Numerical simulation of flow through porous media. A control volume approach in combination with higher order finite elements.

G. TSAMASPHYROS¹ and C. VRETTOS²
Department of Mechanics, School of Applied Mathematics and Physics
National Technical University of Athens
9 Iroon Polytechniou St., Zografou Campus 15 773
GREECE

tsamaph@central.ntua.gr¹, christovrettos@yahoo.com²

Abstract: This work deals with the numerical simulation of the well known resin transfer molding process which is implemented in the composites manufacturing industry. A numerical scheme based on the traditional finite element / control volume technique is adopted. Emphasis has been given on the use of higher order isoparametric h-elements. Numerical implementation has shown that quadratic triangular elements provide a satisfactory level of accuracy in terms of filling time prediction when compared to linear elements. The key of success behind the whole idea is that the mid-side nodes are only used for the calculation of the pressure field but no control volumes are formed around them. The gain from this formulation is that the number of iterations required to fill the mould cavity is significantly lower than the number of iterations that would be required from a linear element mesh with the same number of nodes, thus requiring a much lower CPU time.

Key-Words: porous media, control volume, finite elements, higher order

1. Introduction
The resin transfer molding process is a relatively old method for the manufacturing of composite parts of complex geometries. The use of the method for the manufacture of advanced polymer matrix composites can be traced back in the mid 1970s [1]. Since then, a lot of progress has been made in the technology of the method and by now the number of applications is numerous. The basic principle of the method is that the reinforcing fibers are positioned dry in a closed or semi-closed mould and resin, in liquid form, is injected under pressure into the mould to impregnate all the fibers (filling phase). Then the part is inserted into an environment of elevated temperature and is left to cure (curing phase). It is evident that the number of parameters that affect the success of such a process is numerous and research is focused on the numerical simulation of all the aspects of the process.

The filling phase of the resin transfer molding process is a typical problem of flow through porous media. In order to solve this type of problem, the continuity equation is combined with Darcy’s law. Then the Galerkin method is applied to the resulting equation, which leads to the finite element formulation of the problem. The scheme that is actually used is the combined finite element / control volume method. In this technique, the finite element method is used in every filling (or time) step to solve for the pressure field and the nodal flow rates. The control volumes, which are formed at every node, are then adopted in order to estimate the new position of the flow front. Nowadays, this is considered a traditional method and many numerical codes have been based on this technique [2, 5].

The filling phase of the rtm process is also a moving boundary problem which has been dealt either by having a fixed mesh [2, 5] or by re-meshing the impregnated (wet) part of the domain at every filling step. In both cases, though, linear triangular, quadrilateral or brick elements are used with nodes situated at the vertexes of the elements. Numerical accuracy is increased with increased mesh density but with the cost of increased computational time.

In the present study, a fixed mesh is used to model the mould cavity, filled with higher order h-elements of triangular shape. By the term higher order, it is meant that the elements have nodes at the mid-sides. The key of success behind the whole idea is that the mid-side nodes are only used for the calculation of the pressure field but no control volumes are formed around them. The control volumes, as with traditional linear elements, are formed around the vertex nodes only.
Numerical implementation and comparison with analytic solutions show that for the same element density or node density, the higher order elements predict the filling time with similar or even better accuracy, in some cases, than the linear elements.

2. Governing equations
The resin transfer molding process is a typical problem of incompressible flow of a liquid through a porous medium. For such a flow the mass balance at a point within the domain is expressed by the equation of continuity (1):

$$\phi \frac{\partial s}{\partial t} = -\nabla \cdot \vec{u}$$

(1)

where $0 < \phi < 1$ is the porosity of the pre-form, $0 \leq s \leq 1$ is the saturation level, $\vec{u}$ is the velocity vector of the liquid and $t$ is the time. Yet, if we are examining one single step of the filling phase, the process can be considered as quasi steady state and the transient term on the left of equation (1) can be omitted. Thus, for the saturated media, equation of continuity (1) is reduced to:

$$\nabla \cdot \vec{u} = 0$$

(2)

For liquid flow through a porous medium, which is an incompressible flow of very low Reynolds number, the pressure / velocity relationship, at every point of the domain, is sufficiently expressed by Darcy’s law (3):

$$\vec{u} = -\frac{k}{\mu} \nabla p$$

(3)

where $k$ is the permeability tensor of the pre-form, $\mu$ is the dynamic viscosity of the liquid resin and $p$ is the pressure.

