Computational Mechanics 15 (1995) 511–520 © Springer-Verlag 1995

A procedure for advection and diffusion in thin cavities

F. Ladeinde

Abstract In this paper a finite element formulation is proposed for the calculation of advection and diffusion in a thin cavity. For these kinds of systems, very high aspect ratio elements are necessary for cost-effective simulation. Locally, element dimensions, say in x and y, are comparable, whereas the dimension in the transverse direction z is orders of magnitude smaller than those for x and y. In our formulation, the three-dimensional basis functions for interpolation are constructed as a tensor product of the basis functions that span the lateral (x, y) plane of an element and those that span the transverse direction. Unknowns along the transverse direction are solved implicitly, in a line-by-line fashion, using the tridiagonal matrix algorithm, while the "out-of-line" unknowns are treated either explicitly or semi-implicitly. Several applications to material processing are discussed, as are the computationally-intensive components of three variations of our basic procedure.

1 Introduction

Some problems of interest in engineering involve the flow of a liquid, gas, or molten plastic in very narrow cavities. These are exemplified by the flow of molten polymer in injection molding, compression molding, and in-mold coating; the flow of lubricants in narrow channels, and fluid flow in certain types of heat exchangers. The geometries involved are essentially two-dimensional, locally, but the orientation in space is arbitrary and quite complex. The casings of certain kinds of computer monitors, automobile body panels, and the handset of a telephone, are some examples in material processing. To fabricate these structures, molten plastic is forced to flow in the cavity of a mold for the parts. When geometric complexity is combined with the fact that the Reynolds number of the flows in these systems is low, the finite element method becomes very attractive.

The physics in thin cavity problems usually do not vary significantly in the local lateral plane, compared to the thickness direction. Thus, it is pertinent to pursue an approach that takes an advantage of very large element aspect ratios in order to obtain a cost-effective simulation. For this to happen, means need to be devised to circumvent the significant amount of stiffness, and hence convergence difficulties, that is introduced into the discrete equations by large element aspect ratios. An approach that accomplishes this is the subject of this paper.

The analyses of interest in this work are reminiscent of the shell theory in structural mechanics, where a form of the solution through the thickness of a shell element is assumed in place of an explicit three-dimensional discretization. In the structural case, the degrees of freedom are the three displacements (x, y, z) and three rotations associated with the changes in the curvature of the part. However, for the fluid mechanics problem of interest to us, an assumed solution in the thickness direction does not appear to be a prudent way to do things. Some reasons for this include the complications associated with the (hydrodynamic) nonlinearity of the advection terms at high Péclet or Schmidt numbers, and the rather complicated dependence of viscosity on temperature and strain rate, which vary strongly in the transverse direction.

Communicated by Y. Jaluria, 20 September 1994

Department of Mechanical Engineering,
SUNY at Stony Brook,
NY 11794-2300, USA

This work was supported by a Grant from Technalysis Incorporated, Indianapolis, Indiana, USA. This support is gratefully acknowledged.

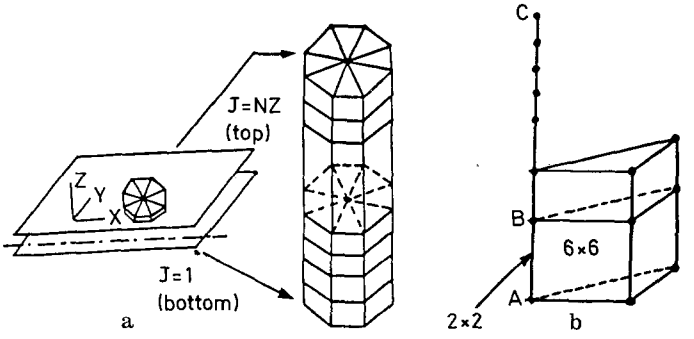

Fig. 1. Figure 1(a) shows the way three-dimensional elements are stacked in thin cavity formulations. Here, the base elements are triangles while the three-dimensional elements are prisms. Figure 1(b) shows two typical three-dimensional elements and the one-dimensional elements in the thickness direction. The one-dimensional elements are edges of the three-dimensional elements

The approach we propose starts out with shell elements, by way of mesh generation but, internally, the code stacks three-dimensional elements on top of the base elements (Fig. 1 (a)). We use a separate set of basis functions for the lateral (x, y) plane and the thickness (z) direction, with a tensor product of the two sets constituting the basis functions for an element. However, even with this, and unlike shell theory where the bandwidth is determined by grid topology in the (x, y) plane, the bandwidth of our equations will still be governed by three-dimensional elements, rendering the procedure quite unattractive for realistic engineering systems. To remedy this, a line method is proposed in this paper, in which the unknowns in the thickness direction are solved implicitly, using the tridiagonal matrix algorithm. Various treatments of the "out-of-line" nodes are possible, some of which are discussed critically in this paper. Several applications to material processing are discussed, as are the computationally-intensive components of three variations of our basic procedure.

