The calculation of scalar transport during the injection molding of thermoset polymers

Foluso Ladeinde

SUNY at Stony Brook, Stony Brook, NY, USA

Hasan U. Akay[a]

Technalysis Incorporated, Indianapolis, IN, USA

In this paper we present a numerical scheme for the calculation of the equation governing the advection of scalars such as temperature T and cure α during the injection mold filling of thermoset polymers. A finite-element line method is presented, with variations intended to cover a variety of processing conditions. Sample calculations are presented for the Garcia[10] problem and the encapsulation of a Motorola computer chip. We also share our experience with some of the peculiar numerical difficulties associated with the simulation of injection molding for realistic systems. Some of these are related to mesh "quality," time step size selection, and (numerical) degeneracy that could result from some otherwise "physical" material models. The foregoing are issues that have not received a great deal of attention in the literature.

Keywords: transport, injection molding, thermoset polymers

1. Introduction

The process of injection molding of polymers seems to have benefited greatly from emerging computer technology. For example, with commercially available computer codes such as PLASTEC,[1] the filling simulation could predict short shots, weld lines, air trapping, overheating, the number of gates and their locations for optimum design, balancing of runners, optimization of injection pressure and clamp force requirements, calculation of pressures, temperatures, shear rate, shear stress, velocity distribution, etc. Further, postfilling processes (packing, in the case of thermoplastics) can also be simulated to provide part shrinkage and the initial state of stress (needed for subsequent structural analysis of the part).

The simulations involve the solution of the momentum equations and the transport equations for one or more scalars (depending on whether the polymer is a thermoplastic or a thermoset). For thermoplastics, simulation complications include: (a) the need for an accurate non-Newtonian characterization of the polymer melt as it goes through the molding process—viscosity as a function of shear rate, temperature, and sometimes pressure, (b) complicated geometry—thin parts arbitarily oriented in three-dimensional space which, sometimes, are combined with full three-dimensional parts, (c) moving fluid front, (d) fountain flow phenomenon at the front and, finally, (e) fiber orientation, as in Reifschneider, et al.[1] Some examples of numerical simulation of injection molding include Broyer et al.,[2] Hieber and Shen,[3] Kamal et al.,[4] Wang et al.,[5] Ladeinde et al.,[6] and Subbiah et al.[7]

Most of the computer simulation capabilities mentioned in the first paragraph have been applied to thermoplastics and, to a lesser extent, to thermosetting polymers. However, we know that thermosetting polymers are the natural choice in applications where heat resistance is needed, an example being the use of epoxy molding compound (EMC) for the encapsulation of computer chips (to protect the delicate metallic layout from the heat that is generated). Thermosets present additional difficulties for simulation, including (a) the need to model the degree of cure (to be defined later in this paper), (b) a viscosity model that must now incorporate the effect of the degree of cure, and (c) the more significant effect of fountain flow on cure development, this being a consequence of the absence of dissipative effects, particularly at the top/bottom walls of the mold. There have been previous studies on computer simulation of thermoset polymers, these being exemplified by the works of Gonzalez et al.,[8] Shen,[9] Garcia,[10] Garcia et al.,[11] Turng et al.,[12] and Nguyen et al.[13]

This paper focuses on thermosets, the filling of which involves the flow of a reactive mixture into molds. The viscosity of the reactants is initially low, to promote mixing and permit the filling of larger parts; but the

[a] Also at Indiana University–Purdue University, Indianapolis, IN, USA

Address reprint requests to Dr. Ladeinde at SUNY at Stony Brook, Stony Brook, NY 11794, USA,

Received 1 March 1993; revised 22 October 1993; accepted 29 November 1993

viscosity increases and the mixture ultimately gels and solidifies as a result of chemical cross-linkage or phase separation. For this kind of process, an optimum filling rate must be used, which should be slow enough to allow uniform (laminar) filling, but sufficiently high to prevent premature gelling. (In fact, EMC is the preferred material for microelectronic encapsulation because of its low initial viscosity, which allows it to flow over delicate lead frame and wire bonds without causing a large deformation, and its ability to undergo an additional (polymerization) reaction (during filling and postfilling), to acquire the mechanical and thermal properties that protect the electronic part from the environment.) We have observed in our work that the filling rate has a serious consequence for numerical procedures, one that is complicated by the moving boundary.

Most injection molding processes involve low Reynolds number and thin-walled parts in which the lateral dimensions (x, y) are much larger than the local thickness (z). Thus, the pressure is assumed constant along z and the Hele–Shaw approach is invoked, leading to the replacement of the momentum equations by a Poisson equation for pressure (Richardson[14] and Hieber and Shen[3]). The equation, which is locally two-dimensional (in the plane of the part), is then solved, usually with the finite-element or finite-difference method. To advance the melt front, most people use the control volume approach in Wang et al.[5]

The pressure (melt front advancement) part of injection molding simulation is cheap compared with the calculation of the transport of temperature T or the cure α. Some reasons for this include the presence of nonlinear, nonsymmetric convective operators in the scalar transport equations, the high Péclet number, the need for an accurate time integration, and the requirement for a full three-dimensional calculation in thin-walled geometries that are arbitrarily oriented in a three-dimensional space. Moreover, as discussed in Ladeinde et al.,[6] direct methods or fully explicit formulations are not desirable.

We are aware of two main approaches for calculating temperature (and that are also applicable to the equation for cure). The first is ad hoc and involves the assumption of a profile across the thickness, without solving the conservative equations. Such simplicity was necessary decades ago when computing was very primitive. For obvious reasons, this procedure is prone to errors, and we feel it should be discontinued. The second approach used to calculate temperature combines a finite difference in the thickness direction with a finite-element procedure that is used to solve the pressure equation. Thus, this is a hybrid approach. In an implementation of this approach,[3,5] terms of the temperature equation (such as $\partial^2 T/\partial z^2$ or $\mathbf{u}\cdot\nabla T$) are defined as element quantities, and element contribution to a node is assumed to depend on the volume of the element. The procedure is tedious and does not appear flexible in terms of providing variations that could be explored to optimize the calculations for a variety of processing conditions.

A contribution of this work is the presentation of a procedure that uses the finite-element method in all three coordinate directions and in a manner that obviates the foregoing difficulties. For example, we present variations of the scheme that allow the use of small or large time step sizes, as demanded by the filling rate. The usefulness of this capability will become more apparent when we discuss the crucial issue of time step size selection. Because we use the finite-element method in all three coordinate directions, we inherit the simplicity of applying complicated boundary conditions and arbitrary mesh grading in all coordinate directions. We will share our experience with some of the general difficulties associated with the simulation of injection molding for realistic systems. Some of these are related to mesh "quality," time step size selection, and the (numerical) degeneracy that could result from some otherwise "physical" material models. These issues have not received much attention in the literature, leaving the reader (or a "code-user") with the impression of "robust" and trouble-free calculations.

The mathematical formulation is presented in Section two, and the viscosity models of interest are discussed in Section three. Section four details the numerical procedure; in Section five, we discuss code validation. Concluding remarks are presented in Section six. An appendix is provided in which the elements of certain matrices are defined.

