

An adaptive remeshing method in the simulation of Resin Transfer Molding (RTM) process

Whie Chang and Noboru Kikuchi

*Department of Mechanical Engineering and Applied Mechanics, The University of Michigan,
Ann Arbor, MI 48109, USA*

Received 7 December 1992

Revised manuscript received 8 January 1993

In this paper, an adaptive remeshing method for the numerical analysis of a mold filling process is presented. After a brief discussion of the derivation of governing equations using the homogenization method, the adaptive remeshing method is developed which incorporates error measures and an automatic mesh generator. The method is successfully combined with the finite element/control volume method to solve the governing equations. Two model problems are provided to examine the performance of the method.

1. Introduction

Resin transfer molding (RTM) is a relatively new process for manufacturing fiber reinforced materials. It is one of several processes that have been developed for the mass production of continuous fiber reinforced materials. In the RTM process, the mold is packed with dry fiber mats which act as reinforcements. These fiber mats are impregnated progressively by the resin injected through injection ports of the mold. After the mold is completely filled, the resin solidifies and the product is taken out of the mold.

To simulate the mold filling process, first one has to model the resin flow within a mold cavity which is packed with a reinforcing fiber mat. The flow of resin is usually slow, viscous dominant and can be modeled as Stokes' flow. But the governing equations obtained from Stokes' model is almost impossible to solve directly due to the complexity of the flow domain. The flow domain, which consists of the cavity between the reinforcing fibers, becomes much more complex when the reinforcing fibers are made of much thinner fibers. However, this complexity can be removed by introducing the homogenization method which incorporates a special averaging technique. By applying the homogenization method to the governing equation, one obtains a set of simplified governing equations consisting of a 'microscopic equation' and a 'macroscopic equation'. The microscopic equation provides the permeability of the fiber mat which characterizes the flow domain, whereas the macroscopic equation gives the pressure distribution and the averaged flow field. Although the mold filling process is a transient process, the macroscopic equation can be solved at each time step with a quasi-steady state assumption because it is a relatively slow process.

Once the simplified governing equation is obtained, the next step is the numerical implementation. The numerical implementation of the microscopic equation can be easily achieved by the conventional finite element method (FEM). In contrast, the numerical implementation of the macroscopic equation poses several difficulties. One major difficulty is keeping track of the transient free flow front. The shape and size of the flow domain within the mold cavity is constantly changing, and it is necessary to redefine the geometry of the flow domain at each time step. In the conventional FEM, this can be done by remeshing processes at each time step which are the most time-consuming part of the simulation (see [1]). In order to avoid remeshing at each time step, several numerical methods have been developed which are capable of performing simulations on a fixed grid. Examples of these methods are the fluid

Correspondence to: Whie Chang, Computational Mechanics Lab., Department of Mechanical Engineering and Applied Mechanics, University of Michigan, Ann Arbor, MI 48109, USA.

analysis network (FAN) method [2] and the finite element/control volume (FE/CV) method [3]. In these fixed grid methods, one initial grid is used from the beginning of the simulation to the end. In this paper, the FE/CV method is utilized to solve the macroscopic equation.

Although these fixed grid methods have advantages over conventional finite element methods, the use of one grid for the entire simulation can also reduce the computational efficiency; especially when the size and shape of the flow domain changes significantly throughout the simulation, which happens quite often in the mold filling process. For example, smaller meshes, which are needed for the accuracy of the solution in the initial filling stages, will require longer computing time, yet may no longer be needed at the half filled stage. Thus, at a certain point of the simulation, it is desirable that the meshes be modified to minimize the computing time while maintaining the accuracy of the solution. This can be achieved by an adaptive scheme.

Most of the adaptive schemes applied in mechanics are adaptive refinements, which simply: refine the mesh size (*h*-version); adjust the location of nodes (*r*-version); or increase the order of the interpolation function (*p*-version) in the vicinity of large error. However, in the simulation of the RTM process, an adaptive remeshing scheme is preferred which completely regenerates finite element meshes. This is because (a) in the mold filling simulation, the flow domain is constantly changing and the modification over the whole domain will lead to fewer interruptions to modify meshes; and (b) the modification includes not only refinements of the meshes but also enlargements which are not easy to do if the modification is confined to a small portion of the flow domain. We consider a complete adaptive remeshing scheme in this paper.

At the time when the concept of the adaptive method was first introduced by Sewell, Babuška and others, the adaptive remeshing method was not attractive because the method requires a reliable automatic mesh generation program which was difficult to find at that time. Thus, most of the research was conducted on the adaptive refinement method. Although the need for a complete remeshing was recognized by many researchers in many nonlinear problems, only after considerable research effort on the automatic mesh generation method did the adaptive remeshing method become popular. Among many researchers who worked on adaptive remeshing scheme, Bennet and Botkin were the first to make an application [4]. There they applied the adaptive remeshing scheme to the shape optimization of an elastic structure where the domain under consideration was varying significantly in the optimization process. This idea was then adopted by the group of Zienkiewicz in Swansea [5, 6]. An application of the adaptive remeshing method to the metal forming process can be found in [7].

An adaptive remeshing method requires, by its very nature, a very reliable and fully automatic mesh generation program that can generate finite element meshes from the data of domain boundaries and mesh densities. The first research work on such mesh generation method is Suhara and Fukuda [8] which became the foundation of many other mesh generation methods. It is also noted that implementation of an automatic mesh generation program was announced by Tezuka [9].

In this paper, we investigate the application of the method to the simulation of the mold flow problem. An adaptive remeshing method is obtained by combining the following four modules: (a) a module capable of calculating finite element approximation errors; (b) a module capable of generating sub-domains by grouping finite elements according to the approximation errors; (c) a module capable of generating finite element meshes based on the sub-domains; and (d) a module capable of transferring the nodal information to newly generated finite element meshes. The method controls the mesh sizes during the simulation so that the error indicators at each mesh are within certain error bounds. During the simulation, error indicators, which are measures of finite element approximation error, are calculated at each finite element at each time step. These error indicators are checked to determine whether or not they are in the error bound which is set in terms of a lower error bound and an upper error bound. If an error indicator deviates from the error bound, the simulation is interrupted and the adaptive routine is invoked. In the adaptive routine, a new mesh is generated and the nodal information is transferred to the newly generated mesh by linear interpolation. Then the simulation continues with the new finite element meshes.

Two examples are provided to investigate the performance of the adaptive method. In the examples, the meshes near the free flow front are kept reasonably small to provide a smooth flow front profile even when the error indicators suggest larger meshes there. As a result the error indicators along the

flow front are often smaller than the lower error bound which causes frequent remeshing. In order to resolve this problem, remeshing is not performed until the total number of elements with error indicators lower than the lower bound reaches a certain point. It is found that the adaptive remeshing method enhances computational efficiency considerably when the error bounds are set properly. When the interval between the error bounds is too small, remeshing is performed too frequently and the CPU time required for mesh generation cancels out the CPU time saved by the adaptive method. On the other hand, when the interval is too large, remeshing is not performed enough, leaving room for further improvement.

2. Modeling and the homogenization method

The flow of resin through the cavity between reinforcing fibers can be characterized as a transient flow through porous media with moving free boundaries. A fiber mat, which is the domain of a resin flow, can be considered porous media. Furthermore, it can be considered a double porous media if the reinforcing fiber is actually a fiber bundle (which is a porous media itself) made of much thinner fibers. Recently, many investigations [10–13] were reported on the flow through anisotropic fiber mats. In those investigations, Darcy's law is the most popular equation in modeling the flow through fiber mats. Darcy's law states that the resin velocity \bar{V} is proportional to the pressure gradient

$$\bar{V} = -\frac{1}{\mu} \bar{K} \nabla P, \quad (1)$$

where \bar{K} is the permeability tensor (m^2), P is the resin pressure (Pa), and μ is the resin viscosity (Pa s). However, most of these investigations were limited to a resin flow through porous media, and few studies can be found on a flow through a double porous fiber mat. Also in investigations above, experiments had to be conducted to obtain the permeability tensor. In the petroleum industry and agriculture, modeling of the flow through a double porous media has been studied extensively [14–16]. However, most of the research in these fields is not directly applicable to the mold flow of the RTM process. In this section, we briefly discuss the derivation of the governing equation for the resin flow through double porous media using the homogenization method. Details of the derivation will be presented in another paper.

The homogenization method appeared in the 1970s and has been the subject of considerable research in different areas of applied mathematics. Applications of the method to the flow and transport through porous media were first done in papers by Keller [17] and Tartar [18] in which the derivation of Darcy's law as a macroscopic equation from Stokes' equation as a micro-model was shown. Advantages of the homogenization method include: the permeability tensor can be calculated from the pore structure of fiber mats; the flow field within the pore structure can be calculated.

2.1. Modeling

The resin flow through a double porous fiber mat can be modeled as Stokes' flow (which is an inertia-less viscous flow) through a double porous medium. The inertia effect is negligible because the Reynold's number of the resin flow is small. Furthermore, the surface tension is also negligible compared to the dominant viscous force. See Fig. 1 for the configuration of a mold. To simplify the problem, we make the following assumptions:

- (1) The fiber mat in the mold cavity is rigid and is not shifted by the resin flow.
- (2) One dimension of a mold is much smaller than the other two, so that a two-dimensional model is applicable.
- (3) The resin flow is an isothermal flow.

The governing equations of the resin flow are

$$\nabla \cdot \bar{v} = 0, \quad (2)$$

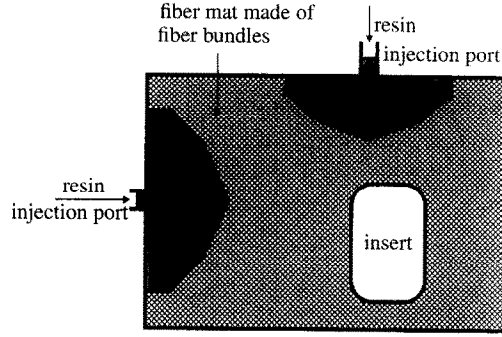


Fig. 1. Top view of a mold cavity.

$$-\nabla P + \mu \nabla \cdot (\nabla \bar{v}) + \bar{f} = 0, \quad (3)$$

$$\bar{v} = 0 \quad \text{along solid boundaries}, \quad (4)$$

where \bar{v} is the velocity, P the pressure, μ the viscosity, and \bar{f} the specific weight of the resin. Solving (2)–(4) directly is very difficult due to the complex geometry of the flow domain which has double porous microstructures. However, this difficulty can be removed by applying the homogenization method to (2)–(4).

2.2. Homogenization method

The homogenization method is applied to the governing equations noting that there are two different regimes of flow domain. As a flow domain, a double porous fiber mat has micro structures in two different scales, namely, the ‘first level microstructure’ and the ‘second level microstructure’. The first level microstructure refers to the pore structure between reinforcing fiber bundles, whereas the second level microstructure refers to the pore structure within fiber bundles. In both levels of microstructure, it is assumed that the microstructures are made of repetitions of a unit cell, which implies that they have periodic structures (see Fig. 2). It is also assumed that the scale length l of the first level microstructure is much smaller than the macroscopic scale length L , and that the scale length l' of the second level microstructure is much smaller than l .

In order to capture the details within the microstructures, two stretched scales, \bar{y} and \bar{z} , are employed. The stretched coordinate \bar{y} is used for the first level microstructure whereas the coordinate \bar{z} is used for the second level microstructure (see Fig. 3). These coordinates are defined by

$$\bar{y} = \frac{\bar{x}}{\varepsilon}, \quad \text{where } \varepsilon = \frac{l}{L}, \quad (5)$$

$$\bar{z} = \frac{\bar{y}}{\delta}, \quad \text{where } \delta = \frac{l'}{l}. \quad (6)$$

First, the homogenization method is applied to governing equations (2)–(4) in the second level microstructure which describe the flow within fiber bundles. As a result, two equations are obtained, the macroscopic equation and the microscopic equation. The macroscopic equation, which is identical to Darcy’s law, is the governing equation for the flow within fiber bundles. The microscopic equation is the equation for the permeability of Darcy’s law. Formal derivations of the macroscopic equation and the microscopic equation are omitted for simplicity (refer to [17, 18] for details). The governing equation obtained is

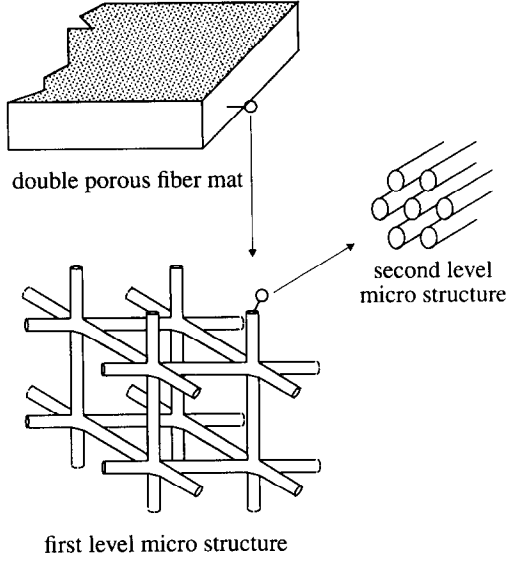


Fig. 2. Microstructures of a double porous fiber mat.

