Parallel Computations of 3D Flows in Resin Transfer Molding

Whie Chang*

(Received January 21, 1996)

In this work, fast 3D non-isothermal mold flow simulation programs that take advantage of the new generation parallel supercomputers such as Cray C90 and TMC CM-5 are developed and evaluated. The codes are based on the FE/CV formulations and utilize "overfill and redistribute scheme" to enhance the flow front movement. They rely on recently developed fast iterative solvers: the generalized minimal residual solver and the quasi minimal residual solver. In the examples, the performance of the iterative solvers and the overfill and redistribute scheme is evaluated, and a 3D simulation using 104, 000 HEXA8 element mesh is performed.

Key Words: Parallel Computation, Three Dimensional Mold Flow Analysis, Resin Transfer Molding

1. Introduction

Resin Transfer Molding (RTM) is a newly developed mass production method for continuous fiber reinforced materials. In RTM, the mold cavity is packed with dry fiber preforms which act as reinforcements. These preforms are impregnated progressively by the resin entered through the injection ports of the mold. After the mold is completely filled, the resin solidifies in the curing phase and the product is taken out of the mold. When the mold is properly designed, RTM can produce high quality composite parts with tight tolerance and low void fraction.

The analysis of a mold filling process is a very important step in the product development stages of RTM. It can provide useful information to a mold designer such as pressure distributions, possible location of defects, hard to fill regions, proper locations of injection ports and air tabs. These information can be used to save product development cost and time by reducing the number of necessary experiments.

Many researchers have investigated the mold filling processes in the context of RTM. Gonzalez, Castro, and Macosko(1985) studied an 1D mold flow using a disk shaped mold. Coulter and Guceri(1988) studied the effects of anisotropic reinforcements on the mold flow. They considered a 2D isothermal flow and developed a finite difference code to simulate the mold flow utilizing boundary fitted curvilinear coordinate systems. Young, Han, Fong, Lee and Liou(1991) investigated the effects of permeability variations in a similar 2D setting. They simulated the mold flow utilizing Finite Element / Control Volume (FE/CV) method. Trochu and Gauvin(1992) also simulated a 2D flow using non-conforming finite element method. Bruschke and Advani (1994) considered a non-isothermal flow and solved a 2D momentum balance equation with a 3D heat balance equation using the FE/CV method. Young(1994) considered a non-isothermal mold flow and developed a 3D simulation code based on FE/CV method.

In this work, parallel computations of the non-isothermal 3D mold flow simulation is presented. The computation of 3D simulation is an order of magnitude larger than that of 2D simulation. To accomplish 3D simulation for meshes with large number of elements, the simulation is carried out on new generation parallel super-com-

* Department of Mold and Die Design, Seoul National Polytechnic University

puters such as Cray C90 and TMC CM-5.For the shared memory super-computer Cray C90, the code is optimized to achieve maximum vectorization, whereas, for the massive parallel super-computer CM-5, the code is optimized to minimize inter-processor data transfer while maintaining work load balance among processors. The program utilizes fast iteration solvers developed in the 1980s and 1990s such as the Generalized Minimal Residual (GMRES) solver and the Quasi Minimal Residual (QMR) solver.

The simulation code incorporates Darcy's flow equation, a heat balance equation, and a constitutive equation relating the resin viscosity to the temperature. Parameters of the equations, such as fiber preform permeability and effective heat conductivity, are obtained by solving the flow behavior within the micro-structure of the composite using the homogenization method. The equations are discretized by FE/CV method and an "overfill and redistribute flow front updating scheme" is utilized to reduce the computation amount significantly.

In the example, the performance of the GMRES and QMR solvers are evaluated, and the effectiveness of the overfill and redistribute scheme is demonstrated. Finally, a 3D non-isothermal simulation is performed using a mesh with 104,000 HEXA8 elements. The computations are done on Cray C90 and TMC CM-5 supercomputers. The computational costs are provided along with the flow front profiles and temperature distributions.

2. Averaged Flow Equations

In RTM, the resin flows through the complex network of pores between the reinforcing fibers. The flow is often treated as flow through porous media and is modeled by Darcy's law. The Darcy's law states that(see Kaviany, 1991):

$$\nu_i^D = \frac{K_{ij}}{\mu}\left(f_j - \frac{\partial P}{\partial x_j}\right) \tag{1}$$

where ν_i^D is defined as flux per unit area and is called "Darcean velocity". μ is the viscosity, f_j the specific weight of the resin, and P the pressure. K_{ij} is the permeability tensor. Combining (1) with the continuity equation, $\nabla \cdot \bar{\nu}^D = 0$ gives:

$$\frac{\partial}{\partial x_i}\left(\frac{K_{ij}}{\mu}\left(f_j - \frac{\partial P}{\partial x_j}\right)\right) = 0 \tag{2}$$

The boundary conditions are: pressure is set to zero at the free flow front, and the normal component of ν_i^D is set to zero at the mold walls. Either pressure or flow rates are prescribed at the injection ports depending on the port types.