2
Numerical procedure

The generic form of the equations of interest in this paper is

$$\frac{\partial\phi}{\partial t}+u\frac{\partial\phi}{\partial x}+v\frac{\partial\phi}{\partial y}=\kappa\left(\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}+\frac{\partial^2\phi}{\partial z^2}\right)+Q \tag{1}$$

Equations such as this arise from some conservation laws which, in this case, states that the rate of change with time t of a quantity ϕ at a fixed point (Eulerian) is equal to the net advection of ϕ away from the fixed point, the net molecular diffusion of ϕ to the fixed point, and the volumetric rate of generation Q of ϕ at the fixed point. u and v are the components of velocity, which advect ϕ in the x- and y- coordinate directions respectively, and κ is the diffusivity for molecular diffusion of ϕ. For now, we will assume that velocities u and v are prescribed; the way to obtain them depends on the application, as we discuss later in this paper. It should be noted that multiple equations such as the complete set of equations governing fluid flow under non-isothermal conditions can, in principle, be solved using our scheme, especially when combined with the segregated approah to solution. Some applications are discussed later in this paper in which two equations of the form in Eq. (1) and some other equations are solved.

We will now discuss our procedure for calculating ϕ. First, to obtain nonlinear stability of the calculations we introduce the so-called streamline upwind Petrov-Galerkin method (see Hughes and Brooks (1982)), application of which gives the modified equation:

$$\left[\frac{\partial\phi}{\partial t}+\mathbf{u}\cdot\nabla\phi-\frac{1}{2}h_s v_s\frac{\mathbf{u}}{|\mathbf{u}|}\cdot\nabla(\mathbf{u}\cdot\nabla\phi-Q)\right]=\nabla\cdot\kappa\nabla\phi+Q, \tag{2}$$

where 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,

$$Pe=\frac{|\mathbf{u}|h_s}{\kappa}.$$

We introduce a nonstandard finite element interpolation of the form

$$\phi(x,y,z,t)=\sum_{j=1}^{3}\sum_{n=1}^{2}N_j(x,y)M_n(z)\phi_{jn}(t), \tag{3}$$

where N and M are vectors of basis functions in the (x, y) plane and z-direction, respectively. For N, we use the linear basis functions of a three-node triangle while, for M, linear, one-dimensional basis functions of the Lagrange types are used. A similar weighting function (that is, $N_i(x, y)M_k(z)$) is employed. The ordinary differential equation from the Galerkin procedure is integrated using either the backward Euler or the trapezoid rule, for which the algebraic equation can be written as

$$
\begin{aligned}
&[M_{ijkn} + \Delta t\theta_1 G^1_{ijkn}(\phi_m^{l+1}) + \Delta t\theta_2 G^2_{ijkn}(\phi_m^{l+1}) + N_{ijkn}(\phi_m^{l+1})]\,\Delta\phi^{l+1}_{m,jn} \\
&\quad = (1-\theta_3)F^l_{ik} + \theta_3 F^{l+1}_{ik}(\phi_m^{l+1}) + [M_{ijkn} - \Delta t(1-\theta_1)\,G^1_{ijkn} - \Delta t(1-\theta_2)G^2_{ijkn}]\,\phi^l_{jn} \\
&\qquad - [M_{ijkn} + \Delta t\theta_1 G^1_{ijkn}(\phi_m^{l+1}) + \Delta t\theta_2 G^2_{ijkn}(\phi_m^{l+1})]\,\phi^{l+1}_{m,jn} \qquad (4)
\end{aligned}
$$

so that

$$
\phi^{l+1}_{m+1,jn} = \phi^{l+1}_{m,jn} + \Delta\phi^{l+1}_{m,jn} \qquad (5)
$$

m denotes iteration level within a time step and l denotes the time step. N_{ijkn} represents the Newton-Raphson correction to the regular coefficient matrix, which is zero if a scheme which 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 that appear in Eq. (4) are defined below:

$$
M_{ijkn} = \int_\Omega N_i(x,y)\left[\int_z M_k(z)\,M_n(z)\,dz\right]N_j(x,y)\,dx\,dy,
$$

$$
G^1_{ijkn} = \int_\Omega N_i(x,y)\left[\int_z M_k(z)\,M_n(z)\,dz\right]\left\{\bar{u}\frac{\partial N_j(x,y)}{\partial x} + \bar{v}\frac{\partial N_j(x,y)}{\partial y}\right\}dx\,dy,
$$

$$
\begin{aligned}
G^2_{ijkn} &= \int_\Omega \frac{\partial N_i(x,y)}{\partial x_\lambda}\left[\int_z M_k(z)\,\kappa M_n(z)\,dz\right]\frac{\partial N_j(x,y)}{\partial x_\lambda}\,dx\,dy \\
&\quad + \int_\Omega N_i(x,y)\left[\int_z \frac{\partial M_k(z)}{\partial z}\,\kappa\frac{\partial M_n(z)}{\partial z}\right]N_j(x,y)\,dx\,dy \\
&\quad + \frac{1}{2}\frac{h_s v_s}{|\mathbf{u}|}\int_\Omega\left\{\bar{u}\frac{\partial N_i(x,y)}{\partial x} + \bar{v}\frac{\partial N_i(x,y)}{\partial y}\right\}\left[\int_z M_k(z)M_n(z)\,dz\right]\left\{\bar{u}\frac{\partial N_j(x,y)}{\partial x} + \bar{v}\frac{\partial N_j(x,y)}{\partial y}\right\}dx\,dy,
\end{aligned}
$$

