A novel approach to reduce calculation time during simulations of resin transfer molding

Sofiane Soukane — Edu Ruiz — François Trochu

Centre de Recherches Appliquées Sur les Polymères (CRASP)
Dpt de Génie Mécanique
École Polytechnique de l'Université de Montréal
H3C 3A7, Canada

{sofiane.soukane; eduardo.ruiz;trochu@polymtl.ca}

ABSTRACT. *The present work aims to reduce the computational burden associated with 3D simulations of Resin Transfer Molding (RTM). The goal is achieved by minimizing the number of elements required to describe the geometrical domain. A 2D mesh refinement technique based on edge subdivision and swapping is coupled to a mesh extrusion algorithm to generate a three-dimensional computational domain composed of multiple layers of planar elements. The refinement optimizes the mesh by stretching or concentrating the elements in desired locations while the full geometry is constructed from relatively low number of elements. Beyond the extrusion algorithm robustness, the solid mesh extruded through the thickness of the composite mirrors the detailed structure of the laminate, including sandwich components. Mold filling calculations are carried out on the extruded mesh using a non-conforming finite element formulation.*

KEYWORDS: *Resin Transfer Molding, mesh refinement, mesh extrusion, prismatic element, non-conforming finite element.*

1. Introduction

Because they tend to replace classical materials with a lighter weight alternative and equivalent or improved mechanical properties, composites have nowadays reached a critical importance. Engineering advances quickly and composite uses have already extended to boat manufacture, sports equipment, biomedical components, underground pipes, and notably automotive components and aerospace structures. In the last fifteen years a set of processes have been developed to manufacture composite parts and termed, *Liquid Composite Molding* (LCM). As a member of LCM family of processes, RTM involves injecting a liquid resin through fibrous reinforcements previously placed in the cavity of a closed and rigid mold. From the resin injection to curing of the final part, several physical phenomena affect quality at each stage of the process. Process improvement has been the driving force for a better understanding of these phenomena and has resulted in the ability to predict mold filling (Trochu *et al*., 1993), saturation (Ruiz *et al*., 2004, Bréard *et al*., 2003) and heat exchanges in the mold during resin injection and cure (Ruiz *et al*., 2004). Modeling and simulation are thus key steps in the design and optimization of RTM molds.

Much like many engineering physics problems, the filling and curing stages in RTM can be modeled by a set of partial differential equations (PDEs). The mold often has a complex shape and the PDEs cannot be solved by analytical means. Moreover, complex interactions occur during the filling stage due to simultaneous momentum, heat and mass transfers. Hence, numerical methods are used to seek a solution and predict the evolution in time of physical process parameters, such as pressure and temperature, throughout the mold cavity. Several approaches can be taken depending on the degree of coupling assumed. In general, for a given discretization of the domain under consideration, the more complex the coupling, the larger the computational time required to analyze the process**.** A successful simulation also results from a proper domain discretization. Indeed, mesh generation associates degrees of freedom, (*i.e*., unknown values of the variables to a finite number of points judiciously selected in the geometric domain). The mesh has to be fine enough to capture the physical phenomena investigated and coarse enough to allow rapid calculations. Model simplification also allows running simplified simulations during the design stage before a full analysis is carried out. However, several situations exist where no simplification is possible and all previously mentioned physical phenomena need to be considered in the model. This typically occurs when the mold is heated at a different temperature than the resin. In which case, the resin viscosity changes through the thickness of the part. This results in irregular flows, usually faster inside the skin when the mold is maintained at a higher temperature than the resin. Yet, through thickness flow may also occur if the parts contain multi-layer fibrous reinforcement or high fiber volume contents. In such situations full coupled three-dimensional analyses have to be conducted. Otherwise model simplification would lead to poor predictions.

The goal of this study is to minimize the number of elements required in the simulation of resin transfer molding. Numerical simulation efficiency is improved by using a remeshing technique that refines the mesh or stretches the elements in desired geometric regions. The development of 3D remeshing techniques is still at the leading edge of mesh generation and implementation requires tremendous efforts. This paper examines how 2D remeshing is undertaken. This capability is enhanced by an extrusion algorithm to generate 3D multi-layer data mesh structures composed of prismatic elements. The mesh extrusion performed through the thickness of the composite not only has the ability to build the 3D meshes, but can also reflect accurately the structure of the laminate. A new non-conforming prismatic element has been developed. It is shown that this type of element can lead to more efficient and accurate RTM mold filling simulations.

2. The resin transfer molding process

As mentioned above, RTM consists of injecting a resin through fibrous reinforcements already placed in the mold cavity. The main assumptions made about RTM modeling are summarized as follows:

– the reinforcement is considered as a porous medium. Fibrous reinforce-ment includes empty spaces (inner tow and inter fiber spaces), is permeable to a wide variety of fluids and can then be considered porous (Dullien 1979);

– the medium is assumed to be homogeneous. Volume averaging techniques based on intermediate characteristic lengths between the size of inter-fiber spaces to the overall extent of the material have been successfully applied to model porous media (Parnas 2000). However, this assumption has limitations, for example, when the heat transfer in the liquid phase (resin) follows a different regime than in the solid phase (fibers). However this type of situation can still be handled using empirical or semi-empirical correlations to include hydrodynamic or thermal dispersion;

– the resin flow is assumed to be in a creeping regime. This is almost always true. Indeed, resin injection is slowly and carefully conducted to ensure proper fiber impregnation and prevent the formation of voids and confinement of air during the filling stage;

– the resin is assumed to behave as a Newtonian fluid, (i.e., during flow, the fluid strain rate is equal to the velocity gradient defined perpendicularly to the flow direction).

In the case of the above-mentioned assumptions, the resin flow is governed by the widely known Darcy's equation (Darcy 1856). It states that the amount of fluid (resin) flowing through a particles bed (fibers) confined in a cavity (mold) is proportional to the gradient of fluid pressure. This law establishes a relationship between the average fluid velocity (Darcy's velocity) $\langle \vec{v} \rangle$, (*i.e.*, the flow rate per unit area), and the pressure gradient ∇P as follows:

$$\langle \vec{v} \rangle = -\frac{[K]}{\mu} \langle \nabla P \rangle \qquad [1]$$

where [*K*] is the permeability tensor and μ the resin viscosity,

Equation [1] represents an undetermined system since both the velocity and the pressure are unknowns. Another relation must be established to close the system and solve for the resin flow in the fibrous reinforcement. The continuity equation is used here with the following assumptions:

– the mold is considered rigid. This is always valid in RTM since the mold is usually metallic (steel or aluminum). One need only ensure that the mold cover is not allowed to move in any direction;

– the reinforcement placed in the cavity is also considered rigid. Although this is not always true, the fiber bed can be considered as non-deformable as long as it is properly confined in the mold cavity and its deformations remain negligible;

– the resin is incompressible, which is true for liquids.

