ELSEVIER

Composites: Part A 31 (2000) 87–94

composites
Part A: applied science and manufacturing

www.elsevier.com/locate/compositesa

Numerical simulation of injection/compression liquid composite molding. Part 1. Mesh generation

K.M. Pillai[a,1], C.L. Tucker III[a,*], F.R. Phelan Jr[b]

[a]Department of Mechanical and Industrial Engineering, University of Illinois, 1206 W. Green Street, Urbana, IL 61801, USA
[b]Polymer Composites Group, Polymers Division, Building 224, Room B108, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA

Accepted 14 June 1999

Abstract

This paper presents a numerical simulation of injection/compression liquid composite molding, where the fiber preform is compressed to a desired degree after an initial charge of resin has been injected into the mold. Due to the possibility of an initial gap at the top of the preform and out-of-plane heterogeneity in the multi-layered fiber preform, a full three-dimensional (3D) flow simulation is essential. We propose an algorithm to generate a suitable 3D finite element mesh, starting from a two-dimensional shell mesh representing the geometry of the mold cavity. Since different layers of the preform have different compressibilities, and since properties such as permeability are a strong function of the degree of compression, a simultaneous prediction of preform compression along with the resin flow is necessary for accurate mold-filling simulation. The algorithm creates a coarser mechanical mesh to simulate compression of the preform, and a finer flow mesh to simulate the motion of the resin in the preform and gap. Lines connected to the top and bottom plates of the mold, called spines, are used as conduits for the nodes. A method to generate a surface parallel to a given surface, thereby maintaining the thickness of the intermediate space, is used to construct the layers of the preform in the mechanical mesh. The mechanical mesh is further subdivided along the spines to create the flow mesh. Examples of the three-dimensional meshes generated by the algorithm are presented.

Keywords: Liquid composite molding (LCM); E. Resin transfer molding (RTM)

1. Introduction

Liquid composite molding (LCM) is emerging as an important technology to make net-shape parts of polymer–matrix composites [1]. In any LCM process, a preform of reinforcing fibers is placed in a closed mold, then a liquid polymer resin is injected into the mold to infiltrate the preform. When the mold is full, the polymer is cured by a crosslinking reaction to become a rigid solid. Then the mold is opened to remove the part. LCM processes offer a way to produce high-performance composite parts using a rapid process with low labor requirement.

This paper deals with a particular type of LCM process called injection/compression liquid composite molding (I/C-LCM). In I/C-LCM, unlike other types of LCM processes, the mold is only partially closed when resin injection begins. This increases the cross-sectional area available for the resin flow, and decreases flow resistance by providing high porosity in the reinforcement. Often, the presence of a gap at the top of the preform further facilitates the flow. After all of the resin has been injected, the mold is slowly closed to its final height, causing additional resin flow and saturating all portions of the preform. The I/C-LCM process fills the mold more rapidly, and at a lower pressure than the other LCM processes that use injection alone (see Ref. [2] for more details of the process).

Complete filling of the mold with adequate wetting of the fibers is the primary objective of any LCM mold designer; incomplete filling in the mold leads to production of defective parts with dry spots. There are many factors which affect the filling of the mold: permeability of the preform, presence of gaps in the mold to facilitate resin flow, arrangement of inlet and outlet gates, injection rates of resin from different inlet ports, etc. [3–5]. Often it is not possible for the mold designer to visualize and design an adequate system for resin infusion by intuition alone, and mold filling simulations [2,5–7] are used to optimize mold performance. The situation in I/C-LCM is more complex than ordinary LCM because of compression of the mold during the filling operation. As a result, numerical simulation of the mold filling process in I/C-LCM becomes all the more important.

* Corresponding author. Tel.: + 1-217-333-2692; fax: + 1-217-244-6534.

[1] Present address: Center for Composite Materials, University of Delaware, Newark, DE 19716, USA.

1359-835X/00/$ - see front matter
PII: S1359-835X(99)00052-4

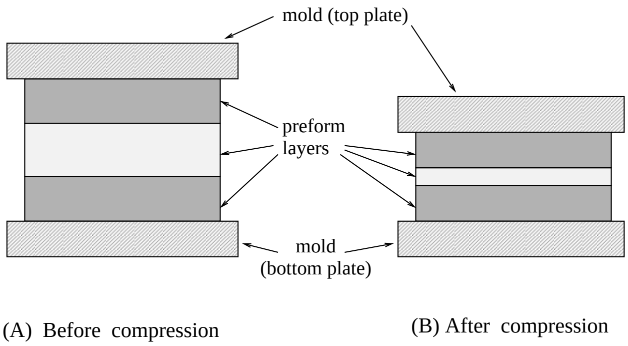

Fig. 1. Uneven deformation of preform layers under compression.

I/C-LCM fiber preforms frequently comprise layers of different reinforcing materials such as biaxial woven fabrics, stitch-bonded uniaxial fibers, random fibers. Each type of material has a unique behavior as it is compressed in the mold. When such different materials are layered to form the preform, each of them will compress by different amounts as the mold is closed. This behavior is illustrated in Fig. 1, which shows a small piece of a mold. Here the lighter center layer deforms much more than the darker outer layer as the mold is closed.