As the resin flows inside the mold cavity, heat transfer takes place between the resin, fiber preform, and the mold walls creating temperature variations. The temperature variations influence the flow pattern by changing the resin viscosity, which is a strong function of temperature. In the heat transfer analysis, it is assumed that the resin and fiber reach local thermodynamic equilibrium instantly at each point. This is justified because the heat transfer coefficient between the resin and the fiber is fairly large compared to the thickness of the fiber. With this assumption, the heat balance equation becomes:

$$(\rho_f c_{Pf}\phi_f + \rho_r c_{Pr}(1-\phi_f))\frac{\partial T}{\partial t} + \rho_r c_{Pr}\nu_i^D\frac{\partial T}{\partial x_i}$$
$$= (1-\phi_f)\,\dot{s} + \chi_{ij}^H\frac{\partial T}{\partial x_i \partial x_j} \tag{3}$$

where x_i is the spatial coordinate and t is the time. T is the temperature. ρ_r, c_{Pr} and ρ_f, c_{Pf} are the density and heat capacity of the resin and of the fiber respectively. ϕ_f is the volume fraction of the fiber. χ_{ij}^H is the effective heat conductivity of the resin and fiber mixture. $\dot{s}$ is the specific heat generated in the resin due to curing. The boundary conditions are: specified resin temperature at the injection ports and at the mold walls. For a more complete analysis, mixed type boundary conditions can be assigned at the mold walls to account for the mold plates and heating/cooling pipes. At the flow front, a convection type boundary condition is used to take into account the initial heat energy of the fiber preform. The actual derivation of (3) is omitted for simplicity. See Chang and Kikuchi(1995) for details.

The heat balance Eq. (3) is coupled with the momentum Eq. (2) in terms of the resin viscosity. The viscosity can be approximated by an Arrhenius type relation which gives the viscosity, μ, as an exponential function of temperature:

$$\mu = \mu_0 \exp(-a\theta) \tag{4}$$

where $\theta = T - T_0$ with T_0 being the reference temperature at which the viscosity is μ_0. In addition to being a function of temperature, the viscosity also depends on the degree of cure. However, we assume that the curing starts after the mold is completely filled and do not consider curing effects in the current analysis. If this effect becomes significant, a chemorheological model need to be introduced to Eq. (4). An analysis which take into account a chemorheological model can be found in Bruschke and Advani (1994) and in Young(1994).

In the mold flow simulation, Eqs. (2) ~ (4) are solved inside the impregnated part of the mold cavity using parameters including the permeability and the effective conductivity tensors. The permeability and conductivity tensors can be obtained either by experiments or by calculation. In the subsequent sections, the calculation of the parameters by considering the flow behavior inside the micro-structures of the fiber preform is given. The ability to calculate the parameters can be very valuable to the mold designer, since it enables him to try out all possible fiber arrangements to come up with a small number of good design candidates which can, then, be tested by experiments.

3. Microscopic Flow Equations

Here, the derivations of the permeability and the average conductivity from the governing equations for the flow inside the fiber preform micro-structure are given. The governing equations are obtained by considering the resin flow in the pore structures of the fiber preform accounting the fibers as boundary conditions. From these equations, the parameters are derived along with the locally averaged flow Eqs. (2) ~ (3).

Configuration of the fiber preform micro-structure is shown in Fig. 1. The resin flow through the pores between the fibers is usually very slow and can be treated as Stoke's flow which is inertia-less viscous flow. The momentum balance equation and the continuity equation of the flow can be written as:

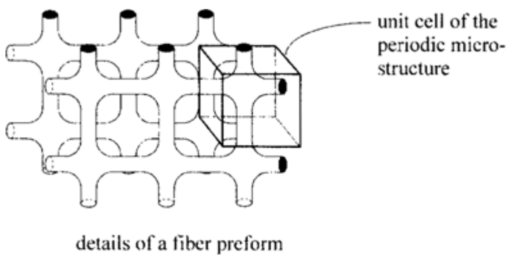
unit cell of the periodic micro-structure
details of a fiber preform

Fig. 1 Fiber preform micro-structure.

$$-\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu \frac{\partial v_i}{\partial x_j}\right) + f_i = 0 \tag{5}$$

$$\frac{\partial v_i}{\partial x_i} = 0 \tag{6}$$

where v_i is the actual velocity of resin. Here, it is assumed that the fiber preform is not deformed or shifted by the resin flow. It is also assumed that the fiber preform has a periodic micro-structure: in other words, it is made of spatial repetitions of a unit cell. The normal component of v_i is set to zero at the solid surface which includes the fibers and the mold wall boundaries.

The heat balance equation is:

$$\rho c_P \frac{\partial T}{\partial t} \rho c_P \left(v_i \frac{\partial T}{\partial x_i}\right) = \frac{\partial}{\partial x_i}\left(\chi_{ij} \frac{\partial T}{\partial x_i}\right) + \dot{s} \tag{7}$$

where χ_{ij} is the heat conductivity. The coefficients ρ, c_P are equal to ρ_r, c_{Pr} at the resin and equal to ρ_f, c_{Pf} at the fiber.

It is almost impossible to solve Eqs. (5) ~ (7) directly due to the complexity of the flow domain which consists of the pores between the fibers. Here, we apply the homogenization method to (5) ~ (7) and get locally averaged governing equations (2) ~ (4) along with the expressions for the permeability and the averaged heat conductivity tensors. The homogenization method takes advantage of the periodic nature of the fiber preform micro-structure and considers the flow behavior within the unit cell of the micro-structure. In the method, a stretched coordinate system, $\bar{y}$ ($\bar{y} = \bar{x}/\varepsilon$, for small ε) is used to capture the details within the unit cell. The configuration of a unit cell in the stretched coordinate is shown in Fig. 2. In addition, a special averaging technique

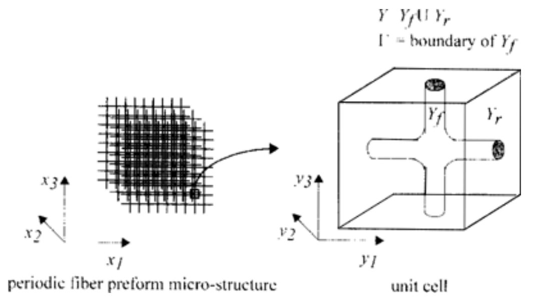

Fig. 2 Unit cell shown in stretched coordinate. $\bar{y}$.

