Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

COMPUTER SIMULATION OF CAVITY FILLING

DURING INJECTION MOLDING PROCESS

by
Sumit Banerjee

Thesis submitted to the Faculty of the Graduate School of
the New Jersey Institute of Technology in partial fulfillment of
the requirements for the degree of
Master of Science in Mechanical Engineering
1983

APPROVAL SHEET

Title of Thesis : Computer Simulation of Cavity Filling During Injection Molding Process

Name of Candidate : Sumit Banerjee
Master of Science in Mechanical Engineering

Thesis and Abstract Approved : _____ _____
Dr. Richard C. Progelhof Date
Professor
Dept. of Mech. Engineering

Signatures of other members of the thesis committee _____ _____
Date

_____ _____
Date

VITA

Name : Sumit Banerjee

Degree and date to be conferred : M. S. M. E , 1983

Collegiate institutions attended	Dates	Degree	Date of Degree
Jadavpur University, India	'75-'80	B.S.M.E	August, 1980
New Jersey Institute of Technology	'81-'83	M.S.M.E	October,1983

Major : Mechanical Engineering

Positions held : Graduate Assistant September '81 - May, '83

New Jersey Institute of Technology

323 High Street, Newark, N.J. 07102

ABSTRACT

Title of Thesis : Computer Simulation of Cavity Filling During Injection Molding Process

Sumit Banerjee, Master of Science in Mechanical Engineering, 1983

Thesis directed by : Dr. Richard C. Progelhof
Professor, Department of Mechanical Engineering

A numerical technique is proposed for the simulation of cavity filling process during injection molding of glass-bead filled polypropylene. The mold cavity is of cylindrical shape. Marker And Cell (MAC) method is utilized for solving the transient flow phenomena, after a mathematical simulation of the flow model is carried out by using the relevant continuity and momentum equations governing the system. The complexity of the equations involved, results in the simplifying assumption of incompressible and isothermal flow process. A computer program is written on the basis of finite-difference equations developed during the application of MAC method under the prevailing conditions.

The numerical results yield significant data on the progression of the melt front, the velocity profiles in both axial and transverse directions and the pressure distributions at different times and positions in the cavity.

To Pitu And My Beloved Parents

ACKNOWLEDGMENTS

The author wishes to thank his advisor, Dr. Richard C.Progelhof, for his numerous suggestions and invaluable encouragement in the development of this work. He also wishes to thank the staff of the Mechanical Engineering department for their ideas and support.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

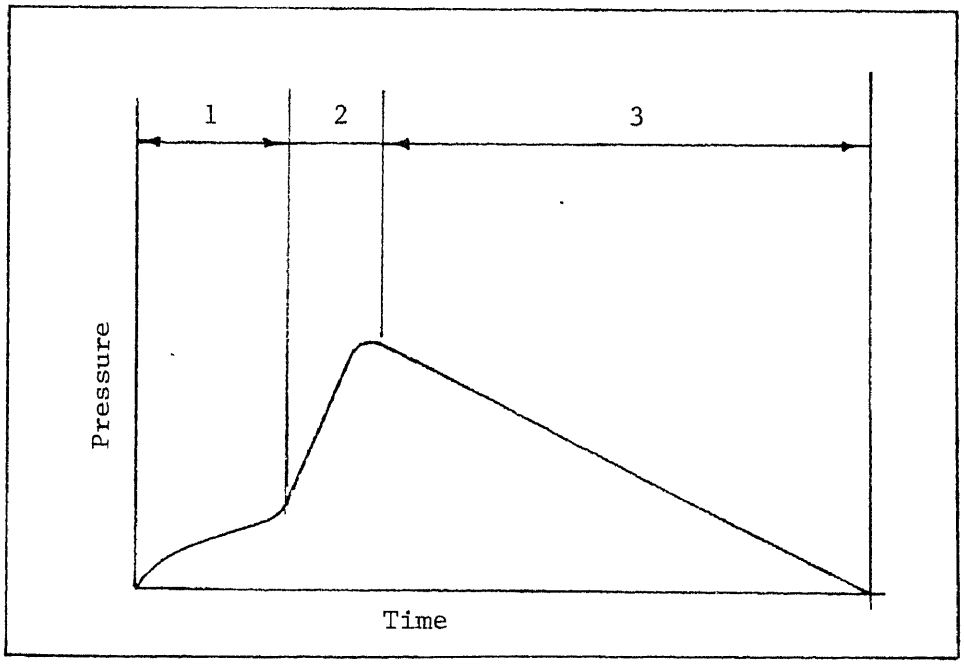
INTRODUCTION

Injection molding is considered to be a very important industrial process for the manufacturing of plastics objects. The molding cycle is composed of three stages : filling, packing and cooling. During filling, the molten polymer, which is produced by the shearing action of a rotating screw combined with external heating, is introduced into the mold. After filling, extra material is packed under high pressure to compensate for shrinkage as the material cools. After the injection pressure is removed, cooling continues, and the pressure decreases. Fig. 1.1[1] shows an injection molding pressure-time cycle schematically Experimental studies suggest that mold filling process determines the final product properties to a very great extent. Manzione[2], in a very recent study of simulation of cavity filling in Reaction Injection Molding, suggests that such simulation of reactive cavity filling is important in the determination of moldability because it allows prediction of the viscosity and temperature rise.

Flow visualization studies of cavity filling have been reported in the literature. The first reported study was made by Spencer and Gilmore[4], who have studied visually the filling of the mold and derived an empirical equation for the determination of filling time. Huang[6] simulated cavity filling by highly viscous thermoplast melt flowing between two rigid parallel plate boundaries with the gate at one edge. Kamal and Lafleur[5], in their report last year, used semi-circular and rectangular mold geometries.

Figure 1.1

Variation of Pressure With Time in Injection Molding



1. Filling
2. Packing
3. Cooling

The present study involves a thermoplast melt flowing through a mold of cylindrical shape. The front region has a free fluid boundary and time dependent velocity components.

Cavity filling is simulated by using a modification of simplified marker and cell method [7] developed by researchers at Los Alamos Scientific Laboratory. This method can be applied for numerical solution of problems concerning the time-dependent, viscous flow of an incompressible fluid in several space dimensions. The cells are characterized according to whether they are solid-fluid boundaries, free-slip or no-slip boundaries, inflow or outflow surfaces.