2. Mathematical formulation

2.1 Governing equations

This paper focuses on the filling stage of the injection molding of thermoset polymers in three-dimensional thin parts. The Eulerian approach to the calculation of both T and α is presented, for which the governing equations are

$$\nabla\cdot S\nabla p = 0 \tag{1}$$

$$\rho C_p\left(\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y}\right) = k\underbrace{\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)} + \frac{\partial^2 T}{\partial z^2}\Bigg) + \frac{\partial \alpha}{\partial t} Q_m + \eta\dot{\gamma}^2 \tag{2}$$

$$\frac{\partial \alpha}{\partial t} + u\frac{\partial \alpha}{\partial x} + v\frac{\partial \alpha}{\partial y} = \frac{d\alpha}{dt} \tag{3}$$

Above, S is the fluidity, which is inversely proportional to melt viscosity, p is the pressure, ρ is the melt density, C_p is the specific heat at constant pressure, T is temperature, and t is time. We also have k, which is the thermal conductivity of the melt; Q_m is heat generation from the exothermic cure reaction, η is viscosity, $\dot{\gamma}$ is strain rate, (x, y, z) are the spatial coordinates, and (u, v) are the velocity components in the local (x, y) directions. α is the degree of cure, defined as the ratio of the heat already released at time t under isothermal conditions, to the total heat that is "releasable" if the reaction is allowed to continue indefinitely, under the same conditions.

Two models for $d\alpha/dt$ have been used in our work. The first (Model 1) has been suggested by Garcia et al.[11] and

is given by

$$\frac{d\alpha}{dt} = \frac{A}{C_0} \exp\left(-\frac{E}{R_{gas}}\right)(1-\alpha)^m C_0^m \tag{4}$$

The second model (Model 2) is more common and has been shown to work very well. It is due to Kamal and Sourour[15] and has the form:

$$\frac{d\alpha}{dt} = (K_1 + K_2\alpha^{m_1})(1-C)^{m_2} \tag{5}$$

where

$$K_i = K_i(T) = a_i \exp(-E_i/T),\ i = 1, 2 \tag{6}$$

In these models, A, C_0, E, R_{gas}, m, K_1, K_2, a_1, a_2, E_1, E_2, m_1, and m_2 are model constants. Model 1 can be expressed in terms of Model 2 as shown in *Table 1*. Also shown in *Table 1* are the input values used in the present work to validate the proposed solution method and the equivalence between the heat of reaction in the two models.

2.2 Velocities and fluidity

The fluidity S is defined as

$$S = \int_0^1 \frac{b^3\phi^2}{\eta(x, y, \phi)}\, d\phi$$

for planar parts or,

$$S = \frac{\pi}{2}\int_0^1 \frac{R^4\phi^3}{\eta(x, y, \phi)}\, d\phi$$

for pipes/ducts. (For noncircular pipes/ducts R represents the equivalent radius, which is determined to give the same linear velocity as a circular pipe of radius R.) The velocities are

$$u(x, y, \phi) = -\frac{\partial p}{\partial x}\int_\phi^1 \frac{b^2\phi}{\eta(x, y, \phi)}\, d\phi,$$

$$v(x, y, \phi) = -\frac{\partial p}{\partial y}\int_\phi^1 \frac{b^2\phi}{\eta(x, y, \phi)}\, d\phi$$

Above, ϕ is the vertical distance normalized by local (half gap) thickness, b. Consistent with the procedure used to obtain equation (1), we can define depth-averaged velocities:

$$\bar{u} = -\frac{\partial p}{\partial x}\frac{S}{b},\ \bar{v} = -\frac{\partial p}{\partial y}\frac{S}{b}$$

For the thin geometries considered the strain rate can be written as

$$\dot{\gamma} = \left[\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2\right]^{1/2},\ \geq 0$$

for planar parts, whereas for ducts we approximate $\dot{\gamma}$ by

$$\dot{\gamma} = -\frac{\partial u_{axial}}{\partial r};\ \geq 0$$

(u_{axial} is the velocity component in the axial direction of pipe.) Finally, conduction in the lateral plane (shown with underbrace in equation (2)) is very small for most applications and could be neglected in the equations. Also, neglecting them is consistent with the Hele–Shaw approach used to obtain the flow equation. (An adiabatic condition must be specified at surfaces where the conduction terms are neglected.) However, we retain both capabilities (Dirichlet and adiabatic), because our procedure can be applied to regular, nonthin parts, as we show later in the discussion of code validation.

2.3 Boundary conditions

The boundary conditions for pressure are: At sidewalls and inserts we use $\partial p/\partial n = 0$, at free surface $p = 0$, and at cavity inlet we specify either $p = p(t)$ or $Q = Q(t)$, where $Q = 2\oint_C - S\nabla p \cdot \mathbf{n} dC$. For temperature, we have $T_w(x, y, z; t)$ at the top wall, $\partial T/\partial n = 0$ at midplane, $T = T_{inject}$ at cavity inlet, and $T = T_{side}$ or $\partial T/\partial n = 0$ at the side walls. The boundary conditions for the degree of cure are $\partial\alpha/\partial n = 0$ at the top, $\partial\alpha/\partial n = 0$ at midplane, $\alpha = \alpha_{inject}$ at cavity inlet, and $\partial\alpha/\partial n = 0$ at the side walls. The temperature and the degree of cure at the free surface are found by a fountain flow model that corresponds to the "simplified front" in Garcia et al.[11] with the understanding that a more accurate model is required for fast filling. A schematic of the various boundary conditions for pressure and temperature is shown in *Figure 1*.

3. Viscosity models

The applications require proper viscosity characterization for the melt, which usually implies dependence on strain rate, temperature, cure rate, and sometimes, pressure. Some materials/processes do not require more than the power law type models, whereas the "relaxational" feature of models such as the modified Cross are needed in many cases. Three viscosity models are used in our work, the first of which was introduced

Table 1. Equivalence of parameters for cure models 1 and 2

Parameter	Model 1	Model 2	Input value	Unit
1	—	m_1	0.0	—
2	m	m_2	2.0	—
3	AC_0^{m-1}	a_1	2.545×10^7	1/sec
4	E/R_{gas}	E_1	6.399×10^3	K
5	—	a_2	0.0	1/sec
6	—	E_2	0.0	K
Heat of reaction	$C_0\Delta H_r$	Q_m	2.3208×10^9	ergs/cm^3

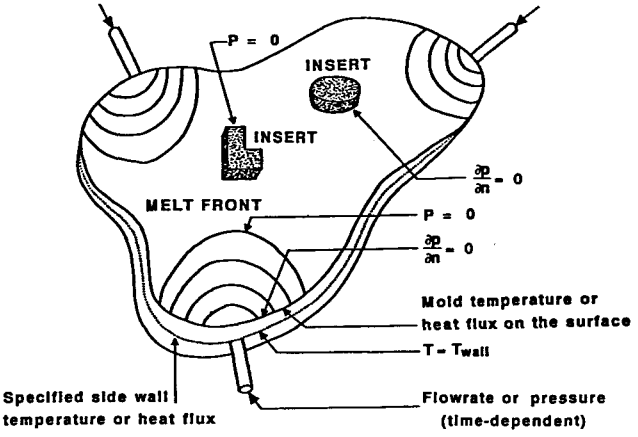

Figure 1. Schematic of the various boundary conditions for pressure and temperature.

by Castro and Macosko[16] and has the form:

$$\eta = A_\mu \exp\left[\frac{E_\mu}{R_g T}\right]\left[\frac{\alpha_{gel}}{\alpha_{gel}-\alpha}\right]^{a+b\alpha}$$

Here, A_μ, E_μ, R_g, α_{gel}, a, and b are the parameters of the model. The second viscosity that we have incorporated into our code is based on the modified Cross model:

$$\eta = \frac{\eta_0(T,P)}{1+G_2(\eta_0\dot{\gamma})^{n-1}}\left[\frac{\alpha_{gel}}{\alpha_{gel}-\alpha}\right]^{a+b\alpha}$$

The (additional) parameters η_0, G_2, and n appear in this model. Finally, we have used a model from Garcia:[10]

$$\eta = A_\mu \exp\left[\frac{1}{C+d(T-T_g)}\right]\left[\frac{\alpha_{gel}}{\alpha_{gel}-\alpha}\right]^{a+b\alpha}$$

C, d, and T_g are model parameters; T_g is the glass transition temperature, for which the form quoted in Shen[9] has been used.