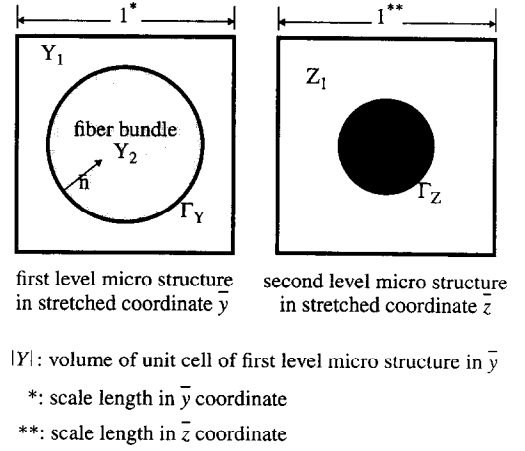


Fig. 3. The cross sections of unit cells.

$$\frac{\partial \hat{v}_i}{\partial x_i} = 0, \quad \text{where } \hat{v}_i = (\varepsilon \delta)^2 \frac{k}{\mu} \frac{\partial P}{\partial x_i}, \quad (7)$$

where \hat{v}_i is Darcy's velocity and k is the permeability which is obtained from the microscopic equation.

Now, we have two different types of flow within a mold cavity interacting with each other. One is Darcy's flow within fiber bundles and the other is Stokes' flow between fiber bundles. From now on, we apply the homogenization method to the governing equations of Stokes' flow to obtain the macroscopic equation and the microscopic equation of the mold flow. The macroscopic equation obtained will be Darcy's law with an added sink term which represents the flux absorbed into fiber bundles. Again the macroscopic equation is the governing equation for the mold flow and the microscopic equation is the equation for the permeability tensor coming from the first level microstructure of the fiber mat.

The pressure and velocity distributions of the mold flow tend to have locally periodic distribution because of the periodic first level microstructure. Thus, we assume that pressure and velocity distributions have two explicit dependencies on space variables. One is the dependency in the macroscopic level, namely \bar{x} , and the other is the dependency in the microscopic level, namely \bar{y} ($\bar{y} = \bar{x}/\varepsilon$). Furthermore we assume that the dependency on \bar{y} is Y -periodic (Y denotes the unit cell of the first level microstructure). This periodicity is represented by ε which is very small.

The objective of the homogenization method is to remove the dependency on \bar{y} by introducing an averaging technique over a unit cell of the first level microstructure. With this objective in mind, we expand the pressure distribution P^ε and velocity distribution \bar{v}^ε into the formal power series of ε (here, we drop the argument on time, t , for simplicity of notation).

$$P^\varepsilon(\bar{x}) = P^{(0)}(\bar{x}, \bar{y}) + \varepsilon P^{(1)}(\bar{x}, \bar{y}) + \varepsilon^2 P^{(2)}(\bar{x}, \bar{y}) + \dots, \quad (8)$$

$$\bar{v}^\varepsilon(\bar{x}) = \varepsilon^2 \bar{v}^{(0)}(\bar{x}, \bar{y}) + \varepsilon^3 \bar{v}^{(1)}(\bar{x}, \bar{y}) + \varepsilon^4 \bar{v}^{(2)}(\bar{x}, \bar{y}) + \dots. \quad (9)$$

As mentioned, $P^{(i)}$ and $v^{(i)}$ are periodic functions with respect to argument \bar{y} . In the asymptotic expansion above, the differential operator applied to $\phi^\varepsilon(\bar{x}, \bar{y})$ becomes

$$\frac{\partial \phi^\varepsilon(\bar{x})}{\partial x_i} = \frac{\partial \phi^\varepsilon(\bar{x}, \bar{y})}{\partial x_i} + \frac{1}{\varepsilon} \frac{\partial \phi^\varepsilon(\bar{x}, \bar{y})}{\partial y_i}. \quad (10)$$

By introducing (8)–(10) into (2)–(4) and applying the homogenization method, the microscopic equation and the macroscopic equation are obtained:

Microscopic equation:

$$\chi_i^m \in V_y: \int_{Y_1} \mu \frac{\partial \chi_i^m(\bar{y})}{\partial y_j} \frac{\partial \delta v_i^{(0)}(\bar{y})}{\partial y_j} dY = \int_{Y_1} \delta v_m^{(0)}(\bar{y}) dY \quad \forall \delta v_i^{(0)} \in V_y, \quad (11)$$

where

$$V_y = \left\{ w_i \mid w_i \in (H^1(Y_1))^3, Y\text{-periodic}, w_i = 0 \text{ on } \Gamma_Y, \frac{\partial w_i}{\partial y_i} = 0 \right\}; \quad (12)$$

Macroscopic equation:

$$\frac{\partial \tilde{v}_i}{\partial x_i} + \frac{\varepsilon^2}{|Y|} \int_{\Gamma_Y} v_i^1 n_i d\Gamma = 0, \quad (13)$$

where

$$\tilde{v}_i = \varepsilon^2 K_{ij} \left(f_j - \frac{\partial \tilde{P}}{\partial x_j} \right), \quad (14)$$

$$K_{ij} = \frac{1}{|Y|} \int_Y \chi_j^i dY, \quad (15)$$

$$\tilde{P} = \frac{1}{|Y|} \int_Y P^{(0)} dY, \quad (16)$$

$$P|_{Y_1} = P|_{Y_2} \quad \text{along } \Gamma_Y, \quad (17)$$

$$\int_{\Gamma_Y} v_i^{(1)} n_i d\Gamma = - \left(\frac{\delta}{\varepsilon} \right)^2 \int_{\Gamma_Y} k \frac{\partial P}{\partial y_i} n_i d\Gamma. \quad (18)$$

The microscopic equation (11) can be solved for χ_j^i using the conventional finite element method. χ_j^i , in turn, is used to calculate the permeability tensor, K_{ij} , by (15). With the permeability tensor at hand, the macroscopic equation (13) which is the governing equation of the mold flow is solved for the pressure distribution, \tilde{P} , and Darcy's velocity, \tilde{v}_i , of the mold flow. With the solutions of the macroscopic equation, the flow field within the microstructure can be obtained by the following equation:

$$v_i^\varepsilon \approx \varepsilon^2 v_i^{(0)}(\bar{x}, \bar{y}) = \varepsilon^2 \chi_j^i \left(f_j - \frac{\partial \tilde{P}}{\partial x_j} \right). \quad (19)$$

While solving (13), one must note that it is coupled with (7) which is the governing equation of the resin flow within the fiber bundle. This reflects the fact that there are two different regimes of flow, the flow between the fiber bundles and the flow within the fiber bundles, interacting with each other. The coupling term, which is the second term of (13), represents the flux absorbed into the fiber bundle and must be evaluated from the solution of (7). This requires that if one is solving equation (13) using a finite element approximation, first one must solve (7) at each node of the finite elements. This means that one has to keep track of the moving free boundary of the resin flow within the fiber bundle while keeping track of the moving free boundary of the mold now. In the examples, instead of solving (7) numerically, an analytical solution is found and utilized.

An analytical solution to (7) can be found when the fiber bundle has a rather simple geometry. For a fiber bundle having a circular cross section as in the examples, the pressure within the fiber bundle is found to have a logarithmic distribution. The analytical solution to (7) for this case is

$$P = \tilde{P} \left(- \frac{1}{\ln(r_i/r_0)} \ln\left(\frac{r}{r_0}\right) + 1 \right), \quad (20)$$

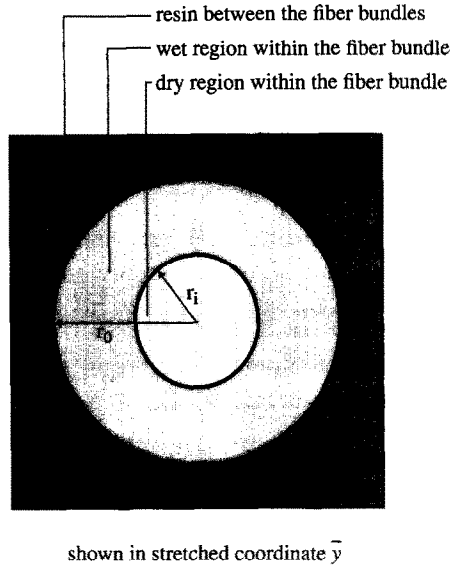


Fig. 4. Details within the fiber bundle.

where P is the pressure within the fiber bundle and \tilde{P} is the resin pressure surrounding the fiber bundle. r_i represents the location of the flow front, and r_0 represents the radius fiber bundle in stretched coordinates. This is illustrated in Fig. 4. The analytical solution (20) can be combined with (13)–(18) to yield

$$\frac{\partial}{\partial x_i} \left[\varepsilon^2 K_{ij} \left(f_j - \frac{\partial \tilde{P}}{\partial x_j} \right) \right] - \frac{2\pi\delta^2 k}{|Y|} \frac{\tilde{P}}{\ln(r_i/r_0)} = 0. \quad (21)$$

In the examples, the decoupled equation (21) is used instead of (13).

A wet ratio can be introduced to indicate the degree of impregnation within the fiber bundle. A wet ratio is defined as

$$\text{wet ratio} = \frac{\text{wet area within the fiber bundle}}{\text{total area of the fiber bundle}} = \frac{\pi r_i^2}{\pi r_0^2}. \quad (22)$$

Since the resin starts to impregnate the fiber bundle as soon as the fiber bundle is included in the flow domain, a wet ratio can be used to determine the location of moving free boundary of the mold flow. One can identify the flow domain by checking if the wet ratio is larger than zero there. In the examples, wet ratio plots are provided to show the location of the moving free boundary of the mold flow as well as the degree of impregnation within fiber bundles.

3. Numerical implementation

Most of the difficulties in solving (7), (13)–(18) come from the fact that the mold filling simulation is a transient problem. To solve (13)–(18) numerically using finite element approximation, one first has to solve (7) and keep track of the moving free boundary within the fiber bundle at each nodal point of the finite elements. This can be done by updating the location of the flow front at each time step using the time increment and Darcy's velocity obtained from the solution of (7). In the examples, an analytical solution to (7) is found and used to update the location of the moving free boundary.

Next, with the solution of (7) one has to solve (13)–(18) over an entire flow domain which is changing constantly throughout the mold filling process. The changing flow domain will require modifications of the finite element meshes at each time step of the simulation, if one is using the finite

element method to solve equations (13)–(18). Modifying the meshes according to the changing flow domain not only becomes the most time-consuming part of the simulation but also addresses several difficulties in locating the new flow front at the next time step (see [1]). To avoid these difficulties, we use the finite element/control volume (FE/CV) method. The FE/CV method is one of the fixed grid methods which use one grid throughout the entire simulation. Also in the FE/CV method, location of the new free flow front can be determined much more easily.

In the FE/CV method, the surface of a thin mold cavity is first divided into three node triangular elements. The centroids of these elements are joined to the midpoints of the three corresponding sides creating polygonal control volumes that surround each vertex node. This procedure is shown in Fig. 5. The thickness of an element is specified at the three nodal points and is linearly interpolated within each element. The nodal thicknesses are assigned separately at each element in order to accommodate possible sudden changes of the thickness between neighboring elements.

The FE/CV method is based on the ‘conservation of mass principle’ applied to each control volume. The governing equations of the mold flow, i.e. (13), can be interpreted as: ‘the net flux entering the control volume must be equal to the amount of flux absorbed into the fiber bundles within the control volume’. This interpretation is used to construct a set of linear equations which are discretized versions of (13).

The net flux entering a control volume can be obtained by integrating the flux along the boundary of the control volume. As shown in Fig. 5, the control volume assigned to node *A* is accommodated by several elements surrounding that node. Within each element, pressure distribution is interpolated using linear shape functions. Then the gapwise volumetric flow rate, which is proportional to the gradient of a pressure distribution, becomes constant within each individual element. The constant flow rate within each element is integrated along the boundaries of the control volume to yield the net flow rate into the control volume. The net flow rate within the *i*th element, entering the control volume assigned to the node *A*, can be represented as

$$q_i = \int_S \left(-K_{ij} \frac{\partial N_l}{\partial x_k} P_l \hat{n}_j t \right) ds, \quad (23)$$

where P_l is the nodal pressure, N_l is the linear shape function, \hat{n}_j is the normal vector, and t is the thickness. The net flux entering a control volume can be obtained by summing q_i 's over each element accommodating the control volume.

