1 &= \frac{1}{8}(1+r)(1-s)(1+t) & N_2 &= \frac{1}{8}(1+r)(1+s)(1+t) \\ N_3 &= \frac{1}{8}(1-r)(1+s)(1+t) & N_4 &= \frac{1}{8}(1-r)(1-s)(1+t) \\ N_5 &= \frac{1}{8}(1+r)(1-s)(1-t) & N_6 &= \frac{1}{8}(1+r)(1+s)(1-t) \\ N_7 &= \frac{1}{8}(1-r)(1+s)(1-t) & N_8 &= \frac{1}{8}(1-r)(1-s)(1-t) \end{aligned} \quad (23)$$

By combining (21), (22), and (23), the displacement derivatives with respect to the reference coordinate system are obtained using (24).

$$\begin{aligned} \frac{\partial u}{\partial X} &= \frac{\partial N_i}{\partial X} u_i & \frac{\partial u}{\partial Y} &= \frac{\partial N_i}{\partial Y} u_i & \frac{\partial u}{\partial Z} &= \frac{\partial N_i}{\partial Z} u_i \\ \frac{\partial v}{\partial X} &= \frac{\partial N_i}{\partial X} v_i & \frac{\partial v}{\partial Y} &= \frac{\partial N_i}{\partial Y} v_i & \frac{\partial v}{\partial Z} &= \frac{\partial N_i}{\partial Z} v_i \\ \frac{\partial w}{\partial X} &= \frac{\partial N_i}{\partial X} w_i & \frac{\partial w}{\partial Y} &= \frac{\partial N_i}{\partial Y} w_i & \frac{\partial w}{\partial Z} &= \frac{\partial N_i}{\partial Z} w_i \end{aligned} \quad (24)$$

The mapping from the local coordinate system  $(r, s, t)$  to the reference system is defined by the Jacobian (25), where  $(X_j, Y_j, Z_j)$  are the coordinates of the node in the reference configuration.

$$\vec{J} = \begin{bmatrix} \sum_{j=1}^8 \frac{\partial N_j}{\partial r} X_j & \sum_{j=1}^8 \frac{\partial N_j}{\partial r} Y_j & \sum_{j=1}^8 \frac{\partial N_j}{\partial r} Z_j \\ \sum_{j=1}^8 \frac{\partial N_j}{\partial s} X_j & \sum_{j=1}^8 \frac{\partial N_j}{\partial s} Y_j & \sum_{j=1}^8 \frac{\partial N_j}{\partial s} Z_j \\ \sum_{j=1}^8 \frac{\partial N_j}{\partial t} X_j & \sum_{j=1}^8 \frac{\partial N_j}{\partial t} Y_j & \sum_{j=1}^8 \frac{\partial N_j}{\partial t} Z_j \end{bmatrix} \quad (25)$$

Using (25), the derivatives of the shape function with respect to the reference coordinate system are defined in (26).

$$\begin{bmatrix} \frac{\partial N_i}{\partial X} \\ \frac{\partial N_i}{\partial Y} \\ \frac{\partial N_i}{\partial Z} \end{bmatrix} = \vec{J}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial s} \\ \frac{\partial N_i}{\partial t} \end{bmatrix} \quad (26)$$

Substituting (25) and (26) into (21),  $\vec{F}$  can be determined at each point in the cell using (27):

$$\vec{F} = \vec{I} + \begin{bmatrix} \frac{N_i(r, s, t)u_i}{\partial X} & \frac{N_i(r, s, t)u_i}{\partial Y} & \frac{N_i(r, s, t)u_i}{\partial Z} \\ \frac{N_i(r, s, t)v_i}{\partial X} & \frac{N_i(r, s, t)v_i}{\partial Y} & \frac{N_i(r, s, t)v_i}{\partial X} \\ \frac{N_i(r, s, t)w_i}{\partial X} & \frac{N_i(r, s, t)w_i}{\partial Y} & \frac{N_i(r, s, t)w_i}{\partial Z} \end{bmatrix} \quad i = 1, \dots, 8 \quad (27)$$

The strain energy density can be found using (2) and (4). Finally, the total strain energy (5) can be found using the Gauss–Legendre integration method given by (28):

$$\begin{aligned}
 U &= \int_{\Omega} w dV = C_1 \int_{\Omega} (I_1 - 3) \det(\vec{\mathcal{J}}) dr ds dt \\
 &\approx C_1 \sum_{i=1}^8 (I_1 - 3)_{r=r_i, s=s_i, t=t_i} w_i \det(\vec{\mathcal{J}}_i)
 \end{aligned} \tag{28}$$

where  $r_i, s_i, t_i$  represent a set of eight different evaluation points; each coordinate can take on the value  $\pm\sqrt{1/3}$ . Each point  $r_i, s_i, t_i$  is multiplied by its corresponding weight  $w_i = 1$  respectively, and  $C_1$  is set arbitrarily.<sup>‡</sup>

#### 4. THE CONJUGATE GRADIENT METHOD

As described above, the initial configuration is the one that introduces the minimum change in strain energy when each pair of nodes is merged into a single node, where each merged node is initially positioned at the *mean* of the original coordinates of the nodes. To further improve the initial merged configuration, the conjugate gradient method is used to minimize the strain energy of the merged cells, by adjusting the position of the merged nodes from their initial configuration.

The conjugate gradient method requires the computation of the gradient of the strain energy  $w$ , with respect to the displacements of the nodes  $\vec{u}_k$  (the nodes that correspond to the merged faces), and is given in (29):

$$\frac{\partial w}{\partial \vec{u}_k} = C_1 \frac{\partial(I_1)}{\partial \vec{u}_k} \tag{29}$$

Using the finite element relation (30), the first invariant with the modification of the deformation gradient tensor (4) can be determined using (31):

$$\begin{aligned}
 \vec{u} &= \sum_{i=1}^8 N_i \vec{u}_i \\
 \vec{\bar{u}} &= \sum_{i=1}^8 N_i \vec{\bar{u}}_i
 \end{aligned} \tag{30}$$

$$\begin{aligned}
 I_1 &= \vec{C}' \cdot \vec{I} = (\vec{F}'^T \vec{F}') \cdot \vec{I} \\
 &= (\vec{\mathcal{J}}^{-1/3} \vec{F}'^T J^{-1/3} \vec{F}') \cdot \vec{I} \\
 &= \vec{\mathcal{J}}^{-2/3} (\vec{F}'^T \vec{F}') \cdot \vec{I} \\
 &= \vec{\mathcal{J}}^{-2/3} \vec{C} \cdot \vec{I}
 \end{aligned} \tag{31}$$

where  $\vec{\mathcal{J}} = \det(\vec{F})$ .