Capturing this deformation behavior during compression is critical to the accuracy of any I/C-LCM process model. Resin flows through the preform at all stages of compression, and the porosity and permeability of the preform are critical in determining the resin flow. The ratio of deformed volume to initial volume determines the porosity of each preform layer, and from this one can determine the layer's permeability, either from a theoretical prediction or a correlation of experimental data [8,9]. Because of this strong coupling between the state of compression in a preform layer and its permeability, computations for fluid flow and preform compression have to be done simultaneously for mold filling simulations in I/C-LCM.

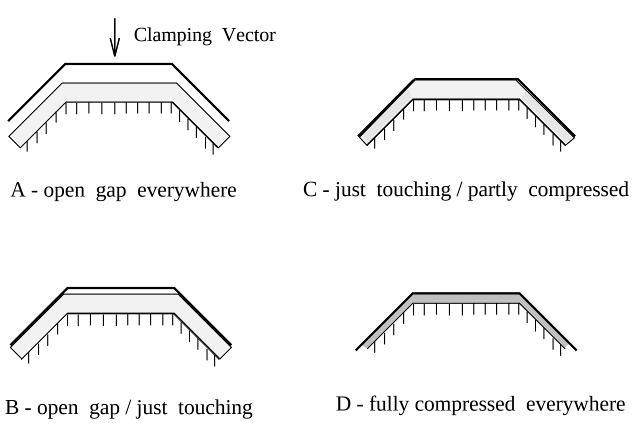

Fig. 2. A schematic describing the various stages of the compression/injection molding process. The top plate of the mold moves along the clamping vector, while the bottom plate is stationary. Stages A–C are three possible starting positions of the top plate. Stage D shows the final configuration of the mold when it is fully compressed.

Significant steps have already been taken to computationally model the mold filling in the I/C-LCM process. A computer program called CRIMSON [2,10], is capable of isothermal mold filling simulation which involves simultaneous fluid flow and preform compression computations in the flow domain. But the initial capacity of CRIMSON is limited to two-dimensional (2D) planar geometries where prediction of preform compression is straightforward. Deformation of the preform is modeled using the incremental linearized theory of elasticity; the mathematics simplifies due to reduction in the number of degrees of freedom (DOF) associated with displacement from the usual three to one along the thickness direction. However parts made by the I/C-LCM process typically have complicated three-dimensional shapes and this reduction of the mathematical complexity is no longer possible. The present paper describes our effort to expand the capability of CRIMSON by enabling it to tackle any arbitrary non-planar three dimensional (3D) mold geometry.

Most injection molding simulation programs read for the mold geometry in the form of a shell mesh [2,6,7]. Even if it were possible to transmit the full geometrical information about the mold through a 3D mesh, it still is difficult to incorporate all the information of relevance to the process engineer. The latter needs to know the thicknesses of various layers of fiber mats and their corresponding porosities at each time step. As a result, it is very important that elements representing different layers of preform in the 3D finite element mesh fall within separate layered regions. Overlap of an element onto more than one region is not acceptable as the element has to carry the material properties, such as porosity, permeability, of only one fiber mat. Mesh-generators in state-of-the-art commercial software such as PATRAN [11] are not designed to generate such a 3D mesh. Consequently, we decided to create a preprocessor suitable for I/C-LCM mold filling simulation.

The objectives of this paper are to introduce basic ideas about modeling mold filling in 3D I/C-LCM parts, and to introduce an algorithm to generate a 3D finite element mesh from a given 2D shell mesh for preform and flow computations. In subsequent papers [12,13], we will model finite deformation of preform using the non-linear theory of elasticity, and use this information to model resin flow in an I/C-LCM mold.

2. Generating a 3D mesh from the given 2D shell mesh

Our aim is to develop a preprocessor that can generate 3D finite element meshes for flow computations starting from a 2D shell mesh. We wish to allow the I/C-LCM process engineer to include all relevant information such as thicknesses of the layers of the preform, thickness of the gap, into the mesh.

Fig. 2 describes the three possible starting mold configurations (A–C) for a typical angular part geometry. Case A

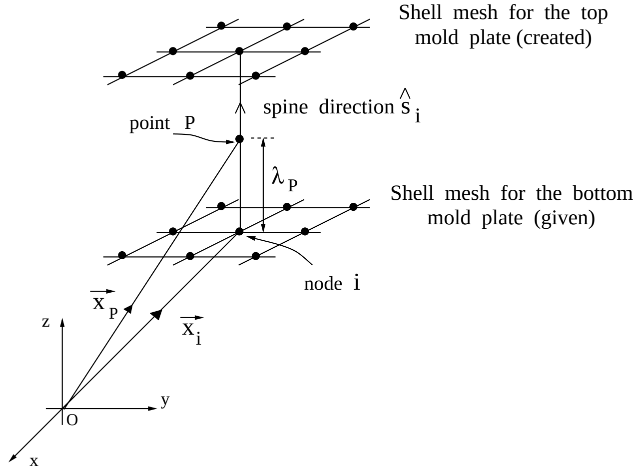

Fig. 3. Expressing the position vector $\vec{x}_P$ of a point P of the mechanical mesh in terms of the spine direction $\hat{s}_i$, the node position $\vec{X}_i$, and the spine parameter λ_P.