$$
\begin{aligned}
F_{ik} &= \bar{u}_\lambda\frac{1}{2}\frac{h_s v_s}{|\mathbf{u}|}\int_\Omega N_i(x,y)\left[\int_z M_k(z)\,Q_{,\lambda}\,dz\right]dx\,dy + \int_\Omega N_i(x,y)\left[\int_z M_k(z)\,Q\,dz\right]dx\,dy \\
&\quad + \int_S N_i(x,y)\left[\int_z M_k(z)\,\kappa\left\{\frac{\partial\phi}{\partial x}(z)\eta_x(z) + \frac{\partial\phi}{\partial y}(z)\eta_y(z)\right\}dz\right]dA_\Omega \\
&\quad + \int_\Omega N_i(x,y)\left[M_k(z)\,\kappa\frac{\partial\phi}{\partial z}(x,y)\,\eta_z(x,y)\right]_{z=0}^{z=b}dx\,dy.
\end{aligned}
$$

The velocities $\bar{u}$ and $\bar{v}$ that appear in these equations are the arithmetic averages of the nodal values for an element, while η_x, η_y, and η_z are components of the surface normals. We allow specified ϕ or flux of ϕ at the lateral walls, and at the bottom or top wall. Symmetry conditions are used at the centerline of the cavity so that only half of the domain is used, with $z = 0$, b corresponding to the centerline and top of cavity respectively.

One cannot hope to solve Eq. (4) explicitly because of a stability restriction associated with small grid size in the transverse direction. Moreover, we have actually experimented with the explicit procedure but encountered either unphysical solutions or lack of convergence. A fully-implicit approach is also not a viable alternative as this demands too much memory and operation counts, in addition to the stiffness problem mentioned earlier in this paper. We use an implicit scheme in the transverse direction,

in the manner outlined below. To this end, we rewrite Eq. (4) as

$$\mathbf{Ax} = \mathbf{f} \tag{6}$$

where

$$\phi_{i,m+1}^{l+1} = \phi_{i,m}^{l+1} + x_{i,m}^{l+1} \tag{7}$$

Next, it is pointed out that Eq. (6) can be interpreted in several ways, among which are 1) as an element equation, 2) as an equation for the entire domain, and 3) as an equation for a line in the transverse direction. In this paper, we will choose the third option. The way to obtain the assembled equations for this case are described as follows:

(a) "Convert" the (6 × 6) matrix problem for each three-dimensional element to a (2 × 2) matrix problem for one-dimensional element **on a side** of the three-dimensional element (Fig. 1). The coefficient matrix for the two problems are denoted by A_{ij}^6 and A_{ij}^2, respectively. We will retain the notation x and f for the unknowns and the right-hand side vector in the new system.

(b) Collect and assemble the (2 × 2) matrices from all lateral (three-dimensional) elements having an edge on the line and at all layers in a stack. Thus, the assembled system will represent the governing discrete equations for the line ABC in Fig. 1(b).

(c) Solve the assembled system of equations implicitly with the tridiagonal matrix algorithm.

2.1
Variations of the scheme

The process of "converting" the (6 × 6) matrix A_{ij}^6 to the (2 × 2) matrix A_{ij}^2 can be done in several ways, including 1) treating the out-of-line nodes explicitly; that *i*, using their values at the previous time step. 2) treating the out-of-line nodes implicitly; that is, using their yet-to-be determined values and, 3) using the most recent values of the out-of-line nodes; that is, in the Gauss-Seidel manner. For the explicit treatment, the equation for a two-node element on line ABC in Fig. 1(b) can be written as

$$A_{ij}^2 x_j = f_i^2 = f_i^6 - \sum_{\lambda=\lambda_0}^{n_1} A_{i\lambda}^6 x_\lambda \quad i,j = 1,2,\ldots n_2 \tag{8}$$

Above, n_1 is the size of A_{ij}^6, which is six, and n_2 is the size of A_{ij}^2, which is two. $\lambda_0 = n_2 + 1$. For the implicit treatment, we use the current values of the solution at the out-of-line nodes. Since these are not known, a condensation procedure is implied, a detailed discussion of which is available on pages 155 and 156 of Huebner and Thornton (1982). When applied to the present problem, the equation for a two-node element on line ABC can be written such that

$$A_{ij}^2 = A_{ij}^6 - \left\{ \sum_{m=1}^{n_1-n_2} \left[\sum_{k=1}^{n_1-n_2} A_{i,n_2+k}^6 H_{km} \right] A_{m+n_2,j} \right\}, \tag{9}$$

with the right-hand side as

$$f_i^2 = f_i^6 - \left\{ \sum_{m=1}^{n_1-n_2} \left[\sum_{k=1}^{n_1-n_2} A_{i,n_2+k}^6 H_{km} \right] f_{m+n_2} \right\}. \tag{10}$$

Above, H is an inverse matrix associated with the variables that have to be "condensed" out of A^6.