<sup>‡</sup>A suggested value is the same order of magnitude as the Young's modulus of the material.

The derivatives of  $I_1$  with respect to  $\vec{u}_k$  are then computed, as shown in (32):

$$\frac{\partial I_1}{\partial \vec{u}_i} = -\frac{2}{3} \mathcal{J}^{-5/3} \frac{\partial J}{\partial \vec{u}_i} (\vec{C} \cdot \vec{I}) + \mathcal{J}^{-2/3} \frac{\partial (\vec{C} \cdot \vec{I})}{\partial \vec{u}_i} \quad (32)$$

The two terms in (32) are handled differently. The Cauchy Green tensor  $\vec{C}$  is expressed in terms of the finite element mapping given by (33):

$$\begin{aligned} \vec{C} &= \vec{F}^T \vec{F} = \left[ \vec{I} + \left[ \sum_{j=1}^8 \frac{\partial N_j}{\partial \vec{X}} \vec{u}_j \right]^T \right] \left[ \vec{I} + \sum_{j=1}^8 \frac{\partial N_j}{\partial \vec{X}} \vec{u}_j \right] \\ &= \vec{I} + 2 \sum_{j=1}^8 \frac{\partial N_j}{\partial \vec{X}} \vec{u}_j + \left[ \sum_{j=1}^8 \frac{\partial N_j}{\partial \vec{X}} \vec{u}_j \right]^T \left[ \sum_{j=1}^8 \frac{\partial N_j}{\partial \vec{X}} \vec{u}_j \right] \\ &= \vec{I} + \left( 2 \sum_{j=1}^8 \frac{\partial N_j u_j^m}{\partial X_k} \right) \vec{e}_m \otimes \vec{e}_k \\ &\quad + \left( \sum_{j=1}^8 \frac{\partial N_j u_j^m}{\partial X_k} \right) \vec{e}_k \otimes \vec{e}_m \left( \sum_{j=1}^8 \frac{\partial N_j u_j^l}{\partial X_n} \right) \vec{e}_l \otimes \vec{e}_n \\ &= \vec{I} + \left( 2 \sum_{j=1}^8 \frac{\partial N_j u_j^m}{\partial X_k} \right) \vec{e}_m \otimes \vec{e}_k \\ &\quad + \left( \sum_{j=1}^8 \frac{\partial N_j u_j^m}{\partial X_k} \right) \left( \sum_{i=1}^8 \frac{\partial N_i u_i^m}{\partial X_n} \right) \vec{e}_k \otimes \vec{e}_n \end{aligned} \quad (33)$$

(33) is substituted into (29) giving (34):

$$\begin{aligned} \vec{C} \cdot \vec{I} &= 3 + 2 \left( \sum_{j=1}^8 \frac{\partial N_j u_j^m}{\partial X_m} \right) + \left( \sum_{j=1}^8 \frac{\partial N_j u_j^m}{\partial X_m} \right) \left( \sum_{i=1}^8 \frac{\partial N_i u_i^m}{\partial X_m} \right) \\ &= 3 + 2 \left( \sum_{i=1}^8 \frac{\partial N_i u_i^m}{\partial X_m} \right) + \left( \sum_{i=1}^8 \frac{\partial N_i u_i^m}{\partial X_m} \right)^2 \end{aligned} \quad (34)$$

The derivative of (33) is taken with respect to  $\vec{u}_k$  to produce (35):

$$\begin{aligned} \frac{\partial (\vec{C} \cdot \vec{I})}{\partial \vec{u}_k} &= \frac{\partial (\vec{C} \cdot \vec{I})}{\partial u_k^m} \vec{e}_m \\ &= 2 \left( \frac{\partial N_k}{\partial X_m} \right) \vec{e}_m + 2 \left( \sum_{j=1}^8 \frac{\partial N_j u_j^m}{\partial X_m} \right) \left( \frac{\partial N_k}{\partial X_m} \right) \vec{e}_m \end{aligned} \quad (35)$$

Note that (35) is in a vectorial form.

The next term to derive is  $\partial \vec{\mathcal{J}} / \partial \vec{u}_k$  as shown in Equation (36):

$$\frac{\partial \vec{\mathcal{J}}}{\partial \mathbf{u}_k} = \frac{\partial \vec{\mathcal{J}}}{\partial u_k} \mathbf{e}_1 + \frac{\partial \vec{\mathcal{J}}}{\partial v_k} \mathbf{e}_2 + \frac{\partial \vec{\mathcal{J}}}{\partial w_k} \mathbf{e}_3 \quad (36)$$

following the definition of  $\vec{\mathcal{J}} = \det(\mathbf{F})$ , combined with Equation (21) the terms  $\partial \vec{\mathcal{J}} / \partial u_k$ ,  $\partial \vec{\mathcal{J}} / \partial v_k$ ,  $\partial \vec{\mathcal{J}} / \partial w_k$  are shown in Equation (37):