An interlaced grid system is used where the velocity components are centered at the cell sides and pressures placed at cell centers. This minimizes the involvement of neighboring cells for rigorous momentum conservation. At the same time it also decreases the amount of averaging usually required to provide variable values at grid points where the variables are not explicitly defined. The unique exact form of continuity equation can only be achieved by such a grid.

The first step in the setup of Marker And Cell (MAC) method is to provide the informations regarding the number of cells, viscosity, time step, marker particles density and other parameters pertaining to to the problem.The next step is to build up the cells. Here the input informations concerning the system boundaries include the shape of the cells and types of boundaries represented. Such informations are used

to identify all cells in the system. If there are no particles in the system, all interior cells are flagged as EMP cells (Empty Cells). The final step is to set up the marker particles which will represent the fluid flow. The coordinates of these particles are stored for future reference. The velocity field of the fluid flow is calculated and stored into the appropriate cells in the system. Consequently, the cells that contain particles are flagged as FUL or SUR cells (Full or Surface cell).

The fluid flow is then advanced through a series of time cycles, each of finite length δt. During each time cycle the output information is taken from the previous cycle and then the time is advanced by an increment of δt. Then the cells are checked to see if any of the previously EMP cells now contain fluid, or if any of the earlier SUR cells are now EMP or FUL. Flagging and changing of variables are done accordingly.The next step is to calculate the new velocity field from previously known velocities or input conditions. Finally, the marker particles are moved with a weighted average of the four nearest cell velocities. These time cycles are repeated as long as the problem is of any interest.

The following chapters describe the mathematical simulation of the problem,which involves writing the relevant continuity and momentum equations governing the system with appropriate boundary and initial conditions representing the prevailing conditions in the cavity, as well as the essential solution techniques utilizing MAC method.

CHAPTER II

MATHEMATICAL SIMULATION

To illustrate the Marker And Cell method for solving free-surface problems, we restrict ourselves to two-dimensional motions in a plane.

A cylindrical coordinate system (r,θ,z) is chosen with r measured in the radial direction and z taken normal to the r-axis. The effects in the θ-direction are assumed to be negligible.

2.1 Equation Of Continuity

Assuming the fluid to be non-Newtonian and incompressible in an isothermal flow process, we have

$$D = (1/r)(\partial/\partial r)(rV_r) + (\partial/\partial z)(V_z) = 0 \qquad \text{---------- (1)}$$

where D should vanish everywhere during the finite-difference calculation for the conservation of mass for an incompressible fluid.

2.2 Equation Of Momentum

For two-dimensional motion we can write

r-component

$$\rho((\partial V_r/\partial t)+V_r(\partial V_r/\partial r)+V_z(\partial V_r/\partial z)) = -(\partial P/\partial r)-((1/r)(\partial/\partial r)(r\tau_{rr}) -(\tau_{\theta\theta}/r) + (\partial \tau_{rz}/\partial z)) \qquad \text{---------- (2)}$$

z-component

$$\rho((\partial V_z/\partial t)+V_r(\partial V_z/\partial r)+V_z(\partial V_z/\partial z)) = -(\partial P/\partial z)-((1/r)(\partial/\partial r)(r\tau_{rz}) +(\partial \tau_{zz}/\partial z) \qquad \text{---------- (3)}$$

2.3 Constitutive Equation

In order to solve the equations of change, a rheological constitutive equation, which relates the stress tensor to the velocity field, is required.For a non-Newtonian fluid we can write

$$\underset{\sim}{\tau} = -\eta \underset{\sim}{\Delta} \qquad \text{---------- (4)}$$

in which the non-Newtonian viscosity η , a scalar quantity, is a function of rate of deformation or strain tensor, $\underset{\sim}{\Delta}$. A large amount of data available in the literature indicate that the Ostwald-de Waele or the Power Law Model is successful in describing the stress-rate of strain relationship for polymer melts, especially within narrow ranges of shear rate.Thus, we can represent this particular situation by :

$$\eta = m \left| \left(\tfrac{1}{2}\left(\underset{\sim}{\Delta} : \underset{\sim}{\Delta}\right)\right)^{\frac{1}{2}} \right|^{n-1}$$

---------- (5)

in which m, n are empirical fluid parameters. The rate of strain tensor is expressed as below :

$$\underset{\sim}{\Delta} = \begin{vmatrix} 2(\partial V_r/\partial r) & 0 & (\partial V_r/\partial z) + (\partial V_z/\partial r) \\ 0 & 2V_r/r & 0 \\ (\partial V_r/\partial z)+(\partial V_z/\partial r) & 0 & 2(\partial V_z/\partial z) \end{vmatrix}$$

---------- (6)

and $\underset{\sim}{\Delta} : \underset{\sim}{\Delta} = 4\,(\partial V_r/\partial r)^2 + 2(\partial V_r/\partial z + \partial V_z/\partial r)^2 + 4(V_r/r)^2 + 4(\partial V_z/\partial z)^2$

consequently

$$\eta = m \left| 2(\partial V_r/\partial r)^2 + ((\partial V_r/\partial z)+(\partial V_z/\partial r))^2 + 2(V_r/r)^2 + 2(\partial V_z/\partial z)^2 \right|^{\frac{n-1}{2}} \qquad \text{---------- (7)}$$

It is evident from equation (6) that the injection molding process is not a steady process dominated by either shear flow or extensional flow, but clearly a dynamic process with some combination of both types of flows (represented by a rate of deformation tensor with both non-zero diagonal and off-diagonal components) .

The choice of the fluid parameters, m and n in equation (7) may show the shear viscosity data from the viscometry measurement.
The two dimensional flow model in the front region is only a small part in the total flow compared with the fully developed region which is mostly shear flow in essence. The motivation for this choice is due to its success in describing experimentally measured material function for bulk polymers in simple shearing flows. The applicability of equation (7) to extensional viscosities both uniaxial and biaxial is proved experimentally by Denson[9], to be suitable for a low rate of extension where the fluid behavior approaches Newtonian viscosity.