For Gauss-Seidel treatment of out-of-line nodes, we simply use the most-recent values of the solutions at these nodes. Other options for solution are derivable from those presented above. Three of these options, which we have experimented with extensively, and will be reported in this paper, are 1) implicit out-of-line nodes with only one iteration per time step (IOL1), 2) implicit out-of-line nodes with multiple iterations per time step (IOLM), and 3) explicit treatment of out-of-line nodes (EOL). We have also experimented with the Gauss-Seidel method but found it to be unacceptably unstable. In the next section we will discuss code validation and some applications. This will be followed by a comparison of CPU time requirements for the three methods above.

The need for the present approach could be put to question, since Subbiah et al. (1989), for example, have somehow managed to solve the problem using finite difference. The motivation in our work comes from the general preference of the finite element method for structural problems and low Reynolds number flows, where quite accurate and dependable general-purpose codes are possible because of the flexibility with boundary conditions and complicated geometries. Moreover, in spite of recent advances

in numerical grid generation, it is still quite tedious using finite difference for the kinds of geometries involved in many of these applications. An alternative procedure to Subbiah's or ours, for that matter, involves a hybrid, with finite element in the lateral plane and finite difference in the transverse directions. However, this hybrid procedure is limited in the sense that some (adhoc) assumptions have to be made regarding the contribution of an element to the solution at a node (Ladeinde et al. 1989). Other advantages of our procedure can be found in the use of the Streamline Upwind Petrov-Galerkin (SUPG) Method to control nonlinear instability. The theory and effectiveness of SUPG are well developed (Hughes and Brooks 1982; Johnson and Saranan 1986; and Johnson and Szepessy 1987). Upwind methods in other procedures require, among other things, that one determine (explicitly) if a node is upwind or downwind of an element (at each time step); and these have not been analyzed. Finally, variations of the basic scheme provided here are easy to obtain because of the flexibility inherent in finite element interpolations.

3
Code validation and applications

Because the interpolation in Eq. (3) is not the standard one for finite element, the need arises to show that it works. The steady state limit of Eq. (1) in which transport of ϕ is due entirely to molecular diffusion is used, in which case, $\partial^2\phi/\partial x^2 + \partial^2\phi/\partial y^2 + \partial^2\phi/\partial z^2 \equiv \nabla^2\phi = 0$. Our calculations were tested against analytical solutions in Carslaw and Jaeger (1959) (Eq. (9) and (10) on page 178). The domain for the test is $(x, y, z) \in (0, 1) \times (0, 1) \times (0, 1)$. $\phi = 0$ on the boundary except at $x = 0, 1$, where $\phi(0, y, z) = 10$ and $\phi(1, y, z) = 5$ respectively. The grid used is $(10 \times 10 \times 10)$. The agreement was excellent. This test not only shows that the interpolation in (3) works fine for thin cavities, but for thick cavities, as well. To test the code for advection-diffusion problems, we calculated non-isothermal flow of molten plastic in a thin cavity mold for injection molding of thermoplastics. Comparisons are made with the results in Subbiah et al. (1989), who have used finite difference with numerically-generated grid. For injection molding applications, the Hele-Shaw type assumption (Hieber and Shen 1978, 1980) is invoked, which allows the replacement of the momentum equations by a Poisson equation for pressure and algebraic equations (for (u, v)) that are determinable from the pressure solution. Convection of momentum is usually negligible in most injection and compression molding problems, beause of the large melt viscosity. However, convection of temperature is quite significant (Hieber and Shen 1978, 1980; Ladeinde et al. 1989), as the Péclet number is large. Thus, the temperature distribution in injection molding can be modeled as in Eq. (1), with ϕ being temperature, and Q, volumetric heating rate. The control-volume/fraction-of-fill procedure in Wang, Hieber, and Wang (1986) is used to advance melt-front. The injection rate and temperature are 30 cm/sec and 563 K, while the mold wall temperature, part thickness, melt density, specific heat at constant pressure, and thermal conductivity are $T_w = 358$ K, $h = 0.25$ cm, $\rho = 0.94$ g/cm^3, $C_p = 2.1 \times 10^7$ ergs/(g $-$ K), and $k = 1.5 \times 10^4$ ergs/(sec.cm.K). The Carreau viscosity model is used:

$$\eta = \eta_1 \exp\left[-\frac{A_n(T - T_0)}{T_0}\right],$$

with

$$\eta_1 = \eta_\infty + (\eta_0 - \eta_\infty)[1 + (\lambda\dot{\gamma})^2]^{(n-1)/2}.$$

The model parameters are $A_n = 0.0$, $T_0 = 453$ K, $\eta_\infty = 0$, $\lambda = 1.04$, $n = .398$, and $\eta_0 = 1.4 \times 10^4$ Ns/m^2. The theoretical fill time is 2.32 seconds. The computational mesh used in our calculation is shown in Fig. 2, where the arrow in the figure indicates the injection location. The mesh consists of 1130 triangular elements and 641 nodal points in the lateral plane; 10 three-dimensional elements are used in the thickness direction.