$$\begin{aligned} \frac{\partial \vec{\mathcal{J}}}{\partial u_k} &= \frac{\partial N_k}{\partial X} \left(1 + \frac{\partial N_i}{\partial Y} v_i\right) \left(1 + \frac{\partial N_j}{\partial Z} w_j\right) - \frac{\partial N_k}{\partial Y} \left(\frac{\partial N_i}{\partial X} v_i\right) \left(1 + \frac{\partial N_j}{\partial Z} w_j\right) \\ &\quad + \frac{\partial N_k}{\partial Z} \left(\frac{\partial N_i}{\partial X} v_i\right) \left(\frac{\partial N_j}{\partial Y} w_j\right) \\ \frac{\partial \vec{\mathcal{J}}}{\partial v_k} &= \frac{\partial N_k}{\partial Y} \left(8 + \frac{\partial N_k}{\partial X} u_i\right) \left(1 + \frac{\partial N_i}{\partial Z} w_i\right) - \frac{\partial N_k}{\partial Z} \frac{\partial N_i}{\partial Y} w_i - \frac{\partial N_k}{\partial X} \left(\frac{\partial N_i}{\partial Y} u_i\right) \left(1 + \frac{\partial N_j}{\partial Z} w_j\right) \\ &\quad + \frac{\partial N_k}{\partial Z} \left(\frac{\partial N_i}{\partial X} w_i\right) + \frac{\partial N_k}{\partial X} \left(\frac{\partial N_i}{\partial Z} u_i\right) \left(\frac{\partial N_j}{\partial Y} w_j\right) - \frac{\partial N_k}{\partial Y} \left(\frac{\partial N_i}{\partial X} w_i\right) \\ \frac{\partial \vec{\mathcal{J}}}{\partial w_k} &= \left(8 + \frac{\partial N_k}{\partial X} u_i\right) \frac{\partial N_k}{\partial Z} \left(1 + \frac{\partial N_i}{\partial Z} w_i\right) - \frac{\partial N_k}{\partial Y} \frac{\partial N_i}{\partial Z} v_i - \frac{\partial N_k}{\partial Z} \left(\frac{\partial N_i}{\partial Y} u_i\right) \left(\frac{\partial N_j}{\partial X} v_j\right) \\ &\quad + \frac{\partial N_k}{\partial Z} \left(\frac{\partial N_i}{\partial X} w_i\right) + \frac{\partial N_k}{\partial Y} \left(\frac{\partial N_i}{\partial Z} u_i\right) \left(\frac{\partial N_j}{\partial X} v_j\right) - \frac{\partial N_k}{\partial X} \left(1 + \frac{\partial N_i}{\partial Y} v_i\right) \end{aligned} \quad (37)$$

Combining (37) and (35) into (32), (38) is obtained.

$$\frac{\partial I_1}{\partial \vec{u}_k} = \left\{ -\frac{1}{9} \vec{\mathcal{J}}^{-5/3} \frac{\partial \vec{\mathcal{J}}}{\partial \mathbf{u}_k} (\vec{C} \cdot \vec{I}) + 2 \vec{\mathcal{J}}^{-2/3} \left[ \left( \frac{\partial N_k}{\partial X_m} \right) + \left( \sum_{j=1}^8 \frac{\partial N_j u_j}{\partial X_n} \right) \left( \frac{\partial N_k}{\partial X_n} \right) \right] \right\} \vec{e}_m \quad (38)$$

(38) is substituted into the complete form of the strain energy  $U$ , and taking its derivatives with respect to  $\vec{u}_k$ , and using the Gauss–Legendre integration method, (39) is obtained.

$$\frac{\partial U}{\partial \vec{u}_k} = C_1 \sum_{i=1}^8 \left( \frac{\partial I_1}{\partial \vec{u}_k} \right)_{r=r_i, s=s_i, t=t_i} \det(\vec{\mathcal{J}}_i) \quad (39)$$

All the components of (39) have been developed for a single cell with respect to the displacements applied to the nodes. In order to work directly with the coordinates of the new merged nodes, the relation between the coordinates is defined in (40), where  $\vec{x}_m$  corresponds to the new coordinates of node  $m$ .

$$\begin{aligned} \vec{u}_k^1 &= \vec{x}_m - \vec{X}_k^1 \\ \vec{u}_k^2 &= \vec{x}_m - \vec{X}_k^2 \end{aligned} \quad m = 1, \dots, 4 \quad (40)$$

```

1  Select Faces to Merge;
2      Inverse map the signal and the cell into the reference configuration;
3      Find the faces with the maximum signal gradient generated by the
      adjacent cell;
4  Select Configuration;
5      Compute the initial displacements of the merged nodes for every pairing
      configuration;
6      Compute the change in strain energy for every configuration;
7      Assign MAX ENERGY to each state that fails to pass the condition in
      equation (7) ;
8  Conjugate Gradient Search;
9      For all 4 merged nodes:
10         Compute the strain energy gradient with respect to the coordinates
            of a single node;
11         Displace the node in the direction of the conjugate gradient;
12         Compute the total strain energy;
13         If the total strain energy has reached a plateau:
14             then stop;
15         else return to line 9;
    
```

Figure 2. Pseudo-code outline of the algorithm.

The total energy gradient with respect to  $\vec{x}_m$  is given by (41):

$$\begin{aligned}
 \frac{\partial U^{\text{total}}}{\partial \vec{x}_m} &= \frac{\partial U}{\partial \vec{u}_k^1} \frac{\partial \vec{u}_k^1}{\partial \vec{x}_m} + \frac{\partial U}{\partial \vec{u}_k^2} \frac{\partial \vec{u}_k^2}{\partial \vec{x}_m} \\
 &= \frac{\partial U}{\partial \vec{u}_k^1} + \frac{\partial U}{\partial \vec{u}_k^2}
 \end{aligned} \tag{41}$$

Figure 2 contains pseudo-code outlining each step in the algorithm.

## 5. COMPUTATIONAL COST

In this section the computational cost of executing the new merging approach is analyzed. Usually, in any type of mesh generation method, the cost must be thoroughly analyzed to assess the quality of a particular method. In this section, a comparison will be made between a method that utilizes a lookup table versus the new approach introduced in this paper. The comparison will be based on the complexity of the different methods.

The paving approach has been shown to be robust and reliable, and able to successfully cope with complex geometries. The paving method *directly* creates quadrilateral elements on 3-D surfaces [48]. It starts by creating quadrilateral elements on the boundary that progressively advance inwardly. At some point, element boundaries will either collide or will intersect. A proximity factor which utilizes a heuristic equation is used to detect nearby or intersecting elements. The algorithm then uses the concept of *sticky space*. Any edge that falls within a predefined volume around a given edge will be attracted to it and will form a connection. The *sticky space*  $S$  is defined by  $S = \alpha(l_1 + l_2)/2$ , where  $l_1$  and  $l_2$  are the length of the line segment. The heuristic parameter  $\alpha$  is defined as the likelihood that any given two edges will be merged. Once two edges have been selected to be merged, three conditions (i.e. midpoint-to-midpoint, edge-to-midpoint, and edge-to-edge) are evaluated against a proximity equation. Depending on the latter value, a connection between the edges is formed. Using  $n$  as the total number of edge pairs that have been merged together, the complexity of the *sticky space* merging process is  $O(3n)$ , as there are three conditions which have been tested.