4. Numerical procedure

4.1 The "lay-flat" transformation

Equations (1)–(3) are in terms of some global cartesian coordinates. However, to solve these equations for a complex (thin) part with arbitrary orientations in space, each element is transformed so that the part as a whole is "laid-flat," to reconcile coordinates (variables, fluxes, etc.) between adjacent elements. Thus, the equations are solved with x and y as the local in-plane (element) coordinates, with z in the direction of the local thickness.

4.2 Discretization

The pressures are solved using the depth-averaged Poisson equation (Hele–Shaw) procedure in Hieber and Shen,[3] in conjunction with the control volume filling approach in Wang et al.[5] We use linear, three-node triangular elements for pressure (and linear, six-node prism elements for T and α). The corresponding elements for pipes are linear, two-node, one-dimensional elements and linear, four-node axisymmetric elements, respectively. More details of the pressure solution procedure, as well as the moving front simulation, are available in the cited literature.

The procedure for calculating T or α, which we now describe, is one of the original contributions of the present work. We will discuss the T equation, with the understanding that the procedures are also completely applicable to the equation for α. To obtain nonlinear stability of the calculations we introduce the so-called streamline upwind Petrov–Galerkin method (see Hughes and Brooks[17]). The modified energy equation becomes

$$\rho C_p\left[\frac{\partial T}{\partial t}+\mathbf{u}\cdot\nabla T-\frac{1}{2}h_s v_s\frac{\mathbf{u}}{|\mathbf{u}|}\cdot\nabla\left(\mathbf{u}\cdot\nabla T-\frac{\left(\Phi+\frac{\partial\alpha}{\partial t}Q_m\right)}{\rho C_p}\right)\right]=\nabla\cdot k\nabla T+\Phi+\frac{\partial\alpha}{\partial t}Q_m \tag{7}$$

where Φ is viscous dissipation, $=\eta\dot{\gamma}^2$, h_s is a characteristic length in the flow direction, and v_s is a parameter of order one, depending on local Péclet number, Pe,

$$Pe=\frac{|\mathbf{u}|h_s}{\kappa},\qquad \kappa=\frac{k}{\rho C_p}$$

Standard finite-element procedures are not appropriate for the present applications because of the thinness of the cavity compared to the lateral dimensions. We present a finite-element line method as well as a couple of variations of the approach. To this end, we introduce an interpolation (within an element) of the form:

$$T(x,y,z)=\sum_{K=1}^{J}\sum_{Q=1}^{2}N^K(x,y)M^Q(z)T^{KQ} \tag{8}$$

where $J=2$ for ducts and 3 for triangular pressure elements, and N and M are vectors of basis functions in the (x,y) plane and z-direction, respectively. For N, we have used the standard linear basis functions for a three-node triangle, whereas for M, linear, one-dimensional basis functions of the Lagrangian type are used. A similar weighting function (that is, $N^L(x,y)M^P(z)$) is used. (Of course, N and M are identical for circular/duct parts.) The ordinary differential equation from the Galerkin procedure is integrated using either the backward Euler or the trapezoid rule. The algebraic equation from this can be written as

$$\begin{aligned}&[M^{LP}_{KQ}+\Delta t\theta_1 G^{1,LP}_{KQ}(T^{l+1}_m)+\Delta t\theta_2 G^{2,LP}_{KQ}(T^{l+1}_m)\\&\quad+N^{LP}_{KQ}(T^{l+1}_m)]\Delta T^{l+1,LP}_m\\&=\Delta t(1-\theta_3)F^{l}_{KQ}+\Delta t\theta_3 F^{l+1}_{KQ}(T^{l+1}_m)\\&\quad+[M^{LP}_{KQ}-\Delta t(1-\theta_1)G^{1,LP}_{KQ}\\&\quad-\Delta t(1-\theta_2)G^{2,LP}_{KQ}]T^{l,LP}\\&\quad-[M^{LP}_{KQ}+\Delta t\theta_1 G^{1,LP}_{KQ}(T^{l+1}_m)\\&\quad+\Delta t\theta_2 G^{2,LP}_{KQ}(T^{l+1}_m)]T^{l+1,LP}_m\end{aligned} \tag{9}$$

so that

$$T_{m+1}^{l+1,LP} = T_m^{l+1,LP} + \Delta T_m^{l+1,LP} \tag{10}$$

where m denotes iteration level within a time step and l denotes the time step. N_{KQ}^{LP} represents the Newton–Raphson correction to the regular coefficient matrix, which is zero if a scheme that is explicit in the convective terms is used. Δt^l is the time step size for step l. The θ's are the usual parameters for weighting the contribution of previous (explicit) and current (implicit) values of the terms, to give, among other schemes, the backward Euler and the trapezoid rule.

The matrices and vectors appearing in equation (9) are given in the Appendix.

4.3 Semi-implicit line method

A completely explicit solution approach to equation (9) is bound to be very expensive because of stability restriction associated with small grid size in the transverse direction. We have actually experimented with the explicit procedure but encountered either unphysical solutions or lack of convergence. A fully implicit approach is not a good alternative either as this demands too much memory and operation counts. We use an implicit scheme in the transverse direction, in the manner outlined below. To this end, we observe that equation (9) can be written as

$$\mathbf{S}\Delta\mathbf{T} = \mathbf{f} \tag{11}$$

where

$$T_i^{l+1,m+1} = T_i^{l+1,m} + \Delta T_i^{l+1,m} \tag{12}$$

In the semi-implicit procedure, **T** in the equation above refers to the array of temperatures for vertical nodes "stacked" on a pressure node. That is, we loop through pressure nodes (in the order in which the associated control volumes are filled) and solve for the temperatures at the vertical nodes stacked on a given pressure node (*Figure 2*). The procedure to assemble equation (11) is as follows:

1. Convert the (6 × 6) matrix problem for each three-dimensional temperature element to a (2 × 2) matrix problem for a one-dimensional element on a side of the three-dimensional elements. The coefficient matrices for the two problems are denoted by K_{ij}^6 and K_{ij}^2, respectively.

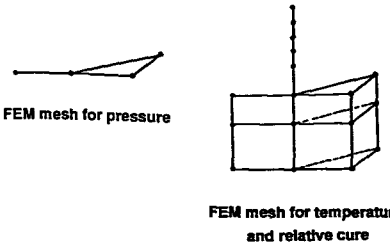

Figure 2. "Stacking" of temperature elements on the pressure elements, showing linear, one-dimensional and linear, two-dimensional pressure elements; and linear, four-node, axisymmetric and linear, six-node, prism elements for temperature.