Equation (4) may then be rewritten as

$$\tau_{rr} = -\eta\,(\,2\,\partial V_r/\partial r\,) \qquad \text{---------- (8)}$$

$$\tau_{\theta\theta} = -\eta\,(\,2V_r/r\,) \qquad \text{---------- (9)}$$

$$\tau_{zz} = -\,(\,2\,\partial V_z/\partial z\,) \qquad \text{---------- (10)}$$

$$\tau_{rz} = \tau_{zr} = -\eta\,((\partial V_z/\partial r) + (\partial V_r/\partial z)) \qquad \text{---------- (11)}$$

With the help of equations (1), (8), (9), (10), (11) we can now develop equations (2) and (3) as

$$\begin{aligned}\partial V_r/\partial t = &-(1/r)\,(\partial(rV_r^2)/\partial r) - (\partial(V_rV_z)/\partial z) - (\partial(P/\rho)/\partial r)\\ &+(\eta/\rho)(\partial^2V_r/\partial r^2) + (1/r)(\partial V_r/\partial r) - V_r/r^2 + \partial^2V_r/\partial z^2)\\ &+2(\partial V_r/\partial r)(\partial(\eta/\rho)/\partial r) + ((\partial V_z/\partial r)+(\partial V_r/\partial z))(\partial(\eta/\rho)/\partial z)\end{aligned}$$

---------- (12)

$$\begin{aligned}\partial V_z/\partial t = &-(1/r)(\partial(rV_rV_z)/\partial r) - \partial V_z^2/\partial z - \partial(P/\rho)/\partial z\\ &+(\eta/\rho)((\partial^2V_z/\partial r^2)+(1/r)(\partial V_z/\partial r)+(\partial^2V_z/\partial z^2))\\ &+2(\partial V_z/\partial z)(\partial(\eta/\rho)/\partial z)+((\partial V_z/\partial r)+(\partial V_r/\partial z))(\partial(\eta/\rho)/\partial r)\end{aligned}$$

---------- (13)

Like any other specific problem, it is necessary here to provide an appropriate set of initial and boundary conditions. We are particularly concerned with a prescribed set of rigid walls that may be no-slip or free-slip, and with inflow and outflow boundaries.

In addition to the prescribed boundary specifications, there will be boundary conditions to apply at the free surface, whose position varies with time in a previously unknown manner.

The rigid wall boundary conditions follow directly from momentum equations. For a free-slip boundary, the normal velocity component must vanish; for a no-slip boundary,the tangential components must, in addition, vanish. Consequently, boundary conditions on pressure are obtained through equations (12) and (13). However, we do not write these differential boundary conditions in detail here, as, for the numerical calculations, it is necessary to derive the finite-difference analogies to the boundary conditions directly from the finite-difference momentum equations.

Conditions along an inflow boundary are similarly derived; the only difference is that the velocity components are prescribed in the following arbitrary manner, rather than forced to vanish.

Here, for the sake of simplicity, the inlet velocity profile at the entrance of the mold is assumed to be that of a steady one dimensional fully developed flow model. This can be derived from equation (2). Consequently, the equation of motion becomes

$$(\partial P/\partial r) + (\partial \tau_{rz}/\partial z) = 0$$

Thus, the velocity profile becomes

$$V_z = \overline{V}\ ((2n+1)/(n+1))\left(\ 1-(r/(H/2))^{\frac{1}{n}+1}\ \right)$$

where H= diameter of the cylindrical mold

r = 0 represents the centerline

r = H/2 represents at wall

$\overline{V}$ = Average velocity

$$= \frac{\text{Volumetric flow rate}}{\text{Area of the entrance of mold}}$$

Pressure boundary conditions then follow from equations (12) and (13) in such a way as to again insure consistency with the momentum balance. The following principles form the basis for the free-surface boundary conditions :

i) Stress tangential to the surface must vanish.

ii) Stress normal to the surface must exactly balance any externally applied normal stress.

The detailed application of the boundary condition to the finite difference form of the Marker And Cell method will be discussed in the next chapter.

CHAPTER III

THE SOLUTION TECHNIQUE

In the preceding chapter we utilized the momentum equation to describe the dynamics of non-Newtonian melt through the cylindrical flow region. The complexity of the equations involved results in simplifying assumptions of incompressible isothermal flow and the resort to computer simulation and numerical solutions of these equations. The principal method we use here to solve this particular problem is the adaptation of a numerical technique based on the Marker And Cell (MAC) method. The preference for this particular numerical solution technique has been essentially governed by the presence of a free surface, at the front fluid boundaries, whose position varies with time and also by the nature of the two dimensional flow at the front flow region.

3.1 Representation of The Fluid Flow Field

In the development of any computing method for fluid dynamics problems, there are two interacting considerations that must be taken into account :

i) How are the fluid and its environment to be represented ?

ii) How are changes through time to be calculated ?

Many representations can be visualized for calculating the flow of an incompressible fluid with a free surface. The MAC arrangement appears to be one of very few approaches which enable us to achieve rigorous mass and momentum conservation.

There are, in effect, two coordinate systems used in MAC-method calculations : The primary one covers the entire domain of interest with rectangular grid of cells, each of dimensions δr by δz. The cells are numbered by indices i and j, with i counting the columns in the r-direction and j counting the rows in the z-direction. The field variable values describing the flow field are directly associated with these cells. Their point of definition, relative to a cell, are shown in Fig. 3.1 .

Actually the true field would, in general, have different set of field-variable values for every infinitesimal point in the fluid. The representation used for computing, however, must be restricted to a finite number of values, each approximating an average through the