The new energy-minimization merging approach can be applied to either 2-D quadrilateral elements or 3-D hexahedral elements. For the 2-D case (to compare with the paving approach), initially, two elements are selected to be merged based on their proximity. Next, one edge from each element is selected to be merged, again based on proximity. Since there are 2 edges, and each edge terminates in a node, the total number of possible merging configurations is 2. Next, the total strain energy associated with each configuration is calculated, and the configuration with the minimum energy is selected. Finally, the merged nodes are displaced to further minimize the strain energy. The complexity of the last step (the total number of iterations) depends on the orientation of the element prior to being merged. Averaging this parameter over multiple cases gave an average value of 8.2 iterations. Thus, the overall complexity of the new method is  $O(10n)$ .

The CPU time required for the merging process is analyzed next. Figure 8(c) shows a topological object, comprising hexahedral elements that have been merged utilizing the new approach. The mesh generation scheme was computed on an *AMD Athlon* 32 bit CPU with 1.81 GHz clock speed. Once a pair of elements were selected to be merged, the time from the beginning to end of the merging process was recorded. Figures 3 and 4 show the results of a series of experiments where 13 486 different samples were merged, with their corresponding compute times. In Figure 3, the

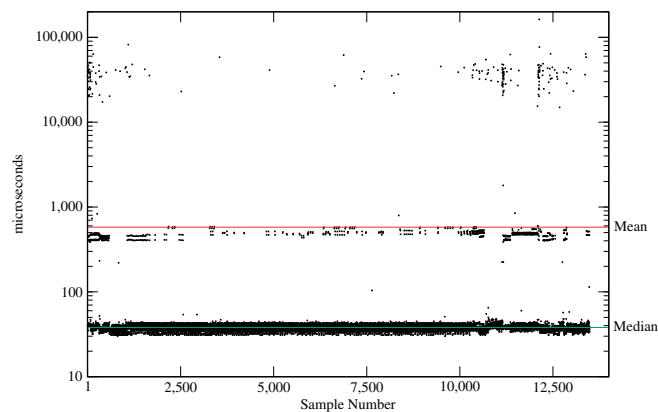


Figure 3. CPU-time for 13 486 different pairs of merged elements.

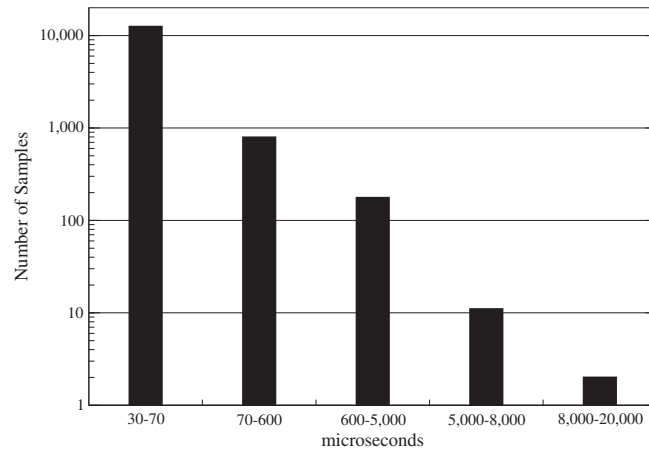


Figure 4. Histogram of CPU-time for 13 486 different pairs of merged elements.

$y$ -axis represents a logarithmic time scale measured in microseconds. The  $x$ -axis corresponds to the test number. The median time was  $38\ \mu\text{s}$ . The average time was  $578\ \mu\text{s}$ . In most situations, the measured time was on the order of  $40\ \mu\text{s}$ . However, a number of tests required longer times ( $\sim 400\ \mu\text{s}$ ), and a much smaller number required even longer times ( $\sim 8000\ \mu\text{s}$ ). A histogram of the times is shown in Figure 4.

The different time periods depend on the number of iterations required for the conjugate gradient method to converge. The number of iterations is highly dependent on the initial relative orientation between the elements to be merged.

## 6. RESULTS

The merging method introduced here is capable of successfully merging cells regardless of their arrangement and is completely general.

To illustrate the method, three examples were selected, shown in Figures 5–7. The portion of each figure identified with the letter (a) illustrates cells selected for merging after testing for proximity or intersection. The portion of each figure identified with the letter (c) contains a chart showing the distribution of strain energy among the 24 possible initial merging configurations. The  $y$ -axis indicates the strain energy corresponding to the configuration number on the  $x$ -axis. The configuration with the minimum energy has been selected as the initial state of merging. This state sets the initial location of the new merged nodes, which are located mid-way between their original positions.

The portion of each figure identified with the letter (d) illustrates the energy optimization process through the conjugate gradient method. The  $y$ -axis corresponds to the strain energy while the  $x$ -axis corresponds to the number of iterations it takes to reach global energy minimization. Notice that the number of iterations is relatively small. This is because only the lowest energy configuration of merged nodes is selected as the starting point for the optimization. The high rate of convergence is also a direct result of the closed-form analytical expression derived for the energy gradient.