The total amount of flux absorbed into the fiber bundles within each control volume can be obtained by integrating the second term of (13) (or (21) if the analytical solution (20) is used) over the control volume.

The boundaries of the flow domain in the mold filling process include the mold walls, the injection

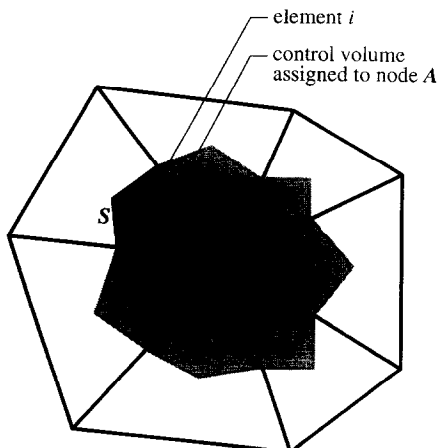


Fig. 5. Polygonal control volume assigned to node *A*.

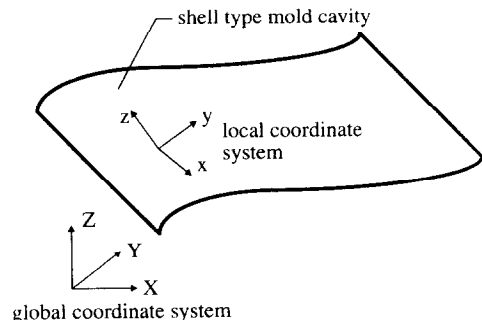


Fig. 6. Coordinate systems used in shell type mold.

ports, and the flow front. At the mold walls, the normal component of the flux is set to zero so that the flux is always parallel to the walls. At the injection ports, either pressure is specified or incoming flux is assigned according to the type of injection ports. At the flow front, a parameter f is used to represent the status of each control volume. If a control volume is empty, f is set to zero. If a control volume is filled up with resin, f is set to one. If a control volume is partially filled, f is set to the volume fraction of the resin within the control volume. The control volumes with f between one and zero are considered to be the flow front and zero pressure is assigned to the corresponding node.

A set of algebraic equations can be constructed by considering the mass conservation principle with the boundary conditions above. Since both the net flux entering a control volume and the net flux absorbed into the fiber bundles within the control volume can be represented in terms of nodal pressures P_i , the resulting equations are in terms of nodal pressures. Thus, the pressure distribution can be obtained by solving the set of linear equations. The pressure distribution is then used to determine the velocity of the moving flow front which is needed to determine the location of the moving flow front at the next time step.

The mold filling process can be regarded as a quasi-steady state process which assumes a steady state condition at each time step. At each time step, pressure distribution is calculated and the velocity of the flow front is determined by (14). With the velocity of the flow front and the time increment, the location of the flow front is updated at each time step. The selection of the time increment for each quasi-steady state is based on two considerations. One is that each time increment allow only one control volume to be completely filled and the other is that each time increment be small enough to assure the accuracy of the solution within the fiber bundles. This restriction of the time increment ensures the stability of the quasi-steady state approximation.

The current formulation can be extended to a mold flow problem with a shell type mold by introducing a proper coordinate transformation. This is demonstrated in the second example. A coordinate transform can be made from a global coordinate system to a local coordinate system whose z -axis is always normal to the surface of the shell type mold (see Fig. 6). Once the transformation is made, the shell type problem is reduced to a two-dimensional problem and all the formulations derived in the earlier sections are applicable.

4. Adaptive remeshing method

Although the FE/CV method has increased computational efficiency by removing the most time-consuming part of the simulation, i.e. generation of meshes at each time step, using one grid throughout the simulation can also cause computational inefficiency. This is because in an RTM process the size and shape of the flow domain is changing significantly and the finite element grid adequate at a certain stage of the simulation is not likely to be adequate at other stages of the simulation. Thus, it is desirable that the meshes be modified, although not very frequently, as the simulation proceeds. This can be achieved by an adaptive remeshing method which monitors the finite element approximation error of the solution and regenerates the meshes when the current meshes are found to be inadequate.

An adaptive remeshing method controls the mesh sizes during the simulation so that the finite approximation errors at each element are within a certain bound. In an adaptive remeshing method, error indicators, which are measures of finite element approximation errors, are calculated at each element and checked to determine if they are within a certain error bound (given a priori). If an error indicator deviates from the error bound, the simulation is interrupted and a remeshing routine is invoked. In the remeshing routine, a new finite element mesh is generated according to the current mesh sizes and error indicators, and all the nodal information is transferred to the new mesh by linear interpolation. Then the simulation continues with the newly generated mesh.

The adaptive remeshing method developed in this paper is obtained by combining these four modules: (a) a module capable of calculating finite element approximation errors; (b) a module capable of generating sub-domains by grouping finite elements according to the approximation errors; (c) a module capable of generating finite element meshes based on the sub-domains; and (d) a module

capable of transferring the nodal information to the newly generated finite element meshes. Each module is discussed in detail in the following.

4.1. Error indicators

In the adaptive remeshing method, the error indicators, which are measures of discretization error, are used to determine the adequateness of the finite element meshes. If error indicators show very small error in a certain region, it implies that the finite element meshes are unnecessarily small in the region, causing longer computing time. On the other hand, if they show very large error, it implies that the meshes in that region need to be refined to reduce the error.

The error indicators should be able to represent the discretization error while requiring least amount of computational effort to evaluate. This is because the error indicators are calculated at each element at each time step during the simulation. If the computational overhead needed to calculate the error indicators is too high, it may cancel out the benefits of the adaptive remeshing method. To reduce the computational overhead, the error indicator should be an a posteriori error measure, which means that the evaluation of the error indicators should be done utilizing the known information during the solution process and involve local rather than global computations. In the following, derivations of several error indicators used in the examples are presented.

It is noted that the derivation of the error indicators in the following is based on the finite element approximation error although the FE/CV method is used in the current simulation. This can be justified because the FE/CV formulation for this diffusion type problem results in discretization equations identical to those which arise from using Galerkin or variational techniques which the finite element method is based on. See [19] for further information.

The weak form of the governing equation (21) used in the examples can be represented as

$$P \in V_g: \int_{\Omega} \left(\frac{\partial \bar{P}}{\partial x_i} \varepsilon^2 K_{ij} \frac{\partial P}{\partial x_j} + \bar{P} c P \right) d\Omega = \int_{\Omega} \frac{\partial \bar{P}}{\partial x_i} K_{ij} f_j d\Omega - \int_{\Gamma} \bar{P} v_i n_i d\Gamma \quad \forall \bar{P} \in V_0, \quad (24)$$

where V_g is the set of all functions which have square integrable first derivatives and satisfy g -type boundary conditions, that is $v = g$ ($v \in V_g$) for a specified pressure g on $\partial\Omega_g \subset \partial\Omega$. V_0 is the same as V_g except that it has zero values along the g -type boundary $\partial\Omega_g$. Constant c represents the coefficients of \bar{P} in (21).

Equation (24) can be rewritten as

$$u \in V_g: a(u, w) = b(w) \quad \forall w \in V_0, \quad (25)$$

where $a(\cdot, \cdot)$ and $b(\cdot)$ are defined as

$$a(u, w) = \int_{\Omega} \left(\frac{\partial w}{\partial x_i} \varepsilon^2 K_{ij} \frac{\partial u}{\partial x_j} + w c u \right) d\Omega, \quad (26)$$

$$b(w) = \int_{\Omega} \frac{\partial w}{\partial x_i} K_{ij} f_j d\Omega - \int_{\partial\Omega} w v_i n_i d\Gamma. \quad (27)$$

Let V_{gh} and V_{0h} be finite element approximations of V_g and V_0 , respectively, such that $V_{gh} \subset V_g$ and $V_{0h} \subset V_0$. Let u_h be the solution to the discrete problem

$$u_h \in V_{gh}: a(u_h, w_h) = b(w_h) \quad \forall w_h \in V_{0h}. \quad (28)$$

Here it is assumed that the domain Ω is exactly covered by finite elements and that $a(\cdot, \cdot)$ and $b(\cdot)$ are exactly integrated in V_{gh} and V_{0h} . Then the finite element approximation error, e , can be defined as

$$e = u_h - u. \quad (29)$$

The most natural choice of the error norm would be

$$\|e\|^2 = a(e, e) = a(u_h - u, u_h - u) = \sum_i a_{\Omega_i}(u_h - u, u_h - u), \quad (30)$$

where $a_{\Omega_i}(\cdot, \cdot)$ is the same as $a(\cdot, \cdot)$ except that the integration is over an element, Ω_i . The amount of the error within each element can be represented by

$$\|e_i\| = \sqrt{a_{\Omega_i}(u_h - u, u_h - u)}. \quad (31)$$

Error indicators are estimations of the local error defined by (31). Although the evaluation of $\|e_i\|$ requires an exact solution which is often not available, there are several different estimation methods which lead to different error indicators. In the following, we discuss three different methods to estimate $\|e_i\|$.

4.1.1. The error indicator using approximation error, $\epsilon_i^{[1]}$

One way to estimate $\|e_i\|$ is to use the interpolation error. Substituting $w = w_h$ in (25) and subtracting from (28) yields the orthogonality condition of the approximation error to the finite element approximation space V_{0h} :

$$a(u_h - u, w_h) = 0 \quad \forall w_h \in V_{0h}. \quad (32)$$

Using this orthogonality condition, it is found that

$$\|e\|^2 = a(u_h - u, u_h - u) = a(u_h - u, w_h - u) \leq a(w_h - u, w_h - u) \quad \forall w_h \in V_{0h}. \quad (33)$$

If w_h in (33) is identified with the interpolation of the solution, u , the finite element approximation error is bounded by the interpolation error. This means that the interpolation error can be used to estimate the amount of error given in (30). Thus, instead of estimating the discretization error, one can estimate the interpolation error and use it as an error indicator. The error indicator, $\epsilon_i^{[1]}$, can be represented by

$$\epsilon_i^{[1]} = \|e_i^{\text{interpolation}}\| = \sqrt{a_{\Omega_i}(w_h - u, w_h - u)}. \quad (34)$$

The interpolation error in (34) can be estimated from the finite element solution, u_h . A configuration of an element used in the simulation is shown in Fig. 7. Using Taylor's series expansion, the exact solution can be represented as

$$u = u|_c + \frac{\partial u}{\partial x} \Big|_c (x - x_c) + \frac{\partial u}{\partial y} \Big|_c (y - y_c) + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} \Big|_c (x - x_c)^2 + \frac{1}{2} \frac{\partial^2 u}{\partial y^2} \Big|_c (y - y_c)^2 + \frac{\partial^2 u}{\partial x \partial y} \Big|_c (x - x_c)(y - y_c) + \text{higher order terms}, \quad (35)$$

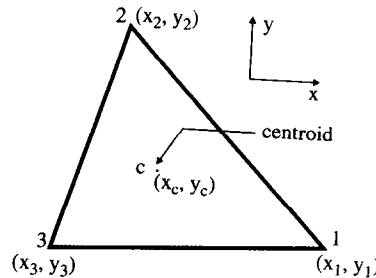


Fig. 7. The i th element used in the simulation.

whereas the interpolation of the exact solution can be approximated using u_h as

$$w_h \approx u_h|_c + \frac{\partial u_h}{\partial x}(x - x_c) + \frac{\partial u_h}{\partial y}(y - y_c). \quad (36)$$

The interpolation error can be estimated if the first and second derivatives in (35) can be obtained.

Using u_h , the first derivatives in (35) can be approximated as

$$\frac{\partial u}{\partial x}\Big|_c \approx \frac{\partial u_h}{\partial x}, \quad \frac{\partial u}{\partial y}\Big|_c \approx \frac{\partial u_h}{\partial y}. \quad (37)$$

Since the first derivatives of u_h have constant value within each element, the second derivatives in (35) cannot be obtained directly from the first derivatives of u_h . However, continuous linear versions of the first derivatives can be obtained using a weighted average scheme. Then the second derivatives can be obtained by simply differentiating them. The weighted average scheme can be used to determine the value of the first derivatives at each node. The value of a first derivative at node i can be determined by

$$\frac{\partial u}{\partial x}\Big|_{\text{node } i} \approx \sum_{\alpha=1}^n \frac{\partial u_h}{\partial x}\Big|_{\text{element } \alpha} l_{\alpha i}, \quad (38)$$

where n is the total number of elements containing node i , and $l_{\alpha i}$ is the weight of the α th element at node i . The continuous linear first derivatives can be constructed by interpolating the nodal values using linear interpolation functions. With the first derivatives and second derivatives of (35) at hand, one can construct an error indicator by evaluating (34).