2. Collect and assemble the (2 × 2) matrices from all lateral (three-dimensional) elements and at all layers in a stack.
3. Solve the assembled system of equations implicitly with the tridiagonal matrix algorithm (TDMA).

The conversion of the (6 × 6) three-dimensional matrix problem to the (2 × 2) one-dimensional problem could be done in such a way that, at a vertical layer, the remaining (four lateral temperature) nodes of a three-dimensional element could be treated explicitly or implicitly,[b] in the manner shown below. The advantage of this, of course, is that schemes of different stability characteristics can be produced, which could be used as demanded by the processing conditions.

4.4 Explicit contribution of neighboring nodes

When the neighboring nodes contribute explicitly to the (2 × 2) matrix problem, we use their values at the previous time level, and the one-dimensional problem for the vertical nodes becomes:

$$K_{ij}^2 \Delta T_j = F_i^2 = F_i^6 - \sum_{\lambda=\lambda_0}^{n_1} K_{i\lambda}^6 \Delta T_\lambda$$

$$i, j = 1, 2, \ldots, n_2 \tag{13}$$

Above, n_1 is the size of K_{ij}^6, which is six, and n_2 is the size of K_{ij}^2, which is two. $\lambda_0 = n_2 + 1$.

4.5 Implicit contribution of neighboring nodes

In this case we use the current values of the solution at the neighboring nodes. Because these are not known, a condensation procedure is implied.[18] In this method, a matrix system such as $\mathbf{KT} = \mathbf{F}$ is partitioned as

$$\begin{bmatrix} [K11] & [K12] \\ [K21] & [K22] \end{bmatrix} \begin{bmatrix} \{T_1\} \\ \{T_2\} \end{bmatrix} = \begin{bmatrix} \{F_1\} \\ \{F_2\} \end{bmatrix}$$

where $[K11]$, $[K12]$, $[K21]$, and $[K22]$ are themselves matrices, and $\{T_1\}$ and $\{T_2\}$ are vectors. To condense out the degrees of freedom associated with $\{T_2\}$ we have

$$[[K11] - [K12][X][K21]]\{T_1\} = \{F_1\} - [K12][X]\{F_2\}$$

where

$$[X] = [K22]^{-1}$$

Interpreted for the current problem, the coefficient matrix for the one-dimensional problem is

$$K_{ij}^2 = K_{ij}^6 - \left\{ \sum_{m=1}^{n_1-n_2} \left[\sum_{k=1}^{n_1-n_2} K_{i,n_2+k}^6 H_{km} \right] K_{m+n_2,j} \right\} \tag{14}$$

where **H** is an inverse matrix associated with the variables that have to be "condensed" out of $\mathbf{K}^6$. For the

[b] By implicit contribution of neighboring nodes we imply a solution procedure, for a node with n neighbors, that gives the same accuracy as a direct method that solves, simultaneously, for the $n + 1$ unknowns.

right-hand side of the matrix equation we have

$$F_i^2 = F_i^6 - \left\{ \sum_{m=1}^{n_1-n_2} \left[\sum_{k=1}^{n_1-n_2} K^6_{i,n_{2+k}} H_{km} \right] F_{m+n_2} \right\} \quad (15)$$

Analogous procedures are used for the circular parts of the domain. Finally, the reader will observe that the foregoing schemes are reminiscent of the successive line overrelaxation (SLOR) and the alternating direction implicit (ADI) methods in finite difference. Our semi-implicit approach differs from existing procedures in the following ways:

1. We use a finite-element interpolation in all three coordinate directions, as opposed to the hybrid approach in previous schemes. Thus, in our procedure, cell contributions to the assembled system of equations are handled in a straightforward manner, without any ad hoc treatments. Existing procedures such as those in Wang et al.[5] are very tedious, and approximations are involved that require that cell contributions be weighted on the basis of, say, cell volume. Moreover, unlike previous methods, ours has the advantage of ease of boundary condition specification at the "top" and "bottom" of the mold, while arbitrary grid spacing along z poses no special considerations.
2. We use the streamline upwind Petrov–Galerkin method to control nonlinear instability. The theory and effectiveness of this approach are well developed.[17,19,20] Upwind methods in existing procedures require, among other things, that one determine (explicitly) if a pressure node is upwind or downwind of an element (at each time step); and these have not been analyzed.
3. As we show later in this paper, our procedure allows the use of one of three variations of the schemes, with potentially different accuracy and stability characteristics.
4. The interpolation in equation (8) is not the standard one for finite elements, as equation (8) is composed of the tensor product of the basis functions in the in-plane (x, y) coordinate directions and that for variation along z. However, we need to verify the appropriateness of equation (8) for our type of problem. The heat conduction problem seems appropriate for this purpose.

5. Code validation

The filling (flow, pressure) part of the present code has been validated in Ladeinde et al.,[6] so that the focus in the present work is on the transport equation for T or α. The present code is routinely being applied to realistic industrial systems with arbitrary complex (but thin-walled) geometries, including microchip encapsulation. However, as discussed earlier, we need to show that equation (8) gives accurate interpolation. We also need to establish the accuracy of the complete code, by comparisons with published computational and experimental work. (We are not aware of an exact, closed-form solution for injection molding equations, even in a trivial geometry.) For the purpose of validation, we will discuss heat conduction in nonthin parts, and the numerical results in Garcia et al.[11] on curing in a thin-walled rectangular part. We will then present results for a simulation involving a complicated geometry, using the encapsulation of a microchip.

5.1 Three-dimensional conduction test

What is really being tested with this conduction problem is not mold filling, but the interpolation in equation (8). The equation solved is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \equiv \nabla^2 T = q = 0 \quad (16)$$

The solid is $(x, y, z) \in (0, a) \times (0, b) \times (0, c)$ with boundary conditions $T(0, y, z) = T_0$, $T(a, y, z) = T_a$, and $T(x, 0, z) = T(x, b, z) = T(x, y, 0) = T(x, y, c) = 0$. We have used $a = b = c = 1$ and $T_0 = 10$, $T_a = 5$.

The series solution for this problem is

$$\frac{16}{\pi^2} \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} \left[\frac{\{T_0 \sinh l(a-x) + T_a \sinh lx\}\{\sin[(2p+1)\pi y/b]\}\{\sin[(2q+1)\pi z/c]\}}{(2p+1)(2q+1)\sinh la} \right] \quad (17)$$

where

$$l = \frac{(2p+1)^2\pi^2}{b^2} + \frac{(2q+1)^2\pi^2}{c^2} \quad (18)$$

The series solution and the numerical solution from the present work are compared in *Table 2* for an interior region. Note that the treatment in the code of singularities at the corners and their exclusion in the series solution make the numerical solutions appear to be more accurate than the series solution for the mesh used ($10 \times 10 \times 10$ grid points, with $q_\infty = p_\infty = 50$). For this reason, it is more appropriate to compare results at interior points.