## A NOVEL ENERGY-BASED APPROACH

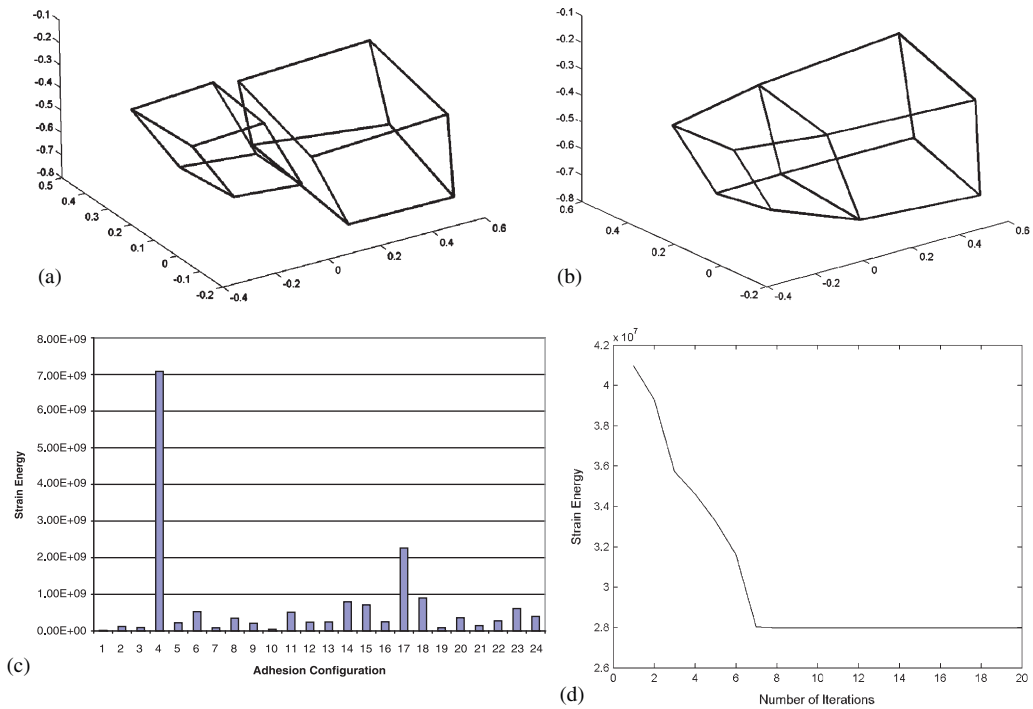


Figure 5. First example: (a) two non-overlapping, but nearby, cells prior to merging; (b) the two cells subsequent to merging; (c) the energy states of all 24 possible merging configurations. The minimum one is selected; and (d) the strain energy minimization using the conjugate gradient method.

The portion of each figure identified with the letter (b) presents the final configuration of the cells after the merging process. The distortion of the cells, in order to achieve merging, is minimal and results in elements that are suitable for further computation. It can also be seen that the method is robust in that it successfully merges nodes and attached cells in a variety of initial arrangements.

## 7. EXAMPLES

Figure 8 shows three-dimensional structures that were generated using an evolutionary technique<sup>§</sup> that utilized the new merging approach. These structures have complex topological characteristics such as non-convexity and non-regularity. The odd shapes of elements comprising the structure demonstrates the ability of the method presented here to successfully merge hexahedral elements.

The mesh quality for the three different structures was quantified following the node Jacobian matrix definition presented by P.M. Knupp [6]. The Jacobian determinant for each node was

<sup>§</sup>The application that generated these structures is beyond the scope of this paper [49, 50].

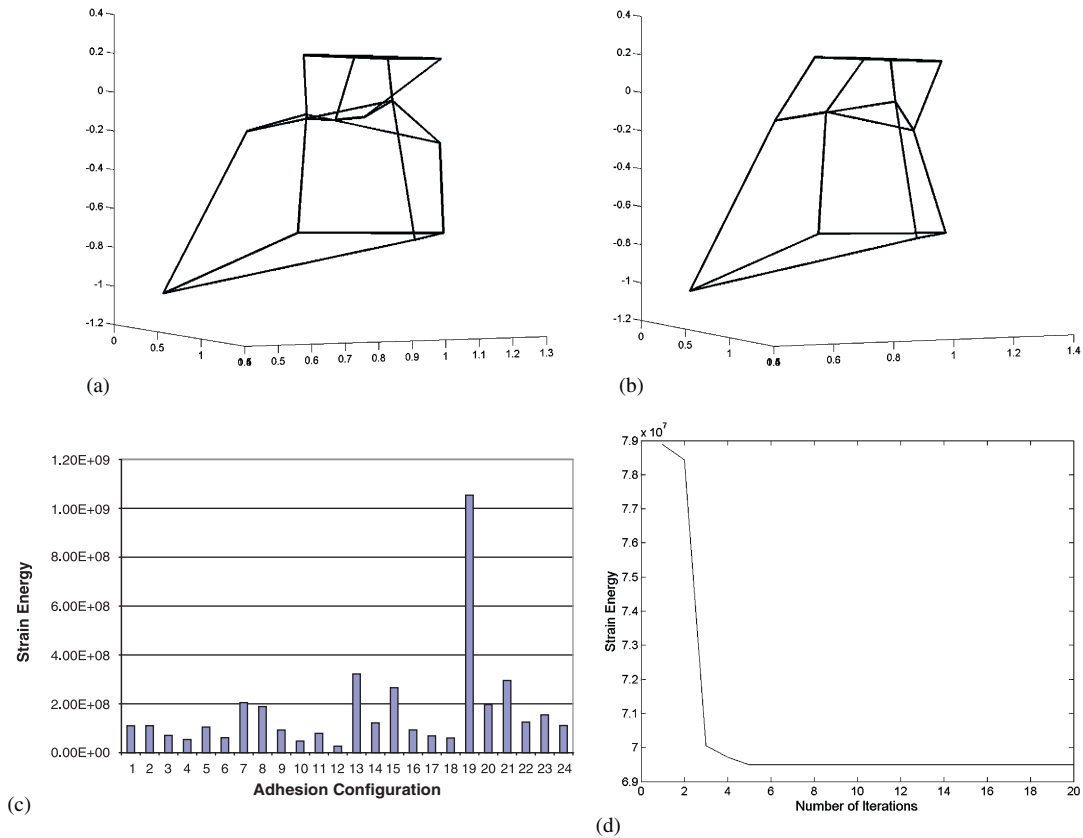


Figure 6. Second example: (a) two cells prior to merging; (b) the two cells subsequent to merging; (c) the energy states of all 24 possible merging configurations. The minimum one is selected; and (d) the strain energy minimization using the conjugate gradient method.

computed such that a value 3.0 corresponds to a perfect node with orthogonal outward vectors. The mesh quality is thus quantified by taking the average determinant of all nodes comprising the structures. The structures presented in Figure 8(a)–(c) have the following mesh quality measurements: 3.151, 3.258, 3.31 respectively. These measurement values show the capability of the new merging approach to generate robust meshes with various cell-to-cell orientations.