4.1.2. The error indicator using higher order approximation solution, $\epsilon_i^{[2]}$

Another way to estimate $\|e_i\|$ is to identify the exact solution u with the solution obtained by approximation using higher order interpolation functions. In the examples, a six node triangular element solution, \hat{u}_h , is identified as the exact solution while three node triangular elements are used to simulate the mold filling process. The six node triangular element solution is obtained by solving the following discrete problem defined in Ω_i independently of other finite elements without any assembling:

$$\begin{aligned} \hat{u}_h \in V_i: \quad & \int_{\Omega_i} \left(\frac{\partial w_h}{\partial x_i} \epsilon^2 K_{ij} \frac{\partial \hat{u}_h}{\partial x_j} + w_h c \hat{u}_h \right) d\Omega = \int_{\Omega_i} \frac{\partial w_h}{\partial x_i} K_{ij} f_j d\Omega - \int_{\partial\Omega_i} w_h v_i^\wedge n_i d\Gamma, \\ \forall w_h \in V_i, \end{aligned} \quad (39)$$

where

$$V_i = \left\{ v_h = \sum_{i=1}^6 v_{hi} N_i(l_1, l_2): v_{hi} = u_{hi} \text{ for } i = 1, \dots, 3 \right\},$$

v_i^\wedge is the average of flux at the boundary with the adjacent element and

$$\begin{aligned} N_1 &= l_1 - \frac{1}{2} N_6 - \frac{1}{2} N_4, & N_2 &= l_2 - \frac{1}{2} N_4 - \frac{1}{2} N_5, \\ N_3 &= (1 - l_1 - l_2) - \frac{1}{2} N_5 - \frac{1}{2} N_6, & N_4 &= 4l_1 l_2, \\ N_5 &= 4l_2(1 - l_1 - l_2), & N_6 &= 4l_1(1 - l_2 - l_1). \end{aligned}$$

Here l_1 and l_2 are the area coordinates within each finite element. The approximation of the exact solution \hat{u}_h is the solution to the discrete problem in an element Ω_i with boundary condition $v_i = v_i^\wedge$ while having the same value with u_h at nodes of a three node triangular element.

Using the approximation of the exact solution, the amount of error at each element is estimated by

$$\|e_i\| = \sqrt{a_{\Omega_i}(u_h - u, u_h - u)} \approx \sqrt{a_{\Omega_i}(u_h - \hat{u}_h, u_h - \hat{u}_h)}. \quad (40)$$

Thus, the error indicator at the i th element can be given by

$$\varepsilon_i^{[2]} = \sqrt{a_{\Omega_i}(u_h - \hat{u}_h, u_h - \hat{u}_h)}. \quad (41)$$

4.1.3. The error indicator using residual at each element, $\varepsilon_i^{[3]}$

Another way to estimate $\|e_i\|$ is to use the residual within each element, Ω_i . The residual, r_i within Ω_i , can be represented as

$$r_i = \nabla \cdot (\varepsilon^2 K(f - \nabla u_h)) + cu_h. \quad (42)$$

Assuming that this residual is almost zero within each element, the estimation will lead to the error indicator

$$\varepsilon_i^{[3]} = \left(\frac{ch}{|K|^{0.5}} \int_{\partial\Omega_i} J^2 d\Gamma \right)^{0.5}, \quad (43)$$

where c is a constant, h is the size of each element, and J is the jump of normal flux ($=\varepsilon^2 K(f - \nabla u_h) \cdot n$) across the element boundaries. The derivation of (43) is omitted for simplicity (see [20] for details).

4.2. Generation of sub-domains

Once the error indicators show that remeshing is necessary, data for an automatic mesh generation program are prepared. These data include the description of the domain and the size of new meshes within that domain. In the current method, this is achieved by dividing the flow domain into several sub-domains and assigning a mesh size to each of them. Within each sub-domain, uniform meshes are generated according to the mesh sizes in the mesh generation stage.

The new mesh size, h_i^{new} , is determined at each element of the current mesh according to its mesh size, h_i^{old} , and the error indicator, ε_i . The new mesh size is given by

$$h_i^{\text{new}} = h_i^{\text{old}} \left(\frac{\varepsilon_{\text{ref}}}{\varepsilon_i} \right)^p, \quad (44)$$

where ε_{ref} is a reference error, and p is a constant which depends on the choice of error indicator and the choice of mesh generation program. Usually ε_{ref} is set to the average value of the upper and lower error bounds. This will reduce the mesh size if ε_i is larger than the reference error, and increase the mesh size if ε_i is smaller than the reference error.

According to their new mesh sizes, elements of the current mesh are classified into several groups. Each group is a collection of elements having similar mesh sizes, h_i^{new} . A group may consist of several continuous regions which are sub-domains, each of which can be identified by the following procedure.

- (1) Make a list of line segments for each group by decomposing the sides of elements within the group.
- (2) Determine line segments lying on the boundaries of each sub-domain. This can be achieved by examining the occurrence of every individual segment in the line segment list. If a line segment appears once in the list, it is on the boundary of a sub-domain. If it appears twice, it is inside a sub-domain.
- (3) Create loops by connecting line segments lying on the boundaries of sub-domains.
- (4) Determine the geometric relationships between each loop.
- (5) Construct sub-domains.

This procedure is illustrated in Fig. 8.

Within each sub-domain, the average value of the mesh sizes, h_i^{new} , is calculated. This average, along with the boundary description of each sub-domain, is used as input data of the mesh generation program.

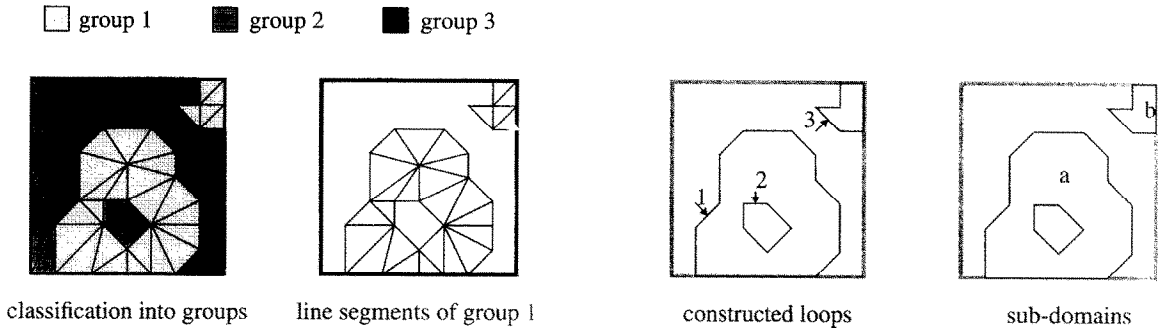


Fig. 8. Generation of sub-domains.

4.3. Mesh generation

Most of the automatic mesh generation method can be classified into the following four types: conformal mapping, elliptic, serendipity, and algebraic mesh generation method. Among these, the algebraic mesh generation method is the most flexible method and can be used to generate the finite element meshes from the description of the sub-domain boundaries and the corresponding mesh densities. The key idea of this method is to generate finite element meshes using a rectangular net or a triangular net.

In the algebraic mesh generation method, meshes are first generated within the shrunk domain of each sub-domain. A shrunk domain is slightly smaller and has the same shape as the sub-domain. Meshes within shrunk domains are generated by the following procedures:

- (1) Calculate the number of grid points in the horizontal and vertical direction using the dimension of the sub-domain and corresponding mesh size.
- (2) Generate grid points at nodal points of the triangular net or rectangular net which covers the whole sub-domain.
- (3) Remove grid points outside the shrunk domain.
- (4) Connect grid points systematically to generate finite element meshes.

After the mesh generation within each shrunk domain, meshes are generated within boundary strips between them on a trial and error basis.

The size and shape of generated meshes changes abruptly between neighboring regions because (a) elements are generated quite differently in the shrunk domain and in the boundary strip, and (b) elements within each shrunk domain are generated independently from neighboring shrunk domains. In order to transform these meshes to smoothly graded ones, smoothing schemes are applied to the meshes. Two smoothing schemes, Gaussian and Winslow, are used in the current mesh generation program. The Gaussian smoothing scheme relocates each node to the center of neighboring nodes, and the Winslow smoothing scheme relocates each node to the center of the centroids of surrounding elements.

After the smoothing process nodes are renumbered to minimize the jump of the node numbering between adjacent nodes. This reduces the bandwidth of the matrix in discretized equations and reduces the computational effort in the simulation.

4.4. Transferring old mesh nodal data to new mesh

After the generation of the new finite element mesh, input data to the simulation program must be prepared. These data contain the initial and boundary conditions for the simulation of the next subsequent time steps. They are obtained by transferring the old mesh nodal information to the newly generated mesh. This can be achieved by linear interpolation. Then, the simulation continues with the newly generated mesh checking error indicators.

5. Examples

The implementation of the adaptive remeshing method is examined in the following two examples. In the first example, a rectangular mold geometry with one insert and two injection ports is considered in the simulation. In the second example, an inner reinforcement of the automobile bonnet is considered as an extension of the method to a shell type mold geometry. Each example is simulated three times using three different error indicators discussed in Section 4.1.

In the simulation, the mesh sizes are not allowed to be larger than a certain limit at the moving free boundary in order to provide a smooth flow front profile. As a result, the meshes are much smaller than the size suggested by the error indicator. Because of this, error indicators at the flow front are much smaller than the lower error bound, interrupting the simulation at each time step. To resolve this problem, remeshing is not performed until the total number of elements with error indicators lower than the lower bound reaches a certain point.

In order to examine the performance of the adaptive remeshing method, meshes are plotted at various stages of the simulation, and error indicators are plotted before and after the adaptation process. In both examples, pressure plots and wet ratio plots are obtained as results of the simulation. In addition, the flow field within the microstructure is obtained and plotted in the second example.

5.1. Example 1

The configuration of the mold geometry is shown in Fig. 9. Layers of aligned fiber bundles arranged within the mold cavity are shown in Fig. 10. The dimensions of fiber bundles are shown in Fig. 11. The mold has one octagonal insert and two injection ports, A and B, through which the resin is injected. The gravity is normal to the mold geometry and its effect is neglected. Permeabilities of the fiber mat and the fiber bundle are obtained by solving the corresponding microscopic equations using the finite element method. Parameters used in the first example are as follows.

- dimension of the mold geometry, $1 \text{ m} \times 0.7 \text{ m} \times 0.05 \text{ m}$;
- $\varepsilon = 0.005$, $\delta = 0.01$;
- injection rates: injection port A $7.0 \times 10^{-4} \text{ m}^3/\text{s}$, injection port B $6.9 \times 10^{-4} \text{ m}^3/\text{s}$;
- viscosity of resin, 1 Pa s ;
- computed permeability (see (14)) of the fiber mat,

$$\frac{1}{\mu} K = \begin{bmatrix} 0.28 \times 10^{-2} & 0 \\ 0 & 0.28 \times 10^{-2} \end{bmatrix} \text{m}^2/(\text{Pa s});$$

- computed permeability (see (7)) within fiber bundles,

$$\frac{k}{\mu} = 0.1065 \times 10^{-2} \text{m}^2/(\text{Pa s}).$$

The results of the simulation are shown in Figs. 12–18. Figures 12–14 show pressure and wet ratio

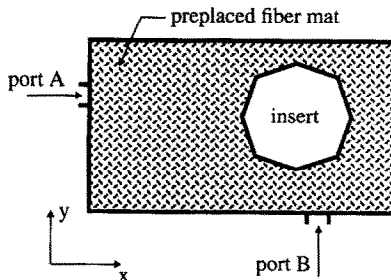


Fig. 9. Configuration of the mold used in Example 1.

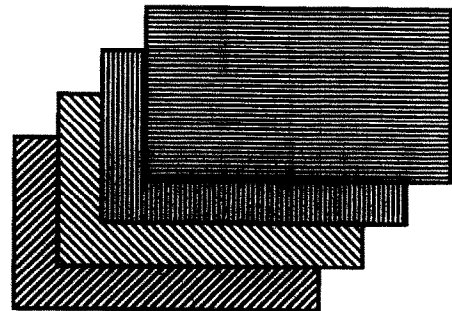


Fig. 10. Layers of assigned fiber bundles.