In general, the agreement between our calculations and those of Subbiah et al. are quite good. The results for the time-dependent domain boundary compare quite well (we have a moving boundary problem), Figs. 3(a) and (b). The contour map of temperature at the centerline ($z = 0$), at the end of fill, is shown in Fig. 4 for our calculation. This also compares well with those in Subbiah et al., within the limits of the errors involved when interpolations are done from contours. We cannot show their results here because they are in color. Our calculations in Figs. 3 and 4, as well as those by Subbiah et al. were done using a Dirichlet boundary condition for the sidewall of the cavity. However, it is the opinion of some injection molding modelers (Hieber and Shen 1978) that an adiabatic condition at the sidewall is suitable. Comparison of our calculations with those obtained from two commercial packages (Professor Cengiz Altan, personal communication) showed an excellent agreement for the case of adiabatic sidewalls.

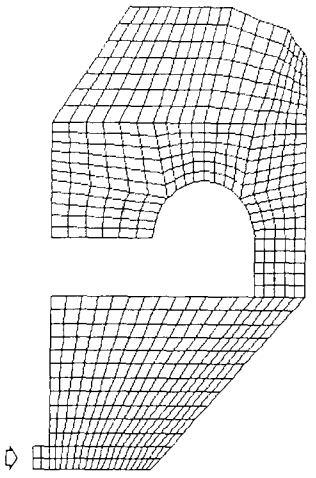

Fig. 2. The mesh for the comparison exercise with Subbiah et al. (1989). There are 1130 base elements, with 641 nodes. 10 three-dimensional elements are stacked on each base element. The arrow shows the location of the injection (gate) of fluid. This is a moving-boundary problem, in which case the flow front advances with continuous injection from the gate

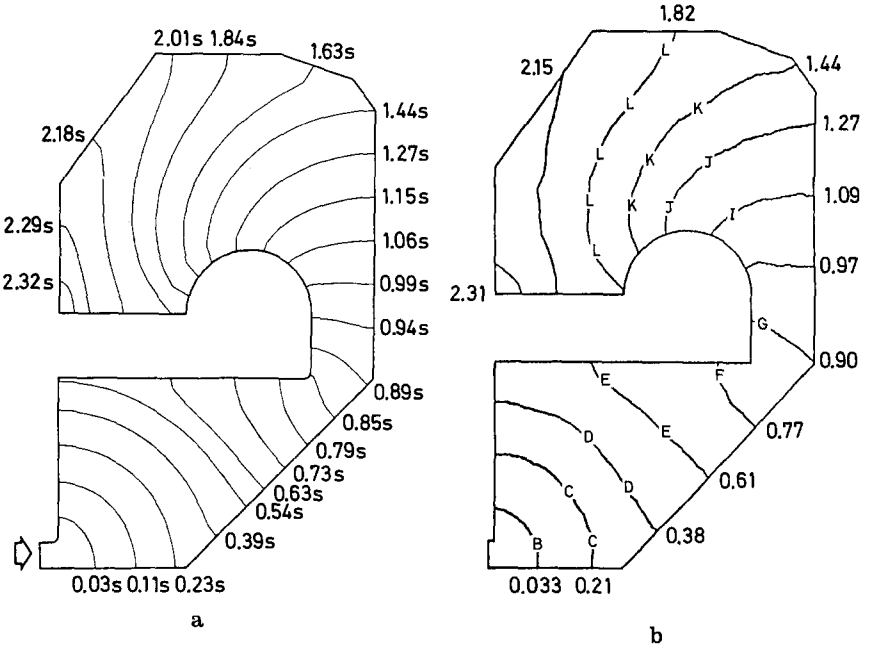

Fig. 3. Comparison of the time-dependent location of the fluid domain boundary. The results of Subbiah et al. are shown in (a) and ours in (b). The theoretical fill time, based on injection rate and mold volume, is approximately 2.32 seconds

In Figs. 5 and 7, we show the results of applying our procedure to simulate compression molding of a real truck hood, using SMC (sheet molding compound). The energy equation for this case also appears in the form of Eq. (1). The friction model of Barone and Caulk (1986) is used, along with a Hele-Shaw-like approximation. The thermophysical properties and processing conditions include a mold temperature of 366 K and a charge temperature of 300 K. The specific heat at constant pressure, density, and thermal conductivity are $C_p = 1.17 \times 10^7$ ergs/(g K), $\rho = 1.6$ g/cm^3, and $k = 3.5848 \times 10^4$ ergs/(sec cm K), while κ, the hydrodynamic friction coefficient, for the friction between charge and the mold is 1.5252×10^5g/(sec cm^2). Press speed is .254 cm/sec and charge thickness (at highlighted nodes in Fig. 5) is 1.2 cm. The desired part thickness is .125 cm at charge locations and .28 cm at other locations. The mesh consists of 1152 triangular elements (626 nodes) in the lateral plane, with a stack of 8 three-dimensional elements in the transverse direction. Depth-averaged temperature using Dirichlet and adiabatic boundary conditions at the sidewall are shown in Figs. 6 and 7. It is pointed out that Eq. (1) is in terms of some global Cartesian coordinates. However, to solve this equation for a complex (thin) part with arbitrary orientations in space (such as in Fig. 5), 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.