represents the starting configuration for the open mold injection/compression (I/C) molding, with ample gap between the top plate and preform. Cases B and C occur when the gap is partly or completely eliminated before the start of the injection process. In the former, the preform is completely uncompressed with gaps at a few places. In the latter, the gap is removed at the cost of partial compression of the preform in certain regions. In the present paper, mesh generation for configuration A only will be addressed. Once this mesh is created, cases B and C can be generated by solving for the mechanical compression of the preform [12].

As we shall see in the subsequent papers, six-noded wedge elements and eight-noded brick elements are adequate for modeling both the resin flow and preform compression. Our mesh generation algorithm is designed to generate such elements from the three- and four-noded triangular and quadrilateral elements of the shell mesh.

2.1. Mechanical and flow meshes

Development of the 3D mesh for flow computations from a given 2D shell mesh, representing the part geometry, is divided into two stages. In the first stage, an intermediate *mechanical* mesh is created, where the number of layers of elements equals the number of fiber mats in the lay-up, with the thickness of the mats equal to the height of those elements. Such a coarse mesh is adequate to track deformation of the mats during compression of the mold. In the second stage, the mechanical mesh is further subdivided along the thickness direction to create a more refined mesh, called the *flow* mesh, which is used for flow calculations.

3. Basic concepts of mesh generation algorithm

We first introduce two basic ideas that form the backbone of our mesh generation algorithm: spines and parallel surfaces.

3.1. Use of spines

One of the salient features of our mesh generation technique is the use of *spines* to track the nodes of the 3D mechanical mesh. This is similar to the use of spines in the free boundary problems [6,14] where they have been used to adapt the computational mesh with time. These spines are lines connecting node points of the top mold surface to their counterparts of the bottom mold surface.

In our scheme, nodes representing the preform between the two surfaces are constrained to move along these lines. Fig. 3 demonstrates the use of a spine to track the position of a point P of the mechanical mesh. The position of point P can be expressed as:

$$\vec{x}_P = \vec{X}_i + \lambda_P \hat{s}_i \tag{1}$$

where λ_P, $\hat{s}_i$ and $\vec{X}_i$ are the spine parameters, a unit vector representing the spine direction, and the position vector of node i of the shell mesh, respectively. $\vec{x}_P$ is the position

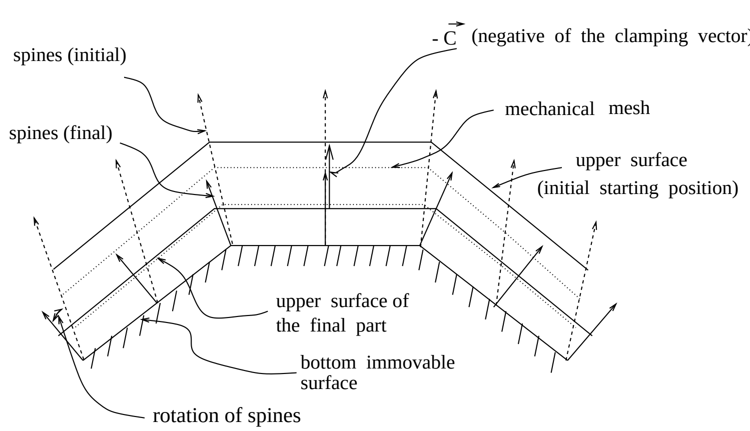

Fig. 4. Spines and their role in the creation of meshes.

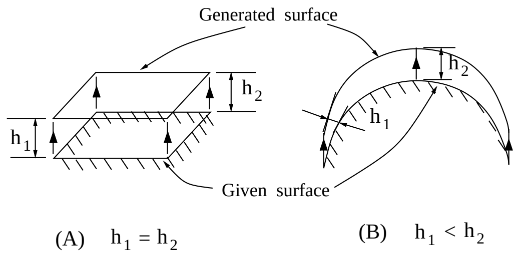

Fig. 5. Simple translation of a surface can be used to generate a parallel surface if the given surface is planar (part A). This method fails if the given surface does not lie in a plane (part B).

vector of point P. Since $\vec{X}_i$ is given and $\hat{s}_i$ can be computed, use of spines leads to a reduction in the nodal degree of freedom for mesh generation from the typical three position vector components to a single scalar variable λ (the spine parameter), representing the distance from the bottom surface along the spine.

Since the spines are connected to the upper and lower plates of the mold, they may rotate as the I/C-LCM mold is compressed, as indicated in Fig. 4. This is relevant in modeling preform compression that will be considered in our next paper [12].