Figure 9 shows the growth of a helical-shaped structure from a single ‘cell’. The merging process was critical as cells frequently merged during the growth process.

## 8. CONCLUSIONS

In this paper a novel and general approach for merging two 3-D finite elements was introduced. Using first principles, including minimum potential energy and cell signaling, a model was developed that merges any two finite elements, regardless of their configuration or deformation.

## A NOVEL ENERGY-BASED APPROACH

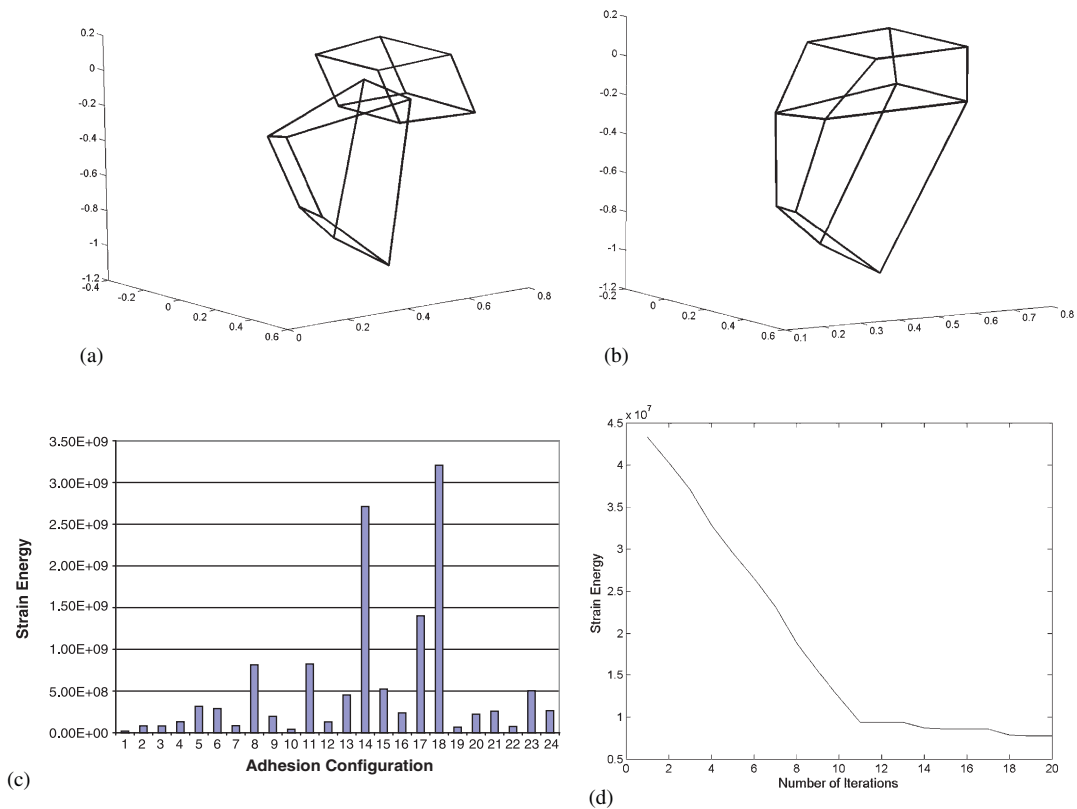


Figure 7. Third example: (a) two intersecting cells prior to merging; (b) the two cells subsequent to merging; (c) the energy states of all 24 possible merging configurations. The minimum one is selected; and (d) the strain energy minimization using the conjugate gradient method.

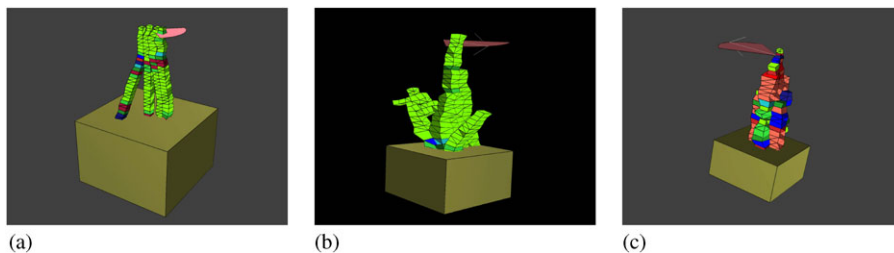


Figure 8. Three different structures developed by a three-dimensional synthesis method [49, 50].

This model may be used in any application that requires the merging of cells or finite elements. One of the major applications that may utilize this approach is the *advancing front* method for generating 2-D or 3-D meshes of convex and non-convex bodies. We have shown how this new method has been successfully employed in a 3-D growth model. The major advantage