Under these considerations, the equation of mass conservation for the fluid phase simplifies into:

$$div(\rho . v_r) = 0 \qquad [2]$$

where ρ is the density of the resin injected and v_r is the superficial resin velocity, *i.e*., the velocity of the flow front during mold filling. Note that this latter term is related to Darcy’s velocity via the porosity ϕ of the porous medium by:

$$v_r = \phi v \qquad [3]$$

Finally injecting equation [1] in [2] leads to the partial differential equation that governs the fluid flow:

$$div\left(\frac{[K]}{\mu} \langle \nabla P \rangle \right) = 0 \qquad [4]$$

Considering isothermal filling conditions the viscosity of the resin is assumed to be constant. Equation [4] has been solved successfully using a finite element/finite volume mixed formulation with conforming (Bruschke *et al*., 1990, Fracchia *et al*., 1989) or non-conforming shell elements (Trochu *et al*., 1993).

3. Finite element formulation

To solve the elliptic PDE stated in equation [4] that governs the resin flow a weak formulation can be expressed in terms of weighted residuals (Trochu *et al*., 1993, Remacle *et al*., 1998). Defining the shape function as scalar *p* and introducing a space *F*(Ω) of test functions *w*, equation [4] is replaced by an equivalent integral on the domain Ω, namely:

$$\int_{\Omega} w \cdot div\left(-\frac{1}{\mu}[K].\nabla p \right) \partial\Omega = 0 \qquad \forall w \in F(\Omega) \qquad [5]$$

Integrating by parts and applying Green's theorem leads to:

$$\int_{\Omega} \nabla w \cdot \left(-\frac{1}{\mu}[K].\nabla p \right) \partial\Omega = \int_{\Gamma_d} \left(w \cdot (\hat{n} \cdot \bar{v}) \right) \partial\Gamma_d \ , \quad \forall w \in F(\Omega) \, , \ \Gamma_d = \Gamma_1 \cup \Gamma_2 \qquad [6]$$

where Γ_1 and Γ_2 represent the domain boundaries where natural and essential boundary conditions are to be assigned respectively.

The finite element method consists of finding a set of functions *w* that approximate the scalar potential *p* on the whole domain Ω. The domain Ω is divided into sub-domains (elements) of simple geometrical shape, where *p* is approximated by a linear combination of shape functions. For general shape functions s_n and test functions s_j associated to the group of nodes in the finite sub-domain Ω_e , the scalar *p* is then estimated using:

$$p(x) = \sum_{n=1}^{N} p_n s_n(x) \qquad [7]$$

where $s_n(x)$ is a piecewise linear shape function and *x* denotes the position vector in the domain Ω. The Galerkin formulation gives:

$$\sum_{n=1}^{N_e} \left[p_n \int_{\Omega_e} \nabla s_j \cdot \left(\frac{[K]}{\mu} \nabla s_n \right) \partial\Omega_e \right] = \int_{\Gamma_e} \left(s_j \cdot (\hat{n} \cdot \bar{v}) \right) \partial\Gamma_e \, , \qquad \forall s_j \in \Omega_e \qquad [8]$$

where N_e are the element nodes and Γ_e denotes the boundary of the finite sub-domain Ω_e . In matrix notation, equation [8] can be rewritten as follows:

$$[M]\{p\} = \{R\} \qquad [9]$$

where [*M*] represents the *N*x*N* stiffness matrix of the scalar potential field. An element of matrix *M* is defined as:

$$m_{jn} = \int_{V_e} \nabla s_j \cdot \left(\frac{[K]}{\mu} \nabla s_n \right) \partial \Omega_e \ , \quad \text{for} \quad j,n = 1,...,N \qquad [10]$$

and $\{R\}$ is a vector $\{r_j\}$ containing Neumann boundary conditions, *i.e.*, the injection flow rates imposed along some parts of the boundary:

$$r_j = \int_{\Gamma_{1e}} \left(s_j \cdot (\hat{n} \cdot \bar{v}_s) \right) \partial \Gamma_{1e} \ , \quad j = 1,...,N \ \text{ and } \Gamma_{1e} = \Gamma_e \cap \Gamma_1 \qquad [11]$$

where $\bar{v}_s$ is the velocity specified on Γ_1. The approximate solution of the scalar potential p is then obtained by solving the linear system [9] of N equations with N unknowns.

3.1. *Non-conforming triangle*

In the present approach, the finite element is considered to represent the control volume where the mass balance equation is solved. The fluid mass transported between adjacent elements is the scalar product of the normal vector at the interface by the mass velocity $\rho \cdot \bar{v}$ (Figure 1a). In order to ensure mass conservation at elements interface, non-conforming finite element approximations are used. A discontinuous shape function may be obtained by interpolating the pressure field on the element edges as depicted in Figure 1c in the 2D case. In the 3D case, the interpolation is performed on the faces of each solid element. This non-conforming interpolation ensures a continuous mass flux, *i.e.*, a constant $\hat{n} \cdot \rho \bar{v}$ across the interface (Trochu *et al.*, 1993).

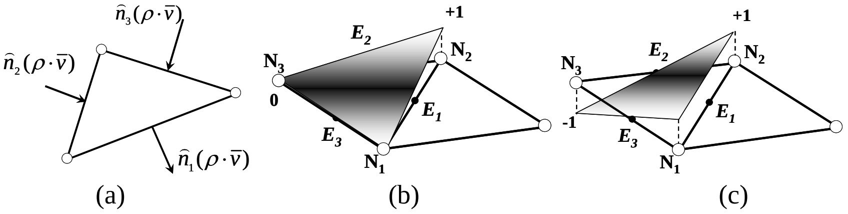
$\hat{n}_3(\rho \cdot \bar{v})$
$\hat{n}_2(\rho \cdot \bar{v})$
$\hat{n}_1(\rho \cdot \bar{v})$
(a)
+1
E_2
N_3
N_2
0
E_1
E_3
N_1
(b)
+1
N_3
E_2
N_2
-1
E_1
E_3
N_1
(c)

Figure 1. *Triangular finite element: (a) resin flow; (b) conforming shape function; (c) non-conforming shape function*

For the triangular non-conforming finite element of Figure 2a, the three degrees of freedom are assigned at the mid-points of the element edges.