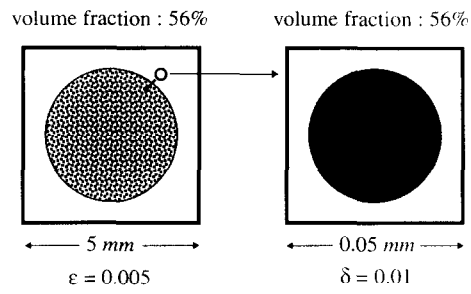
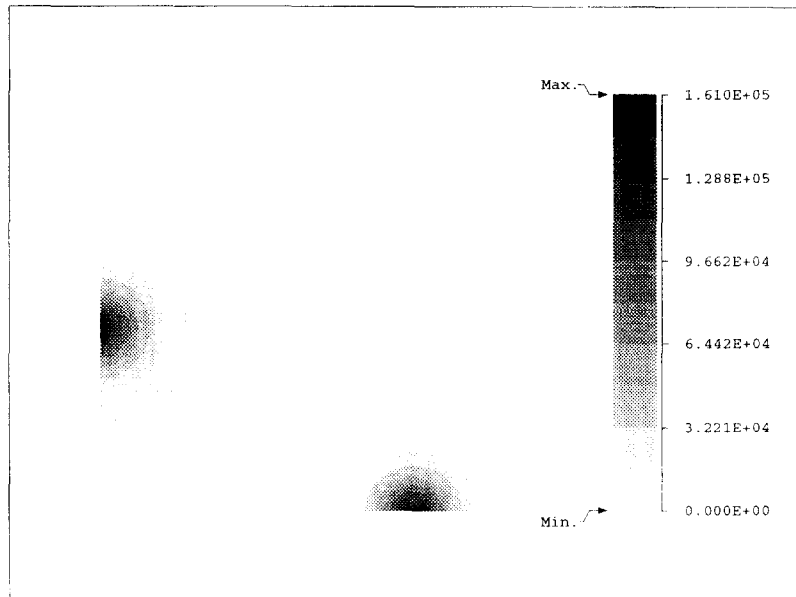
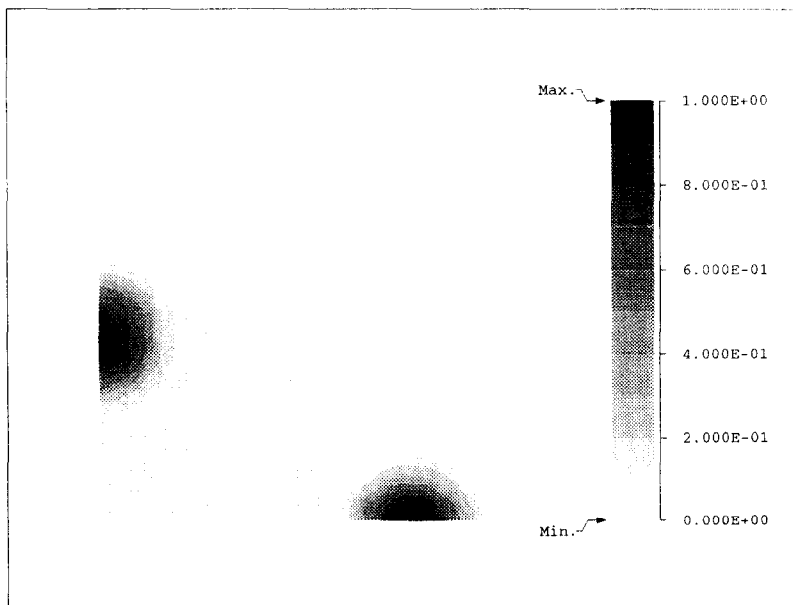


Fig. 11. Unit cell cross section of the fiber mat.



<pressure>



<wet ratio>

Fig. 12. Pressure (Pa) and wet ratio plots when 20% full.

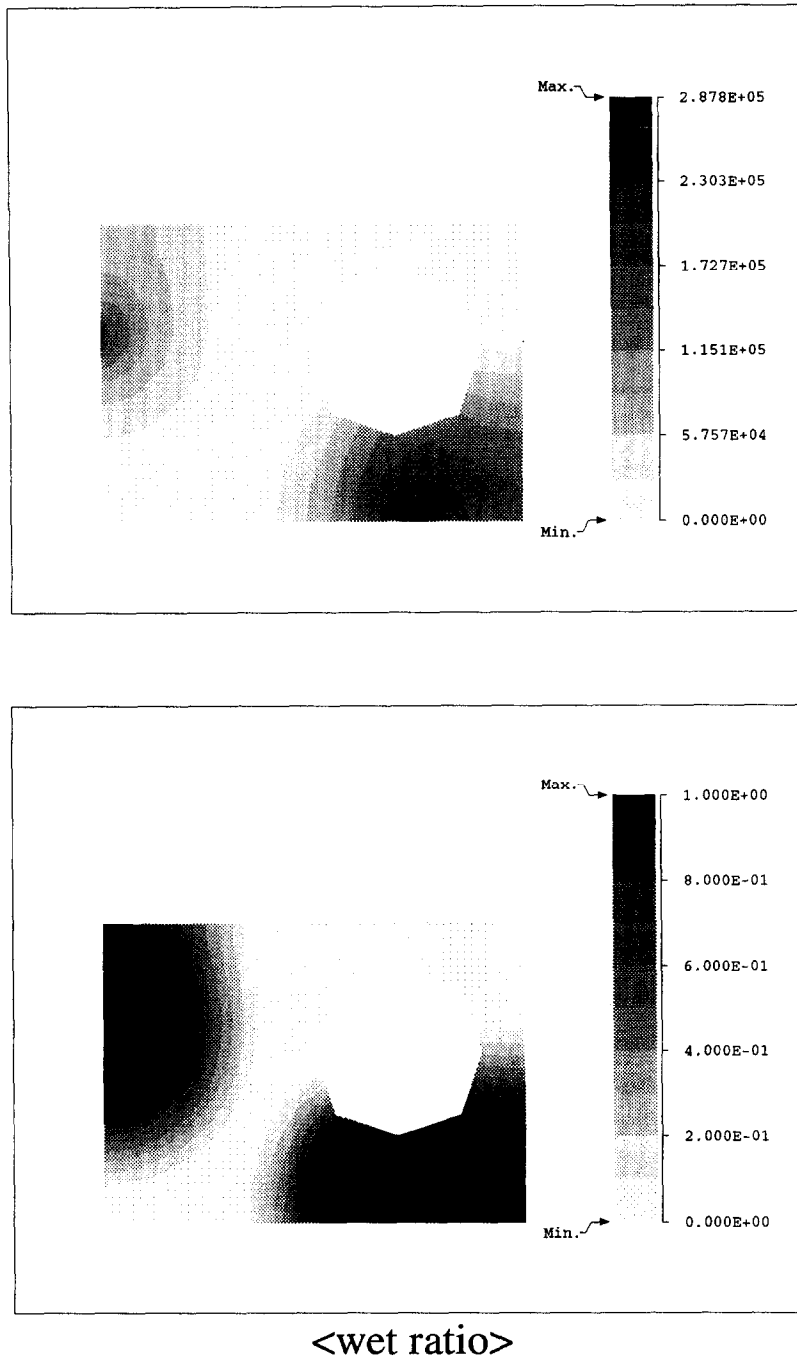


Fig. 13. Pressure (Pa) and wet ratio plots when 66% full.

distributions at three different stages of the mold filling process. These pressure plots can provide operating parameters of the RTM process, and wet ratio plots can provide valuable information for the mold design such as locations of the merging flow fronts, proper location of the air tabs, etc. By checking if the wet ratio is greater than zero at each point, one can identify the flow domain since the resin starts to impregnate the fiber bundle as soon as it is included in the flow domain. Figures 15–17 are some of the adapted finite element meshes generated during the simulation using three different error indicators discussed in Section 4.1. It is clear that rather refined meshes are generated in the neighborhood of the injection ports in order to approximate large pressure gradients accurately. As

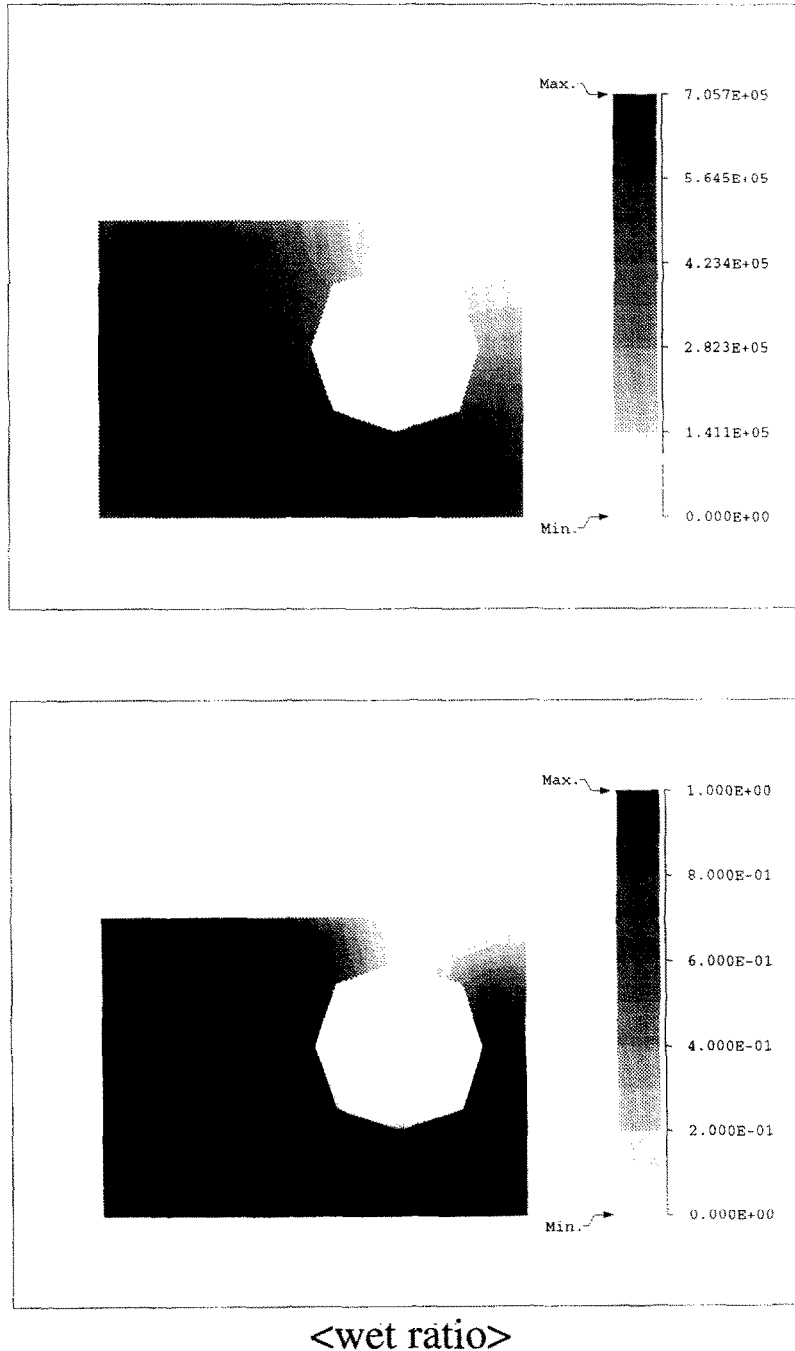
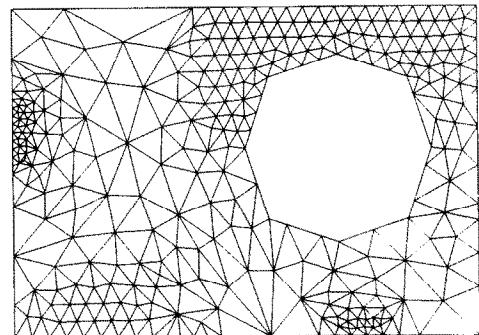
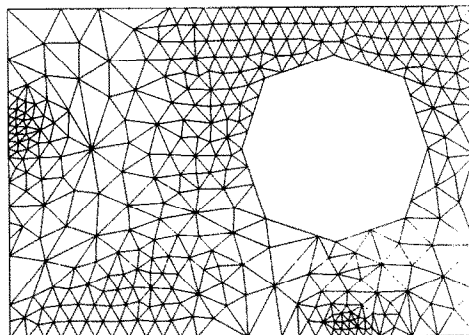
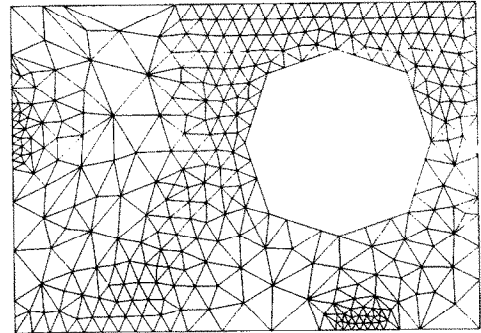
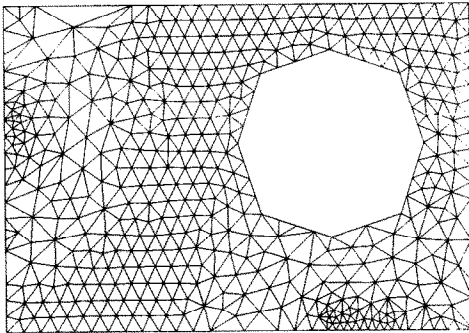
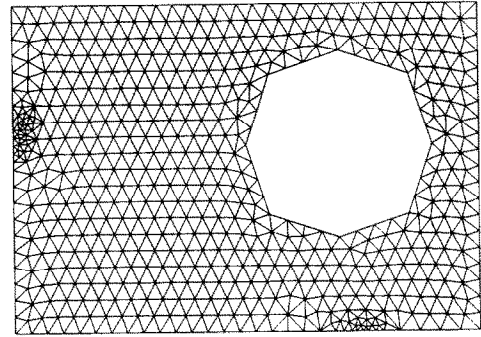
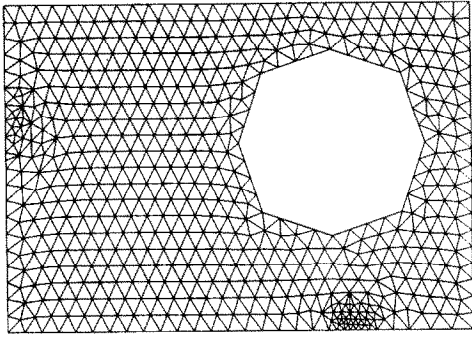


Fig. 14. Pressure (Pa) and wet ratio plots when 96% full.

mentioned earlier, while the neighborhood of the flow front does not involve a large amount of approximation error, de-refinement is not performed to maintain a smooth flow front profile. It is also notable that there are little differences between the adapted meshes based on different error indicators, which means that the choice of the error indicator is not so critical as long as the error indicator can provide a good estimation of the finite element approximation error. Figure 18 shows the distribution of errors in the meshes before and after the adaptation. From the plot, it is clear that the error after the adaptation has more uniform distribution.