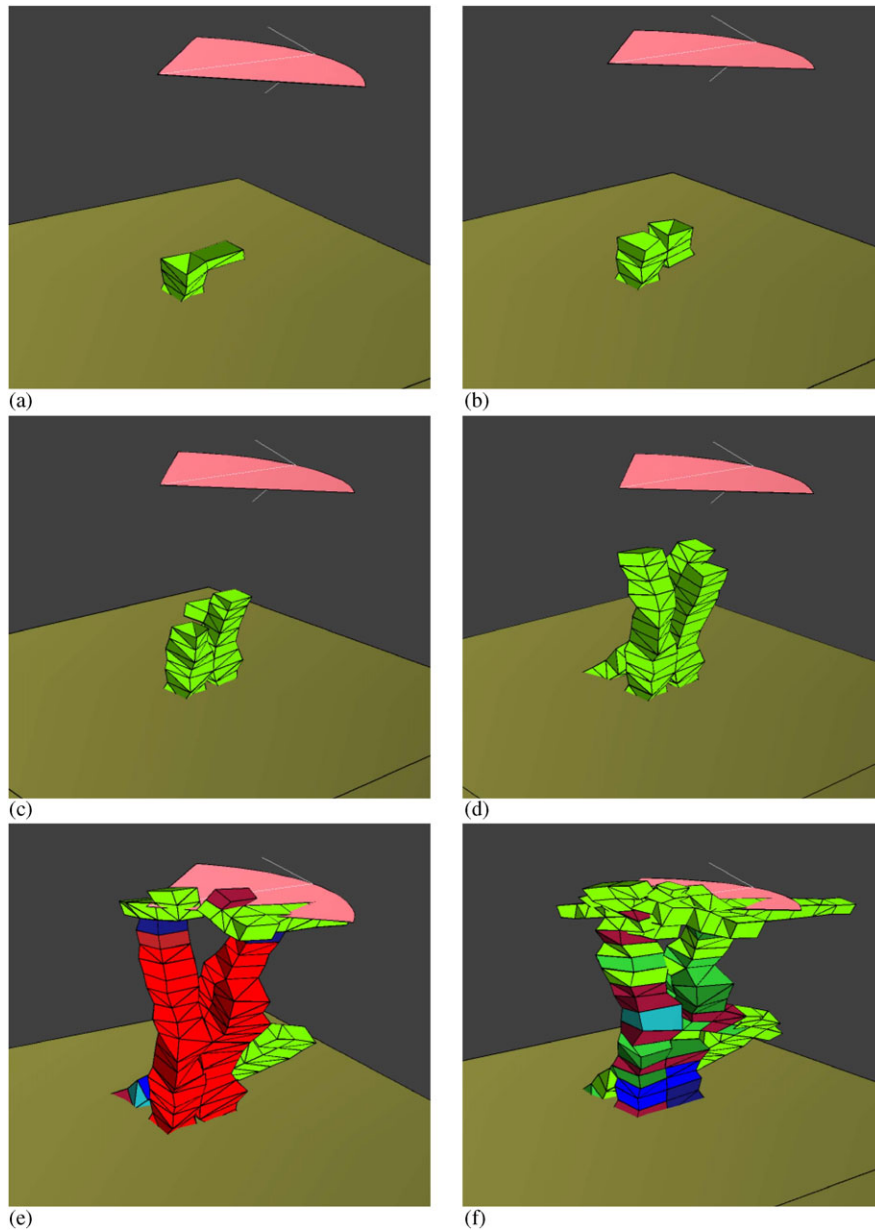


Figure 9. Six stages in the growth and development of the structure shown in Figure 8(c).

of our approach is the absence of any heuristic rules commonly used in these problems. The approach introduced here imitates nature and works smoothly and robustly, and produces well-conditioned merged meshes. This approach, with minor adjustments, is suitable for most types of 3-D elements. Several structures, generated by utilizing the merging approach, have been

presented in this paper. The mesh quality characteristics for each structure was analyzed. The mesh quality values indicates the ability of the method to generate good quality elements upon merging. The CPU-time required for the merging approach has been calculated and analyzed, as it is a major component that needs to be considered in every mesh generation method. The complexity of our new energy-minimization merging approach is  $\sim 3$  times larger than the *sticky space* approach. However, the generality of our approach makes it more robust and more general, and it can accommodate distorted elements with complex relative orientations prior to being merged.

## REFERENCES

1. Blacker TD, Stephenson MB. Paving: a new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering* 1991; **32**(4):811–847. Special Issue: Adaptive Meshing. Issue Edited by Olgierd C. Zienkiewicz, Richard H. Gallagher, Ronald W. Lewis.
2. Lo SH. Volume discretization into tetrahedra—II. 3D triangulation by advancing front approach. *Computers and Structures* 1991; **39**(5):501–511.
3. Owen SJ, Saigal S. H-Morph: an indirect approach to advancing front hex meshing. *International Journal for Numerical Methods in Engineering* 2000; **42**(1–2):289–312.
4. Chong C, Kumar AS, Lee H. Automatic mesh-healing technique for model repair and finite element model generation. *Finite Elements in Analysis and Design* 2007; **43**(15):1109–1119.
5. Freitag LA, Knupp PM. Tetrahedral mesh improvement via optimization of the element condition number. *International Journal for Numerical Methods in Engineering* 2002; **53**(6):1377–1391.
6. Knupp PM. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II—A framework for volume mesh optimization and the condition number of the Jacobian matrix. *International Journal for Numerical Methods in Engineering* 2000; **48**(8):1165–1185.
7. Shephard MS, Georges MK. Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering* 1991; **32**(4):709–749. Special Issue: Adaptive Meshing. Issue Edited by Olgierd C. Zienkiewicz, Richard H. Gallagher, Ronald W. Lewis.
8. Yerry MA, Shephard MS. Automatic three-dimensional mesh generation by the modified octree technique. *International Journal for Numerical Methods in Engineering* 1984; **20**(11):1965–1990.
9. Sundar H, Sampath RS. Low-constant parallel algorithms for finite element simulations using linear octrees. *SC '07: Proceedings of the 2007 ACM/IEEE Conference on Supercomputing*, Reno, NV, 2007; 1–12.
10. Bielak J, Ghattas O, Kim EJ. Parallel octree-based finite element method for large-scale earthquake ground motion simulation. *CMES* 2005; **10**(2):99–112.
11. Yuan J, Zhang L, Li Z. A step-by-step approach for three-dimensional finite element mesh generation. *IEEE Transactions on Magnetics* 1998; **34**(5):3375–3378.
12. Baker TJ. Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Engineering with Computers* 1989; **5**(3–4):161–175.
13. Choi H-S, Kim H-S, Hahn S-Y. An adaptive mesh generator based on nodal errors. *IEEE Transactions on Magnetics* 2000; **36**(4):1619–1622.
14. Lawson CL. Software for  $C^1$  surface interpolation. In *Mathematical Software III*, Rice JR (ed.). Academic Press: New York, 1977; 161–194.
15. Owen SJ. A survey of unstructured mesh generation technology. *Proceedings of the 7th International Meshing Roundtable*, Dearborn, MI, 1998. Available from: <http://www.andrew.cmu.edu/user/sowen/survey/index.html>.
16. George P-L, Hecht F, Saltel E. Automatic mesh generator with specified boundary. *Computer Methods in Applied Mechanics and Engineering* 1991; **92**(3):269–288.
17. Weatherill NP, Hassan O. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering* 1994; **37**(12):2005–2039.
18. Cheng S-W, Poon S-H. Three-dimensional Delaunay mesh generation. *Discrete and Computational Geometry* 2006; **36**(3):419–456.
19. Yu X, Gu L, Lv S, Liu J, Huang P, Kong X. A novel biomedical meshing algorithm and evaluation based on revised delaunay and space disassembling. *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Lyon, France, August 2007; 5091–5094.