3.2. Generating a parallel surface

A basic action of the mesh generation algorithm is to generate a shell mesh for a surface that is parallel to the shell mesh of another surface. The generated surfaces represent either fiber mat interfaces in the preform, or the moving upper plate of the I/C-LCM mold. As shown in Fig. 5, it is not possible to generate the parallel surface by simple translation if the given surface is non-planar. In the following section, we propose a method to generate the parallel surface, based on the principle of minimizing error between the given and projected heights. This method yields a simple analytical solution in terms of spine parameters.

Using linear interpolation, the position of any point inside a triangular or quadrilateral shell element can be represented by:

$$\vec{X} = \sum_{i}^{n_{\text{nie}}} N_i \vec{X}_i \tag{2}$$

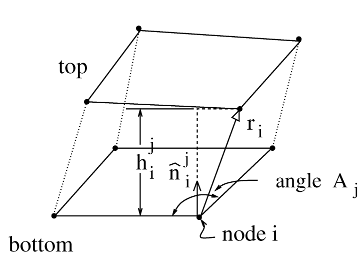

Fig. 6. Definition of the height h_i^j between the top surface with respect to the bottom surface expressed in terms of the normal and the displacement $\vec{r}_i$. The included angle A_j acts as a weighting factor in Eq. (5).

where $\vec{X}$ is the position vector of a point inside the element, N_i $(= N_i(\xi, \eta))$ the shape function, $\vec{X}_i$ the position vector of node i and n_{nie} the total number of nodes in the element. ξ and η are the local coordinates within the element.

Now at any point in the element a vector $\vec{\mathcal{N}}$ normal to the surface is given by:

$$\vec{\mathcal{N}} = \left(\sum_{i}^{n_{\text{nie}}} \frac{\partial N_i}{\partial \xi} \cdot \vec{X}_i \right) \times \left(\sum_{i}^{n_{\text{nie}}} \frac{\partial N_i}{\partial \eta} \cdot \vec{X}_i \right). \tag{3}$$

The corresponding unit normal $\hat{n}$ can be computed by dividing $\vec{\mathcal{N}}$ by its magnitude.

If $\hat{n}_i^j$ denotes the unit normal at node i of element j, then the direction $\vec{S}_i$ of the spine passing through the node is computed as the average of normals of the surrounding elements:

$$\vec{S}_i = \frac{1}{n_{\text{surr}}} \sum_{j}^{n_{\text{surr}}} \hat{n}_i^j \tag{4}$$

where n_{surr} is the number of elements surrounding node i. Once again, the corresponding unit spine direction $\hat{s}_i$ is computed by dividing $\vec{S}_i$ by its magnitude.

The shell mesh of the second surface is generated by minimizing the net deviation of projected heights from the given height as follows:

$$I = \sum_{i}^{n_{\text{nodes}}} \left(\sum_{j}^{n_{\text{surr}}} (h_{\text{i}}^j - h_{\text{o}}^j)^2 A_j \right) \tag{5}$$

where I is the error function which is to be minimized, A_j a weighting factor which is equal to angle subtended by element j at node i, h_{i}^j is the height of the top surface with respect to the bottom surface, and h_{o}^j is the specified thickness of the final part at element j.

Let $\vec{r}_i = \lambda_i \hat{s}_i$, where $\vec{r}_i$ is the displacement vector shown in Fig. 6. By projecting $\vec{r}_i$ onto $\hat{n}_i^j$, the heights can be computed as follows:

$$h_{\text{i}}^j = (\hat{n}_i^j \cdot \hat{s}_i) \lambda_i. \tag{6}$$

On substituting Eq. (6) into Eq. (5), simplifying, and setting $\partial I / \partial \lambda_i$ equal to zero (see Ref. [15] for details), one gets a closed-form expression for the spine parameter

$$\lambda_i = \frac{\hat{s}_i \cdot \left(\sum_{j}^{n_{\text{surr}}} h_{\text{o}}^j \hat{n}_i^j A_j \right)}{\sum_{j}^{n_{\text{surr}}} (\hat{n}_i^j \cdot \hat{s}_i)^2 A_j}. \tag{7}$$

Since $\partial^2 I / \partial \lambda_i^2 > 0$, the set of spine parameters thus obtained ensures a minimum of I. Substituting the λ_i into Eq. (1) gives the nodal coordinates of the desired parallel surface.