By introducing equation (3) into equation (2), the pressure equation (4) is obtained:

$$\nabla \cdot \left( \frac{k}{\mu} \nabla p \right) = 0$$

(4)

The above equation is accompanied by the following boundary conditions:

a) At the injection gate(s):

$$p = p_0 \text{ for constant pressure}$$

or

$$u = u_0 \text{ for constant velocity (or flow rate)}$$

b) At the flow front:

$$p = p_{vent}$$

(5.c)

where $p_{vent}$ is the value of the pressure at the vents

c) At the mold boundary:

$$k_n \frac{\partial p}{\partial n} + k_t \frac{\partial p}{\partial t} = 0$$

(5.d)

(Here the subscripts $n,t$ denote the normal and tangential direction respectively).

The solution of equation (4) at any time (or filling) step, together with the boundary conditions (5.a-d), provides the pressure distribution over the impregnated part of the domain at the specific time step.

3. Control volume approach
In order to solve numerically equation (4), the Galerkin finite element method is implemented. For that purpose, the weak form of equation (4) is formed by multiplying equation (4) by an arbitrary weight and integrating over the volume $V^e$ of element $e$. After quite some algebra, one ends with the following relation which relates the element nodal pressures with the element nodal flow rates:

$$K^e p^e = Q^e$$

(6)

where

$$K^e = \int_{V^e} \nabla N^T \frac{k}{\mu} \nabla N dV^e$$

(7)

is the element hydraulic conductivity matrix

$p^e$ is the vector of the element nodal pressures

$$Q^e = \int_{S^e} N^T \Phi dS^e$$

(8)
is the vector of element nodal flow rates
and \( N \) is a vector containing the element’s shape functions.

By assembling the total hydraulic conductivity matrix for all the elements that belong to the main flow or to the flow front, we obtain the final linear system:

\[
Kp = Q
\]

where \( K \) is the hydraulic conductivity matrix of all the impregnated (wet) elements, \( p \) is the nodal pressure vector and \( Q \) is the vector of nodal flow rates. At this point it is worth mentioning that the nodal flow rate is zero for nodes that belong in the main flow. For inlet gates, the nodal flow rate is positive, while for nodes that belong on the flow front the flow rate is negative. Furthermore, at every time step, the sum of the nodal flow rates of the inlet gates is equal, in terms of absolute values, with the sum of the flow rates of the flow front nodes.

In the present finite element / control volume technique, the pressure equation (9) is solved at every filling step and the flow front is updated accordingly according to the algorithm presented below.

Step 0:
Read input data (element mesh, material properties, boundary conditions etc). Assemble and store the global hydraulic conductivity matrix. For isothermal analysis, this action may be performed only once, at the beginning of the calculations.

Step 1:
Form a control volume around the inlet gate(s). At this moment the filling time is zero: \( T_{\text{fill}} = 0 \)

Step 2:
Apply boundary conditions on the final linear system (9). Solve for the pressure field. Multiply the calculated pressures by the original hydraulic conductivity matrix (without boundary conditions imposed, Step 0) in order to obtain the nodal flow rates. Thus, pressure gradient calculations inside the element are not necessary for the estimation of the incoming and outgoing flux into the control volume from the surrounding control volumes.

Step 3:
For all the control volumes that belong to the flow front, find how much time they need in order to become full. The time that is required to fill the dry portion of control volume \( i \) is:

\[
\Delta t_i = \frac{Q_i}{V_i^\text{dry}} = \frac{Q_i}{V_i(1 - f_i)} \quad \text{for } i = 1...N_{\text{flow front}}
\]

where \( Q_i \) is the flow rate at control volume \( i \), \( V_i \) is the volume that corresponds to control volume \( i \) and \( f_i \) is the filling fraction of volume \( i \) at the current time and is defined below.

\[
f_i = \frac{V_i^\text{wet}}{V_i}
\]

By definition, the filling fraction is limited between 0 and 1:

\[
0 \leq f_i \leq 1
\]

where the value 0 corresponds to a dry (un-impregnated) control volume, value 1 corresponds to a fully impregnated control volume and all the other values correspond to nodes that belong to the flow front.