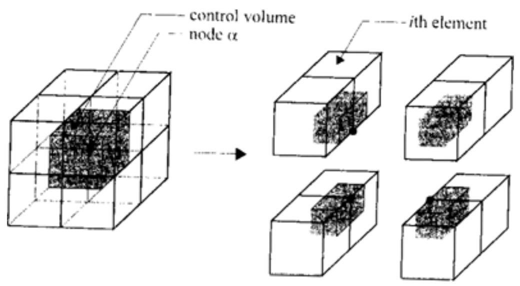

Fig. 3 Control volume assigned to a vertex node.

is applied to get the averaged equations and the parameters.

The permeability and the averaged conductivity tensors are stated here without the formal derivation procedure. See Chang and Kikuchi (1995) for derivation. The permeability tensor is:

$$K_{ij}=\frac{\varepsilon^2}{|Y|}\int_Y \chi_i^j dY \tag{8}$$

where χ_i^j is the solution of:

$$\chi_i^j \in V_y\colon \int_{Y_1}\frac{\partial \chi_i^j}{\partial y_k}\frac{\partial w_i}{\partial y_k}dY=\int_{Y_1} w_j dY \quad \forall w_i \in V_y, \tag{9}$$
$$V_y=\{w_i \mid w_i \in (H^1(Y_1))^3,\ Y\text{-periodic},$$
$$w_i=0 \text{ on } \Gamma,\ \frac{\partial w_i}{\partial y_i}=0\}$$

The averaged heat conductivity is found as:

$$\chi_{ij}^H=\frac{1}{|Y|}\int_Y\left(\chi_{ij}+\chi_{ik}\frac{\partial \Psi^j}{\partial y_k}\right)dY \tag{10}$$

where Ψ^j is the solution of

$$\Psi^j \in W_y\colon \int_Y \frac{\partial \tau}{\partial y_i}\chi_{ik}\frac{\partial \Psi^j}{\partial y_k}dY=-\int_Y\frac{\partial \tau}{\partial y_i}\chi_{ij}dY,$$
$$\forall \tau \in W_y,\ W_y=\{w \mid w \in (H^1(Y))^3,\ Y\text{-periodie}\} \tag{11}$$

4. Numerical Formulation

4.1 Microscopic flow equations-flow parameter equations

The microscopic flow Eqs. (9), (11) are solved first and the solutions are used to evaluate the permeability K_{ij} and the conductivity χ_{ij} in (8) and (10). Using K_{ij} and χ_{ij}, the averaged flow equations are solved in the mold flow simulation. The numerical formulation of (9) and (11) poses no difficulty and is done using the conventional finite element formulation.

4.2 Averaged flow equations-mold flow simulation equations

The numerical formulation of (2) ~ (4) can not be easily done using the conventional finite element method due to the presence of the transient moving boundary where the resin is displacing air in the mold cavity. Since the flow domain is changing its shape constantly and significantly throughout the mold filling process, it is necessary to redefine the flow domain after each time step. If the conventional finite element method is used, one has to develop a scheme to keep the flow front inside the mold cavity and use it to modifiy/ generate the mesh at each time step. Then, the mesh modification/generation process becomes the most time consuming part of the simulation. Moreover, it introduces interpolation errors. Here, we use a modified version of the FE/CV method which uses one initial grid throughout the simulation and provides the flow front as a part of the solution.

4.2.1 Finite element/control volume method

FE/CV method is based on the "conservation of mass principle" applied to each control volume. In the method, the mold cavity is first divided into eight node hexagonal elements. The centroids of each elements are joined to the center of each faces and to the mid points of each edges, creating a polygonal control volume that surround each vertex node. This is shown in Fig. 3. Within each control volume, incoming flux is calculated using tri-linear shape functions and the mass balance is enforced. This yields a set of linear equations which is the discretized version of the governing equations.

To keep track of the moving flow front, an additional parameter ϑ is introduced at each control volume or equivalently at each node. ϑ is the fill factor which represents the volume fraction of resin within the control volume. It is set to zero if a control volume is empty, set to one if the volume is completely filled with resin. At each time step, the new flow front is found by updating fill factors using the flux data and the time step.

4.2.2 Overfill and redistribute scheme

In FE/CV method, the transient mold filling process is regarded as a succession of quasi -steady states. At each time step, velocity field (or the flux field) is calculated and used to update the fill factors. The method allows only one control volume to be completely filled up at each time step. This feature ensures the stability of the method by limiting the time step to a small value. However, the feature gives rise to a problem in the three dimensional simulation. As the number of meshes become larger, the number of time steps required to complete the simulation grows as well as the sized of the discretized matrix equations. Since the matrix equations must be solved at each time step, the amount of computation increases exponentially.

To reduce the computational cost, "overfill and redistribute scheme" is introduced to the FE/CV method. In the scheme, elements are, first, allowed to be overfilled so that multiple control volumes are filled up or overfilled at a time. Then, the overflow is redistributed to the neighboring flow front control volumes depending on their fill factors. This allows entire flow front to advance at a time step. The scheme is based on the conservation of resin mass and is stable by itself.