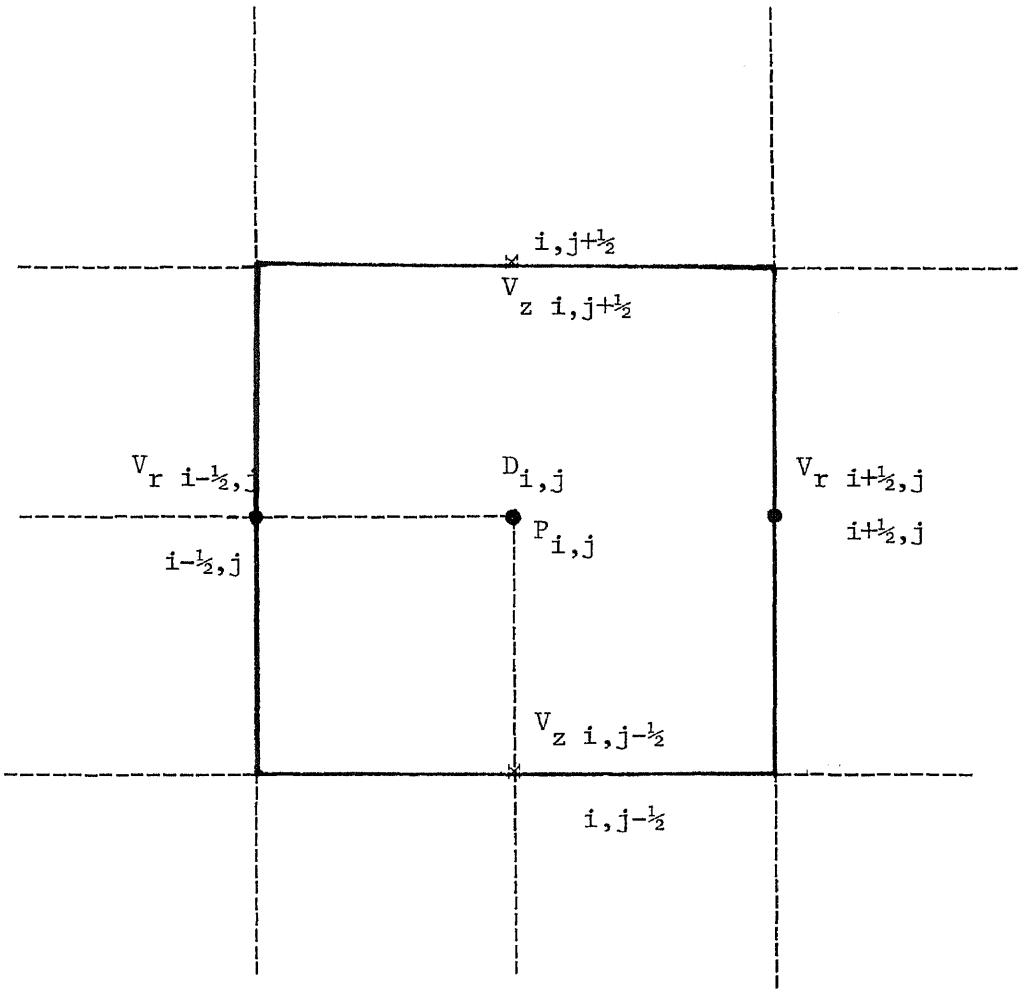


Figure 3.1

Field Variable Layout

immediately adjacent region. It follows that the accuracy of the representation depends strongly upon the fineness of the mesh compared to the macroscopic structure of the flow.

The placement of the field-variable quantities relative to the mesh is of considerable importance to the matter of conservation. It has been observed,by the developers of MAC-method at Los Alamos Scientific laboratory [3], that if the field-variables are placed at the cell centers the following complication is introduced : To attain rigorous finite difference mass conservation, the finite difference equation for pressure would require the involvement of the next layer of cells beyond that which immediately surrounds any central cell. Such an involvement adds enormously to the complexity and inaccuracy of the solution technique.

Apart from the primary-coordinate system of finite-difference cells, there is also a coordinate system of particles whose motions describe the trajectories of fluid elements. These particles serve two purposes : First, they show which cells are surface cells, into which the surface boundary conditions should be applied. Secondly, they show the motion of the fluid and all its distortions as it passes through the computing region. Thus, these particles delineate the fluid surface location and orientation. They move through a network of Eulerian cells, each cell is flagged to denote whether it is an empty cell (E) containing no fluid (hence no particles), a surface cell (S), which contains fluid but is adjacent to an empty cell, or a full cell (F),

which contains fluid and is not next to an empty cell. In addition, the cell network is surrounded by a frame of boundary cells (B). These may also contain particles, in which case they are either inflow or outflow boundaries, or, they may be empty representing rigid walls that may be ' free-slip ' or ' no-slip '. Fig. 3.2 illustrates this labeling. However, it should be noted here that these particles do not contribute anything to the dynamics of the flow. They enter the calculations only when determining the location and orientation of the free-surface. The solution of all the finite-difference approximations is carried out only in the region or regions of the mesh that contain particles.

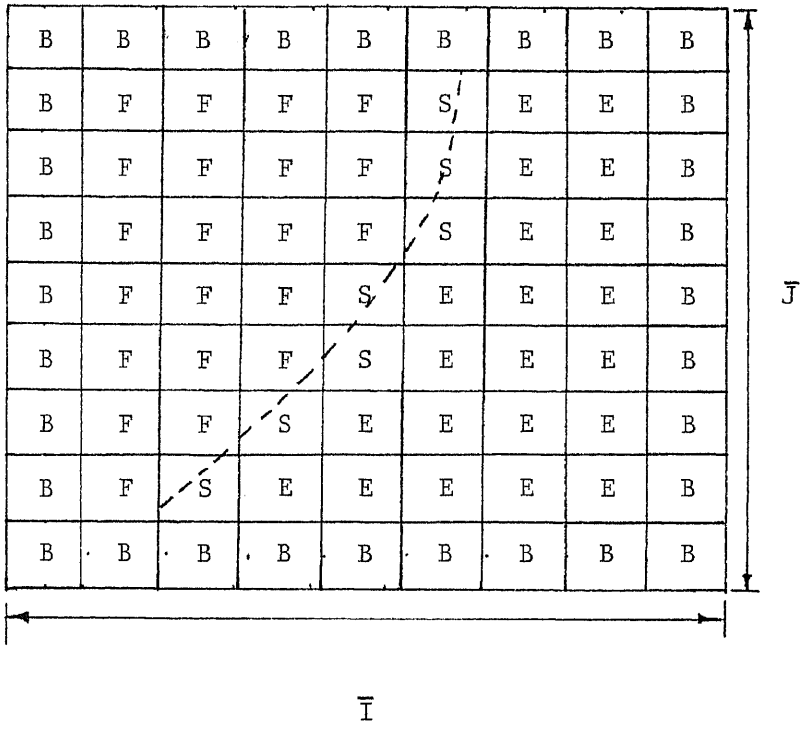
3.2 Outline Of The Computing Method

The cell-and-particle system enables an instantaneous representation of the fluid for any particular time during the evolution of the dynamics. In addition, it is necessary to have a means of actually calculating the changes with time of the fluid representation.With the help of this computing technique the prescribed initial conditions can develop, within the limitations imposed by the boundary conditions, into that subsequent set of configurations that most nearly represent the behavior of a real fluid.