Step 4:
From all the flow front control volumes, select the smallest time-step:

\[
\Delta t_{\text{min}} = \min \left( \Delta t_i \right)_{i=1}^{N_{\text{flow front}}}
\]

Update the filling time:

\[
T_{\text{fill}} = T_{\text{fill}} + \Delta t_{\text{min}}
\]

Update the filling fraction of all the flow front nodes:

\[
f_i = f_i + \Delta f_i = f_i + \frac{Q_i \Delta t_{\text{min}}}{V_i}, \quad i = 1...N_{\text{flow front}}
\]

If for some control volume \( i \) the filling fraction is equal to 1, then this control volume belongs to the
main flow. By using element connectivity new control volumes can join the updated flow front.

Step 5:
If all the control volumes are fully impregnated (filling fraction equal to 1) then end the process otherwise repeat steps 2 to 4.

4. Formation of control volumes
What was presented so far is the traditional procedure for solving isothermal flow through porous media. The procedure described above has been successfully applied to 1, 2 or 3 dimensional problems always using linear h-elements. By the term linear elements, it is meant that there are nodes only at the corners (or vertieces) of the elements and the shape functions are linear along the sides. The control volumes are vertex centered which means that the control volumes are formed around every node and the number of control volumes is equal to the number of nodes (Fig. 1a).

For a mesh consisted of 3 node triangular elements, the control volume for each node \( i \) is formed by joining the mid sides of the surrounding elements with their centroids. However it must be noticed that, in the algorithm implemented herein, it is the size of the control volume that matters. The shape of the control volume is of no importance. Thus for the specific example of Fig. 1a the volume \( V_i \) of control volume \( i \) is:

\[
V_i = \frac{1}{3}V_{e1} + \frac{1}{3}V_{e2} + \frac{1}{3}V_{e3} + \frac{1}{3}V_{e4} + \frac{1}{3}V_{e5} \tag{16}
\]

In the present work the use of quadratic iso parametric h-elements has been investigated. More specifically the 6 node triangular element of Fig. 1b was investigated. The shape functions for these elements are quadratic (second order polyonyms) along the sides. The main problem with the use of these elements is the way the control volumes will be formed.

Initially, we erroneously assumed that both corner nodes as well as mid side nodes must have a control volume formed around them. Yet, when we performed numerical tests we realized that with this control volume formation, the quadratic elements gave more erroneous results than the corresponding traditional linear ones. The filling time of the mold as well as the flow front position were predicted with a significant relative error with respect to analytic and linear FE/CV solutions. After quite some trials we came to the conclusion that the mid side nodes must have no control volumes formed around them. As can be seen on Fig. 1, only the corner nodes have control volumes which are formed by joining the mid sides of the surrounding elements with their centroids. As it will be demonstrated later on, this method for forming the control volumes gives very good results, comparable with the traditional control volume formation scheme, generated by conventional linear elements.

Thus, for the 6 node triangular element mesh of Fig. 2a the correct formulation is:

**corner node** \( i \):

\[
V_i = \frac{1}{3}V_{e1} + \frac{1}{3}V_{e2} + \frac{1}{3}V_{e3} + \frac{1}{3}V_{e4} + \frac{1}{3}V_{e5} \tag{17.a}
\]

**mid side node** \( j \):

\[
V_j = 0 \tag{17.b}
\]

Equation (17.a) for the corner nodes is exactly the same with equation (16) that refers to linear elements. In reality, with the suggested method, the mid side nodes are only used for the calculation of the pressure field at every filling step but they are not taken into consideration for the estimation of the flow front shape and position. At this point, it must also be mentioned that, constant flow rate boundary conditions are applied only on vertex nodes and not on mid side nodes, because the later do not a control volume around them in order to advance the flow front. On the contrary, constant pressure boundary conditions may be applied on both vertex and mid side nodes.

Of course, one might wander what is the gain of using higher – order elements, especially if we take into
account that a larger pressure system must be solved at every filling step, in order to gain only a few percent of precision in filling time / flow front prediction.

Assume three element meshes (Fig. 2) with the following characteristics: Mesh A is consisted of \( N_v \) traditional 3 node linear triangles and has a total of \( N_v \) vertex (corner) nodes. Mesh B is exactly identical with Mesh A but it is consisted of \( N_v \) 6 node quadratic triangles with \( N_v \) vertex (corner) nodes and \( N_m \) mid side nodes. Mesh C uses traditional 3 node linear triangles and it is produced from Mesh B if we join the mid side nodes of the 6 node triangular elements. So, in this case, the mid side nodes of Mesh B have become vertex nodes and every triangular quadratic element has been “broken” into 4 triangular linear elements. Consequently, Mesh C is consisted of \( 4N_v \) 3 node triangular elements and it has \( N_v + N_m \) vertex nodes.