In overfill and redistribute scheme, the choice of the time step is very important. It must be small enough to maintain the validity of the quasi -steady state approximation. It must be large enough to advance the flow front by as many control volumes at a time without sacrificing the mass conservation of the resin. If a time step is too large, some control volumes may become too much overfilled that the overflow can not be redistributed among their neighboring flow front control volumes. This requirement means that a larger time step can chosen as long as no more than one layer of flow front control volume becomes filled up at each time step.

4.2.3 Darcy's flow equation

FE/CV formulation of Darcy's flow Eq. (2) is found by enforcing the mass balance of the resin at each control volume. The net flux is obtained by integrating the flow rate over the control volume boundaries which are accommodated by several elements. Setting the net flux, q_a=entering a control volume a gives:

$$q_a = \sum_{i=1}^{nel} \int_{A_i} -\frac{K_{jk}}{\mu}\left(f_k - \frac{\partial N_l}{\partial x_k} P_l\right) n_j dA = 0 \quad (12)$$

where P_l is the nodal pressure, N_l is the tri-linear shape function and n_j is the outward normal vector at the boundary. *nel* is the number of the elements accommodating the control volume A and A_i is the boundary of the control volume contained in the ith element. Applying the mass balance (12) to each control volume gives a set of linear equations in terms of nodal pressures.

In (12), the mold wall boundary condition is taken care of automatically, and the flow front boundary condition is implemented using fill factor, ϑ. The control volumes with ϑ less than one is considered to be outside the flow domain, and zero pressure is assigned to the corresponding nodes. The injection port boundary condition is done by either specifying nodal pressures or introducing additional flux to (12).

In the simulation, the pressure field is found by solving Eq. (12) and is used to find the velocity field at each time step. The new flow front is determined by updating the fill factors, ϑ according to the flow front velocity and the time step. Then, the simulation code moves to the heat balance equation.

4.2.4 The heat balance equation

The discretization of the heat balance Eq. (3) is done using the FE/CV method. Since each control volume is assigned to a node lying at the center, the heat balance for each control volume yields a set of linear equations in terms of nodal temperatures. The heat balance for each control volume is:

$$\vartheta^{n+1} V_{c\nu}(a+b)(T^{n+1}-T^{n})-\varDelta\vartheta V_{c\nu}(a(T^{n}-T_0)+bT^{n})=\sum_{i=1}^{nel}\int_{A_i}\left(\chi_{jk}^{H}\left(\frac{\partial T}{\partial x_k}\right)_{A_i}^{n+0.5} n_j - \rho_r c_r \nu_j^{D} n_j T_{A_i}^{n+0.5}\right) dA \cdot \varDelta t + \vartheta^{n+0.5} V_{c\nu} b \dot{s} \cdot \varDelta t \tag{13}$$

where superscript n denotes the values at the current time step and $n+1$ denotes the values at the next time step. $V_{c\nu}$ is the volume of each control volume, T_0 is the initial temperature of fiber, and $\varDelta t$ is the time step. a and b are the thermal properties of fiber and resin, respectively and are defined as $a=\rho_f c_f \phi_f$ and $b=\rho_r c_r(1-\phi_f)$. Other variables are defined as:

$$T_{A_i}^{n+0.5}=\frac{1}{2}(T_{A_i}^{n+1}+T_{A_i}^{n}) \tag{14}$$

$$\left(\frac{\partial T}{\partial x_k}\right)_{A_i}^{n+0.5}=\frac{1}{2}\left(\left(\frac{\partial T}{\partial x_k}\right)_{A_i}^{n+1}+\left(\frac{\partial T}{\partial x_k}\right)_{A_i}^{n}\right) \tag{15}$$

A set of linear equations can be obtained in terms of nodal temperatures by applying the heat balance to each control volume.

After the nodal temperatures are found, the simulation code checks if the entire mold cavity is filled up. If not, the code calculates the viscosity field by (4), updates the time step, and goes back to solve Darcy's flow equation for the newly found flow domain. The code repeats this cycle until the mold cavity is completely filled up.

5. Iterative Solution Techniques

In the three dimensional simulation, the discretized equations often have several hundred thousand unknowns and should be solved at each time step. Since it takes hundreds of time steps or more to complete the simulation even with the overfill and redistribute scheme and a lot more time steps without, the computation cost involved is a huge one. Here, fast iterative solvers including Quasi Minimal Residual (QMR) solver and Generalized Minimal Residual (GMRES) solver are used along with the new generation of parallel super-computers to solve the equations. The solvers rely heavily on matrix vector multiplication and can easily be implemented to massive parallel super-computers. They also provide excellent scale-ability.

5.1 GMRES solver

GMRES solver belongs to the class of Krylov subspace method which solves the matrix equation $Ax=b$ by generating approximate solution x_n of the form:

$$x_n=x_0+K_n(r_0, A),\ n=1, 2, \cdots \tag{16}$$

where x_0 is the initial guess of the solution, and $r_0=b-Ax_0$ is the corresponding residual vector. $K_n(r_0, A)$ is the nth Krylov space generated by r_0 and A.

In GMRES, orthonormal bases for the Krylov subspace are generated by Arnoldi process. See Arnoldi (1951). At each iteration step, x_n is computed by solving a least squares problem to minimize the norm of the corresponding residual vector. One advantage of this method is that the least squares problem always has unique solution and thus GMRES can not break down. Moreover, since the residual norm is minimized at each step, the approximate solution x_n shows monotonic convergence behavior to the solution $A^{-1}b$ as the iteration proceeds. See Saad and Schultz (1986) for details. The biggest draw back of GMRES solver is that the work and storage requirements per iteration grow linearly with the iteration index n. In practice, one can not afford to run the full algorithm and have to use the algorithm iteratively, i.e. to restart the algorithm after certain number of iteration steps. This often leads to slower convergence.