Like most other fluid dynamics computing techniques for transient problems, the MAC method works with a time cycle, or ' movie frame ' point of view. This means that the calculation proceeds through a

Figure 3.2

Cell Labels and Computing Mesh



B = Boundary (Inflow,Outflow, Free-slip ,or No-slip)

E = Empty (No Marker Particles)

F = Full(Contains particles and is not adjacent to an Empty cell)

S = Surface(Contain particles but is exposed on one or more sides)

sequence of cycles, each advancing the entire fluid configuration through a small, but finite, increment of time, δt. The results of each cycle act as initial conditions for the next one, and the calculation continues for as many cycles as the investigator wishes. Each cycle itself is subdivided into phases :

i) The pressure for each cell is obtained by solving a finite difference Poisson's equation, whose source term is a function of the velocities. This equation was derived subject to the requirement that the resulting momentum equations should produce a new velocity field that satisfies the incompressibility condition.

ii) The full finite-difference Navier-Stokes equations are used to find the new velocities throughout the mesh.

iii) The marker particles are moved to their new positions, using for their velocities simple interpolated values from the nearby cells.

iv) Bookkeeping processes are accomplished related to the creation or destruction of surface cells, the input or output of particles, the advancement of a time counter, printing or plotting results, and numerous similar matters.

By the end of the cycle, the results have been arranged in the computer memory in such a way that the next cycle can immediately begin.

3.3 The Finite Difference Equations

The finite-difference for equation (1), the discrepancy term, is :

$$D_{i,j} = (1/r_{ij})(1/\delta r)((rV_r)_{i+\frac{1}{2},j} - (rV_r)_{i-\frac{1}{2},j}) + (1/\delta z)(V_{z\ i,j+\frac{1}{2}} - V_{z\ i,j-\frac{1}{2}}) \quad \text{---------- (14)}$$

Thus, the incompressibility condition becomes

$$D_{ij} = 0$$

which we require for every cell at every time step.

The finite -difference forms of equations (12) and (13) can be written as :

$$V^{\alpha+1}_{r\ i+\frac{1}{2},j} = V_{r\ i+\frac{1}{2},j} + \delta t(-(1/r_{i+\frac{1}{2},j})(1/\delta r)(r_{i+1,j}\ V^2_{r\ i+1,j} - r_{i,j}V^2_{r\ i,j}) - (1/\delta z)((V_r V_z)_{i+\frac{1}{2},j+\frac{1}{2}} - (V_r V_z)_{i+\frac{1}{2},j-\frac{1}{2}}) - (1/\delta r)((P/\rho)_{i+1,j} - (P/\rho)_{i,j}) + \beta_{i+\frac{1}{2},j})$$

---------- (15)

$$V_{z\ i,j+\frac{1}{2}}^{\alpha+1} = V_{z\ i,j+\frac{1}{2}} + \delta t(-(1/r_{i,j+\frac{1}{2}})(1/\delta r)((rV_r V_z)_{i+\frac{1}{2},j+\frac{1}{2}}$$

$$-(rV_r V_z)_{i-\frac{1}{2},j+\frac{1}{2}}) - (1/\delta z)(V_{z\ i,j+1}^2 - V_{z\ i,j}^2)$$

$$-(1/\delta z)((P/\rho)_{i,j+1} - (P/\rho)_{i,j}) + \beta_{i,j+\frac{1}{2}})$$

---------- (16)

where

$$\beta_{i+\frac{1}{2},j} = (\eta/\rho)_{i+\frac{1}{2},j}((1/\delta r^2)(V_{r\ i+3/2,j} + V_{r\ i-\frac{1}{2},j} - 2V_{r\ i+\frac{1}{2},j})$$

$$+(1/r_{i+\frac{1}{2},j})(V_{r\ i+3/2,j} - V_{r\ i-\frac{1}{2},j})(1/2\delta r) - (V_{r\ i+\frac{1}{2},j}/r^2_{i+\frac{1}{2},j})$$

$$+(1/\delta z^2)(V_{r\ i+\frac{1}{2},j+1} + V_{r\ i+\frac{1}{2},j-1} - 2V_{r\ i+\frac{1}{2},j}))$$

$$+2(1/2\delta r)(V_{r\ i+3/2,j} - V_{r\ i-\frac{1}{2},j})(1/\delta r)((\eta/\rho)_{i+\frac{1}{2},j}$$

$$-(\eta/\rho)_{i-\frac{1}{2},j}) + ((1/\delta r)(V_{z\ i+1,j} - V_{z\ i,j}) + (1/\delta z)$$

$$(V_{r\ i+\frac{1}{2},j+\frac{1}{2}} - V_{r\ i+\frac{1}{2},j-\frac{1}{2}}))(1/\delta z)(\eta/\rho)_{i+\frac{1}{2},j} - (\eta/\rho)_{i+\frac{1}{2},j-1})$$

$$\beta_{i,j+\frac{1}{2}} = (\eta/\rho)_{i,j+\frac{1}{2}}(1/\delta r^2)(V_{z\ i+1,j+\frac{1}{2}} + V_{z\ i-1,j+\frac{1}{2}} - 2V_{z\ i,j+\frac{1}{2}})$$

$$+(1/r_{i,j+\frac{1}{2}})(1/2\delta r)(V_{z\ i+1,j+\frac{1}{2}} - V_{z\ i-1,j+\frac{1}{2}}) + (1/\delta z^2)(V_{z\ i,j+3/2}$$

$$+V_{z\ i,j-\frac{1}{2}} - 2V_{z\ i,j+\frac{1}{2}}) + 2(1/2\delta z)(V_{z\ i,j+3/2} - V_{z\ i,j-\frac{1}{2}})(1/\delta z)$$

$$((\eta/\rho)_{i,j+\frac{1}{2}} - (\eta/\rho)_{i,j-\frac{1}{2}}) + ((1/\delta r)(V_{z\ i+\frac{1}{2},j+\frac{1}{2}} - V_{z\ i-\frac{1}{2},j+\frac{1}{2}})$$

$$+(1/\delta z)(V_{r\ i,j+1} - V_{r\ i,j}))(1/\delta r)((\eta/\rho)_{i,j+\frac{1}{2}} - (\eta/\rho)_{i-1,j+\frac{1}{2}})$$

The superscript α or $\alpha+1$ refer to a value at time $\alpha\delta t$ or $(\alpha+1)\delta t$, so that α counts the number of time cycles, where the superscript is omitted, α is implied, i.e., the value of the quantity at the beginning

of the cycle.