Fig. 15. Finite element meshes generated using $\epsilon^{[1]}$.Fig. 16. Finite element meshes generated using $\epsilon^{[2]}$.

5.2. Example 2

The configuration of the mold geometry is shown in Fig. 19. The fiber mat is a woven fiber bundle, and the geometry of its representative unit cell is approximated as shown in Fig. 20. The configuration and dimension of fiber bundles are identical to the first example, and the permeability calculated in the first example is used. The permeability of the fiber mat is obtained by solving the microscopic equation. The parameters used in the second example are as follows.

- dimension of the mold geometry, $1.5 \text{ m} \times 1.2 \text{ m} \times 0.02 \text{ m}$;
- $\epsilon = 0.002$, $\delta = 0.025$;
- injection rate, $1.0 \times 10^{-4} \text{ m}^3/\text{s}$;
- viscosity of resin, 1 Pa s ;
- computed permeability (see (14)) of the fiber mat,

$$\frac{1}{\mu} K = \begin{bmatrix} 0.1187 \times 10^{-2} & 0 \\ 0 & 0.1187 \times 10^{-2} \end{bmatrix} \text{m}^2/(\text{Pa s});$$

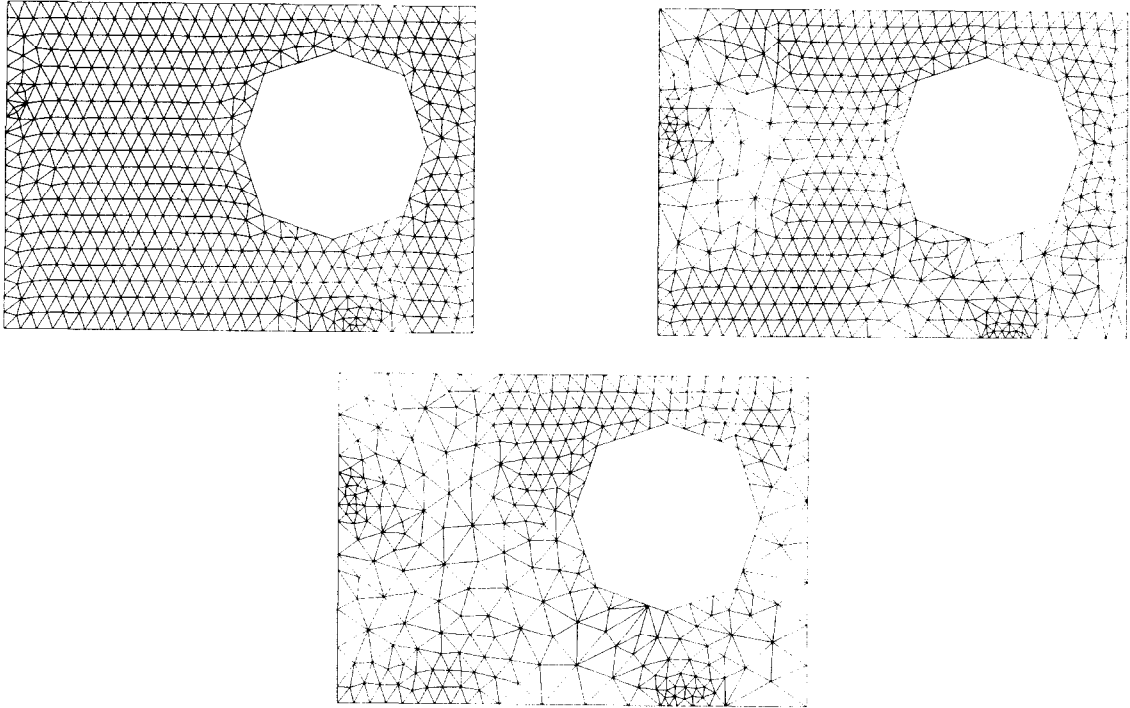


Fig. 17. Finite element meshes generated using $\epsilon^{[3]}$.

– computed permeability (see (7)) within fiber bundles,

$$\frac{k}{\mu} = 0.1065 \times 10^{-2} \text{ m}^2 / (\text{Pa s}).$$

The results are shown in Figs. 21–27. Figures 21–23 are the pressure plots and wet ratio plots at different stages of the mold filling process. Figures 24 and 25 show the local flow field at an arbitrary point A within the first level microstructure. The local flow field is calculated by (19) after the global (or averaged) flow field is obtained. This is one of the main advantages of the homogenization method. With the homogenization method, one cannot only calculate the global flow field but also find the details of the local flow field whenever necessary. Figure 26 shows some of the finite element meshes generated during the simulation. Since similar sets of meshes are generated from three different error

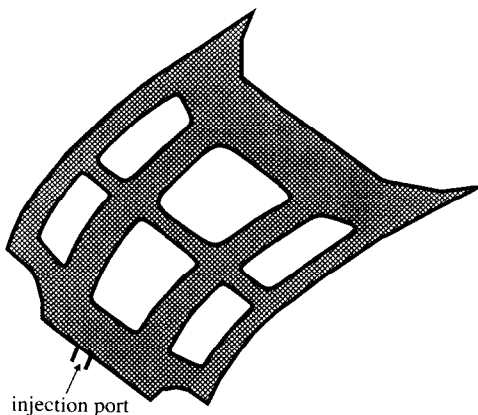
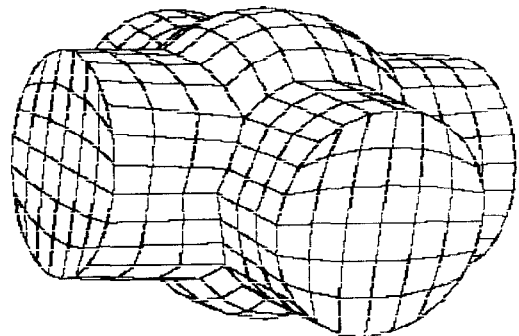
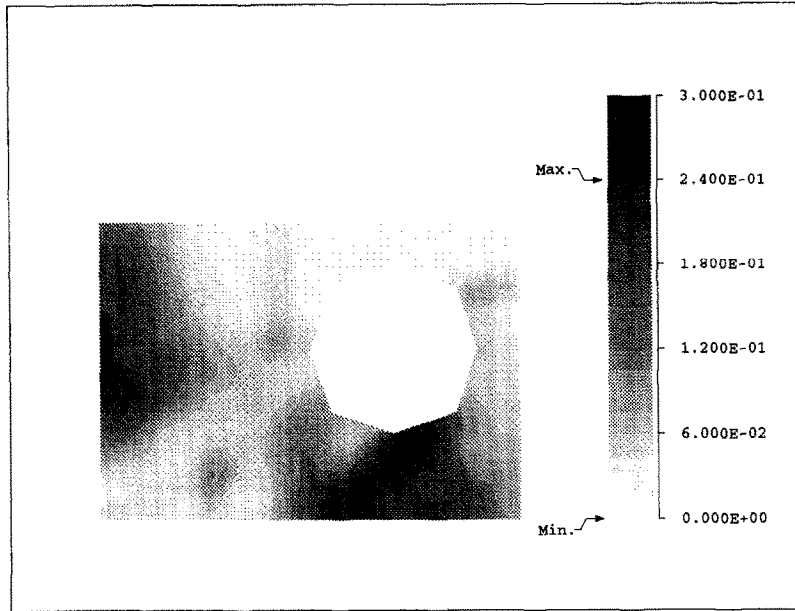


Fig. 19. Configuration of the mold used in Example 2.

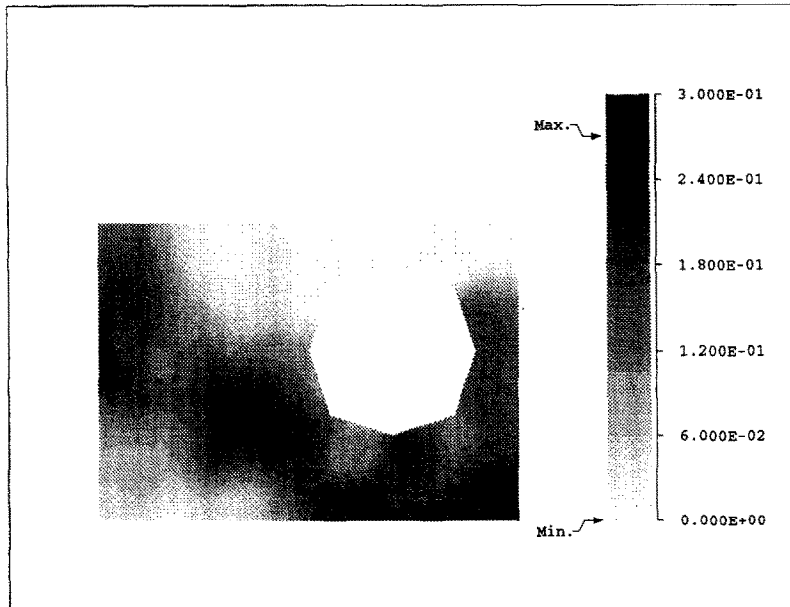


diameter of fiber bundle: 0.5mm
volume fraction of the flow domain: 0.7

Fig. 20. Configuration of fiber bundle, unit cell shown.



<before>



<after>

Fig. 18. Error distribution before and after adaptation.

indicators, only one is shown. Figure 27 shows the error distribution before and after the adaptation. Again the error after the adaptation has more uniform distribution.