GMRES solver is often used with preconditioners. Different preconditioners are selected for different computer architectures. For a massive parallel super-computer (TMC CM-5), simple diagonal preconditioner is found to be most efficient. More sophisticated preconditioners requires intensive inter-processor communication which is expensive in massively parallel machines. For a shared memory super-computer

(Cray C90), sophisticated preconditioners provided in CPACK are used such as the Clustered Element By Element (CEBE) preconditioner by Tezduyar et al. (1992).

CPACK is a general iterative solver package developed in 1991–1992 by Tezduyar research group at the University of Minnesota. It is based on several years of experience and research into the use of iterative solvers with stabilized implicit finite element methods. It has been written to allow researchers to use the package with only a few simple subroutine calls. The user only needs a basic knowledge of the techniques used in CPACK to effectively use the package.

5.2 QMR solver

QMR solver is a Lanczos based Krylov space method. It uses non-symmetric Lanczos algorithm to generate the bi-orthogonal base vectors. The iterates x_n are generated by solving the least squares problem applied to part of the residual norm expression-a "quasi-minimization" process. Since the least squares procedure always has a solution, and possible breakdowns in the underlying Lanczos process are prevented by the "look-ahead" scheme, the QMR solver does not breakdown except for the rare cases of incurable breakdowns. Moreover, the solver can be implemented with low and roughly constant work and storage requirements per iteration because the Lanczos algorithm requires only three term recurrences to generate the base vectors. In the simulation code, QMR solver is used with a diagonal preconditioner.

6. Parallel Implementation

6.1 Massively parallel super-computers

The discretized Eqs. (12), (13) arising from the FE/CV formulation are solved using aforementioned iterative solvers with diagonal preconditioners. The data parallel implementation of these procedures is carried out on massively parallel TMC CM-5 super-computer system using Connection Machine Fortran (CMF). One of major issues that distinguishes the implementation on massive parallel computers from that on traditional shared memory vector computer (such as Cray C90) is the non-uniform costs associated with on-processor memory references and off-processor memory references. Generally, accessing a remote processor memory requires computing resources an order of magnitude greater than accessing on-processor memory. Hence, implementations on massive parallel computers require data structure that minimizes the inter-processor data transfers while maintaining work-load balances among processors. In the actual implementation, all the data related to each element are stored in the same processor so that no inter-processor data transfer is needed to calculate the element level matrix equations. Once the element level equations are made, they are passed to the iteration solvers without being assembled to the global matrix equation.

Calculations within iteration solvers are mainly composed of dot products, SAXPY operations ($\bar{p}=\bar{p}+a\bar{q}$), matrix-vector multiplications and preconditioning steps. Since only diagonal preconditioner is used which does not involve inter-processor data transfers, the dominant portion of the solver is the matrix-vector product. The matrix-vector product is done in three steps: (a) generation of the element level vector from the global vector, (b) element level matrix-vector multiplication, (c) generation of the global vector form element level vector from step (b). This scheme requires expensive inter-processor data transfers in steps (a) and (c) which involve vector operation instead of matrix operation. To handle the data transfer, efficient scatter and gather procedures provided in CMF are used.

Since FE/CV method uses one initial grid throughout the simulation, $K_{jk}(\partial N_i/\partial x_k)$ part of the element level stiffness matrix of Eq. (12) remains constant at each time step. In the code, the redundant part is calculated once for all the elements and, at each time step, is scaled by μ to give the element level stiffness matrix. This saves lots of computations at the cost of memory huge amount of which is available on TMC CM-5.

6.2 Shared memory vector super-computer

Another simulation code is developed for the shared memory vector computer Cray C90.Since the memory is equally accessible from all processors, there is no need for special data structures. The implementation is done using Cray Fortran 77 compiler (cf77) which close to the traditional Fortran 77 with some vetorization and parallelization options. The code is optimized to achieve maximum vectorization and achieves parallelization by work-sharing. work-sharing is a process of splitting loops and code sections into tasks which are queued to processors. New tasks are assigned to processors as they finish previous tasks. In the code, most of the work-sharing is accomplished using compiler directives.

7. Numerical Examples

7.1 Performance of iteration solvers

Figure 4 shows the configuration of a rectangular mold cavity used in the performance test of iteration solvers. The resin is injected along the vertical line located at the left surface of the cavity. Initial and boundary conditions are given independent of z-coordinate in order to make the simulation a two dimensional problem. A uniform $10\times10\times10$ hexagonal mesh is used. The problem can be scaled in z-direction by changing the scale factor without any changes in nodal solution. In the test, discretized governing equations are calcultated and solved using GMRES and QMR solvers for varous scale factors when 62% of the mold cavity is filled. The solvers are used in conjunction with the diagonal preconditioner and are stopped when the norm of the residual is reduced by a factor of 0.1×10^{-5}. The GMRES solver is restarted after every ten iterations.

The test result for the discretized Darcy's equations is shown in Fig. 5.The number of equaitons is 745.The discretized equations have larger condition number which grows rapidly as the scale factor is increased. For both solvers, the performace declines as the condition number increases. The QMR solver is less affected by the increase of condition number and outperfroms GMRES solver when the condition number is large. For this reason, QMR solver is preferred when the simulation is approaching a complete fill state, at which the discretized equations becomes an indefinte system. As the simulation approaches complete fill state, the condtion number of the equations increases rapidly, and the GMRES solver becomes too slow. This is shown in Fig. 6 and Fig. 7.