In an isoparametric approximation, the element shape functions s_e write as follows (Figure 2b):

$$s_e = \begin{cases} s_1 = 1 - 2\eta \\ s_2 = 2(\zeta + \eta) - 1 \\ s_3 = 1 - 2\zeta \end{cases} \qquad 0 \le \zeta, \eta \le 1 \qquad [12]$$

where ζ and η are the local coordinates in a reference system.

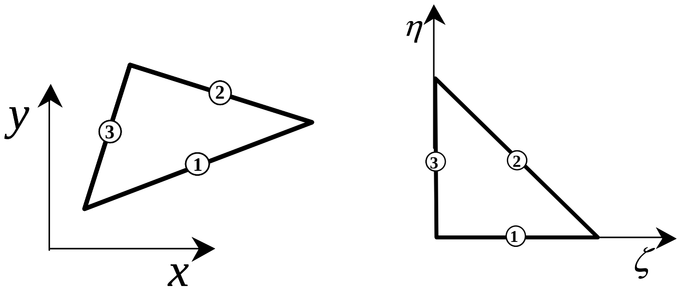

Figure 2. *Non-conforming shape functions on a triangle and reference element: nodes are located at the middle of element edges*

3.2. *Non-conforming prism*

In RTM processing, most composite parts are considered to be thin shells and simulated using 2D finite element approximation with through-thickness averaging. However in the case of laminates, 3D simulations are often required to accurately predict the injection time and verify that no air is entrapped during mold filling. Prismatic elements constructed by extrusion from a triangular mesh simplify considerably the mesh generation task. For this reason, the non-conforming prismatic element (shown in Figure 3) is introduced with degrees of freedom extended on the edges of the bottom and top triangular faces.

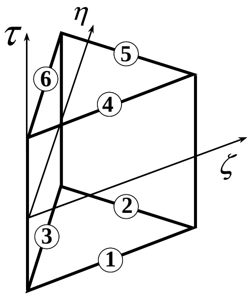

Figure 3. *Non-conforming prismatic element*

The linear function space will have now 6 degrees of freedom. The element shape functions s_e are:

$$s_e = \begin{cases} s_1 = (1-2\eta)\cdot(1-\tau)\cdot 0{,}5 \\ s_2 = (2\zeta + 2\eta - 1)\cdot(1-\tau)\cdot 0{,}5 \\ s_3 = (1-2\zeta)\cdot(1-\tau)\cdot 0{,}5 & 0 \le \zeta, \eta \le 1 \\ s_4 = (1-2\eta)\cdot(1+\tau)\cdot 0{,}5 & -1 \le \tau \le 1 \\ s_5 = (2\zeta + 2\eta - 1)\cdot(1+\tau)\cdot 0{,}5 \\ s_6 = (1-2\zeta)\cdot(1+\tau)\cdot 0{,}5 \end{cases} \qquad [13]$$

where the third dimension τ is referenced from the element midplane.

When 3D simulations are required to predict accurately the injection time and detect local filling problems, this new prismatic element can advantageously replace tetrahedrons to save computer time by reducing the number of degrees of freedom. It provides also a good accuracy with the non-conforming formulation because it ensures a local mass conservation (Ruiz, 2004). This element contains 6 nodes and results from an extrusion of a triangular shell element. Its straightforward generation from triangle extrusion avoids a tedious mesh generation while also allows the simultaneous processing of layers of stacked elements for composite laminate analysis.

The unsteady flow can be solved by considering a succession of approximations in space and time. Every time step the resin front is relocated using the computed velocity field and the previous front location. A technique based on volume of fluids is used to track the resin front using a so-called “saturation coefficient” of the elements (Trochu *et al*., 1993). This “fill factor” is defined by the scalar field *S*(*x*,*t*) on Ω,. For completely filled elements *S* is equal to one, while S is zero in empty elements. For the elements crossed by the front, *S(x,t)* lies between these two values. The fill factor is transported in the partially and fully saturated regions until it finally reaches the value of full saturation (*S*=1) everywhere in the domain.

4. Mesh refinement algorithm

Mesh generation consists of decomposing a geometrical domain into a partition of simple elements such as triangles, rectangles, tetrahedrons, bricks or prisms. Depending on the nature of the elements and the way they are interconnected, the resulting mesh can be either structured or unstructured. In RTM flow simulation, due to the complex mold shapes, unstructured meshes of triangles or tetrahedrons are often needed. However, the construction of 3D unstructured meshes remains a challenging task. Delaunay triangulation (Delaunay, 1934) is usually the most wanted triangulation since it maximizes the minimum angles of the constructed triangles (Green *et al*., 1978). For the triangulation to be a true Delaunay, the circumcircle of each triangle must not contain any point of the triangulation, which is usually called the “empty circle” property (Figure 4).

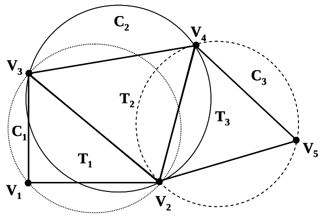

Figure 4. *The Delaunay criterion*

From an initial Delaunay triangulation, that may be obtained using appropriate mesh generation algorithms (Lee *et al.*, 1980, Guibas *et al.*, 1985, Su *et al.*, 1995), refinement techniques (Chew 1989, Chew 1993, Ruppert 1995) are used to finalize the mesh under the empty circle property (Lawson 1977, Bowyer 1981, Watson 1981). More interestingly, a close relationship exists between Delaunay triangulation and triangle edge flipping in 2D (Lawson 1977). Sibson (Sibson 1978) showed that a convex hull triangulation can be converted into a Delaunay triangulation by performing topological modifications based on diagonal swapping.