The finite difference approximation for calculating the viscosity, equation (7),is given by

$$\eta_{i+\frac{1}{2},j} = m \Big| 2((1/2\,\delta r)(V_{r\ i+3/2,j} - V_{r\ i-\frac{1}{2},j}))^2 + ((1/2\,\delta z)(V_{r\ i+\frac{1}{2},j+1} - V_{r\ i+\frac{1}{2},j-1}) + (1/\delta r)(V_{z\ i+1,j} - V_{z\ i,j}))^2 + 2(V_{r\ i+\frac{1}{2},j}/r_{i+\frac{1}{2},j}) + 2((1/\delta z)(V_{z\ i+\frac{1}{2},j+\frac{1}{2}} - V_{z\ i+\frac{1}{2},j-\frac{1}{2}}))^2 \Big|^{(n-1)/2} \qquad \text{---------- (17)}$$

and

$$\eta_{i,j+\frac{1}{2}} = m \Big| 2((1/\delta r)(V_{r\ i+\frac{1}{2},j+\frac{1}{2}} - V_{r\ i-\frac{1}{2},j+\frac{1}{2}}))^2 + ((1/\delta z)(V_{r\ i,j+1} - V_{r\ i,j}) + (1/2\,\delta r)(V_{z\ i+1,j+\frac{1}{2}} - V_{z\ i-1,j+\frac{1}{2}}))^2 + 2(V_{r\ i,j+\frac{1}{2}}/r_{i,j+\frac{1}{2}}) + 2((1/2\,\delta z)(V_{z\ i,j+3/2} - V_{z\ i,j-\frac{1}{2}}))^2 \Big|^{(n-1)/2} \qquad \text{---------- (18)}$$

In the above finite difference equations (eqs. (15), (16), (17) and (18)) , some of the primary cell variables, such as velocities, appear at positions other than those assigned in the Fig.3.1 . In such a case, a special differencing code, referred to as ZIP type, is used.

A description of this technique and a discussion of its advantages are presented by Hirt [10]. The following are some examples of the ZIP code :

$$V^2_{r\ i,j} = V_{r\ i+\frac{1}{2},j}\ V_{r\ i-\frac{1}{2},j}$$

$$V^2_{z\ i,j} = V_{z\ i,j+\frac{1}{2}}\ V_{z\ i,j-\frac{1}{2}}$$

$$V_{r\ i,j} = \tfrac{1}{2}\ (\ V_{r\ i+\frac{1}{2},j} + V_{r\ i-\frac{1}{2},j}\ )$$

$$V_{z\ i,j} = \tfrac{1}{2}\ (\ V_{z\ i,j+\frac{1}{2}} + V_{z\ i,j-\frac{1}{2}}\ )$$

$$V_{r\ i+\frac{1}{2},j+\frac{1}{2}} = \tfrac{1}{2}\ (\ V_{r\ i+\frac{1}{2},j+1} + V_{r\ i+\frac{1}{2},j}\ )$$

It may be noticed that as soon as the pressures are known for all the cells, the equations (15) and (16) become appropriate for the calculation of new velocities, a process accomplished by simple algebraic substitution. To find an equation for the pressures, it is only necessary to manipulate equations (15) and (16) into an expression for the rate of change D_{ij} . Let us define :

$$\phi = P / \rho$$

Then, by substituting equations (15) and (16) into equation (14) we have :

$$\begin{aligned} D^{\alpha+1}_{i,j} = D_{i,j} + \delta t(-Q_{i,j} &-(1/r_i \delta r^2)(r_{i+\frac{1}{2}}(\phi_{i+1,j} - \phi_{i,j}) \\ &-r_{i-\frac{1}{2}}(\phi_{i,j} - \phi_{i-1,j}))-(1/\delta z^2)(\phi_{i,j+1}+\phi_{i,j-1} -2\phi_{i,j}) \\ &+(1/r_i \delta r)(r_{i+\frac{1}{2}}\beta_{i+\frac{1}{2},j}-r_{i-\frac{1}{2}}\beta_{i-\frac{1}{2},j})+(1/\delta z)(\beta_{i,j+\frac{1}{2}}-\beta_{i,j-\frac{1}{2}})) \end{aligned}$$

---------- (19)

where

$$Q_{i,j} = (1/r_i \delta r^2)(r_{i+1} V^2_{r\ i+1,j} + r_{i-1} V^2_{r\ i-1,j} - 2r_i V^2_{r\ i,j}) + (1/\delta z^2)(V^2_{z\ i,j+1} + V^2_{z\ i,j-1} - 2V^2_{z\ i,j}) + (2/r_i \delta r\, \delta z)((rV_rV_z)_{i+\frac{1}{2},j+\frac{1}{2}} + (rV_rV_z)_{i-\frac{1}{2},j-\frac{1}{2}} - (rV_rV_z)_{i+\frac{1}{2},j-\frac{1}{2}} - (rV_rV_z)_{i-\frac{1}{2},j+\frac{1}{2}})$$

The equation obtained by setting $D^{\alpha+1}_{i,j} = 0$ in equation (19) is used for finding the pressures.

$$\phi_{i,j} = (1/\xi)((1/r_i \delta r^2)(r_{i+\frac{1}{2}} \phi_{i+1,j} + r_{i-\frac{1}{2}} \phi_{i-1,j}) + (1/\delta z^2)(\phi_{i,j+1} + \phi_{i,j-1}) + R_{i,j}) \qquad \text{---------- (20)}$$

where $\xi = 2((1/\delta r^2) + (1/\delta z^2))$

and $R_{i,j} = Q_{i,j} - (D_{i,j}/\delta t) - ((1/r_i \delta r)(r_{i+\frac{1}{2}} \beta_{i+\frac{1}{2},j} - r_{i-\frac{1}{2}} \beta_{i-\frac{1}{2},j}) + (1/\delta z)(\beta_{i,j+\frac{1}{2}} - \beta_{i,j-\frac{1}{2}}))$