In solving the discretized heat balance equations, both iteration solvers converge much faster. In addition, the performance of the solvers is less affected by the scaling. The calculation of the

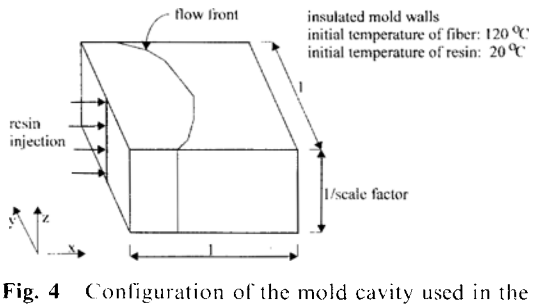
flow front
insulated mold walls
initial temperature of fiber: 120 ℃
initial temperature of resin: 20 ℃
resin injection
l
l/scale factor
l
y
z
x

Fig. 4 Configuration of the mold cavity used in the solver test.

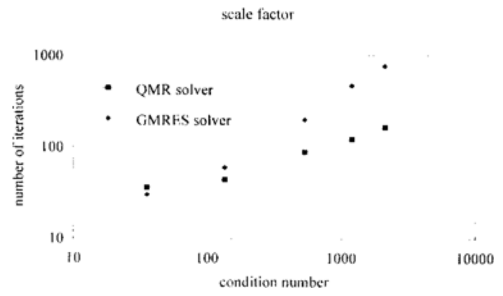

Fig. 5 Perfomance of iteration solvers for discretized Darcy's equations.

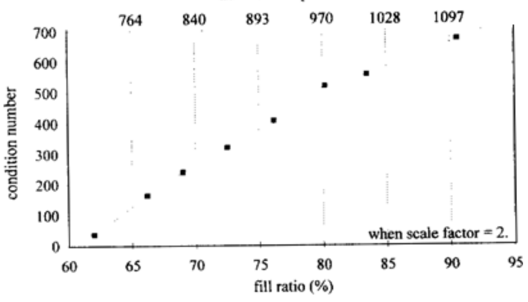

Fig. 6 Condition number vs. fill ratio.

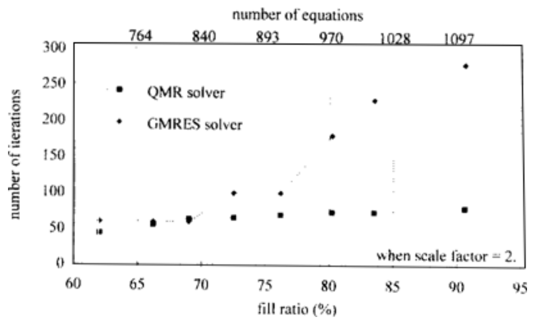

Fig. 7 Performance of iteration solvers when the simulaiton approaches complete fill state.

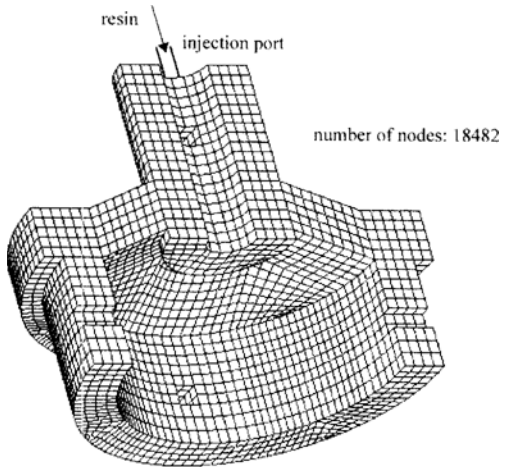

Fig. 8 Mesh of a mold cavity, symmetric half shown.

condition number shows that the equations are well conditioned (condition number of 15.05 when the number of equaions is 814). In this case, GMRES solver is found to be much faster than the QMR solver. The former reduced the residual by 0.2×10^{-11} after ten iterations where as the latter reduced the residual by 0.2×10^{-6} after sixteen iterations.

7.2 Performance of overfill and redistribute scheme

Figure 8 shows the mesh of the mold cavity used for the performance test of the overfill and redistribute scheme. The mold cavity has volume of 1.2×10^{-2} m^3 and packed with the fiber preform, whose volume fraction and isotropic permeability are 0.7 and 1.88×10^{-6} m^2, respectively. Resin is injected from the top of the mold cavity at 1.5×10^{-5} m^3/s. Its viscosity is 0.272 $Pa\cdot s$. Two

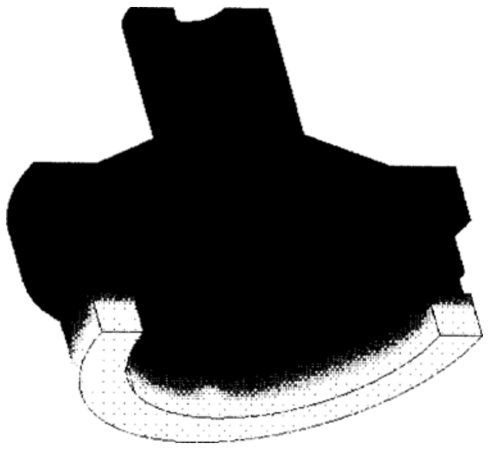

Fig. 9 Flow front profile without overfill and redistribute scheme.