4. Algorithm

The main actions carried out in our mesh generation algorithm are as follows:

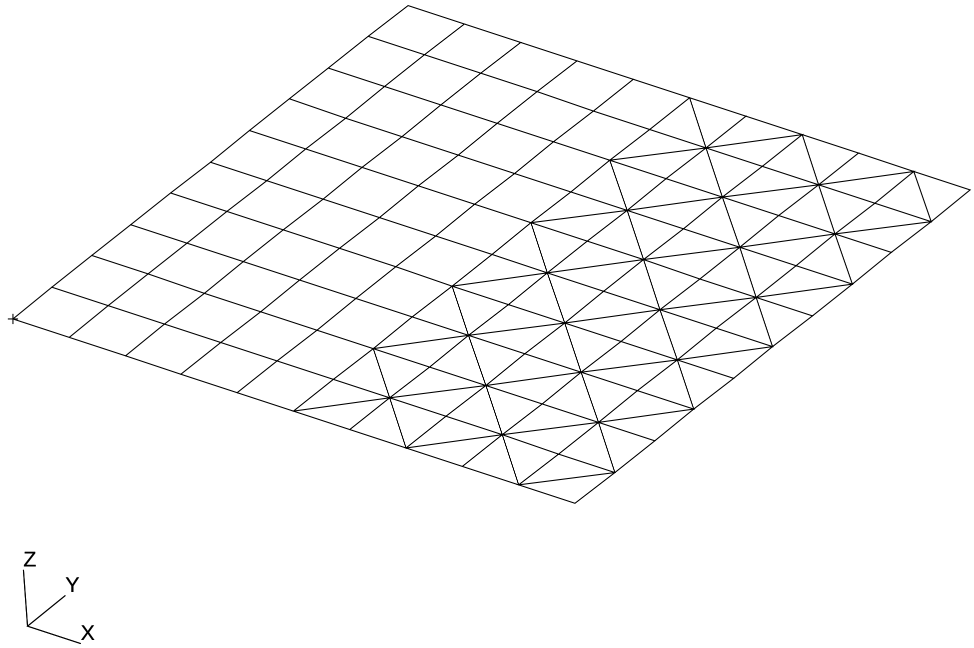

Fig. 7. A shell mesh consisting of a rectangular planar patch of quadrilateral and triangular shell elements.

1. *Read data describing the 2D shell mesh.* The mesh data is read, along with the information important for process modeling such as direction of clamping, properties of fiber mats, initial gap provided at the top of the preform.
2. *Construct the upper surface of the final part.* The upper surface is generated parallel to the input 2D shell mesh which represents the bottom, immovable surface of the mold. The input thicknesses between the given and upper surfaces are taken to be the final thickness of the I/C-LCM mold (equal to the desired part thickness).
3. *Pull out the upper surface to its starting position and solve for the rotated spines.* Movement of the top plate with respect to the fixed bottom plate[2] is responsible for compression in the I/C-LCM mold and distortion in the computational mesh. The motion of the top plate is prescribed by a vector called the clamping vector $\vec{C}$, whose direction and magnitude are equal to the direction and extent of this motion (see Fig. 2). The magnitude of the clamping vector $|\vec{C}|$ is assigned in the beginning and is estimated by translating the top mold plate (or surface) from its final position along the direction antiparallel to the clamping vector, until the height between the two mold plates equals the desired initial height between them (see Fig. 4). This initial height is equal to the sum of thicknesses of all fiber mats in the preform stack in the uncompressed state, plus the height of any initial gap provided above the preform. When the upper mold surface is translated to its initial position, the position of a translated top surface point is $\vec{\bar{X}}_i^{\,t}$ given by

$$\vec{\bar{X}}_i^{\,t} = \vec{X}_i^{t} - |\vec{C}|\hat{c} \tag{8}$$

where $\hat{c}$ is a unit vector along the direction of the clamping vector $\vec{C}$. Once the coordinates of nodes of the translated top surface are known, the directions of new rotated spines can be computed as:

$$\hat{s}_i = \frac{\vec{\bar{X}}_i^{\,t} - \vec{X}_i}{|\vec{\bar{X}}_i^{\,t} - \vec{X}_i|}. \tag{9}$$

4. *Create the mechanical mesh.* The mechanical mesh is generated by once again using the method of generating parallel surfaces. Increments in the spine parameters $\Delta\bar{\lambda}_{i,k}$ are calculated for the new rotated spines, where λ_i of Eq. (7) now represents such increments. Nodes of each interface layers k are created using the Eq. (1) as follows:

$$\vec{\bar{X}}_{i,k} = \vec{\bar{X}}_{i,k-1} + \Delta\bar{\lambda}_{i,k}\hat{s}_i \tag{10}$$

where $\vec{\bar{X}}_{i,k}$ and $\vec{\bar{X}}_{i,k-1}$ are position vectors of node i of shell meshes of layers k and $k-1$, respectively. Now the 3D elements of the mechanical mesh are created on the top of each 2D shell element by joining the corresponding nodes of any two neighboring fiber mat interfaces to create the interfaces of the fiber mats in the preform. The spacing between any two such interfaces is equal to the uncompressed thickness of the corresponding fiber mat. The gap at the top of the preform is treated as an extra layer to generate the gap elements.

[2] This is a convention that we are adopting; it is equally possible for the top plate to remain stationary.

