Flow numerical computation through Bezier shape deformation for LCM process simulation methods

N.Montés [1], F.Sánchez [1*], L.Hilario[1], A.Falcó[1]

[1]*Universidad CEU Cardenal Herrera. C/ San Bartolomé 55, 46115, Alfara del Patriarca, Valencia (Spain)*
URL: www.uch.ceu.es. e-mail: nimonsan@uch.ceu.es;sanchezf@uch.ceu.es

ABSTRACT: The flow front advance computation is commonly used in Liquid Composite Molding simulation for design optimization algorithms, quality process performance, etc. It can be used also to take corrective decisions in on-line control systems for the flow redirection during filling. In these cases, the flow front can be composed by a large number of nodes. In particular, it can be computed by thousands of nodes using a CCD as Finite Element sensor with high resolution [1]. Therefore, a methodology to reduce it in a few significant points is required to improve the computational costs of whatever algorithm that uses the flow front behavior information. In other previous research results, a new mathematical formulation of the flow front using CAGD techniques was used [1], [2]. Therefore, the flow front is formulae in a Bezier curve permitting to represent whatever flow front shape in a few representative points. In particular the common flow front shapes can be represented in just 10 points. The flow front nodes were approximated in a Bezier curve using a least square method to estimate the control points of the Bezier curve and a projection of this into itself. In CAGD fields, there are another interesting techniques that can be used for this propose. In particular, there is an interesting topic known as "Bezier shape deformation" [3]. This topic treats the deformation of a predefined Bezier curve through vectors. Hence, whatever point of the Bezier curve can be moved using the modulus and the direction of this vector. Therefore, the velocity vector field obtained solving the flow kinematics with Finite Element Method (FEM) or Natural Element Method (NEM) can be used to deform a predefined Bezier curve. This flow front parameterization permits a continuous numerical formulation of the flow front avoiding the approximation techniques used in [1], [2]. This computation was also extended to study particle tracking during filling.

Key words: FEM, NEM, RTM, LCM, flow front, Bezier curves, numerical simulation.

1 INTRODUCTION

The flow front tracking is a common tool to compute the control actions in advanced composite manufacturing during filling. When numerical simulation tools are used, such the Finite Element Method or Natural Element Method, the flow front can be composed of a large number of nodes depending on the discretization. Therefore, a methodology to reduce the nodes of the flow front to a few significant points or even to a continuous function is valuable. For this purpose, a mathematical flow front formulation to parametric curves is proposed not only to track the flow front advancement during simulation, moreover it can be applied over artificial vision online control systems used in infusion processes, where the flow front is defined by a number of pixels of the CCD camera treated as nodes [1], [2]. In both cases, continuous functions that locate the nodes of the flow front are approximated to Bezier curves. These kind of parametric curves permit to represent numerous curve shapes with a few control points and are commonly used in CAD/CAM applications.

In this work, a novel technique is applied in order to obtain the numerical computation of the flow front advance through its Bezier curve shape deformation.

In order to analyze the proposed strategy, we have studied the tracking of several fluid particles injected in a porous media simulating the filling of a RTM Mould. The flow front evolution is updated by the shape deformation imposed by the velocity field computed with a Finite Element control volume technique.

2 FLOW MODELLING AND COMPUTING WITH A FEM/CV TECHNIQUE

The resin impregnation of the fiber is described using the flow through porous media theory. The fluid flow problem is defined in a volume Ω,

$$\Omega = \Omega_f(t) \cup \Omega_e(t) \quad (1)$$

where the fluid at time *t* occupies the volume $\Omega_f(t)$.