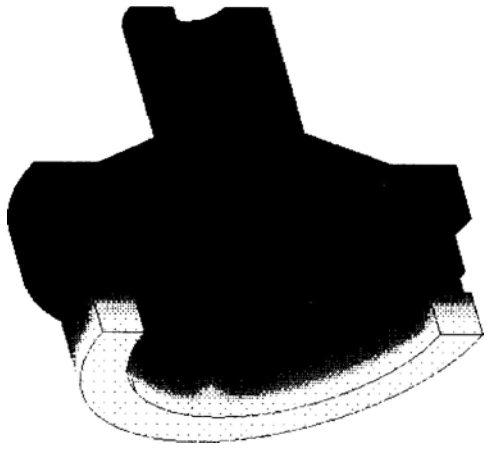

Fig. 10 Flow front profile with overfill and redistribute scheme.

mold flow simulations are performed on 64 node TMC CM-5 super-computer, one without the overfill and redistribute scheme and the other with the overfill and redistribute scheme. In both simulations, isothermal mold flow is assumed, and QMR iteration solver is used.

As a result of the simulations, the fill factor distributions which show the location of the flow front are obtained and plotted in Fig. 9 and Fig. 10. The comparison of the plots shows that both simulations provided virtually identical results. The simulation without the overfill and redistribute scheme provided a little bit smoother flow front but the difference is a subtle one. This means that the scheme have little effect on the

Table 1 Simulation costs on 64 node CM-5 supercomputer.

	Simulation without overfill and redistribute scheme	Simulation without overfill and redistribute scheme
time steps	2974	245
CPU time	156.4 minutes	13.7 minutes

1. allowing 0.5% mass inconsistency

result of the mold flow simulation.

The computational costs of the simulations are compared in Table 1.The comparison shows that the overfill and redistribute scheme significantly reduced the number of time steps and, thus, the CPU time.

In the simulation without the overfill and redistribute scheme, the control volumes are allowed to be overfilled by 0.5%. This is done in an effort to reduce the CPU time--it enables, at each time step, multiple control volumes to be filled up, at the cost of resin mass inconsistency. If the mass conservation is strictly enforced, the gap between the computational costs of the two simulations would have been larger. Note that mass conservation is strictly enforced in the simulation with overfill and redistribute scheme.

7.3 Simulation with large number of elements

A 3D mold flow simulation is carried out to demonstrate the capability of the developed codes. The mold cavity has a rectangular shape (0.6m × 0.5m × 0.4 m) with a cylidrical insert.

The configuration of the mold cavity and the mesh is shown in Fig. 11.The mesh has 112, 750 nodes and 104, 000 elements. The resin (20 °C) is injected from the injection port located on the left surface at 0.196 × 10^{-4} m^3/s. The initial temperature of the fiber preform is 75°C and the mold wall temperature is kept at 75°C. The parameters used in the simulation are:

- resin properties:
  $\rho_r = 1140$ (Kg / m^3)
  $c_{Pr} = 1900$ (J/(Kg°K))
  $\chi_r = 0.2$ (W/m°K))

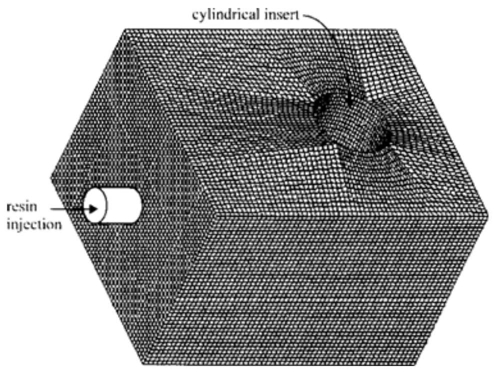

Fig. 11 Configuration of the mold cavity.

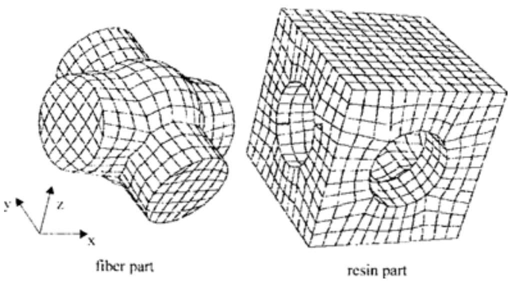

Fig. 12 Fiber preform micro-structure.

$\mu = 2.3 \exp(-0.06 \cdot T)$

- fiber preform properties:
  $\phi = 0.7$
  $\rho_f = 2500$ (Kg/m^3)
  $c_{Pr} = 900$ (J/(Kg°K))
  $k_f = 1.0$ (W/(m°K))

The permeability and the conductivity are calculated by solving the microscopic flow equations. The finite element meshes used in the calculation is shown in Fig. 12.The permeability and the conductivity are:

- permeability:

$$K = \begin{bmatrix} 0.188 & 0.000 & 0.000 \\ 0.000 & 0.188 & 0.000 \\ 0.000 & 0.000 & 0.720 \end{bmatrix} \times 10^{-6} (\mathrm{m}^2)$$

- conductivity:

$$\chi^{H} = \begin{bmatrix} 0.313 & 0.000 & 0.000 \\ 0.000 & 0.313 & 0.000 \\ 0.000 & 0.000 & 0.256 \end{bmatrix} (\mathrm{W/m°K})$$

The micro structure shown in Fig. 12 contains

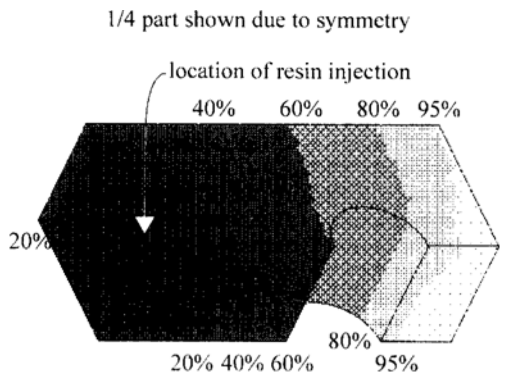

Fig. 13 Flow front locations at various fill ratio.