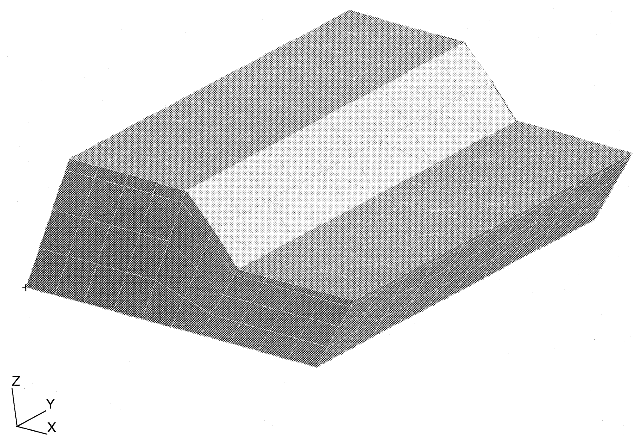

Fig. 8. The mechanical mesh created from the shell mesh shown in Fig. 7. There are three layers of reinforcement with one thin gap layer at the top; the reinforcement layers are of equal thicknesses of 0.1 and 0.05 units in the two material domains, corresponding to the rectangular and triangular elements respectively. The negative of the clamping vector $\vec{C}$ corresponds to the direction $\langle 1, 1, 1 \rangle$.

5. *Create the flow mesh.* The flow mesh is used to compute fluid flow through the multilayered preform, and may be much finer than the mechanical mesh. Therefore, each layer of the mechanical mesh is further subdivided into a pre-specified number of sublayers to create the flow mesh. Either Eq. (1) or Eq. (10) can be used to generate the nodes of intermediate layers of the flow mesh.

5. Examples and discussion

A computer program has been developed to implement the mesh generation algorithm, and tested for its efficacy and robustness. In the following sections, examples of the creation of 3D computational meshes from 2D shell meshes are presented. Since the thicknesses in the I/C-LCM parts are much smaller than their other dimensions, realistic meshes are relatively thin. To highlight important features of the algorithm, the thicknesses of the meshes are scaled up in the following examples. In each example, a gap that is a certain fraction of the total thickness of the uncompressed preform is provided between the upper surface of the preform and the top mold plate.

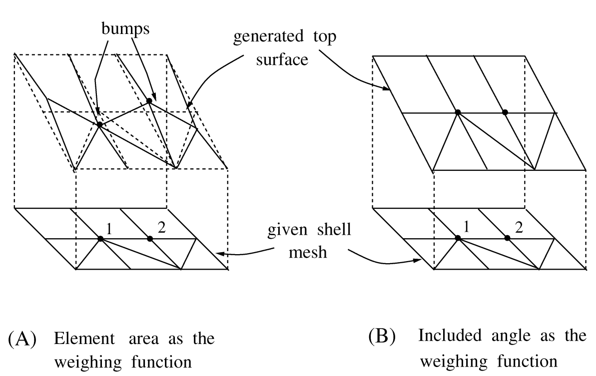

Fig. 9. Effect of weighting factor A_j in Eq. (7) while generating the top surface at the boundary of triangular and rectangular element regions: (a) creation of “bumps” on the surface when A_j is equal to the area of the surrounding element j; (b) bumps disappear when A_j is taken to be the included angle of element j at nodes such as 1 or 2.

5.1. 3D mesh from a planar rectangular shell mesh

Fig. 7 shows a rectangular shell mesh. Fig. 8 shows a mechanical mesh created from a rectangular shell mesh lying in the x–y plane as shown in Fig. 7. The shell mesh contains the two types of elements, quadrilaterals and triangles, which are used most often in LCM simulations [7]. In this example, each type of element corresponds to a separate material domain, and we assign different thicknesses to the layers of the preform in each domain. Due to averaging, the height of the topmost surface varies linearly across the boundary between the two domains. As a result of this averaging, a finer mesh is needed near the boundary for accurate representation of the step change in the mold height. A clamping vector along the $\langle 1, 1, 1 \rangle$ direction is chosen to create a more complicated motion of the top plate.

At this point, the importance of weighting factors while computing direction of spines in Eq. (7) should be mentioned. We discovered that using the areas of elements surrounding a node as weights in these equations leads to formation of “bumps” along an interface where elements of different types (triangles and quadrilaterals) touch (see Fig. 9). This unevenness of the top surface resulted from the difference in the number of elements surrounding alternating nodes along the interface, thereby changing the value of the spine parameter λ. The anomaly was corrected by using the included angle of each surrounding element at a node as the weighting factor A_j in Eqs. (5) and (7).

The mechanical mesh of Fig. 8 is further subdivided along the spine direction to create the flow mesh shown in Fig. 10.