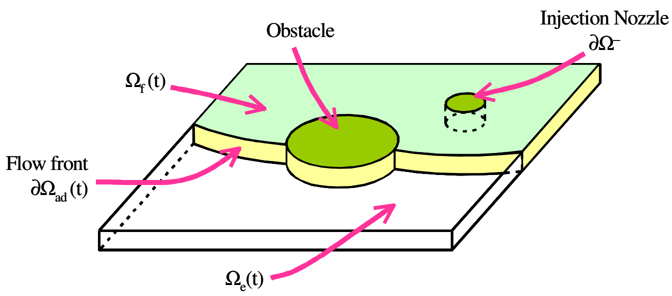

Fig.1 Two-dimensional model of a RTM Mould

Assuming constant and orthotropic preform permeability and a constant resin viscosity, then the variational formulation related to the Darcy flow results

$$\int_{\Omega_f(t)} \left(\nabla p^* \cdot \frac{\underline{\underline{K}}}{\eta} \nabla p \right) d\Omega = 0 \quad (2)$$

where $\underline{\underline{K}}$ is the preform permeability tensor, η the resin viscosity and *p* the pressure.

The location of the fluid into the whole domain Ω is defined by the characteristic function *I* defined by

$$I(\underline{x},t) = \begin{cases} 1 & \underline{x} \in \Omega_f(t) \\ 0 & \underline{x} \notin \Omega_f(t) \end{cases} \quad (3)$$

The evolution of the volume fraction, *I*, is given by the general linear advection equation:

$$\frac{dI}{dt} = \frac{\partial I}{\partial t} + \underline{v} \cdot \nabla I = 0 \quad (4)$$

with *I=1* on the inflow boundary.

The numerical resolution of the governing equations can be performed by means of a conforming finite element Galerkin technique, whereas that for the fluid domain updating a volume of fluid technique will be used. The resolution scheme is based in solving until the complete filling of the mould the three steps:

1. Obtain the pressure field using a finite elements discretisation of the variational formulation given by Eq.2, and imposing null pressure in the nodes not contained by a filled element (i.e nodes 7, 8 and 9 in Fig.2).
2. Compute the velocity field from Darcy´s law.
3. Update the element volume fraction, *I*, integrating the equation 4.

The boundary conditions are given by:

- The pressure gradient in the normal direction to the mold walls is zero, that is, the fluid cannot leave the mold cavity trough the mold walls.
- The pressure or the flow rate is prescribed on the inflow boundary (injection nozzle) $\partial\Omega^-$:
  $p(\underline{x} \in \partial\Omega^-) = P_i \quad or \quad \underline{v}(\underline{x} \in \partial\Omega^-) = \underline{v}_i$
  where $\partial\Omega^- = \{\underline{x} / \underline{v}(\underline{x}) \cdot \underline{n} < 0\}$ and $\underline{n}(\underline{x})$ is the unit outwards vector, defined on the boundary at point $\underline{x}$.
- Zero pressure on the flow front
  $p(\underline{x} \in \partial\Omega_{ad}(t)) = 0$

And if we assume that at time *t=0,* the mold is empty, the initial condition for the function *I* results:

$$I(\underline{x}, t=0) = \begin{cases} 0 & if \ \underline{x} \in \Omega \\ 1 & if \ \underline{x} \in \partial\Omega^- \end{cases}$$

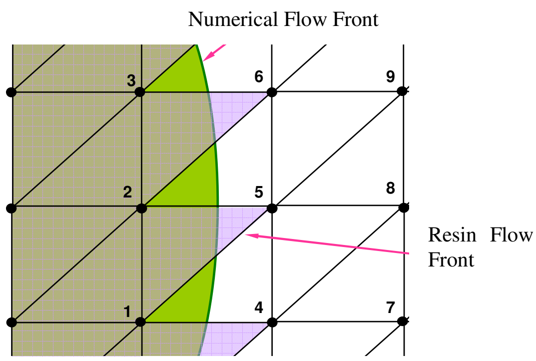

Fig.2. Fixed triangular mesh with control volumes in elements

The domain occupied by the fluid where the governing equations have to be integrated changes continuously, so it has to be defined at each time step during the simulation. The fluid domain evolution is accomplished by the resolution of the hyperbolic transport equation that governs the fluid presence function updating

3 FLOW NUMERICAL COMPUTATION THROUGH BEZIER SHAPE DEFORMATION

In the application presented in this paper, the flow front advance is defined by particle tracking. In each time instant, a few particles are introduced in the mould through the inflow boundary. The particles take the velocity vector of the finite element where are allocated and the new location for the next time instant is computed. In this case we also track the age evolution of each particle. If we assume known a particle position $\underline{x}$ at time t whose age is denoted by $E(t)$, its new age and position at time $t+\Delta t$ results $E=E+\Delta t$ and $\underline{x}=\underline{x}+\underline{v}\Delta t$, respectively.