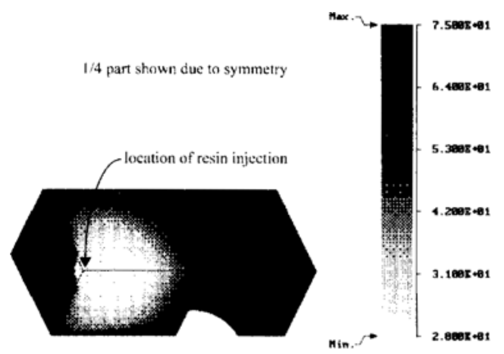

Fig. 14 Temperature distribution when the mold is 95% full.

thin gaps between each layer of the fiber preform. These gaps form continuous flow channels and result in fairly large permeability. The permeabilities of real fiber preforms are usually much smaller because they are densely packed and do not contain such continuous flow channels inside. The permeabilities obtained from the current homogenization method and from other methods are compared in Chang and Kikuchi(1995).

It should be noted that, in the simulation, many simplifications are made due to the lack of experimental data and/or for the sake of simplicity: the effect of surface tension for the flow inside the fiber bundles is neglected; change of viscosity due to the partial cure of resin is neglected; and the mold wall tempe ratures are set to constant. These effects, however, can be added without much difficulty for more accurate analysis. The current example is intended to serve as to demonstrate the strength of the analysis method rather than to give a complete analysis.

The simulations are peformed on TMC CM-5 and Cray C90 supercomputer. With the overfill and redistribute scheme, both simulations take around 700 time steps to fill the mold cavity. QMR solver with diagonal preconditioner is used. On the TMC CM-5 platform, the simulation takes approximately four hours using 256 node configuration. The memory requirement is 1.31 gigabytes. On the Cray C90, the simulation takes about one day.

In the simulation, the flow front profile and the temperature distribution are calculated at various points of the mold filling. Figure 13 shows the flow front profiles within the mold cavity. Figure 14 shows the temperature distribution when 95% of the mold cavity is impregnated. The distribution shows that the temperature is higher near the flow front than near the injection port and higher at the mold walls than at the center of mold cavity as expected, since cold resin is injected into a heated mold.

8. Concluding Remarks

The simulation in 7.3 shows that 3D simulations for large meshes have become feasible, thanks to the development of supercomputers and of fast iterative solvers. The code for massively parallel TMC CM-5 is scaleable and can solve larger problems using more cpu's. It can easily be ported to other supercomputers running standard High Performance fortran. The Cray code has limited scaleabilty due to hardware architechture (shared memory machine) but can also be ported easily to other platforms.

References

Arnoldi, W. E., 1951, "The Principle of Minimized Iteration in the Solution of the Matrix Eigenvalue Problem," *Quart. Appl. Math.*, 9, pp. 17~29.

Bruschke, M. V., and Advani, S. G., 1994, "A numerical Approach to Model Non-Isothermal Viscous Flow Through Fibrous Media with Free Surfaces," *International Journal for Numerical*

Methods in Fluids, 19, pp. 575～603.

Chang W., and Kikuchi, N., 1995, "Analysis of Non-Isothermal Mold Filling Process in Resin Transfer Molding (RTM) and Structural Reaction Injection Molding (SRIM)," *Computational Mechanics,* 16, pp. 22～35.

Coulter, J. P., and Gueri, S. I., 1988, "Resin Impregnation During the Manufacturing of Composite Materials Subject to Prescribed Injection Rate," *Journal of Reinforced Plastics and Composites, 7,* pp. 200～219.

Freund, R. W., and Nachtigal, N. M., 1991, "QMR: A Quasi Minimal Residual Method for Non-Hermitian Linear Systems," Numer. Math., 6, pp. 315～339.

Gonzalez, V. M., Castro, J. M., and Macosko, C. W., 1985, *Polymer Process Engineering,* 3, pp. 173.

Kaviany, M., 1991, *Principles of Heat Transfer in Porous Media,* Springer Verlag, pp. 15 ～42.

Kennedy, J. G., Behr, M., Karlo, V., and Tezduyar, T. E., 1994, "Implementation of Implicit Finite Element Methods for Incompressible Flows on the CM-5," *Comput. Methods Appl. Mech. Engrg.* 119, pp. 95～111.

Saad, Y., and Schultz, M. H., 1986, "GMRES: A Generalized Minimal Residual Algorithm For Solving Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.,* 7 pp. 856～869.

Tezduyar, T. E., Behr, M., Aliabadi, S. K., Mittal, S., and Ray, S., 1992, "A New Preconditioning Method for Finite Element Computations," *Comput. Methods in Appl. Mech. and Engrg.,* 99, pp. 27～42.

Trochu, F., and Gauvin, R., 1992, "Some Issues About The Numerical Simulation of Mold Filling in Resin Transfer Molding," *Advanced Composite Letters,* 1, pp. 41～43.

Young, W. B., Han, K., Fong, L. H., Lee, L. J., and Liou, M. J., 1991, "Flow Simulation in Molds With Preplaced Fiber Mats," *Polymer composites,* 12, pp. 391～403.

Young, W. B., 1994, "Three Dimensional Nonisothermal Mold Filling Simulations in Resin Transfer Molding," *Polymer Composites,* 15 118～127