6. Conclusion

The adaptive remeshing method is developed and its application to the mold filling process is investigated using three different error indicators. The method, combined with FE/CV method, has

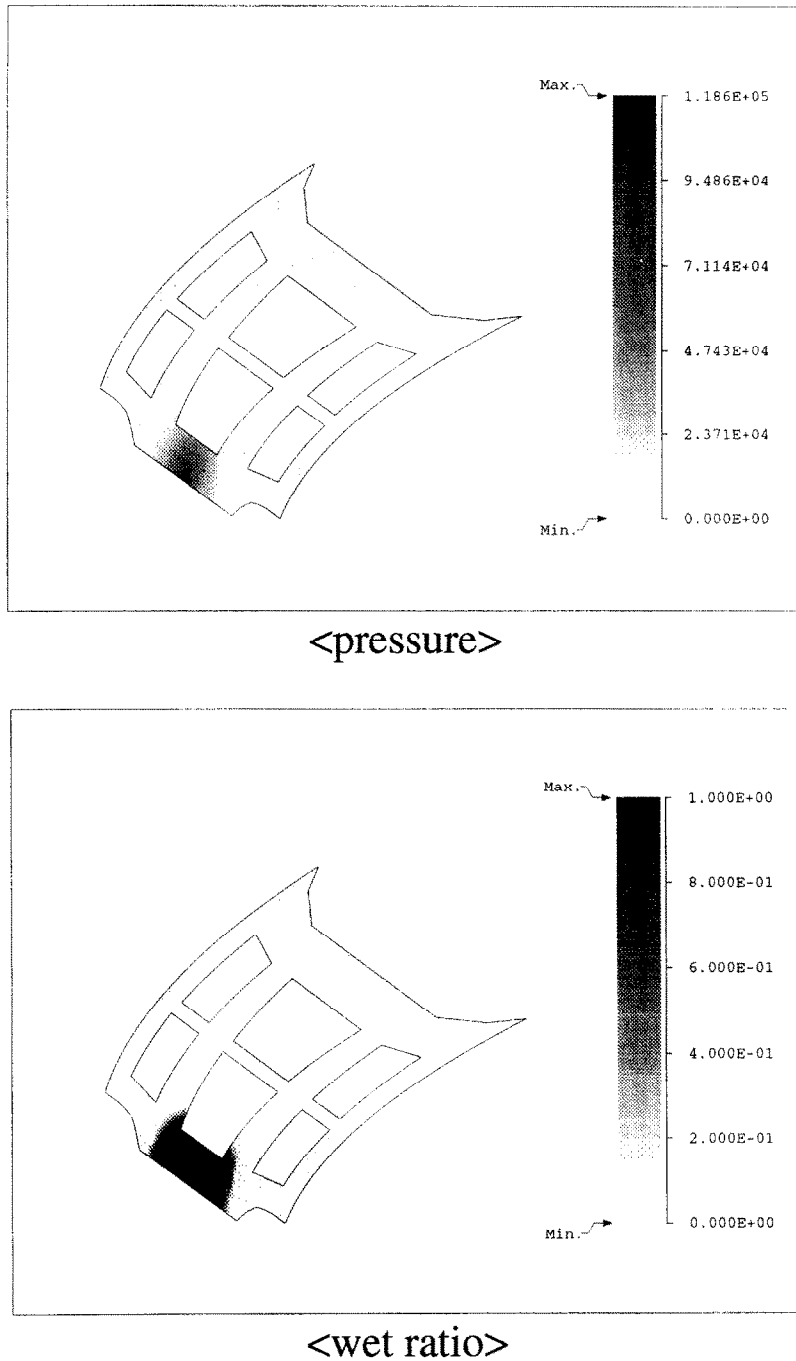
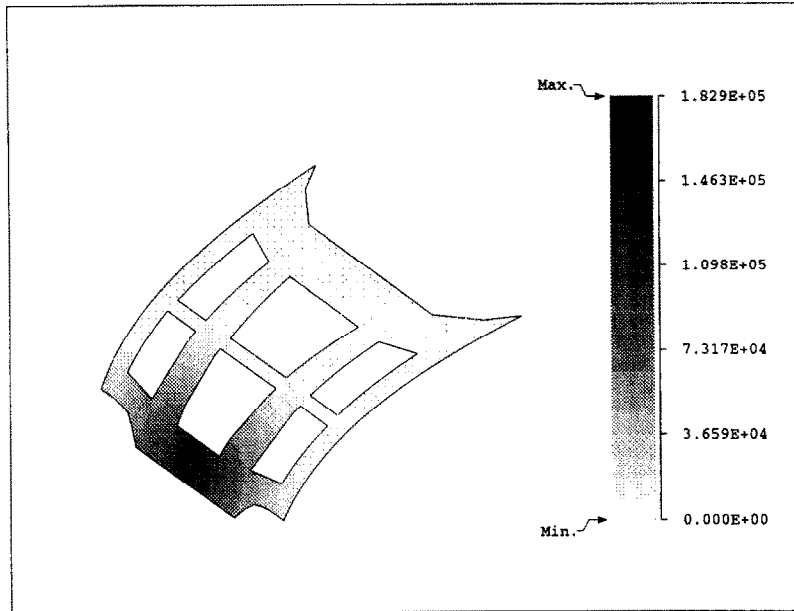
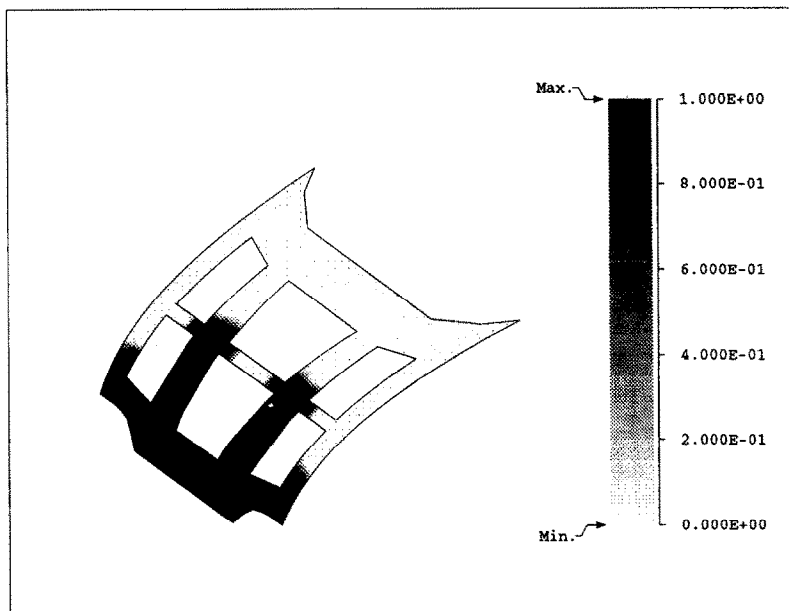


Fig. 21. Pressure (Pa) and wet ratio plots when 10% full.

increased the computational efficiency significantly in the mold filling simulations. As expected, the simulations using three different error indicators provided very similar results. Finite element meshes are modified according to the error indicators at each adaptation stage. Meshes with larger errors are refined whereas meshes with smaller errors are enlarged, resulting in a more uniform error distribution after each remeshing process. Exceptions are the meshes along the moving free boundary. Since the meshes are kept much smaller to provide a smooth flow front, the error indicators have very small values there.



<pressure>



<wet ratio>

Fig. 22. Pressure (Pa) and wet ratio plots when 40% full.

Although the meshes are adjusted to give uniform error distribution at reference error, ε_{ref} , the average value of the error after the adaptation process is lower than the reference error. This is partly because of the smoothing scheme and partly because of the mesh generation program. Since meshes at the flow front are kept small, the smoothing scheme tends to reduce the sizes of the meshes near the flow front. Also within the mesh generation program, enlarging the mesh size is difficult especially when it is confined to a small region. These two effects reduce the mesh sizes and result in smaller error indicators than expected. In order to maximize the performance of the adaptive remeshing method, the reference error should be chosen considering these effects.

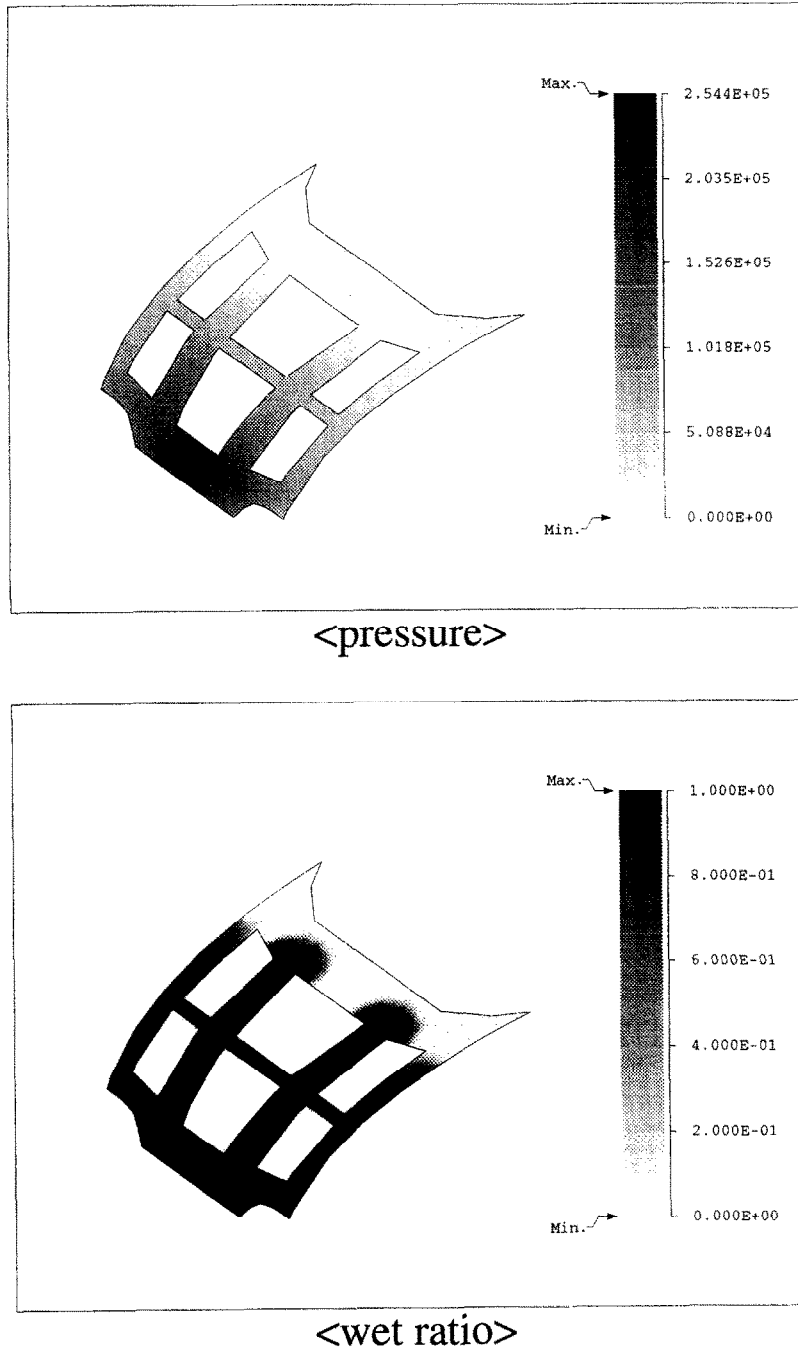


Fig. 23. Pressure (Pa) and wet ratio plots when 73% full.

It is found that the performance of the adaptive remeshing scheme greatly depends on the error bound and the reference error. If the interval between the error bound is too small or if the reference error is not chosen appropriately, the method causes frequent remeshing which cancels out its benefits. On the other hand, if the interval is too large, the remeshing is not performed enough, leaving room for further improvement.

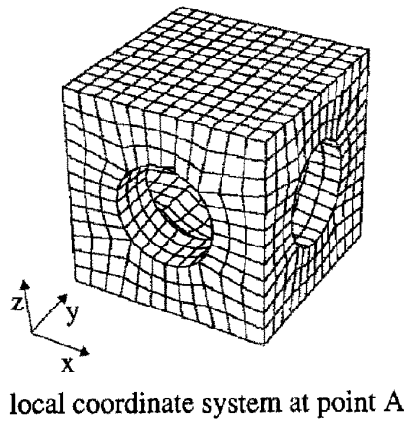
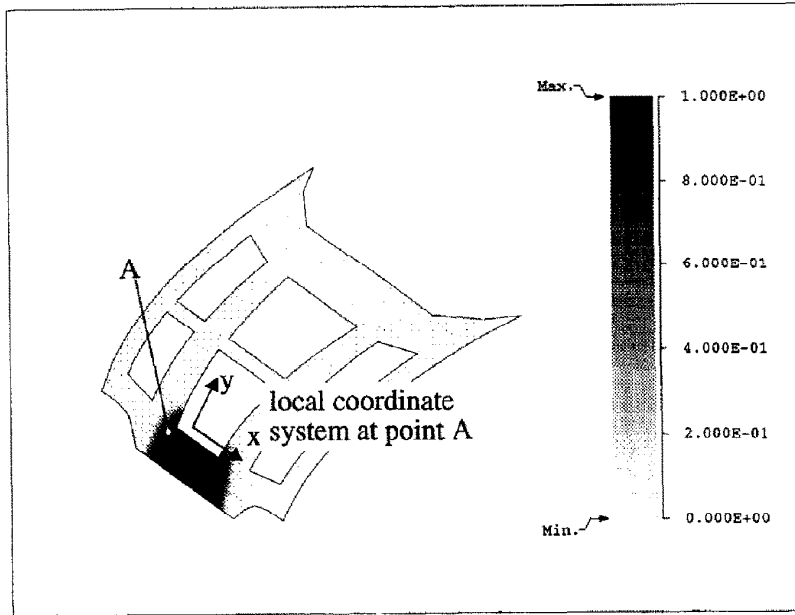


Fig. 24. The flow domain of the first level microstructure at point A when the mold is 10% full.

Acknowledgment

The authors express their appreciation to A. Tezuka and G.-J. Lin who provided the automatic mesh generation routine. During this research both authors were supported by NSF (MSS-9021683). This support is gratefully acknowledged.