5.2. 3D mesh from an arbitrary shell mesh in three dimensions

Fig. 11 shows a shell mesh created from a surface patch that is curved and undulates in three dimensions. The quadrilateral elements making up the shell mesh are no longer co-planar, and the normals of the elements surrounding each node (used to compute the spine parameter in Eq. (7)) are no longer parallel. As a result, this provides a good test of the robustness and general applicability of our method. Fig. 12 shows the mechanical mesh created by stacking three layers of reinforcement, with one thin gap layer at the top. The negative of the clamping vector is once again $\langle 1, 1, 1 \rangle$. As one can see, our method is successful in creating an adequate 3D mesh. This mesh can be further subdivided

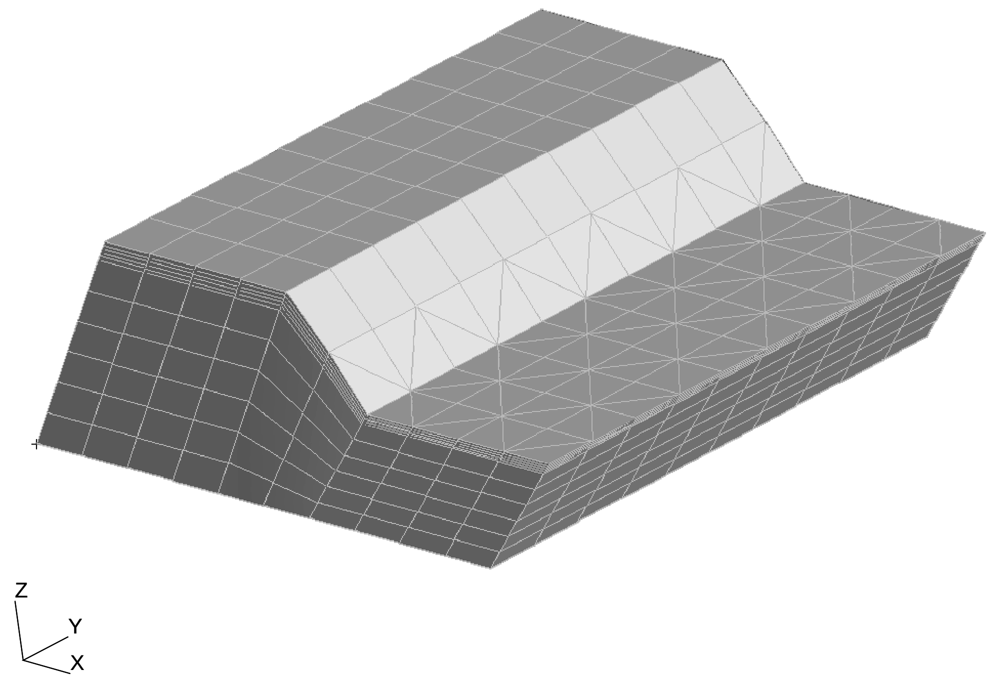

Fig. 10. The flow mesh created by further subdividing the mechanical mesh of Fig. 8 along the spines. The gap and preform layers are divided into five and two sublayers, respectively.

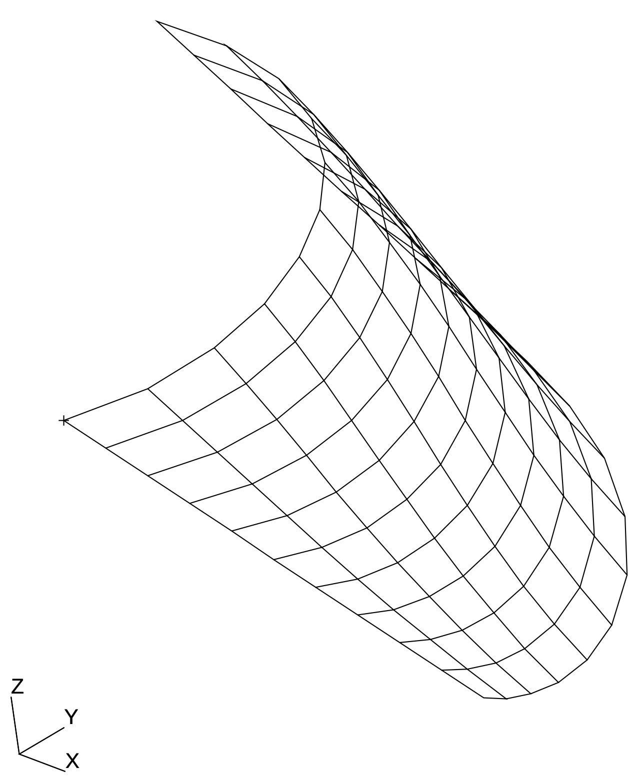

Fig. 11. A representative shell mesh created out of an arbitrary surface patch. Notice that no two elements are coplanar, and consequently the normals of the elements surrounding any node are all dissimilar.

along the spines to create a flow mesh. The aspect ratio of the elements can be improved by refining the mesh.