5.2 The Garcia test problem

We have solved the thermoset problem of Garcia et al.[11] which, in turn, corresponds to the conditions of experiments 6, 7, and 8 for RIM2200 described by Castro and Macosko.[16] Cure Model 1 was used, with the viscosity model of Castro and Macosko. After conversion to the CGS, which is the preferred unit in our code, the following conditions were imposed:

$C_0 = 1.840 \times 10^7$ ergs/(g K), $\rho = 1$ g/cm^3, $k = 1.7 \times 10^4$ ergs/sec cm K), $A_\mu = 1.03 \times 10^{-6}$ g/(cm sec), $E_\mu/R = 4967$ K, $\alpha_{gel} = 0.65$, $a = 1.5$, $b = 1.0$, $Q_m = 2.3208$

Table 2. Comparison of numerical calculation using our code with series solution for three-dimensional heat conduction

x	y	z	Present	Series solution
0.0	0.5	0.5	10.0	9.87
0.1	0.5	0.5	7.76	7.69
0.2	0.5	0.5	5.77	5.67
0.3	0.5	0.5	4.20	4.12
0.4	0.5	0.5	3.12	3.08
0.5	0.5	0.5	2.52	2.50
0.6	0.5	0.5	2.36	2.34
0.7	0.5	0.5	2.59	2.56
0.8	0.5	0.5	3.16	3.12
0.9	0.5	0.5	4.00	3.98
1.0	0.5	0.5	5.00	4.94

$\times 10^9$ ergs/cm^3, $a_1 = 2.545 \times 10^7$/sec, $E_1 = 6.399 \times 10^3$ K, $a_2 = 0$, $E_2 = 0$, $m_1 = 0$, $m_2 = 2$, $T_w = 338$ K, $T_{injec} = 333$ K.

Part dimension is (41.6 cm $\times$ 10.0 cm $\times$ 0.32 cm. Isothermal wall is used at the top. Linear injection rate is $U_0 = 18.6$ cm/sec.

The results presented here were obtained with a mesh consisting of (81 $\times$ 21) nodes in the (x, y) plane and (81 $\times$ 11) in the vertical (x, z) plane, but grid refinement studies have been carried out to establish grid independence, other than the dependence due the moving front (see next section).

We experimented with three variations of the vertically implicit (VI) scheme. In the first case, we use implicit contribution of neighboring nodes (VIIN, for vertically implicit, implicit neighbors), where only one interation per time step is used. In the second approach, we also used implicit neighbors, but allow multiple iterations in a time step (VIMI, vertically implicit, multiple iterations). The third case involves explicit neighbors (VIEN, vertically implicit, explicit neighbors). Because of the multiple iterations, VIMI is expected to be more stable than VIIN; and VIIN more stable than VIEN because of the implicit treatment of neighboring nodes. VIMI and, to a lesser extent, VIIN could be reserved for unusually difficult problems, for example, those involving "unfavorable" filling rate in a "specified" mesh (see below). The cost of VIMI compared to VIIN is a multiple of the number of iterations needed to obtain a convergence of the temperature or cure solution. For small problems (12 $\times$ 20 $\times$ 7 = 1680 temperature nodes) the CPU times for VIIN and VIEN are in the ratio 5:6. However, for larger problems (81 $\times$ 21 $\times$ 11 = 18,711 nodes), the relative CPU time ratio is 11:10. The explicit approach in VIEN has the tendency to increase the number of pressure iterations, and hence, to require more CPU time. However, as the problem size becomes larger the cost of condensation in VIIN becomes larger than that for extra pressure iterations in VIEN. In terms of accuracy, it seems, from our investigations, that more tests on a variety of processing conditions are needed for a more definite conclusion.

The results obtained from our calculations using VIEN agree very well with those reported by Garcia et al. In fact, we reproduced the pressure curve given by *figure 6* in their paper, and the cure level (their table 2) below and above the envelope of maximum residence time are in close agreement. Some of these results are shown in *Figures 3–7* for temperature and the degree of cure at two times during filling. The results of our calculations are consistent with the physical interpretations given in Shen,[9] in terms of the existence (absence) of boundary layer in the temperature (cure) distribution, the location of maximum cure, etc.

In *Figure 8*, we demonstrate the ability of the present approach to handle quite complicated geometries. The sample problem for this case is the encapsulation of a Motorola computer chip, the plan view of which is shown in *Figure 8*. The material is an epoxy thermoset, with the following properties:

$C_p = 1.2059 \times 10^7$ergs/(g K), $\rho = 1.82$ g/cm^3, $k = 6.6992 \times 10^4$ergs/(sec cm K), $\alpha_{gel} = 0.17$, $a = 3.496$, $b = 8.503$, $Q_m = 1.092 \times 10^9$ergs/cm^3, $a_1 = 8.475 \times 10^7$/sec, $E_1 = 7.216 \times 10^3$K, $a_2 = 9.716 \times 10^6$/sec, $E_2 = 8.585 \times 10^3$K, $m_1 = 0.7241$, $m_2 = 1.234$, $T_w = 448$ K, $T_{injec} = 393$ K.

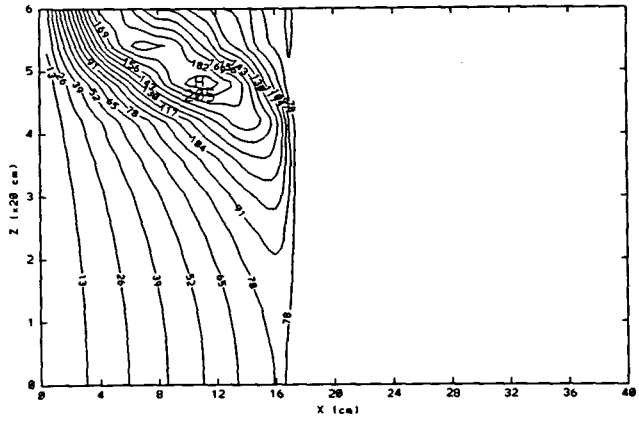

Figure 3. Cure level for the Garcia test problem at $t = 1.11$. The actual values are $\frac{1}{1000}$th of the values shown on the contours. Picture is for the vertical plane (x, z). Note that for *Figures 3–6* and *9* the dimensions of the part in the (x, y) directions are 41.6 cm $\times$ 0.32 cm; the z (thickness) direction has been magnified relative to x.

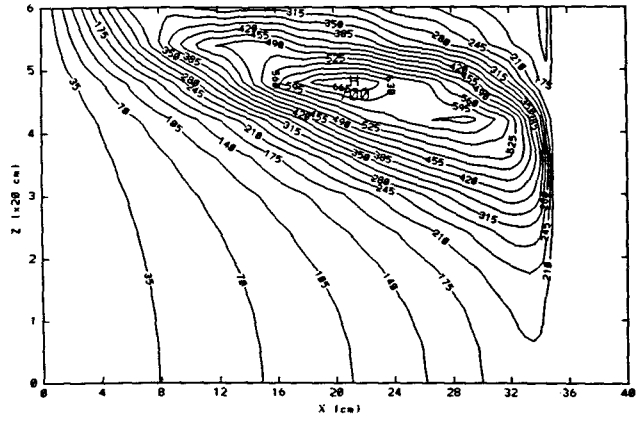

Figure 4. Cure level for the Garcia test problem at $t = 2.2$. The actual values are $\frac{1}{1000}$th of the values shown on the contours. Picture is for the vertical plane (x, z).