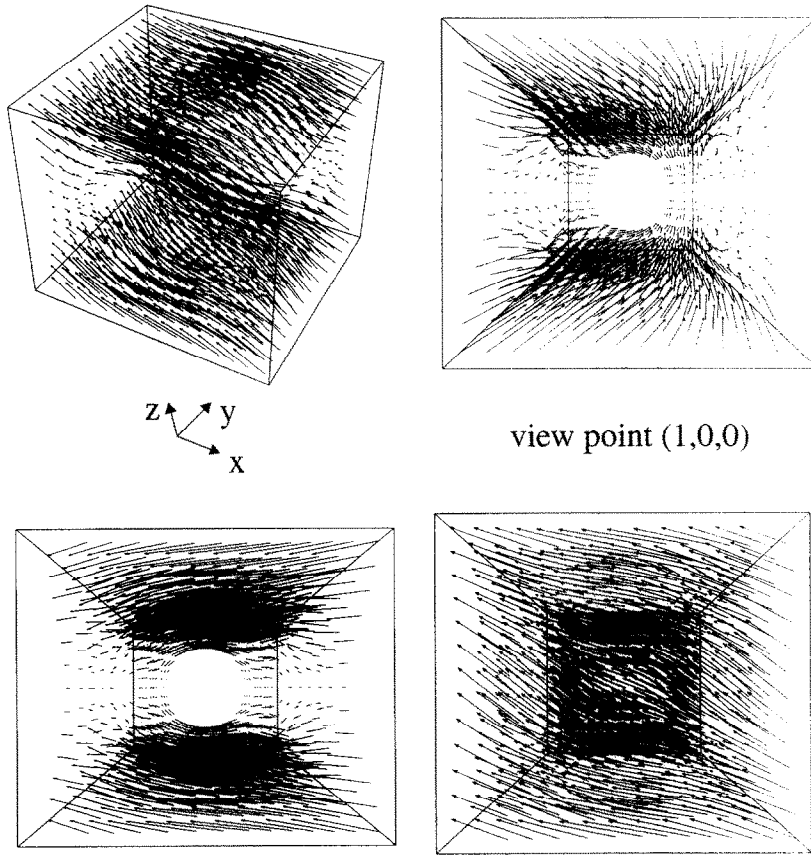


Fig. 25. The flow field at point A when the mold is 10% full.

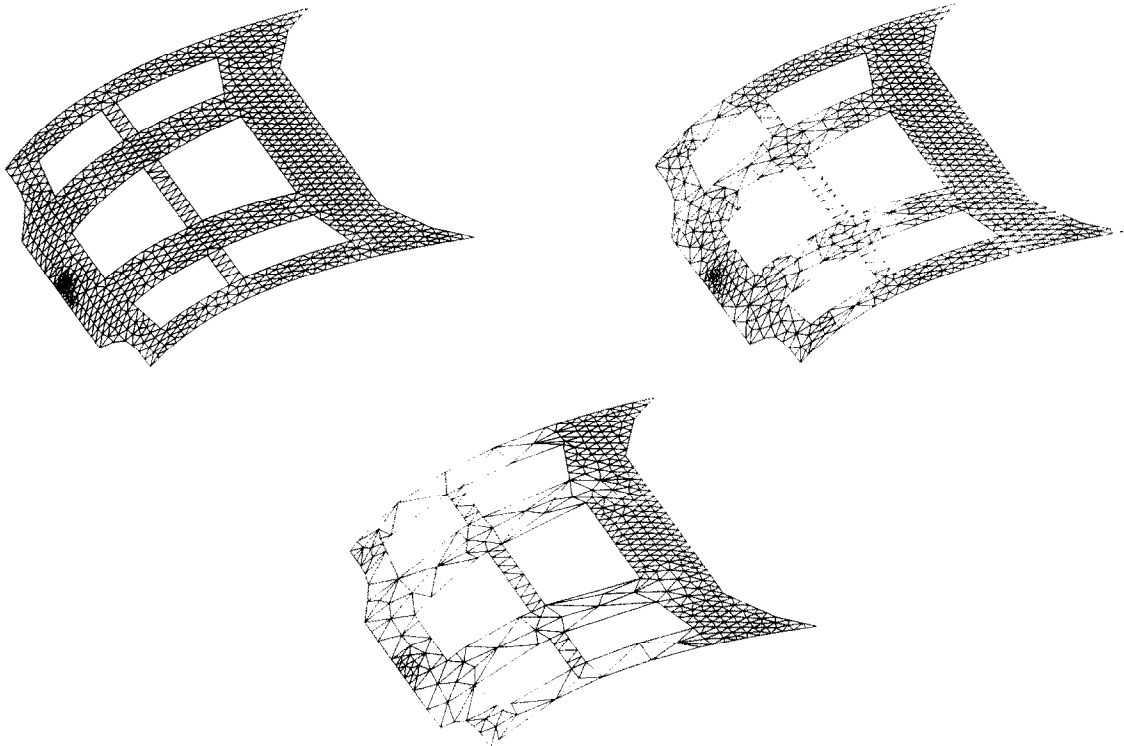
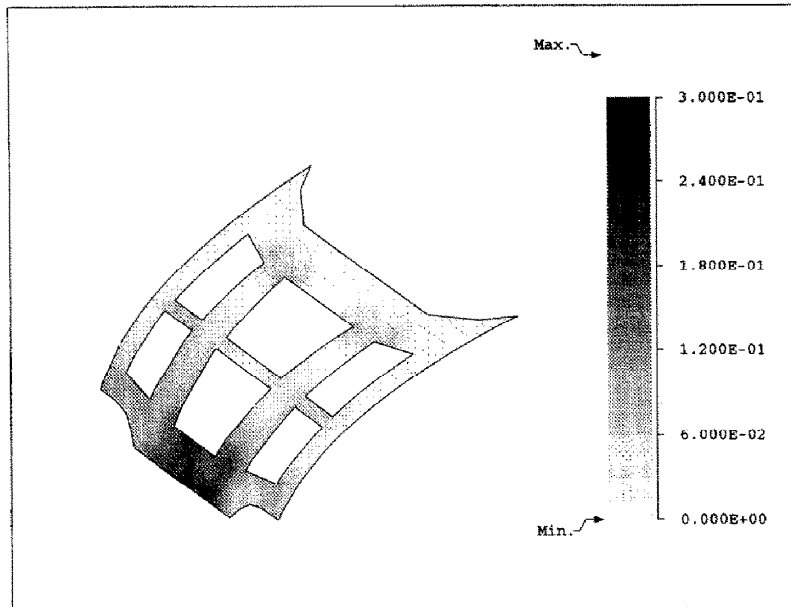
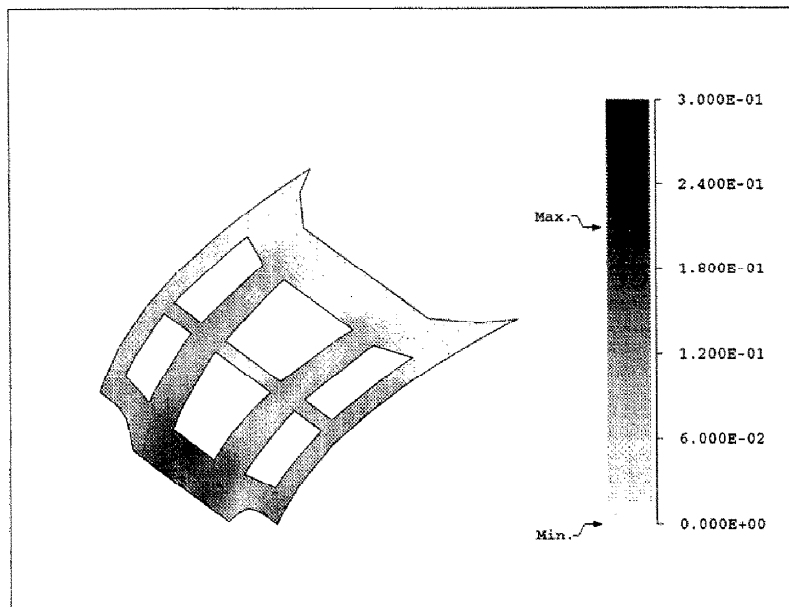


Fig. 26. Finite element meshes used.



<before>



<after>

Fig. 27. Error distribution before and after adaptation.

References

- [1] H. Lee and Hyo-chol Sin, Iterative boundary pressure reflection method for the simulation of injection mold filling, *Polymer Engrg. Sci.* 30 (1990) 1513–1522.
- [2] Z. Tadmore, E. Broyer and C. Gutfinger, Flow analysis network (FAN) – A method for solving flow problems in polymer processing, *Polymer Engrg. Sci.* 14 (1974).
- [3] V.W. Wang, C.A. Hieber and K.K. Wang, Dynamic simulation and graphics for the injection molding of three-dimensional thin parts, *J. Polymer Engrg.* 7 (1986) 21–45.
- [4] J.A. Bennett and M.E. Botkin, Structural shape optimization with geometric description and adaptive mesh refinement, *AIAA J.* 23 (1985).

- [5] O.C. Zienkiewicz and G.C. Huang, Adaptive modeling of transient coupled metal forming processes, in: E.G. Thompson et al., eds., NUMIFORM89 Numerical Methods in Industrial Forming Processes (A.A. Balkema, Rotterdam, 1989) 3–10.
- [6] J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz, Adaptive remeshing for compressible flow computations, *J. Comput. Phys.* 72 (1987) 449–463.
- [7] N. Yukawa, N. Kikuchi and A. Tezuka, An adaptive remeshing method for analysis of metal forming processes, *Adv. Technol. Plastics* 4 (1990).
- [8] J. Fukuda and J. Suhara, Automatic mesh generation for finite element analysis, in: J.T. Oden and Y. Yamamoto, eds., *Advances in Computational Methods in Structural Mechanics and Design* (UAH Press, Huntsville, AL, 1972).
- [9] A. Tezuka, A development of automatic mesh generator with arbitrary geometry-based input description, Dissertation Thesis, Dept. of ME/AM, the Univ. of Michigan, 1989.
- [10] T.S. Lundström, Micromechanical analysis of the permeability of fibrous preform, SICOMP Technical Report 91-001, 1991.
- [11] Raju Davé, A unified approach to molding resin flow during composite processing, *J. Composite Mater.* 24 (1990) 22–41.
- [12] W.B. Young, K. Han, L.H. Fong and L.J. Lee, Flow simulation in molds with preplaced fiber mats, *Polymer Composites* 12 (1991) 391–400.
- [13] A.W. Chan and S.-T. Hwang, Mold filling simulations for the injection molding of continuous fiber-reinforced polymer, *Polymer Engrg. Sci.* 26 (1988) 333–339.
- [14] U. Hornung, Homogenization of miscible displacement in unsaturated aggregated soils, *Composite Media and Homogenization Theory* (1991) 143–152.
- [15] T. Arbogast, Analysis of the simulation of single phase flow through a naturally fractured reservoir, *SIAM J. Numer. Anal.* 26 (1989) 12–29.
- [16] G.I. Barenblatt, Iu.P. Zheltov and I.N. Kochina, Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks [Strata], *PMM* 24 (1960) 852–864.
- [17] J.B. Keller, Darcy's Law for flow in porous media and the two-space method, in: R.L. Sternberg, A.J. Kalinowski and J.S. Papadakis, eds., *Nonlinear Partial Differential Equations in Engineering and Applied Science* (Dekker, New York) 429–443.
- [18] L. Tartar, Incompressible fluid flow in a porous medium – Convergence of the homogenization process, in: E. Sanchez-Palencia, ed., *Non Homogeneous Media and Vibration Theory* (Springer, Berlin, 1980).
- [19] A.M. Winslow, Numerical solution of the quasi-linear Poisson equation in a nonuniform triangle mesh, *J. Comput. Phys.* 2 (1967) 149–192.
- [20] D.W. Kelly et al., A posteriori error analysis and adaptive process in the finite element method, *Internat. J. Numer. Methods Engrg.* 19 (1983) 1593–1619.
- [21] Ching-chih Lee, An improved Flow Analysis Network (FAN) for irregular geometries, *Polymer Engrg. Sci.* 30 (1990) 1607–1614.
- [22] C.A. Hieber and S.F. Shen, Flow analysis of the non-isothermal two dimensional filling process in injection molding process, *Israel J. Technol.* 16 (1978) 248–254.
- [23] C.A. Hieber and S.F. Shen, A finite element/finite difference simulation of the injection-molding filling process, *J. Non-Newtonian Fluid Mech.* 7 (1980) 1–32.
- [24] F. Trochu and R. Gauvin, Some issues about the numerical simulation of mold filling in resin transfer molding, *Advanced Composite Lett.* (1992).