3.4 Pressure Field Calculation

Equation (20) is now applied iteratively over all the points concerned until the $\phi_{i,j}$ computed at each point shows little further change. We can rewrite equation (20) as :

$$\phi^{s+1}_{i,j} = ((1+\lambda)/\xi)(\phi^{s}_{i+1,j} + \phi^{s}_{i-1,j})(1/\delta r^2) + (\phi^{s}_{i,j+1} + \phi^{s}_{i,j-1})(1/\delta z^2) + R_{i,j} - \lambda \phi^{s}_{i,j} \qquad \text{---------- (21)}$$

where λ is an overrelaxation parameter between 0 and 1. If $\lambda = 0$, overrelaxation is eliminated completely. The old values of ϕ (iteration No. s) may be used on the right hand side until a complete set of new values ϕ (iteration No. s+1) are computed. The Gauss-Seidel Method [11], where the most recent values of ϕ are used on the right hand side, is however preferred.

The iteration is considered to have converged when

$$((\phi^{s+1}_{i,j} - \phi^{s}_{i,j})/(\phi^{s+1}_{i,j} + \phi^{s}_{i,j})) < \epsilon$$

has been satisfied for all the concerned points. (Where $\epsilon = 2 \times 10^{-4}$ usually) Computation of pressure in the boundary cells and at the free surface requires certain special considerations which will be discussed later.

3.5 The Particle Movement

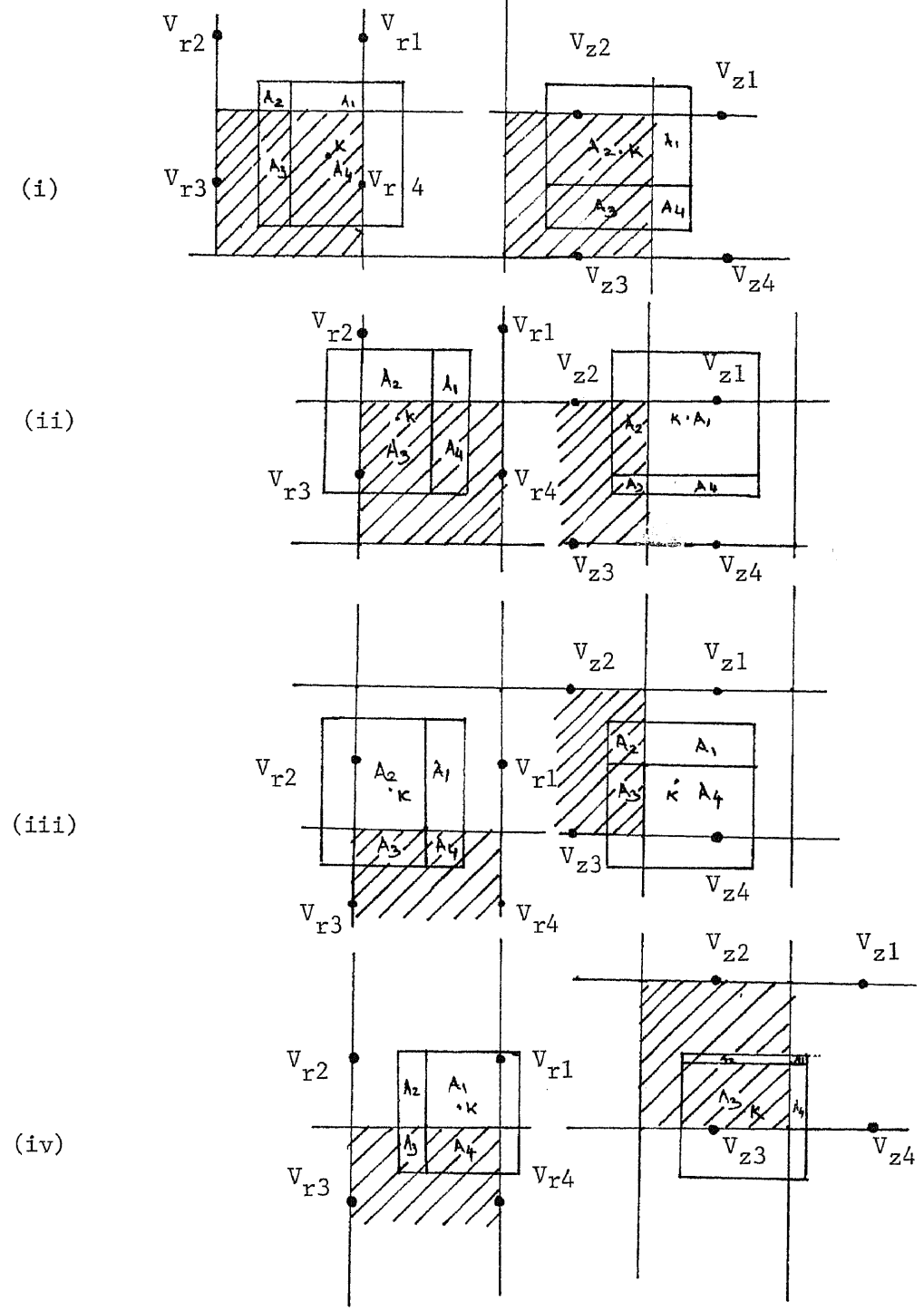
The marker particles, which enter into the solutions of the cell quantity equations, serve only the purpose of showing where the moving, free surface is located, and accordingly which cells should have free-surface boundary conditions imposed in them. In order to keep this information on free-surface position current, it is necessary to move the marker particles each cycle in such a way that they accurately represent the fluid motion.

The technique used by the researchers at the Los Alamos Scientific Laboratory[3] is essentially to find a velocity for the movement of each particle by using a simple area-weighted interpolation method among the nearby cell velocities. Huang[6] used the same area-weighting scheme for the simulation of flow through parallel plates. With reference to Fig.3.3 , the reference cell is shown as the cell lying between the lower pair of V_r velocities when calculating the V_r of the Kth particle (V_{rk}).