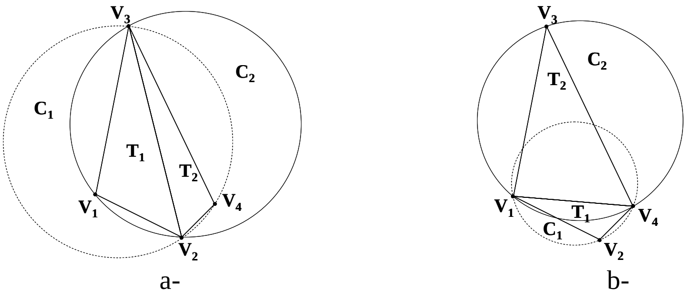

Figure 5. *Example of edge flipping: (a) Delaunay criterion is not verified (b) Delaunay criterion is verified after edge flipping*

Figure 5 shows a case where edge flipping can be applied to ensure Delaunay conformity. Before edge flipping in Figure 5a, all vertices of triangles T$_1$ and T$_2$ lie inside circles C$_1$ and C$_2$. The edge connecting vertices V$_2$ and V$_3$ is removed and replaced in Figure 5b by an edge connecting vertices V$_1$ and V$_4$. The resulting mesh verifies Delaunay criterion since only the vertices of triangle T$_i$ lies inside the circumscribing circle C$_i$.

Based on these efforts, Bechet (Bechet *et al.*, 2002) provided an implementation to refine 3D surface meshes. The algorithm has been successfully extended to handle anisotropic mesh refinement. Local metrics have been introduced to provide more resolution by concentrating the fine scale elements in some specific regions thus minimizing the global mesh size. The adaptive remeshing strategy (Bechet *et al.*, 2003) used in the present work is based on node insertion through edge bisection and local mesh improvement *via* edge flipping combined with anisotropic or isotropic metrics. The main contribution in (Bechet *et al.*, 2003) is to adapt the anisotropic remeshing technique to a moving boundary problem. This technique allows stretching the elements in any desired direction or refining the mesh in a specific region. Anisotropic line metrics can therefore be implemented to create channels in the geometry to account for edge effects (also called "race tracking").

Mesh anisotropy at any point *X* in a triangulated domain *Ω* can be achieved by considering a metric tensor that defines the stretching extent of the elements. In 2D, the tensor is represented by a *d* x *d* positive definite symmetric matrix (*d* = dimension of *Ω*) denoted by:

$$M(X) = \begin{bmatrix} a(X) & b(X) \\ b(X) & c(X) \end{bmatrix} \qquad [14]$$

with $a(X) > 0$, $c(X) > 0$ and $a(X)c(X) - b(X)^2 > 0$. The distance between two points *A* and *B* belonging to *Ω* may be defined as:

$$dist(AB) = \int_0^1 \sqrt{\left(\frac{\partial s(t)}{\partial t}\right) \cdot M(s(t)) \cdot \left(\frac{\partial s(t)}{\partial t}\right)} \cdot dt \qquad [15]$$

where $s(t)$ is the parametric representation of a path connecting *A* and *B* for parameter *t* varying from 0 to 1 (Borouchaki *et al.*, 1997). In an Euclidean space of origin *O*, the path from *A* to *B* is a line segment and $s(t)$ can be expressed as $s(t) = O\vec{A} + t\, A\vec{B}$, so the distance becomes:

$$dist(AB) = \int_0^1 \sqrt{A\vec{B}^T \cdot M(s(t)) \cdot A\vec{B}} \cdot dt \qquad [16]$$

In the case of isotropic remeshing, the metric is given by

$$M = \alpha\, I \qquad [17]$$

where I represents the identity matrix and α is a positive number. Borouchaki (Borouchaki *et al.*, 1997) have related the size of the elements h with the coefficient α by the relation:

$$\alpha = 1/h^2 \qquad [18]$$

In the anisotropic case, the elements are stretched in a given direction. In a base associated to the element, the metric diagonal components are related respectively to the normal and tangential dimensions of the element with respect to the stretching direction.

The remeshing algorithm can be summarized as follows:

1. Classify all triangle edges according to their length, which is estimated using the predefined metric;

2. Create a vertex at the middle of the largest edge (bisection);

3. Insert the vertex in the current topology, which results in a star shaped structure;

4. Check Delaunay conformity for each surrounding triangle. Swap triangle edges if necessary;

5. Resort edges after bisection;

6. Verify if refinement criterion has been reached. If not go back to step 2, otherwise, terminate refinement algorithm.

5. Examples of metrics

The flexibility of the remeshing technique and its ability to concentrate or stretch elements wherever desired is demonstrated through several examples. The key step in achieving any mesh refinement lies in the definition of the appropriate metric that reevaluates the distances between triangle vertices.

5.1. *Isotropic refinement in a square*

The first example as depicted in Figure 6 shows a square meshed with a progressive refinement from the edges to the center. The corresponding metric can be formulated as follows:

$$M \equiv \begin{pmatrix} 1/s^2 & 0 \\ 0 & 1/s^2 \end{pmatrix} \qquad [19]$$

with parameter s given by

$$s = n\, s_{\min} - (L - r) / \lambda \qquad [20]$$

where

n is the ratio between the largest and the smallest edge in the final mesh,

$s_{\min}$ is the length of the smallest edge of the mesh (refinement criterion),

L is a characteristic length of the domain (half of the square diagonal),

r is the distance from the center of the square to a vertex of the mesh

λ is a constant defining the linear progression of the element size.

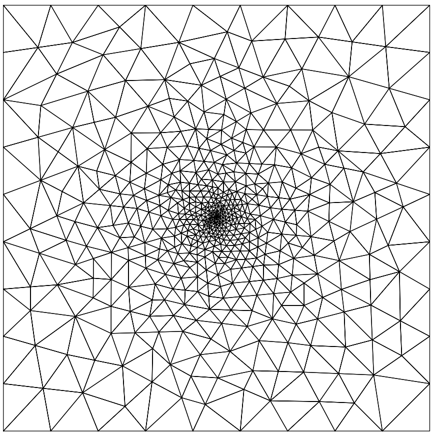

Figure 6. *Pointwise refinement in a square*

5.2. *Isotropic refinement in a rectangle*

The parameter s can be changed to follow any analytical function defined over the domain to be meshed. To achieve an elliptical mesh for example; the distance r_e of a point on the ellipse to its center is first expressed function of the eccentric angle θ by:

$$r_e^2 = \frac{a^2\, b^2}{b^2 \cos^2\theta + a^2 \sin^2\theta} \qquad [21]$$

where a and b are the major and the minor ellipse axis respectively. The ellipse major axis may be expressed as a function of the ellipse eccentricity e and the polar coordinates as

$$a = r_e \sqrt{\frac{1-e^2(\cos\theta)^2}{1-e^2}} \qquad [22]$$

The size distribution of the elements is finally described by equation [20] where the distance *r* is replaced by the above major axis *a* of the ellipse to result in the mesh shown in Figure 7.