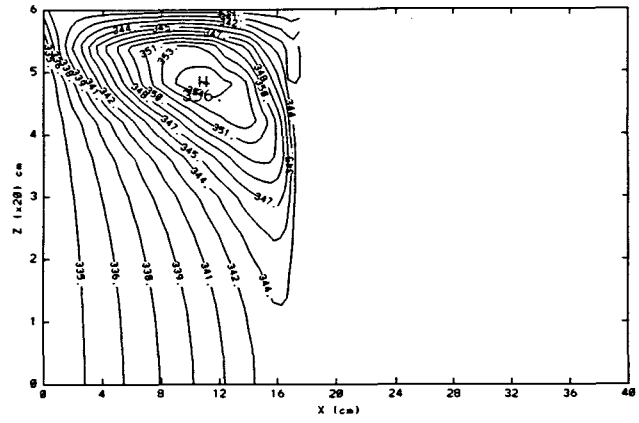

Figure 5. Temperature distribution for the Garcia test problem at $t = 1.11$. Contour levels range from 333 to 354. Picture is for the vertical plane (x, z).

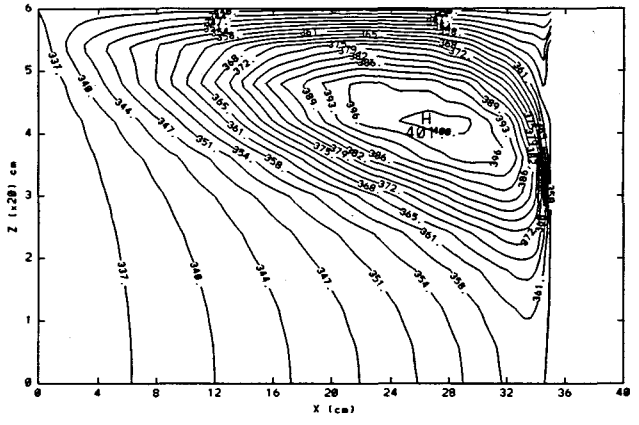

Figure 6. Temperature distribution for the Garcia test problem at $t = 2.2$. Contour levels range from 333 to 400. Picture is for the vertical plane (x, z).

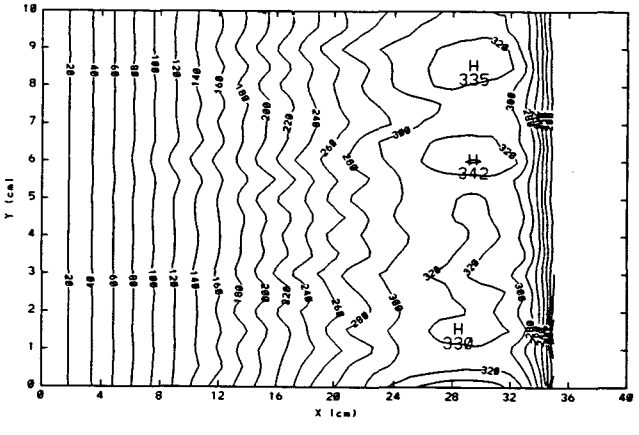

Figure 7. Depth-averaged cure distribution for the (x, y) plane at $t = 2.2$. The actual values are $\frac{1}{1000}$th of the values shown on the contours. The depth-averaged temperature shows a similar distribution, with a range of 333 to 372. Note that the dimensions of the part in the (x, y) directions are 41.6 cm × 10.0 cm; the y direction has been magnified relative to x.

The modified Cross cure model is used with G_2 and η_0 defined as τ^{n-1} and $B \exp(T_b/T)$, respectively. Additional parameters are

$$B = 4.16 \text{ poise}, \quad T_b = 2091K, \quad n = 0.28, \text{ and}$$
$$\tau = 26310 \text{ dynes/cm}^2.$$

The injection region is shown with an arrow in *Figure 8(b)*. The thickness of the part varies from 0.14 to 0.867 cm, depending on location in the lateral plane. Injection rate is such that the part fills in 6 sec. The computational mesh is shown in *Figure 8(a)*; it consists of 2326 pressure nodes, 4080 pressure elements, $2326 \times 11 = 25{,}586$ temperature (cure) nodes, and $4080 \times 10 = 40{,}800$ temperature (cure) elements. The filling simulation is almost exact (5.97 sec vs. the exact value of 6.0 sec). A contour map of depth-averaged temperature is provided in *Figure 8(b)*. Based on measurements from physical experiments at Motorola the computed results are considered to be encouraging. Further work on this problem involves the characterization of the type of boundary conditions that should be used in the code.

Finally, for the Garcia problem, we show, in *Figure 9*, the calculations at $t = 2.08$ sec using VIIN and VIMI. The results, which are similar for the two approaches, are not as accurate as those obtained with VIEN. In fact, the results with VIIN and VIMI resemble those that we have observed for thermoplastics. This is even more so if we observe in these contour maps that the cure level is much lower for VIIN and VIMI compared with VIEN. As of now we cannot explain these differences satisfactorily, but they seem to be related to accumulated truncation errors, as VIMI and VIIN involve many more operations. (A single-precision arithmetic was used for these calculations.) Investigation is continuing, as VIMI and VIIN are of interest from the standpoint of stability.

6. Concluding remarks

We wish to discuss some of our observations that we believe might be helpful to others. The first concerns the relative sensitivity of the filling pattern (and hence the pressure solution) to the grid. Element aspect ratio and grid refinement have significant effects on the control volume approach to simulate filling. The departure of "pressure elements" from equilateral (square, for rectangular elements) causes filling error. In a run in which we (naively) used a grid of $21 \times 21 \times 7$ points for the Garcia test we observed a filling time of over 10 sec, and of about 4 sec with a $41 \times 41 \times 7$ grid. (The expected fill time is 2.29 sec.) These grids produce very "thin" pressure elements, if it is recalled that part dimension in (x, y) is (41.6 cm × 10.0.cm). The possible reasons for the poor performance with aspect ratio of about two or greater can be found in the following features of the filling algorithm. The algorithm does not allow the filling, within a time step, of more than one of those control volumes that are located on the particle path of the flowing melt. This is independent of the flow rate, volume of control volume, and time step size Δt.