The last application that we wish to discuss is the modeling of injection molding of thermosetting plastics. Compared to the thermoplastics case discussed above, in thermosets we have the additional

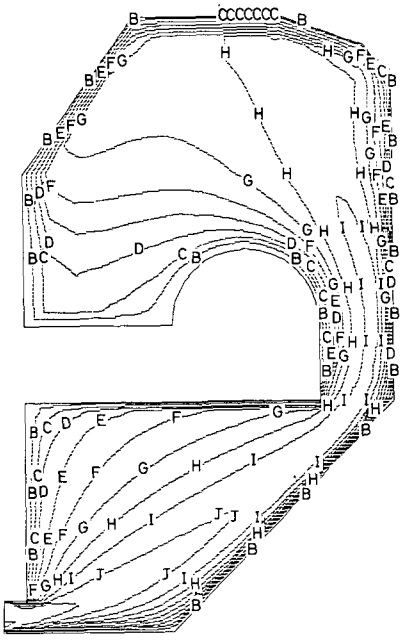

Fig. 4. The contour map of temperature at the centerline of the cavity, as observed from our calculations, using a Dirichlet boundary condition at the sidewalls

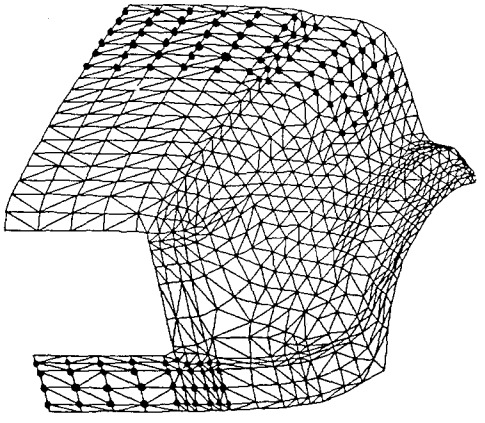

Fig. 5. The mesh for the simulation of compression mold filling of a truck hood. Two charges of sheet molding compound are used, the locations of which are shown with dots. There are 1152 base elements, with 626 nodes. 8 three-dimensional elements are stacked on each base element

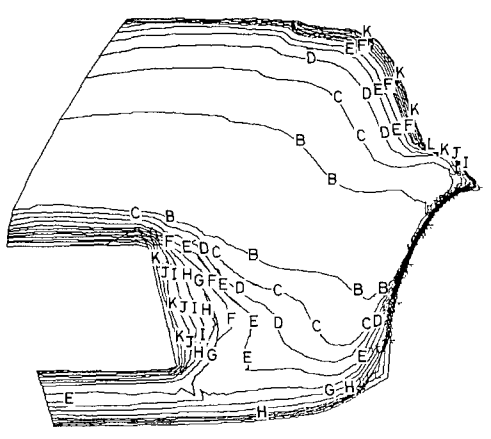

Fig. 6. The contour map of depth-averaged temperature, as observed from our calculations using a Dirichlet boundary condition at the sidewalls

transport equation for the degree of cure α, to which our procedure is applied. Details of thermoset modeling are available in Ladeinde and Akay (1994). Our calculations are compared with the measurements of Garcia et al. (1991). The model for $d\alpha/dt$ used in the test is (Ladeinde and Akay 1994)

$$\frac{d\alpha}{dt} = (K_1 + K_2 \alpha^{m_1})(1 - C)^{m_2}, \tag{11}$$

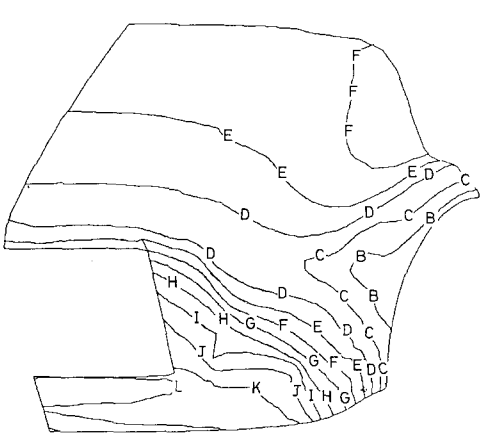

Fig. 7. The contour map of depth-averaged temperature, as observed from our calculations, using an adiabatic boundary condition at the sidewalls

with a viscosity law of the form

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

In these models, K_1, K_2, m_1, m_2, C, and A_μ, E_μ, R_g, α_{gel}, a, and b are constants, with $K_i = K_i(T) = a_i \exp(-E_i/T)$, $i = 1, 2$.