According to the traditional flow front advancement technique, which is also used in our case, the “wet” region of the domain is updated when at least one control volume (vertex node) is full. That means that the maximum number of iterations for mold filling (solution of pressure system, eq. 9) is equal to the number of vertex nodes. As mentioned above, in the proposed formulation the mid side nodes do not have any control volumes around them and are only used for the pressure calculation.

A Fortran code was developed to implement the algorithm presented above. Further more, a graphical interface was developed in Visual Basic for the visualization of the geometry and results. A significant amount of effort has been put and the result is a quite friendly and easy to use RTM simulation software.

5. Numerical implementation

In order to demonstrate the effectiveness of the suggested FE / CV formulation two simple mould geometries were selected. Both cases are described by analytic formulas in terms of mold filling time and flow front position. Numerical results of quadratic element meshes are compared with linear element meshes as well as with analytic solutions.

In both cases the following common properties and boundary conditions were assumed:

<table>
<thead>
<tr>
<th>Property or Boundary Condition</th>
<th>Numerical Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preform Permeability, ( k_x = k_y )</td>
<td>( 1 \times 10^{-9} m^2 )</td>
</tr>
<tr>
<td>Fiber Volume Fraction, ( V_f )</td>
<td>0.55</td>
</tr>
<tr>
<td>Preform Porosity, ( \phi = 1 - V_f )</td>
<td>0.45</td>
</tr>
<tr>
<td>Preform Thickness, ( h )</td>
<td>2 mm</td>
</tr>
<tr>
<td>Resin Dynamic Viscosity, ( \mu )</td>
<td>0.5 Pa·sec</td>
</tr>
<tr>
<td>Vent Pressure, ( P_{vent} ) (atmospheric pressure)</td>
<td>101.325 kPa</td>
</tr>
<tr>
<td>Gate Pressure, ( P_{inj} ) (if prescribed pressure is assumed)</td>
<td>200 kPa</td>
</tr>
<tr>
<td>Gate Flow Rate, ( Q_{inj} ) (if prescribed flow rate is assumed)</td>
<td>200 mm³ / sec</td>
</tr>
</tbody>
</table>

Under this view, it is obvious why even though Mesh B and Mesh C have the same number of nodes, \( N_v + N_m \), Mesh B requires a maximum of \( N_v \) iterations while Mesh C requires a maximum of \( N_v + N_m \) iterations, thus leading to a significant gain in CPU time. It should also be noticed that if the mesh or the flow is symmetric, more than 1 node may become full in one time step, which means that the maximum number of iterations may be less or significantly less than the number of vertex nodes. In the examples that follow, it will be demonstrated that meshes of type B produce results of comparable or sometimes better accuracy with meshes of type A and C.
5.1 Injection into a long strip

Resin is injected into a long narrow mould of constant thickness. The length $L$ of the mold is 500mm and the width $W$ is 50mm. The injection port is situated on one side of the mould while the vent port is situated on the other side of the mold as seen in Fig. 4. Two cases were examined. The first case involves injection under constant pressure while the second case involves injection under constant volumetric flow. Analytic formulas exist for both scenarios and are presented in Table 2, below.

<table>
<thead>
<tr>
<th>Table 2, Analytic formulas for long strip injection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant flow rate at gate ($Q_{inj} = \text{const}$)</td>
</tr>
<tr>
<td>Constant pressure at gate ($P_{inj} = \text{const}$)</td>
</tr>
</tbody>
</table>

In order to investigate mesh convergence two basic meshes, named 1 and 2 were examined, the second one being denser than the first. Meshes of types A, B and C (Fig. 2) were derived from these basic meshes as seen in Fig. 4. Furthermore, the red lines in Fig. 4 represent 1D linear, 2 node runner elements and are used in order to spread the injected resin along the width of the mould and thus provide a more uniform flow front at the beginning of the injection.