20. Shin D, Tjahjadi T. Triangular mesh generation of octrees of non-convex 3D objects. *Proceedings of the 18th International Conference on Pattern Recognition*, vol 3. IEEE Computer Society: Hong Kong, 2006; 950–953.
21. Krysl P, Ortiz M. Variational Delaunay approach to the generation of tetrahedral finite element meshes. *International Journal for Numerical Methods in Engineering* 2001; **50**(7):1681–1700.
22. Löhner R. Progress in grid generation via the advancing front technique. *Engineering with Computers* 1996; **12**(3–4):186–210.
23. Zienkiewicz OC, Taylor RL. *Finite Element Method: Volume 2, Solid Mechanics*. Elsevier: Amsterdam, 2000.
24. Staten ML, Owen SJ, Blacker TD. Unconstrained plastering—a new all hexahedral mesh generation algorithm. In *International Conference on Adaptive Modeling and Simulation (ADMOS 2005)*, Barcelona, Spain, Díez P, Wiberg N-E (eds), 2005.
25. Canann SA. Plastering: a new approach to automated 3-D hexahedral mesh generation. *Proceedings of the 33rd Structural Dynamics and Materials Conference*, Dallas, TX. American Institute of Aeronautics and Astronautics, April 1992.
26. Canann SA. Plastering and optismoothing: new approaches to automated 3D hexahedral mesh generation and mesh smoothing. *Ph.D. Thesis*, Brigham Young University, Provo, Utah, U.S.A., 1991.
27. Hipp J, Lober R. Plastering: all-hexahedral mesh generation through connectivity resolution. *Proceedings of the 3rd International Meshing Roundtable*, Albuquerque, NM, 1994.
28. Blacker TD, Meyers RJ. Seams and wedges in plastering: a 3D hexahedral mesh generation algorithm. *Engineering with Computers* 1993; **9**(2):83–93.
29. Owen SJ, Canann SA, Siagal S. Pyramid elements for maintaining tetrahedra to hexahedra conformability. *Proceedings Special Session on Trends in Unstructured Mesh Generation: 1997 Joint ASME/ASCE/SES Summer Meeting*, Northwestern University, Evanston, IL, AMD-Vol. 220, 29 June–2 July 1997; 123–129.
30. Leland RW, Melander DJ, Meyers RW, Mitchell SA, Tautges TJ. The geode algorithm: combining hex/tet plastering, dicing and transition elements for automatic, all-hexahedral mesh generation. *Proceedings of the 7th International Meshing Roundtable*, Dearborn, MI, 1998; 515–521.
31. Owen SJ. Non-simplicial unstructured mesh generation. *Ph.D. Thesis*, Carnegie Mellon University, 1999.
32. Tautges TJ, Blacker TD, Mitchell SA. The whisker-weaving algorithm: a connectivity based method for constructing all-hexahedral finite element meshes. *International Journal for Numerical Methods in Engineering* 1996; **39**:3327–3349.
33. Folwell NT, Mitchell SA. Reliable whisker weaving via curve contraction. *Proceedings of the 7th International Meshing Roundtable*, Dearborn, MI, 1998; 365–378.
34. Kawamura Y, Islam MS, Sumi Y. A strategy of automatic hexahedral mesh generation by using an improved whisker-weaving method with a surface mesh modification procedure. *Engineering with Computers* 2008; **24**(3):215–229.
35. Staten ML, Kerr RA, Owen SJ, Blacker TD. Unconstrained paving and plastering: progress update. In *Proceedings of the 15th International Meshing Roundtable*, Birmingham, AL, Pébay PP (ed.). 2006; 469–486. Session 5A.
36. Ehmann SA, Lin MC. Accurate and fast proximity queries between polyhedra using convex surface decomposition. In *EuroGraphics 2001*, Chalmers A, Rhyne T-M (eds), vol. 20(3). Blackwell Publishing: Oxford, 2001; 500–510.
37. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P. *Molecular Biology of the Cell* (4th edn). Garland Publishing: New York, London, 2002.
38. Carroll SB, Grenier JK, Weatherbee SD. *From DNA to Diversity: Molecular Genetics and the Evolution of Animal Design* (2nd edn). Blackwell Publishing: Malden, MA, 2004.
39. Yuan KY, Huang Y-S, Yang H-T, Pian THH. The inverse mapping and distortion measures for 8-node hexahedral isoparametric elements. *Computational Mechanics* 1994; **14**(2):189–199.
40. Janmey PA, Discher DE. Developmental biology: holding it together in the eye. *Nature* 2004; **431**:635–636.
41. Flory PJ. Thermodynamic relations for high elastic materials. *Transactions of the Faraday Society* 1961; **57**:829–838.
42. Stasa FL. *Applied Finite Element Analysis for Engineers*. Oxford University Press: Oxford, 1979.
43. Hager WW, Zhang H. A conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization* 2005; **16**(1):170–192.
44. Hager WW, Zhang H. CG descent, a conjugate gradient method with guaranteed descent. *ACM Transactions on Mathematical Software (TOMS)* 2006; **32**(1):113–137.
45. Fletcher R, Reeves CM. Function minimization by conjugate gradients. *The Computer Journal* 1964; **7**(2): 149–154.

## A NOVEL ENERGY-BASED APPROACH

46. Zhang J, Xu C. Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations. *Journal of Computational and Applied Mathematics* 2001; **137**(2):269–278.
47. Spencer AJM. *Continuum Mechanics*. Dover: New York, 2004.
48. Cass RJ, Benzley SE, Meyers RJ, Blacker TD. Generalized 3-D paving: an automated quadrilateral surface mesh generation algorithm. *International Journal for Numerical Methods in Engineering* 1996; **39**(9): 1475–1489.
49. Yogeve O, Antonsson EK. A novel synthesis design approach for continuous inhomogeneous structures. *19th International Conference on Design Theory and Methodology (DTM)*, Las Vegas, NV, September 2007. ASME, Paper Number: DETC2007/DTM-35662.
50. Yogeve O, Antonsson EK. Growth and development of continuous structures. *GECCO 2007, Genetic and Evolutionary Computation Conference*, London, U.K., 2007; 1064–1065.