6. Summary and conclusions

In this paper, we present a methodology to create 3D finite element meshes for modeling mold filling in I/C-LCM. We propose the concept of predicting preform compression using the coarse mechanical mesh, and predicting fluid flow using the finer flow mesh. A mesh-generating algorithm, to create the mechanical and flow meshes from a given shell mesh, is presented. This algorithm incorporates information about the position of fiber mat interfaces in a multi-layered preform, which is crucial for accurate modeling of the filling process. A technique to create surfaces parallel to any arbitrary shell mesh surface enables us to represent the interfaces accurately. Further, the use of spines in mesh generation reduces the number of unknowns at each node from three to one. The algorithm is used successfully to create the mechanical and flow meshes from two different shell meshes; its robustness is demonstrated by creating a 3D mesh from a shell mesh for an arbitrary mold shape. The need to refine the shell mesh in the region of a step change in the thickness of the mold is the main limitation of the algorithm. In subsequent papers, we will use the mechanical and flow meshes to simulate preform compression and resin flow during mold filling in I/C-LCM.

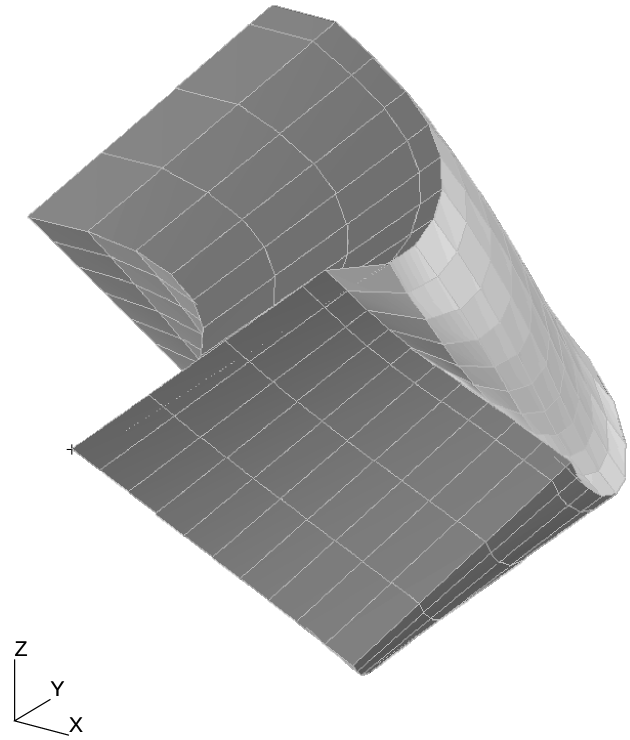

Fig. 12. A mechanical mesh created from the shell mesh of Fig. 11. There are three equally thick reinforcement layers of thicknesses 0.1 units each, with the “gap” layer at the top.

Acknowledgements

Financial support for this work was provided by the National Institute of Standards and Technology.

References

[1] Rudd CD, Long AC, Kendall KN, Magin CGE. Liquid molding technologies, Woodhead Publishing Ltd, 1997.
[2] Phelan Jr FR. Simulation of the injection process in resin transfer molding. Polymer Composites 1997;18(4):460–76.
[3] Pillai KM, Luce TL, Bruschke MV, Parnas RS, Advani, SG. Modeling the heterogeneties present in performs during mold filling in RTM. In: Advanced materials: expanding the horizons, 25th International SAMPE Technical Conference, vol. 25, 1993.
[4] Pillai KM, Advani SG. A model for unsaturated flow in woven or stitched fiber mats during mold filling in resin transfer molding. Journal of Composite Materials 1998;23(19):1753–83.
[5] Advani SG, editor. Flow and rheology in polymer composites manufacturing Amsterdam: Elsevier, 1994.
[6] Tucker III CL, editor. Fundamental of computer modeling for polymer processing Munich: Hanser, 1989.
[7] Bruschke MV, Advani SG. filling simulation of complex three dimensional shell-like structures. SAMPE Quarterly 1991;23(1):2–11.
[8] Lekakou C, Johari MAKB, Bader MG. Compressibility and flow permeability of two-dimensional woven reinforcements in the processing of composites. Polymer Composites 1996;17(5):666–72.
[9] Pearce N, Summerscales J. The compressibility of a reinforcement fabric. Composites Manufacturing 1995;6(1):15–21.
[10] Phelan FR Jr. Analysis of injection/compression liquid composite molding process variants. In: ASME Proceedings, November 1996. Presented at the 1996 ASME International Mechanical Engineering Congress and Exhibition Atlanta, GA.
[11] Patran 3: user manual. PDA Engineering, 2975 Redhill Avenue, Costa Mesa, CA 92626, 1994.
[12] Pillai KM, Tucker CL III, Phelan FR Jr. Numerical simulation of liquid composite injection/compression molding. Part 2. Modeling preform compression. Composites Part A; submitted.
[13] Phelan FR Jr, Pillai KM, Tucker CL III. Numerical simulation of liquid composite injection/compression molding. Part 3. Modeling mold-filling process. Composites Part A; submitted.
[14] Kistler SF, Scriven LE. Coating flows. In: Pearson JRA, Richardson SM, editors. Computational analysis of polymer processing, Barking: Applied Science, 1983.
[15] Pillai KM, Tucker CL III. Progress report: 1 January–31 March. Technical report, University of Illinois, 1997.