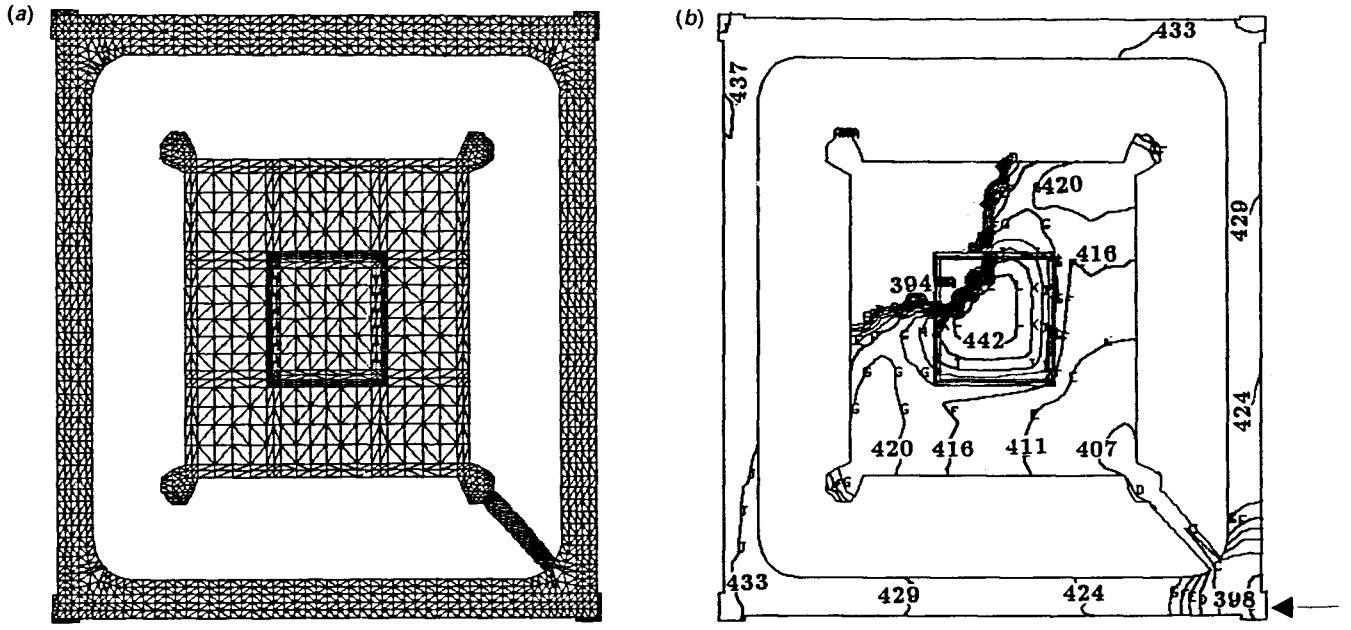

Figure 8. Encapsulation of a Motorola computer chip with thermoset plastics. The computational grip is shown in (*a*), and a contour map of depth-averaged temperature is shown in (*b*).

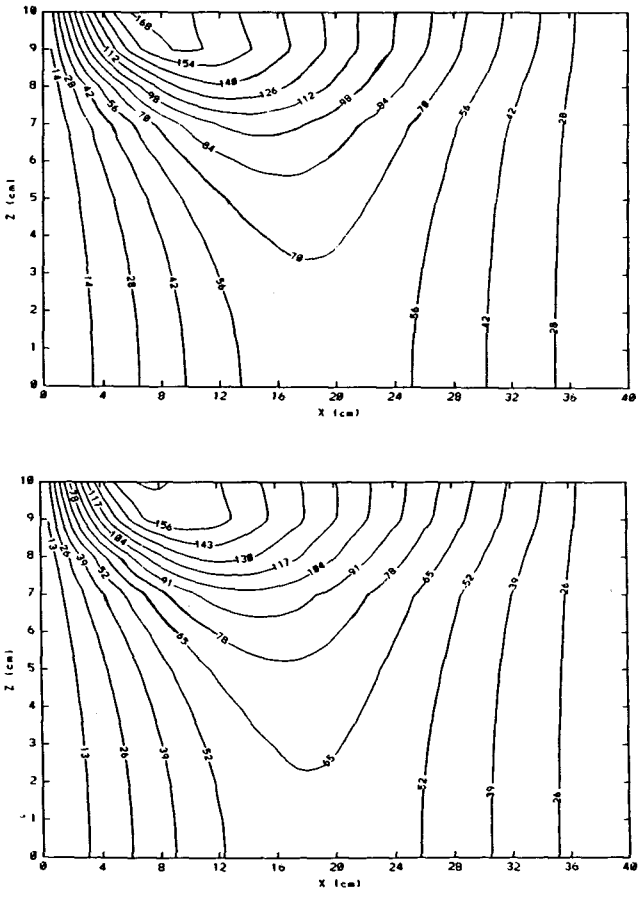

Figure 9. Performance of VIMI and VIIN for the Garcia problem, showing cure in the (x, y) plane at $t = 2.08$. The actual values are $\frac{1}{1000}$th of the values shown on the contours. More details are given in the text.

Further, we do not known the specific location of fluid within a control volume, only the fractional fill is known. With this method, a control volume is considered *filled* when the volume of fluid inside it becomes approximately equal to the volume of the control volume. Finally, because Δt for the current time step is chosen based on the previous time step, there is no way of ensuring that a control volume will *just* fill at the end of the current time step. From the foregoing, we expect that geometry and changes in flow direction will play an important role in filling simulation, including the effect of Δt, as described in the next paragraph. The Garcia results were obtained with a mesh of (81 × 21) nodes in the (x, y) plane, with seven or 11 nodes in the z-direction. This grid is sufficiently fine, with an element aspect ratio of one. The filling time agrees with the expected value.

Another observation is that time step size selection is crucial for accurate calculations, using the control volume approach. It might be necessary in many cases to allow the code to select time step Δt (which requires that at least a control volume be filled per time step), although the mesh must be sufficiently fine to do this. For a coarse mesh a user input time step size might be more appropriate. The speed of signal propagation (injection rate) *must* be used to estimate Δt_{input}, not unlike the Courant number limitation in high-speed flows. This Δt will be estimated from $\Delta x > \Delta t \times U_0$ where Δx is a "typical" mesh size and U_0 is signal speed. Of course, we are now immediately reminded of difficulties that could arise from unnecessary refinement—smaller Δt means more expensive (unnecessary) calculations, with the attendant accumulation of the error associated with filling. As an illustration of some of the foregoing, using 81 × 21 × 7, we experimented with (a) $\Delta t_{input} = 0.01$, (b) $\Delta t_{input} = 0.0267$ and, for case (c), we allowed the code to

select Δt. We also experimented with a fourth case, (d), using a mesh of $121 \times 31 \times 7$, allowing the code to calculate Δt. Both of these grids have an aspect ratio of 1. All tests used isothermal, Newtonian viscosity (which is quite appropriate for filling time but not for pressure values). The total number of time steps required for filling, the fill time, and the CPU time for the four cases (including the time for profiling, which could account for up to half of the total) are, respectively, (359, 1.79, 635), (230, 2.52, 478), (230, 2.49, 479), and (349, 2.53, 1675) units. Note that a Δt of 0.0267 corresponds to that selected from $U_0(=18.6)$ and $\Delta x(=41.6/80 \approx 0.5)$. The use of Δx in this manner once more reminds us of the need for a uniform mesh in (x, y). In conclusion, we want the code to select Δt and make use of the injection rate and mesh dimensions if Δt must be defined from input. Also, "unguided" mesh refinement may be very undesirable!

The final observation that we wish to share pertains to the "numerical degeneracy" that could result from some otherwise "physical" material models. Prior to the convergence of an iterative scheme, the quality of the solution is sometimes not very good, and is not required to be, except the Δt is "small" for the scheme. The intermediate solutions sometimes violate the physics of the problem and, as an example, could lead to a situation where $\alpha > \alpha_g$, which causes overflow in the viscosity models. To remedy this problem, Δt must be small (depending on the scheme) so as to nullify the need for iteration or, if iteration must be done, one could impose bounds on some critical variables during the iterations. Of course, the converged results must be physically correct.

Acknowledgment

The work of the first author was supported by a Grant from Technalysis Inc., Indianapolis, IN, USA. This support is gratefully acknowledged. We had very useful discussions with Professor Alan Kushner, for which we are appreciative. The code described in this work has been used to simulate encapsulation of a Motorola computer chip. We are grateful to Motorola Corporation for giving us the permission to discuss the results. Finally, the comments and recommendations of the referees were quite useful.