Another aspect which is worth mentioning is the modeling of the injection gates. The traditional method of modeling an injection gate is applying a prescribed value of pressure or flow rate at a specific node. Yet, in the recent past, it has been demonstrated [6, 7] that this practice constitutes a poor modeling technique and results in low computational accuracy as it imparts a mathematical singularity. Especially, when prescribed pressure is imposed, the prediction of filling time is much depended on mesh size and shape. In order to overcome these difficulties, other techniques have been suggested and are presented in Fig. 3. Fig. 3b shows that one or more elements can be used to model the injection gate. Even though, this method may not always represent the exact geometry of the gate, it smoothens the mathematical singularity and improves the solution. Fig. 3c represents the correct method for modeling the injection gate but it requires special refinement around the gate. Further to the previous methods, special gate elements have been developed that use logarithmic shape functions to represent the pressure distribution across the element [7]. In the current study, the injection gate was modeled with a single node as in Fig. 3a as well as with multiple nodes as in Fig. 3b where one triangular element was used to model the injection gate (prescribed pressure or flow rate was applied at 3 or 6 nodes depending on the element type.

Table 3 presents the numerical characteristics of the element meshes of Fig. 4, used for the long strip mold simulation.

<table>
<thead>
<tr>
<th>Table 3, Mesh characteristics for long strip test case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh Type</td>
</tr>
<tr>
<td>Mesh 1A</td>
</tr>
<tr>
<td>Mesh 1B</td>
</tr>
<tr>
<td>Mesh 1C</td>
</tr>
<tr>
<td>Mesh 2A</td>
</tr>
<tr>
<td>Mesh 2B</td>
</tr>
<tr>
<td>Mesh 2C</td>
</tr>
</tbody>
</table>

Table 4 presents the results for the filling time prediction. At this point it must be noticed that the total volume of the resin injected into the mold is equal to the volume of the preform plus the volume of the edge runners (red lines in Fig. 4):

\[ V_{\text{resin}} = V_{\text{preform}} + V_{\text{runners}} = 22550\text{mm}^3 \quad (18) \]

The exact solution for the filling time for the case of constant flow rate is:

\[ \begin{align*}
V_{\text{resin}} = V_{\text{preform}} + V_{\text{runners}} & = 22550\text{mm}^3 \\
\end{align*} \]
by taking a good look at the results of Table 4, three main conclusions can be drawn:

- In the case of prescribed pressure, the use of one single node as the injection gate produces poor results with the accuracy ranging between 3.5-6.5 %. The use of one element as the injection gate improves the numerical accuracy in all cases.
- The accuracy provided by the meshes of type B (quadratic elements) is of the same order of magnitude with the meshes of types A and C and in certain cases (prescribed pressure, one element used as the injection gate) even better. The great gain though, is the saving in computational effort. Mesh1B and Mesh1C have the same number of nodes but Mesh1B (quadratic elements) requires a maximum of 205 iterations while Mesh1C (linear elements) requires a maximum of 729 iterations to fill the solution domain. In other words, Mesh1B requires only 28% of the CPU time of Mesh1C. Similarly, Mesh2B requires a maximum of 427 iterations while Mesh2C requires a maximum of 1573 iterations, which means that Mesh2B requires only 27% of the CPU time of Mesh2C.

### 5.2 Injection into a circular disk

Resin is injected into a circular disk mold of constant thickness. The inner radius $r_0$ of the mold is 5mm and the outer radius $R$ is 500mm. The inlet gate was modeled as a hole, like in Fig. 3c. The inner radius represents the injection gate and all the nodes located on the perimeter of the inner radius have either prescribed pressure or prescribed flow rate depending on the case. There are also 3 vent nodes, located on the outer radius of the disk at equal distances among each other. The first case involves injection under constant pressure while the second case involves injection under constant volumetric flow. Analytic formulas exist for both scenarios and are presented in Table 5, below. Table 7 presents the results for the filling time prediction.

### Table 5, Analytic formulas for circular injection

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Gate type</th>
<th>Constant flow rate $T_f$</th>
<th>Constant pressure $T_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh 1A (linear)</td>
<td>1 node</td>
<td>0.018%</td>
<td>5.024%</td>
</tr>
<tr>
<td></td>
<td>1 element</td>
<td>2.705%</td>
<td>-1.758%</td>
</tr>
<tr>
<td>Mesh 1B (quadratic)</td>
<td>1 node</td>
<td>0.559%</td>
<td>6.508%</td>
</tr>
<tr>
<td></td>
<td>1 element</td>
<td>5.880%</td>
<td>0.193%</td>
</tr>
<tr>
<td>Mesh 1C (linear)</td>
<td>1 node</td>
<td>0.071%</td>
<td>3.799%</td>
</tr>
<tr>
<td></td>
<td>1 element</td>
<td>0.479%</td>
<td>1.189%</td>
</tr>
<tr>
<td>Mesh 2A (linear)</td>
<td>1 node</td>
<td>0.860%</td>
<td>4.210%</td>
</tr>
<tr>
<td></td>
<td>1 element</td>
<td>1.579%</td>
<td>0.856%</td>
</tr>
<tr>
<td>Mesh 2B (quadratic)</td>
<td>1 node</td>
<td>0.763%</td>
<td>4.364%</td>
</tr>
<tr>
<td></td>
<td>1 element</td>
<td>1.356%</td>
<td>0.761%</td>
</tr>
<tr>
<td>Mesh 2C (linear)</td>
<td>1 node</td>
<td>0.106%</td>
<td>4.245%</td>
</tr>
<tr>
<td></td>
<td>1 element</td>
<td>0.692%</td>
<td>1.956%</td>
</tr>
</tbody>
</table>