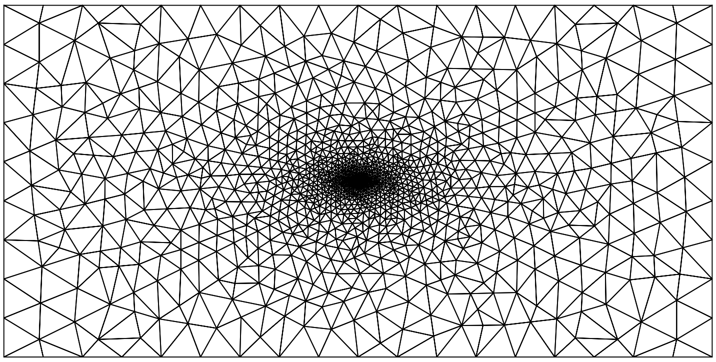

Figure 7. *Elliptic refinement in a rectangle*

5.3. *Anisotropic mesh refinement*

In order to achieve an anisotropic mesh refinement, the metric is reformulated in the following way (Borouchaki *et al.*, 1997):

$$M^* \equiv Q^T M\, Q \qquad [23]$$

where *Q* represents the desired transformation matrix from the global geometry to a local frame centered at each triangle vertex. In a similar way as in matrix [19], *M* is given in 2D by:

$$M \equiv \begin{pmatrix} 1/s_1^2 & 0 \\ 0 & 1/s_2^2 \end{pmatrix} \qquad [24]$$

where, in this case, the two parameters s_1 and s_2 describe the stretch of the element in the two orthogonal directions of a local reference frame. Figure 8a illustrates the case of a stretching direction at 45° with the horizontal.

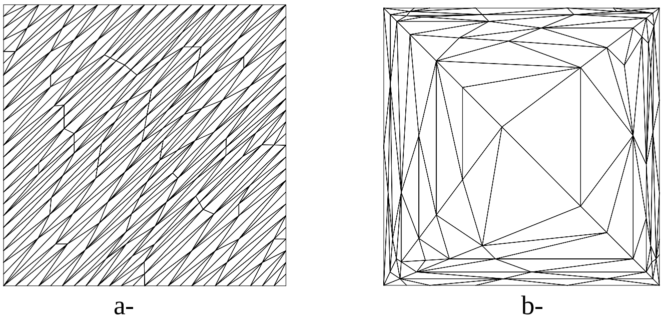

Figure 8. *Anisotropic remeshing: a- Tilted stretch of angle 45, b- Remeshing around a square*

Analogously, in order to create boundary channels, each boundary edge defines its own remeshing direction. Figure 8b shows the inclusion of a channel with two layers of triangles around a simple square, while Figure 9 illustrates the more complex case of a 3D surface of a car hood. As depicted in this figure, remeshing of the hood is performed along the boundaries without altering the initial mesh.

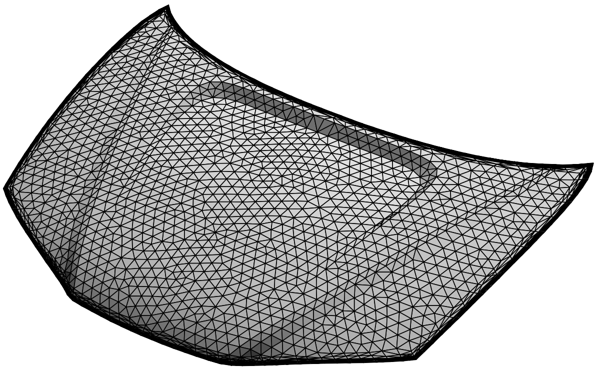

Figure 9. *Remeshing of the hood boundary*

6. Multi-layer mesh generation by extrusion

The mesh refinement technique previously described remains a difficult task in 3D. However, the concept of mesh extrusion offers a wide range of possibilities to improve the geometric modeling of 3D parts in RTM flow simulations. Moreover, composite laminates are most often made out of several stacked layers of materials with different properties, which make extrusion best suited for composite analysis especially since the in-plane and through-thickness properties of the fibrous reinforcement can be easily set for each extruded layer.

A midplane mesh usually constructed for thin shell simulations can be extrapolated in a certain number of layers *Nl* to automatically create a 3D mesh with a multi-layer structure. The extrusion starts by the extrapolation of mesh nodes. The normal vector to the discrete geometry at point *Ni* can be evaluated using the normal vectors to the adjacent elements and their aspect ratios. For a typical element $\boldsymbol{e_1}$, the normal at one of its nodes N_i is deduced from the cross product of the vectors representing the edges sharing this node (Figure 10a), namely:

$$\hat{n}_{e_1} = \frac{\vec{E}_1 \times \vec{E}_2}{\left\| \vec{E}_1 \times \vec{E}_2 \right\|} \qquad [25]$$

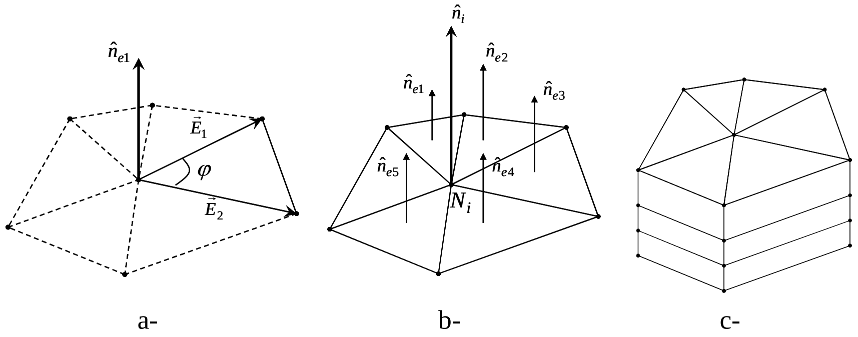

Figure 10. *Estimation of the direction of the normal vector at a node*

From the evaluation of the normal, two different weighted values are estimated in order to account for the elements aspect ratio, namely:

– a normal weighted by the angle φ_{e_1} formed by the edges used in the cross product:

$$\hat{n}_{e_1}^{\varphi} = \hat{n}_{e_1} \cdot \varphi_{e_1} \qquad [26]$$

– a normal weighted by the element area A_{e_1} and the angle φ_{e_1} :

$$\hat{n}_{e_1}^{A} = \frac{\hat{n}_{e_1} \cdot \varphi_{e_1}}{A_{e_1}} \qquad [27]$$

where the angle φ_{e1} is given by:

$$\varphi_{e1} = \arccos\left(\frac{\vec{E}_1 \cdot \vec{E}_2}{\left\|\vec{E}_1\right\| \cdot \left\|\vec{E}_2\right\|}\right) \qquad [28]$$

Denoting by N_e the number of elements sharing node N_i , the generalized angle-based and area-weighted normals at node N_i can be written as follows:

$$\hat{n}_i^{\varphi} = norm\left(\sum_{j=1}^{Nb} \hat{n}_{ej}^{\varphi}\right) = \left\|\sum_{j=1}^{Nb} \hat{n}_{ej} \cdot \varphi_{ej}\right\| \qquad [29]$$

$$\hat{n}_i^A = \sum_{j=1}^{Nb} \hat{n}_{ej}^A = \sum_{j=1}^{Nb}\left(\frac{\hat{n}_{ej} \cdot \varphi_{ej}}{A_{ej}}\right) \qquad [30]$$

Based on Pillai (Pillai *et al.*, 2000) approach that minimizes the net deviation of the projected thickness, the normal vector can be estimated using:

$$\hat{n}_i = \frac{\left(\hat{n}_i^{\varphi}\right)^2 \cdot \hat{n}_i^A}{\sum_{j=1}^{Nb}\left((\hat{n}_j \cdot \hat{n}_i^{\varphi})^2 \cdot \frac{\varphi_j}{A_j}\right)} \qquad [31]$$

Note that the extruded height may be limited by the curvature of the geometry and proper care is needed during implementation of the algorithm to set practical limitations to the extrusion height.

Performing extrusion along the estimated direction of the normal, the height of the layer is used to displace the nodes. The new node position is therefore directly given by:

$$\overline{N}_i^{new} = \overline{N}_i + \hat{n}_i \cdot layer\_height \qquad [32]$$

After the new nodes have been created, the 3D finite element must be generated following the connectivity of the 2D mesh. Note neither material properties nor boundary conditions are necessarily identical for each extrapolated layer.

7. Applications

Since edge effects are important in RTM, they will be considered in application of the proposed modeling strategy. In RTM, prior to resin injection, the fiber bed is cut and placed within the mold cavity. Due to the fibrous nature of the reinforcement, a clearance appears at the edges of the part, which causes a preferential channel flow, *i.e.*, a zone where the resin flow is much faster than in the rest of the cavity (Hammami *et al.*, 1998). This gap is actually a zone that exhibits different characteristics than the porous medium represented by the fibrous reinforcement. In order to model this type of effect, the channel along the boundaries has to be identified.

Two applications are presented in this section:

– A 2D example illustrates the capability of the remeshing algorithm to create channels along the edges of the part and their effect on the total filling time.

– A 3D example demonstrates the coupling of mesh refinement with the extrusion algorithm to generate a 3D mesh of the automotive hood. Filling of the part will also be displayed.

7.1. *2D example*

An analytical approach (Hammami *et al.*, 1998) considers the gap along the part boundaries as a porous medium. An equivalent permeability for the channel is estimated to allow application of Darcy's law in the whole domain. This equivalent averaged permeability has been correlated to the thickness *d* of the gap by the formula:

$$k_{c,ave} = d^2 / 12 \qquad [33]$$

which gives the equivalent permeability in a porous medium of an infinite planar crack of thickness *d*.

A test simulation was performed in Figure 11 for a rectangular plate with process parameters taken from Hammami (Hammami *et al.*, 1998). The clearance between the fabric and the top mold edge was set to 3mm. The isovalues representing the filling time in seconds are displayed in Figure 11 and show how the gap drastically affects the flow pattern. With a uniform resin velocity profile along the line gate on the left side of the rectangular mold, a regular filling with a straight flow front would have been expected for a fabric matching perfectly the mold boundaries. This is not the case as the resin races along the upper edge of the cavity. Edge effects reduce considerably filling times.

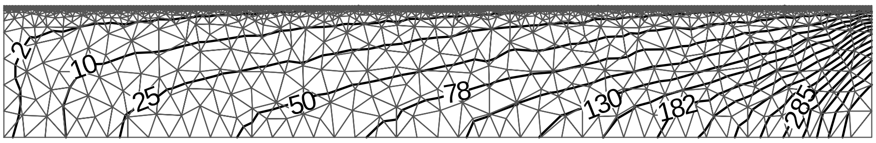

Figure 11. *Filling of a rectangular part with edge effect on the top side*

7.2. *3D example*

A more complex application is considered in order to demonstrate the remeshing and extrusion capabilities of the present technique. The case of the automotive hood with the initial mesh of Figure 12 is considered. A sandwich structure is generated consisting of a central foam layer confined between two fabrics of different permeability. Edge effects are also considered and the mesh boundaries are consequently modified by the remeshing algorithm.

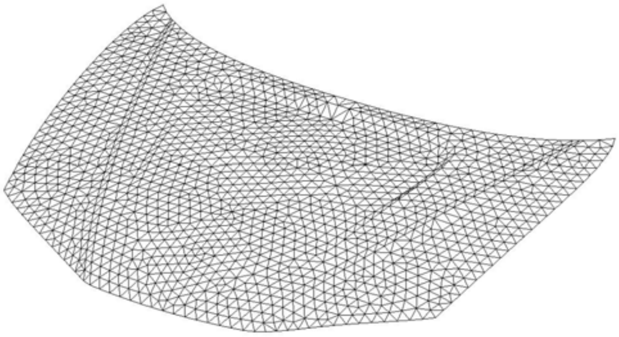

Figure 12. *Initial mesh of the automotive hood*

As shown in Figure 13, the remeshing algorithm stretches the elements of the initial mesh in order to delimitate the central zone 1 where the top and bottom fabrics are separated by the impermeable foam. The same technique is applied to generate zone 3 for the edge effects to represent a clearance between the fibrous reinforcement and the mold boundaries. An injection port is also generated at the center of the part. Finally the mesh is extruded to generate the three desired layers (Figure 14). The characteristics of each zone are described in Table 1 (only permeability is presented in SI units, the permeability of zone 3 is estimated from equation [33]).