For the mould filling analysis, we propose a parameterization of the flow front by Bezier curves defined by the particles introduced in the same time instant. The discrete flow front is translated in a continuous function that can be integrable and differentiable.

Bezier curves can be formulated as

$$\alpha_1(t)=\sum_{i=0}^{n} p_i \cdot B_{i,n}(t)=\sum_{i=0}^{n} p_i \binom{n}{i} t^i (1-t)^{n-i} \tag{5}$$

where p_i are the Bezier control points, t is the Intrinsic parameter defined in the interval $[0...1]$ and n is the Order of the Bezier curve.

The Bezier curve must be capable to modify through velocity vectors defined through the Start, S, and future, T, particle location. The initial particle distribution must be maintained for the correct flow front advancement, see Fig.3.

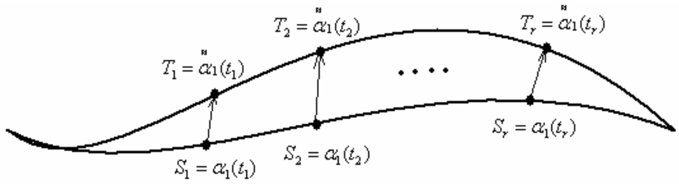

Fig. 3 Bezier Shape deformation with vectors

The new Bezier curve can be formulated as:

$$\tilde{\tilde{\alpha}}_1(t)=\sum_{i=0}^{n}(p_i+\varepsilon_i)\cdot B_{i,n}(t)=\sum_{i=0}^{n}(p_i+\varepsilon_i)\cdot\binom{n}{i} t^i(1-t)^{n-i} \tag{6}$$

The new Bezier control point allocation can be obtained through a constrained optimization method based on Lagrange multipliers [3]. Through this method, it is possible to obtain a new Bezier curve that pass through the future particles location, maintaining the initial particle distribution. In addition, this methodology also permits to establish a cost function to minimize the solution. For our goal, the displacement imposed by the previous FEM/CV resolution permits to ensure the mass conservation. Therefore, the cost function to minimize is:

$$\int_0^1\left(\alpha_1(t)-\tilde{\tilde{\alpha}}_1(t)\right)^2 dt=\int_0^1\left(\sum_{i=0}^{n}\varepsilon_i\cdot B_{i,n}(t)\right)^2 dt \tag{7}$$

The number of velocity vectors that can be taken into account in a Bezier curve displacement is $n-1$. Therefore, for a large number of particles, the order of the Bezier curve is increased implying high computational costs. To solve this problem, it is proposed the concatenation of multiple Bezier curves, m, with low order. The cost function to minimize is translated as

$$\begin{aligned}\int_0^1 & \left(\alpha_1(t)-\tilde{\tilde{\alpha}}_1(t)\right)^2+\cdots\left(\alpha_m(t)-\tilde{\tilde{\alpha}}_m(t)\right)^2 dt= \\ &=\int_0^1\left(\sum_{i=0}^{n}\varepsilon_i^1\cdot B_{i,n}(t)\right)^2+\cdots+\left(\sum_{i=0}^{n}\varepsilon_i^m\cdot B_{i,n}(t)\right)^2 dt\end{aligned} \tag{8}$$

To maintain the derivative property of the curve is imposed to the constrained optimization method the derivative restrictions in the start and end point of the resulting concatenated curve,

$$\alpha_1'(0)-\tilde{\tilde{\alpha}}_1'(0)=0\,,\quad \alpha_m'(1)-\tilde{\tilde{\alpha}}_m'(1)=0 \tag{9}$$