References

1 Reifschneider, L. G., Akay, H. U., and Ladeinde, F. Fiber orientation prediction for three-dimensional injection/compression molded parts. *Computer Modeling and Simulation of Manufacturing Processes*, eds. B. Singh, Y. T. Im, I. Haque, and C. Altan, American Society of Mechanical Engineers. MD-VOL. 20, PED-VOL. 48, 1990

2 Broyer, E., Tadmor, Z., and Gutfinger, C. Filling of a rectangular channel with a Newtonian fluid. *Israel J. Tech.* 1973, **11**(4), 189–193

3 Hiber, C. A. and Shen, S. F. A finite element difference simulation of the injection-molding filling process. *J. Non-Newtonian Fluid Mech.* 1980, **7**, 1–32

4 Kamal, M. R., Goyal, S. K., and Chu, E. Simulation of injection mold filling of viscoelastic polymer with fountain flow. *AICheE J.* **34**(1), 94–106

5 Wang, V. W., Hieber, C. A., and Wang, K. K. Dynamic simulation and graphics for the injection molding of three-dimensional thin parts. *J. Polymer Eng.* **7**(1), 21–45

6 Ladeinde, F., Akay, H. U., and Lynch, W. C. A new formulation for finite element solution of non-isothermal processes in injection/compression molding. *ANTEC '90*, 1989, 1727–1730

7 Subbiah, S., Trafford, D. L., and Güçeri, S. I. Non-isothermal flow of polymers into two-dimensional, thin cavity molds: a numerical grid generation approach. *Int. J. Heat Mass Transfer*, 1989, **32**(3), 415–434

8 Gonzalez, U. F., Shen, S. F., and Cohen, C. Problems of simulating the processing of thermoset polymers. *4th Annual Mtg. Polymer Processsing Soc. Int.*, Orlando, FL, 1988

9 Shen, S. F. Simulation of the processing of thermoset polymers. *Int. J. Numer. Meth. Eng.* 1990, **30**, 1633–1647.

10 Garcia, M. A. Reactive mold filling modeling. Ph.D. Thesis, University of Minnesota, USA, 1991

11 Garcia, M. A., Macosko, C. W., Subbiah, S., and Guceri, S. I. Modeling of reactive filling in complex cavities. *Intern. Polymer Processing* 1991, **VI**(1), 73–82

12 Turng, L. S. and Wang, V. W. On the simulation of microelectronic encapsulation with epoxy molding compound. *Soc. Polymer Eng.* RETEC, Nov. 10–12, 1991, 92–110

13 Nguyen, L. T., Danker, A., Santhiran, N., and Shervin, C. R. Flow modelling of wire sweep during molding of integrated circuits. *ASME Winter Annual Meeting*, Anaheim, CA, Nov. 8–13, 1992

14 Richardson, S. Hele-Shaw flows with a free boundary produced by the injection of fluid into a narrow channel. *J. Fluid Mech.* 1972, **56**(4), 609–618

15 Kamal, M. R. and Sourour, S. Kinetics and thermal characterization of thermoset cure. *Polymer Eng. Sci.* 1973, **13**, 59–64

16 Castro, J. M. and Macosko, C. W. Studies of mold filling and curing in reaction injection molding process. *AIChE J.* 1982, **28**, 250–260

17 Hughes, T. J. R. and Brooks, A. A theoretical framework for Petrov–Galerkin methods with discontinuous weighting functions: application to the streamline-upwind procedure. *Finite Elements in Fluids*, Vol. 4, eds. R. H. Gallagher, D. H. Norrie, J. T. Oden, and O. C. Zienkiwicz, 1982

18 Zienkiwicz, O. C. *The Finite Element Method*, 3rd ed. McGraw-Hill Book Co. (UK) Ltd., London, 1977

19 Johnson, C. and Saranan, J. Streamline diffusion method for the incompressible Euler and Navier–Stokes equations. *Mathematics of Computation* 1986, **47**, 1–18

20 Johnson, C. and Szepessy, A. On the convergence of a finite element method for a nonlinear hyperbolic conservation law. *Math. Comput.* 1987, **49**, 427–444

21 Macosko, C. W. *Fundamentals of Reaction Injection Molding.* Hanser, New York, 1989

22 Kamal, M. R. and Ryan, M. E. Thermoset injection molding. *Injection and Compression Molding Fundamentals*, A. I. Isayev, Marcel Dekker, New York, 1987, Chapter 4

APPENDIX

In this appendix the matrices and vectors appearing in the equation are defined as follows:

$$M_{KQ}^{LP} = \rho C_p \int_\Omega N^K(x, y)\left[\int_z M^Q(z)M^P(z)\,dz\right]N^L(x, y)\,dx\,dy$$

$$G_{KQ}^{1,LP} = \rho C_p \int_\Omega N^K(x, y)\left[\int_z M^Q(z)M^P(z)\,dz\right]\left\{\bar{u}\frac{\partial N^L(x, y)}{\partial x} + \bar{v}\frac{\partial N^L(x, y)}{\partial y}\right\}dx\,dy$$

$$G_{KQ}^{2,LP} = \int_\Omega \frac{\partial N^K(x, y)}{\partial x_j}\left[\int_z M^Q(z)M^P(z)\,dz\right]\frac{\partial N^L(x, y)}{\partial x_j}\,dx\,dy + \int_\Omega N^K(x, y)\left[\int_z \frac{\partial M^Q(z)}{\partial z}\,k\,\frac{\partial M^P(z)}{\partial z}\right]N^L(x, y)\,dx\,dy$$
$$+ \frac{1}{2}\frac{h_s v_s}{|\mathbf{u}|}\int_\Omega \left\{\bar{u}\frac{\partial N^K(x, y)}{\partial x} + \bar{v}\frac{\partial N^K(x, y)}{\partial y}\right\}\left[\int_z M^Q(z)M^P(z)\,dz\right]\left\{\bar{u}\frac{\partial N^L(x, y)}{\partial x} + \bar{v}\frac{\partial N^L(x, y)}{\partial y}\right\}dx\,dy$$

The forces are

$$F_{KQ} = \bar{u}_j\frac{1}{2}\frac{h_s v_s}{|\mathbf{u}|}\int_\Omega \frac{\partial N^K(x, y)}{\partial x_j}\left[\int_z M^Q(z)\left\{\left(\Phi + \frac{\partial \alpha}{\partial t}Q_m\right)\right\}dz\right]dx\,dy + \int_\Omega N^K(x, y)\left[\int_z M^Q(z)\left\{\Phi + \frac{d\alpha}{dt}Q_m\right\}dz\right]dx\,dy$$
$$+ \int_S N^K(x, y)\left[\int_z M^Q(z)k\left\{\frac{\partial T}{\partial x}(z)\eta_x(z) + \frac{\partial T}{\partial y}(z)\eta_y(z)\right\}dz\right]dA_\Omega$$
$$+ \int_\Omega N^K(x, y)\left[M^Q(z)k\frac{\partial T}{\partial z}(x, y)\eta_z(x, y)\right]_{z=0}^{z=b}dx\,dy$$

The velocities $\bar{u}$ and $\bar{v}$ that appear in these equations are the arithmetic averages of the nodal values for an element, whereas η_x, η_y, and η_z are components of the surface normals. We assumed, in the foregoing, that there is no convective flux of heat out of the boundary in the (x, y) plane, other than at the gate where the condition of specified temperature is used. However, we allow specified temperature or heat flux at the lateral wall, and at the bottom or top wall. dA_Ω in the above expression for F_{KQ} stands for an elemental portion of the boundary curve in the (x, y) plane.