By taking a good look at the results of Table 4, three main conclusions can be drawn:

- In the case of prescribed flow rate, all meshes predict the filling time with an accuracy below 0.9 % when the injection gate is modeled with one single node. On the contrary, when the injection gate is modeled with one element, the produced accuracy worsens in all cases and reaches the value of 5.9 % in one case.

$$ t_{exact} = \frac{22550 mm^3}{200 mm^3/sec} = 112.75 \text{sec} $$

Fig. 4, Element Meshes for Long Strip Injection

\[ t_f = \frac{\phi \pi (R^2 - r_0^2) h}{4Q_{inj}} \]
In order to investigate mesh convergence two basic meshes, named 1 and 2 were examined, the second one being denser than the first. Meshes of types A, B and C (Fig. 2) were derived from these basic meshes as seen in Fig. 5. Table 6 presents the numerical characteristics of the element meshes of Fig. 5, used for the circular disk injection.

Table 6, Mesh characteristics for circular test case

<table>
<thead>
<tr>
<th>Mesh Type</th>
<th>Element Type</th>
<th>Element number</th>
<th>Node number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh 1A</td>
<td>Linear</td>
<td>400</td>
<td>231</td>
</tr>
<tr>
<td>Mesh 1B</td>
<td>Quadratic</td>
<td>400</td>
<td>861 (231 vertex)</td>
</tr>
<tr>
<td>Mesh 1C</td>
<td>Linear</td>
<td>1600</td>
<td>861</td>
</tr>
<tr>
<td>Mesh 2A</td>
<td>Linear</td>
<td>800</td>
<td>441</td>
</tr>
<tr>
<td>Mesh 2B</td>
<td>Quadratic</td>
<td>800</td>
<td>1681 (441 vertex)</td>
</tr>
<tr>
<td>Mesh 2C</td>
<td>Linear</td>
<td>3200</td>
<td>1681</td>
</tr>
</tbody>
</table>

The following conclusions may be drawn:

- Again, as in the case of the long strip injection, the quadratic elements produce results of similar accuracy for both prescribed pressure as well as prescribed flow rate boundary conditions. We use the term similar accuracy because it lies within the same order of magnitude. In certain cases it is better and in some others it is worse than the accuracy of the linear elements but in all cases it is satisfactory.
- Keeping in mind the previous lines, we should outline that Mesh1B (quadratic elements) requires a maximum of 231 iterations while Mesh1C (linear elements) requires a maximum of 861 iterations to fill the solution domain. In other words, Mesh1B requires only 27% of the CPU time of Mesh1C. Similarly, Mesh2B requires a maximum of 441 iterations while Mesh2C requires a maximum of 1681 iterations, which means that Mesh2B requires only 26% of the CPU time of Mesh2C.
- Finally, please notice that in all cases, the use of detailed modeling of the injection gate (Fig. 3c) produces much better results and the relative error is only 1.834 % for the prescribed pressure case in contrast to the maximum value of 6.508 % for the long strip injection case.

6. Conclusions

Within the frame of this work, an FE/CV formulation was presented for higher order (quadratic) triangular
elements. The key behind the whole idea is that the midside nodes of the elements are only used for the pressure calculation and no control volumes are formed around them. It was demonstrated that the resulting accuracy by the use of these elements is of the same order of magnitude or even better in certain cases. The main conclusion that is worth mentioning is that if a linear element mesh, having \( N_n \) nodes, is available then there exists a quadratic element mesh of \( N_n \) nodes, with fewer elements, that can produce the same level of accuracy but at the 25 – 30\% of the CPU time of the linear element mesh.

References