And in the joined points of the concatenated curves,

$$\tilde{\tilde{\alpha}}_1'(1)-\tilde{\tilde{\alpha}}_2'(0)=0,\cdots,\tilde{\tilde{\alpha}}_{m-2}'(1)-\tilde{\tilde{\alpha}}_{m-1}'(0)=0 \tag{10}$$

The Lagrange function can be defined as:

$$\begin{aligned}L=&\int_0^1\left(\sum_{i=0}^{n}\varepsilon_i^1\cdot B_{i,n}(t)\right)^2+\cdots+\left(\sum_{i=0}^{n}\varepsilon_i^m\cdot B_{i,n}(t)\right)^2 dt+ \\ &+\sum_{j=1}^{n-1}\lambda_j\cdot\left(T_j^1-\tilde{\tilde{\alpha}}_1(t_j)\right)+\sum_{k=1}^{n-1}\lambda_k\cdot\left(T_k^m-\tilde{\tilde{\alpha}}_m(t_k)\right)+ \\ &+\lambda_r\cdot\left(\alpha_1'(0)-\tilde{\tilde{\alpha}}_1'(0)\right)+\lambda_{r+1}\cdot\left(\alpha_m'(1)-\tilde{\tilde{\alpha}}_m'(1)\right)+ \\ &+\lambda_{r+2}\cdot\left(\tilde{\tilde{\alpha}}_1(1)-\tilde{\tilde{\alpha}}_2(0)\right)+\lambda_{r+3}\cdot\left(\tilde{\tilde{\alpha}}_1'(1)-\tilde{\tilde{\alpha}}_2'(0)\right)+ \\ &+\lambda_{r+2}\cdot\left(\tilde{\tilde{\alpha}}_{m-2}(1)-\tilde{\tilde{\alpha}}_{m-1}(0)\right)+\lambda_{r+2m}\cdot\left(\tilde{\tilde{\alpha}}_{m-2}'(1)-\tilde{\tilde{\alpha}}_{m-1}'(0)\right)\end{aligned} \tag{11}$$

Making $\frac{\partial L}{\partial \varepsilon_i^1},\cdots,\frac{\partial L}{\partial \varepsilon_i^m}$ we obtain a system of linear equations that can be represented in a squared matrix form, A. This matrix only depends on the order of the Bezier curve and the initial particle distribution. Therefore just only is computed at the start of the process. For instance, next figure shows an example where six Bezier of $n=3$ and one of $n=4$ are concatenated.

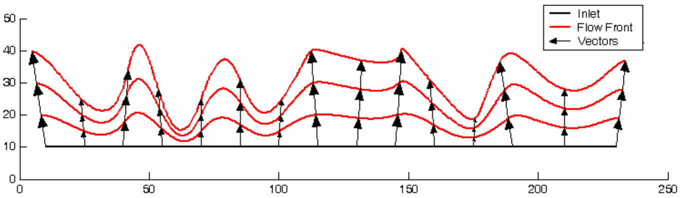

Fig. 4 Flow front advancement through multiple Bezier shape deformation

Due to A matrix is computed only at the start of the process, the flow front evolution is computed in 10 ms in a P IV 2.4 Ghz.

4 NUMERICAL EXAMPLES

In order to show the numerical results of the presented technique, a RTM mould filling with a high permeability area in the center of the piece has been simulated.