After conversion to the *CGS*, which is the prefered unit in our code, the following conditions of the Garcia problem were imposed:

$C_p = 1.840 \times 10^7$ 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, $\alpha_{gel} = 0.65$, $a = 1.5$, $b = 1.0$, $Q = 2.3208 \times 10^9\ \partial\alpha/\partial t$ (ergs/cm^3·sec), $a_1 = 2.545 \times 10^7$/ 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 × 10.0 cm × 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 of (81 × 21) nodes in the (x, y) plane and (81 × 11) in vertical (x, z) plane, although grid refinement studies were carried out to establish grid independence. Good agreement with the measurements of Garcia was observed. A sample plot of the distribution of temperature and α is shown in Fig. 8.

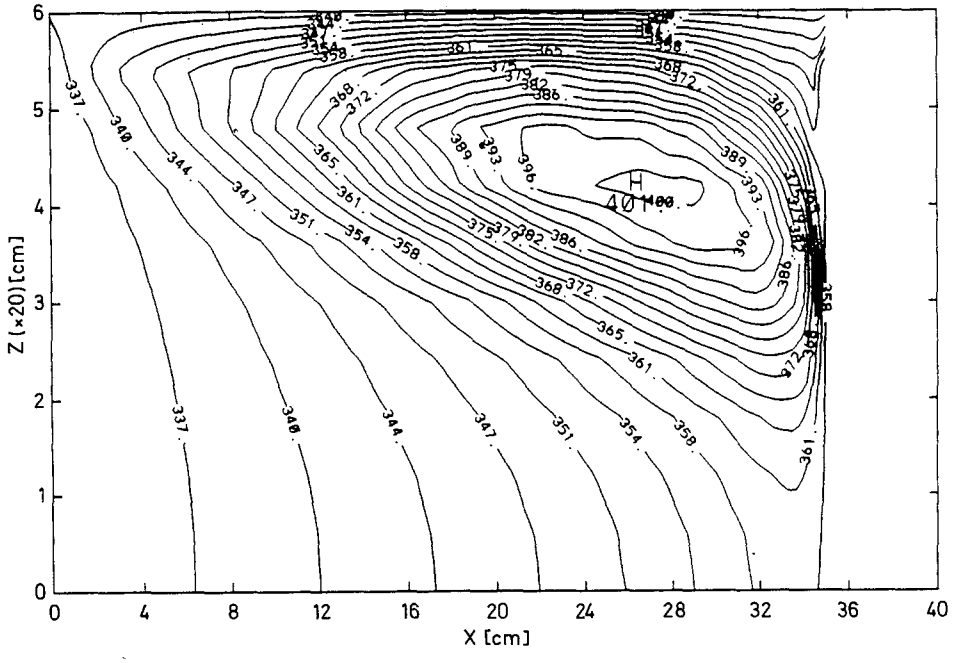

Fig. 8. The distribution of temperature for the Garcia test problem, at $t = 2.2$. Contour levels range from 333 K to 400 K. Picture is for the vertical plane (x, z). The distribution of the degree of cure α is similar to that shown here

Task	IOLM	IOL1	EOL
Managing line-by-line procedure	.0790	.0198	.0182
Calculating 6 × 6 element matrix	.7980	.1995	.2012
Some overheads in 6 × 6 → 2 × 2	.0736	.0184	.0178
Condensation procedure	.2224	.0556	–
Pressure Calculation	< .0631	.0631	.0667
Material Properties Calculation	< .0464	.0464	.0532
Velocity Calculation	< .0218	.0218	.0229
TDMA	.0059	.0014	.0015
Total	< 1.704	.4260	.381

Table 1. CPU time for the computationally-intensive components of our scheme. The unit of CPU time is seconds per time step per (three-dimensional) element.

4
Computational speed

In this section we discuss the CPU time requirement for the computationally-intensive components of our schemes, using the Garcia problem for illustration. CPU times are given in the table 1, in seconds per time step per element. The machine used is *IBM RS/6000 Model 530*. The management of the line-by-line procedure referred to in the table involves looping through the filled part of the domain, calculating the three-dimensional element matrix and implementing part of the condensation procedure. Also included in this are the assemblying of line equations, calling tridiagonal matrix solver routines, and updating solutions. "Overheads" refers to the CPU time spent in reconciling node numbering in (6 × 6) and (2 × 2) elements prior to the conversion of (6 × 6) to (2 × 2). TDMA in the table comprises of the generation of the elements of the tridiagonal matrix and the solution of the matrix. It is pointed out that the CPU times for IOLM is based on that for IOL1, assuming four iterations of the advection-diffusion equations per time step.