Figure 15 shows the flow front at different times during mold filling stage. At 60 sec (Figure 15a), the resin flow is faster on the bottom layer because of its higher permeability. The top and bottom fabrics are separated by the impermeable foam in zone 1. As the resin penetrates into zone 2 in the bottom side, it flows through the stack and reaches the top side of the part (see arrow in Figure 15a). This continues as the resin crosses completely into zone 2 as depicted in Figure 15b. When the resin gets to zone 3 the flow is much faster due to the edge effects. As seen in Figure 15c, almost all the edges of the part are already filled with resin. Finally, Figure 15d shows the flow front near the end of mold filling.

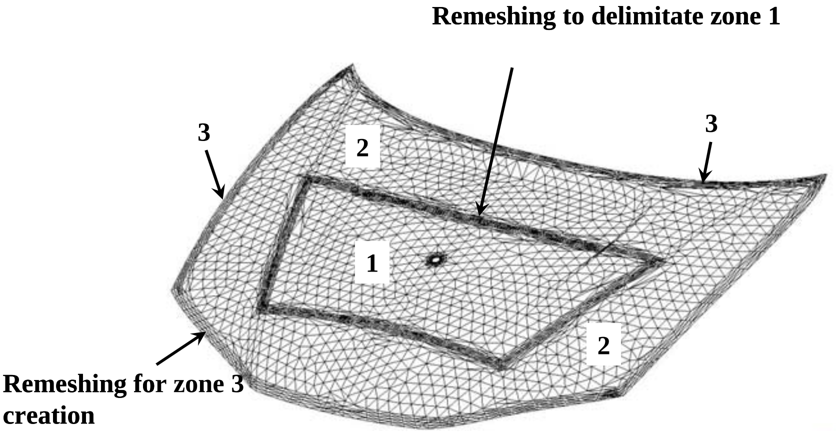

Figure 13. *Remeshed automotive hood*

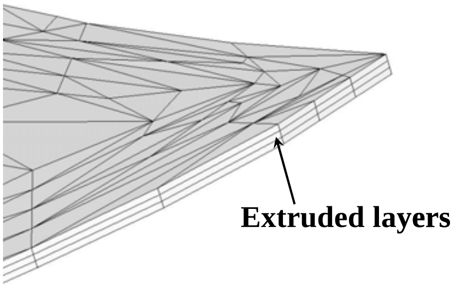

Figure 14. *Zoom of the right top corner of the hood showing extruded layers*

Table 1. *Layers permeability for the different zones*

zone	Top layer	Mid layer	Bottom layer
1	10^{-10}	impermeable	10^{-9}
2	10^{-10}	10^{-10}	10^{-9}
3	10^{-7}	10^{-7}	10^{-7}

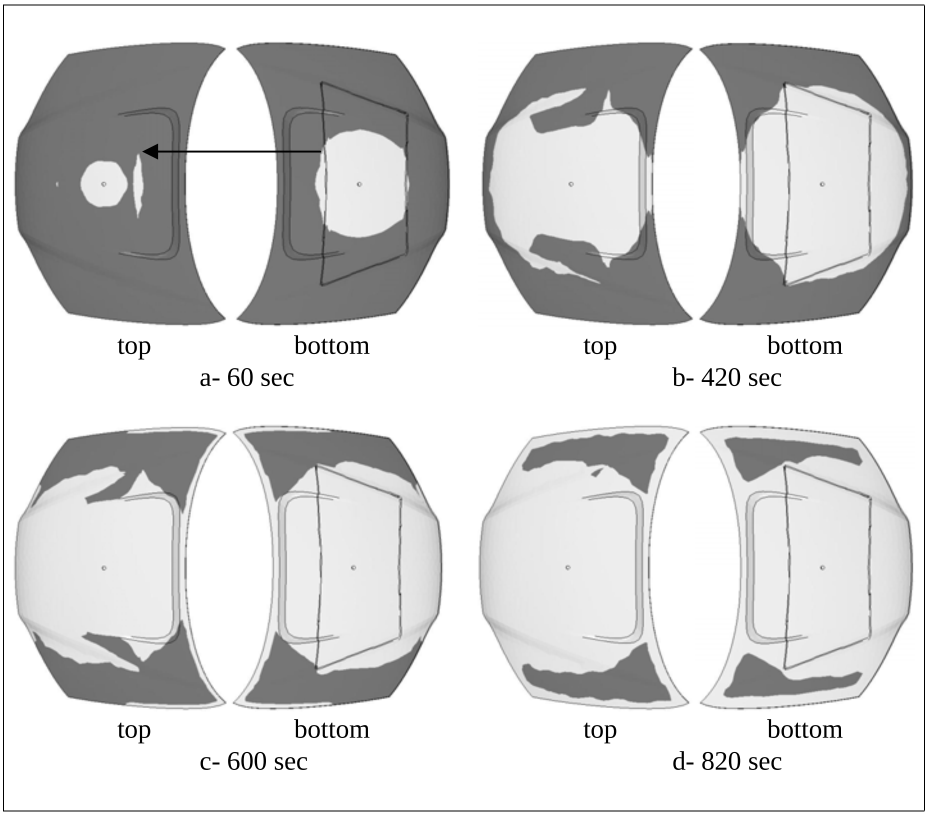

Figure 15. *Flow front at a- 60 sec, b- 420 sec, c- 600 sec and d- 820sec*

8. Conclusion

A 2D remeshing technique has been coupled to a mesh extrusion algorithm to generate appropriate computational domains for the numerical simulation of the RTM process. A remeshing technique was presented based on edge bisection and swapping with different metric examples. The algorithm allowed local refinement of finite element meshes without altering part geometry. A desired level of anisotropy can easily be achieved with proper definition of remeshing metrics. Coupling with the mesh extrusion algorithm allows defining accurately a multi-layer composite laminate structure while ensuring a minimum size of the final 3D mesh. A non-conforming finite element formulation was developed to solve the resin flow in the fibrous reinforcement using triangles and prismatic elements. This approach allows local conservation of the fluid mass on the mesh. The filling of an automotive hood with the inclusion of edge effect was presented as example of application.