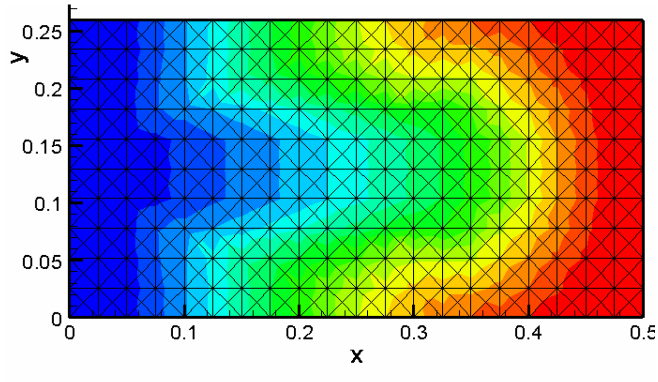

Fig.5. Flow pattern computation with FEM

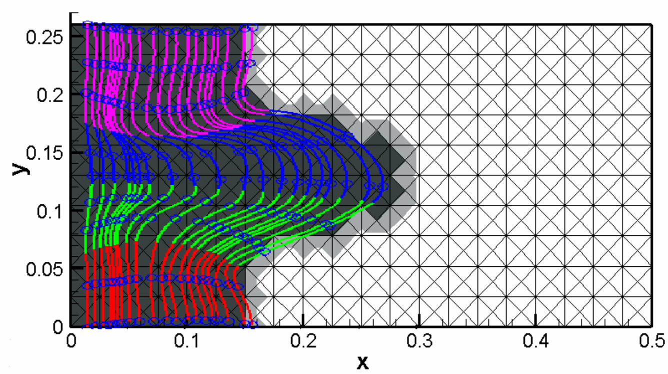

Fig.6. Flow front computation with FEM and Flow front evolution defined by a Bezier curve of 9 particles. The Bezier shape deformation is represented in different time instants.

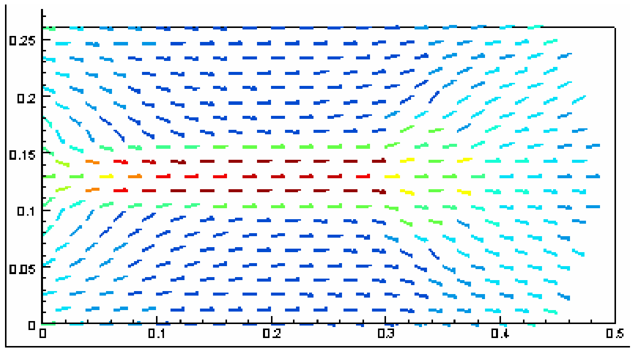

Fig.7. Velocity Field computation with FEM

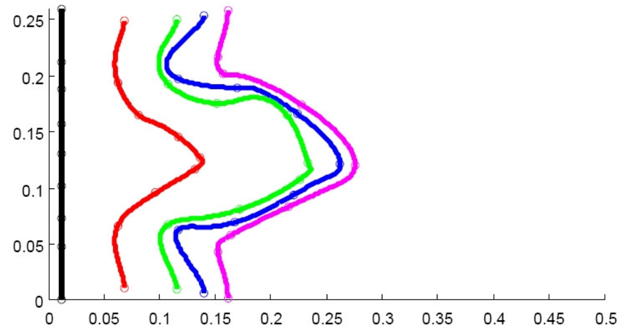

Fig.8. Particle Age evolution through Bezier shape deformation

5 CONCLUSIONS

The large number of nodes involved in the description of the flow front during simulation can be reduced drastically and optimized for computation. For this reason, it has been developed a new mathematical formulation of the flow front using CAGD techniques that allows defining its evolution analytically permitting to represent a curve that can be updated with a vector displacement field.

ACKNOWLEDGEMENTS

This research work is financially supported by Project DPI2007-66723-C02-02 from the Spanish Government and project PRUCH10/08 of the University CEU Cardenal Herrera.

REFERENCES

1. N.Montés, F.Sánchez, A.Falcó, J.Tornero, CAGD Framework and artificial vision on-line control systems of resin infusion processes, preprint
2. N.Montés, F.Sánchez, J.A.García, A.Falcó, J.Tornero, F.Chinesta, Application of Artificial Vision in flow redirection during filling of Liquid Composite Molding processes. ESAFORM, 902-907,(2007).
3. W.Qing-biao, X. Fei-hai, Shape modification of Bezier curves by constrained optimization, Journal of Zhejiang University Science, pp. 124-127, 2005J.J.