The explicit procedure (EOL) requires more iterations of pressure to attain convergence, relative to IOL1 or IOLM. However, as the problem size becomes large, as in this example, the cost of condensation becomes quite significant, making IOL1 (IOLM) much more expensive compared to EOL.

From the table we see that the calculation of the (6 × 6) element matrix for advection-diffusion equation accounts for almost 50% of the total CPU time. The procedure seems costly compared to what one would expect based on the speed with which the pressure equation is calculated. The CPU time for pressure includes the time to form the (3 × 3) element equations for the base triangular elements, as well as the time for frontal solution of the assembled pressue equations, including an average of three pressure iterations per time step. Ways to reduce the CPU time for the calculation of the element matrix for the advection-diffusion equation are needed.

In terms of accuracy of the three variations of the scheme, we have had the most success with EOL, for large problems. As of now we cannot explain the differences in accuracy to satisfaction, but they seem to be strongly related to accumulated truncation errors, as IOL1 and IOLM involve many more operations for large problems. A single-precision arithmetic is currently being used for the calculations.

Finally, because of an extensive data sharing in our implementation of the schemes, the extra cost is merely 20% for the solution of each additional equation of the form in Eq. (1)

5
Concluding remarks

In this paper we have proposed a procedure for the calculation of advection and diffusion in thin cavities. For the applications of interest, elements with very high aspect ratios are necessary for a cost-effective simulation. While our fluid mechanics problem is reminiscent of shell theory in structural mechanics, additional complications exist, and the numerical procedure proposed here are quite different from shell element formulations. It is felt that the present procedure contains sufficient novelty to generate interest in engineers faced with the solution of flow problems in thin cavities. It might be of interest to the reader to know that our procedure, while being mandatory for thin cavities, is actually applicable to domains with any thickness. Further, with a segregated solution approach, the complete set of equations governing fluid flow and scalar transport could be treated in the manner presented here. The foregoing, in our opinion, makes the procedure reported here quite interesting because, in applications, there are sometimes regions for which a thin-cavity approximation does not hold, and in which a full three-dimensional calculation is called for. Thus, it seems worthwhile to pursue the solution of the complete set of Navier-Stokes, energy, and scalar equations, using the approach reported here.

References

Barone, M. R.; Caulk, D. A. 1986: A Model for the Flow of a Chopped Fiber Reinforced Ploymer Compound in Compression Modeling. Journal of Applied Mechanics 53: 361–371

Carslaw, H. S.; Jaeger, C. 1959: Conduction of Heat in Solids. Oxford University Press, 177–178

Garcia, M. A. 1991: Reactive Mold Filling Modeling. *PhD Thesis.* University of Minnesota, USA.

Garcia, M. A.; Macosko, C. W.; Subbiah, S.; Guceri, S. I. 1991: Modeling of Reactive Filling in Complex Cavities. Intern. Polymer Processing VI(1): 73–82

Hieber, C. A.; Shen, S. F.; 1978: Flow Analysis of the Non-isothermal Two-Dimensional Filling Process in Injection Molding. Israel Journal of Technology 16: 284–254

Hieber, C. A.; Shen, S. F.; 1980: A Finite-Element/Finite-Difference Simulation of the Injection-Molding Filling Process. Journal of Non-Newtonian Fluid Mechanics 7: 1–32

Huebner, K. H.; Thornton, E. A. 1982: The Finite Element Method for Engineers. John Wiley and Sons, New York, 155–156

Hughes, T. J. R.; Brooks, A. 1982: A Theoretcial Framework for Petrov-Galerkin Methods with Discontinuous Weighting Functions: Application to the Streamline-Upwind Procedure. Finite Elements in Fluids. Volume 4. Editors: R. H. Gallagher, D. H. Norrie, J. T. Oden, and O. C. Zienkiwicz.

Johnson, C.; Saranan, J. 1986: Streamline Diffusion Method for the Incompressible Euler and Navier-Stokes Equations. Mathematics of Computation 47: 1–18

Johnson, C.; and Szepessy, A. 1987: On the Convergence of a Finite Element Method for a Nonlinear Hyperbolic Conservation Law. Mathematics of Computaion 49: 427–444

Ladeinde, F.; Akay, H. U.; Lynch, W. C. 1989: A New Formulation for Finite Element Solution of Non-Isothermal Processes in Injection/Compression Molding. *ANTEC '90.* 1727–1730

Ladeinde, F.; Akay, H. U.; 1994: The Calculation of Scalar Transport During the Injection Molding of Thermoset Polymers. Applied Mathematical Modeling 18: 347–357

Subbiah, S.; Trafford, D. L.; Güçeri, S. I. 1989: Non-Isothermal Flow of Polymers into Two-Dimensional, Thin Cavity Molds: A Numerical Grid Generation Approach. Int. J. Heat Mass Transfer. 32(3): 415–434

Wang, V. W.; Hieber, C. A.; Wang, K. K.; 1986: Dynamic Simulation of and Graphics For the Injection Molding of Three-Dimensional Thin Parts. Journal of Polymer Engineering 7(1): 21–45