Acknowledgements

The authors would like to thank ESI-Group and National Scientific Research Council of Canada (NSERC) for supporting this research.

9. References

Bechet E., Cuillere J.C., Trochu F., “Generation of Finite Element Mesh From Stereolithography (STL) files”, *Computer-Aided Design*, Vol. 34, No. 1, 2002, pp. 1-17.

Béchet E., Ruiz E., Trochu F., Cuillere J.-C., “Adaptive Mesh Generation For Mould Filling Problems In Resin Transfer Moulding”, *Composites: Part A*, Vol. 34**,** 2003, pp. 813-834.

Borouchaki H., George P.L., *Triangulation de Delaunay et Maillage*, Paris, Editions Hermès, 1997.

Borouchaki H., George P. L., Hecht F., Laug P., Saltel E., “Delaunay Mesh Generation Governed By Metric Specifications, Part I. Algorithms”, *Finite Elements in Analysis and Design*, Vol. 25, 1997, pp. 61-83.

Bowyer A., “Computing Dirichlet Tessellations”, *Computer Journal*, Vol. 24, No. 2, 1981, pp. 162-166.

Bréard J., Saouab A., Bouquet G., “Numerical Simulation of Void Formation in LCM”, *Composites Part A*, Vol. 34, 2003, pp. 517-523.

Bruschke M. V., Advani S. G., “A Finite-Element Control Volume Approach To Mold Filling In Anisotropic Porous-Media”, *Polymer Composites*, Vol. 11, No. 6, 1990.

Chew L. P., Guaranteed Quality Triangular Meshes, Technical Report TR-89-983, Department of Computer Science, Cornell University, 1989.

Chew L. P., “Guaranteed Quality Mesh Generation For Curved Surfaces”, *Proceedings of the Ninth Annual Symposium on Computational Geometry*, San Diego, California, 1993, Association for Computer Machinery, pp. 274-280.

Darcy H., “Les Fontaines Publiques De La Ville De Dijon: Distribution d’Eau Et Filtrage Des Eaux”, *APPENDICE – Note D.* Victor Dalmont, Paris, 1856.

Delaunay B., “Sur la Sphere Vide”, *Bul. Acad. Sci. USSR, Class. Sci. Nat.*, 1934, pp. 793-800.

Dullien F. A. L., *Porous Media – Fluid Transport And Pore Structure*, ACADEMIC PRESS INC. New York, 1979.

Fracchia C.A., Castro J., Tucker III C. L., “A Finite Element/Control Volume Simulation of Resin Transfer Mold Filling”, *Proceedings of the American Society For Composites, Fourth Technical Conference*, 1989, Technomic Publishing Co., INC. Lancaster, PA, pp. 157-166.

Green P., Sibson R., “Computing Dirichlet Tesselations In The Plane”, *Computer Journal*, Vol. 21, 1978, pp. 168-173.

Guibas L.J., Stolfi J., Primitives For The Manipulation Of General Subdivisions And The Computation Of Voromoi Diagrams, *ACM Transactions on Graphics*, Vol. 4, No. 2, 1985, pp. 74-123.

Hammami A., Gauvin R., Trochu F., Touret O., Ferland P., "Analysis of The Edge Effect On Flow Patterns In Liquid Composite Molding", *Applied Composite Materials*, Vol. 5, 1998, pp. 161-173.

Lawson C. L., "Software for C^1 Surface Interpolation", *Mathematical Software III*, John R. Rice, editor, Academic Press, New York., 1977, pp. 161-194.

Lee Der-T., Schachter B. J., "Two Algorithms For Constructing A Delaunay Triangulation", *International Journal of Computer and Information Sciences*, Vol. 9, No. 3, 1980, pp. 219-242.

Parnas R. S., *Liquid Composite Molding*, Hanser Gardner Publications, 2000.

Pillai K. M., Tucker III C. L., Phelan Jr. F. R., "Numerical simulation of injection/compression liquid composite molding, Part 1. Mesh generation", *Composites: Part A*, Vol. 31, 2000, pp. 87-94.

Remacle J.-F., Bréard J., Trochu F., "A Natural Way to Simulate Flow Driven Injections in Liquid Composite Molding", *Computer Methods in Composite Materials CADCOMP 98*, Vol VI., 1998, pp. 97-107.

Ruiz E., De la caractérisation des matériaux et simulation du procédé à l'optimisation de la fabrication des composites par injection sur renfort, Ph.D. thesis, Ecole Polytechnique de Montréal, Département de Génie Mécanique, 2004.

Ruiz E., Achim V., Soukane S., Trochu F., Bréard J., "Optimization of Injection Flow Rate to Minimize Micro/Macro Voids Formation in Resin Transfer Molded Composites", Submitted to *Composites Science and Technology*, August 2004.

Ruiz E., Trochu F., "Coupled Non-conforming Finite Element and Finite Difference Approximation Based on Laminate Extrapolation to Simulate Liquid Composite Molding Processes, Part I: Non-Isothermal Flow", Submitted to *International Journal for Numerical Methods in Engineering*, November 2003.

Ruppert J., "A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation", *Journal of Algorithms*, Vol. 18, 1995, pp. 548-585.

Sibson R., "Locally Equiangular Triangulations", *Computer Journal*, Vol. 21, 1978, pp. 243-245.

Su P., Drysdale R. L., "A Comparison Of Sequential Delaunay Triangulation Algorithms", *Proceedings of the Eleventh Annual Symposium on Computational Geometry (Vancouver, British Columbia, Canada)*, June, Association for Computing Machinery, 1995, pp. 61-70.

Trochu F., Gauvin R., Gao D.-M., "Numerical Analysis Of The Resin Transfer Molding Process By The Finite Element Method", *Advances in Polymer Technology*, Vol. 12, No. 4, 1993, pp. 329-342.

Watson D. F., "Computing The n-Dimensional Delaunay Tessellation With Application To Voronoi Polytopes", *Computer Journal*, Vol. 24, No. 2, 1981, pp. 167-172.