Similarly, the reference cell will be considered as the one lying between the left-most pair of V_z velocities when calculating the V_z of the Kth particle (V_{zk}). The reference cells provide an indexing base for referencing the four V_r s and four V_z s. The donor cell is, on the otherhand, defined as the cell containing the particle before it is moved. Evidently, there is a fifty-percent chance of the reference cell being the donor cell for either V_{rk} or V_{zk} . The reference cell is indexed as (IR,JR) and the donor cell as (ID,JD) . A particle cell of the same dimensions as the computational cell, is constructed around the marker particle of interest and is positioned so that the particle lies at its center. The fractional area of the cell segments A_1,A_2,A_3 and A_4,generated by the intersection of a grid line with a line passing through a reference cell center, are used for weighting the nearest velocities. From Fig.3.3 it is evident that V_{rk} and V_{zk} can be calculated for any of the four possible cases (i,ii,iii and iv) by using the equation on the following page :

Figure 3.3

Area Weighting Scheme For The Particle Velocities V_{rk} and V_{zk}. The Shaded Cell Is The Reference Cell In Each Case.



$$V_{rk} = A_2 \times V_r(ID-1,JR+1) + A_1 \times V_r(ID,JR+1) + A_3 \times V_r(ID-1,JR) + A_4 \times V_r(ID,JR)$$

or

$$= A_2 \times V_{r2} + A_1 \times V_{r1} + A_3 \times V_{r3} + A_4 \times V_{r4}$$

$$V_{zk} = A_2 \times V_z(IR,JD) + A_1 \times V_z(IR+1,JD) + A_3 \times V_z(IR,JD-1) + A_4 \times V_z(IR+1,JD-1)$$

or

$$= A_2 \times V_{z2} + A_1 \times V_{z1} + A_3 \times V_{z3} + A_4 \times V_{z4}$$

After V_{rk} and V_{zk} have been determined, the particles are moved by adjusting its position co-ordinates (r_k, z_k) as follows :

$$r_k^{\alpha+1} = r_k^{\alpha} + V_{rk}\,(t/r) \qquad \text{----------(22)}$$

and

$$z_k^{\alpha+1} = z_k^{\alpha} + V_{zk}\,(t/z) \qquad \text{----------(23)}$$

It should be noted here that the particle co-ordinates are calculated in units of cell distance rather than distance on the basis of mold size. Particles are created from the inflow boundary at the prescribed rate. If a particle moves to an Empty Cell, the receiving cell becomes a Surface Cell. Alternately, if all particles move out of a Full or Surface Cell, it is labelled empty. However, the time interval is normally set in such a fashion that the particle movement per increment never exceeds half a cell. Thus a transition from Full to Empty or Empty to Full can only take place over several time steps during which

time a cell becomes a Surface Cell before turning Empty or Full.

3.6 Boundary Conditions

The type of boundary conditions to be applied depend entirely on the kind of boundary under consideration. Based on the mold geometry involved in this study, we describe the following conditions :

i) At The Center Line (Figure 3.4)

This is a free-slip boundary which has a line of symmetry. The transverse velocity component vanishes at the centerline, and there is no gradient in either the flow direction velocity component or the pressure function P. It is noted that the particle which happens to be at the intersection of the centerline and the front free surface should move to the wall due to the fountain effect. Physically, the particles which originate from the center of the entrance will decelerate along the centerline toward the front surface. Once the particle reaches the intersection of the centerline and the front free surface, it should continue to move ahead, otherwise, there is no room for the next coming particle to move to this point. Consequently, we assign the transverse velocity component at this particular point to be approximately equal to that of the next grid point. Thus,

a) $V_{r\ i+\frac{1}{2},1} = V_{r\ i+\frac{1}{2},2}$

b) $V_{z\ i,1+\frac{1}{2}} = 0$ (except at front surface)

$= V_{z\ i,2+\frac{1}{2}}$ (at the front surface)

c) $P_{i,1} = P_{i,2}$

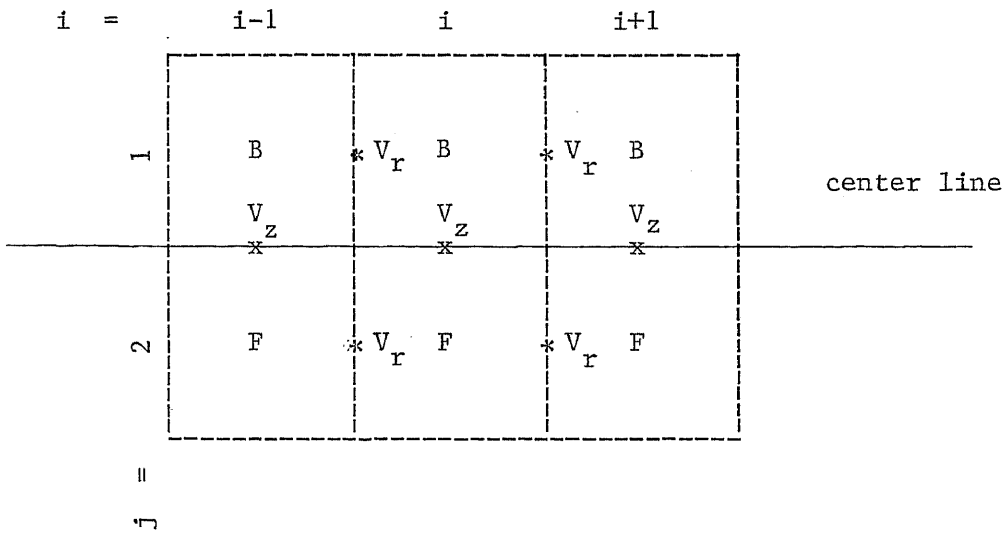


Figure 3.4

Boundary at the centerline

ii) At The Wall (Top And Bottom Side) (Figure 3.5)

Before proceeding to describe the boundary conditions, it should noted here that the present applications of the MAC-method restrict the rigid walls,as well as influx and outflux walls, to follow the cell boundaries. This is inevitable as Eulerian fluid dynamics calculation precludes any other form of representation. In this case we have a no-slip rigid boundary. The velocity components both normal and tangential to the wall are forced to vanish.

a) $V_{z\ i,jp1+\frac{1}{2}} = 0$

b) $V_{r\ i+\frac{1}{2},jp2} = - V_{r\ i+\frac{1}{2},jp1}$

c) $P_{i,jp2} = P_{i,jp1}$

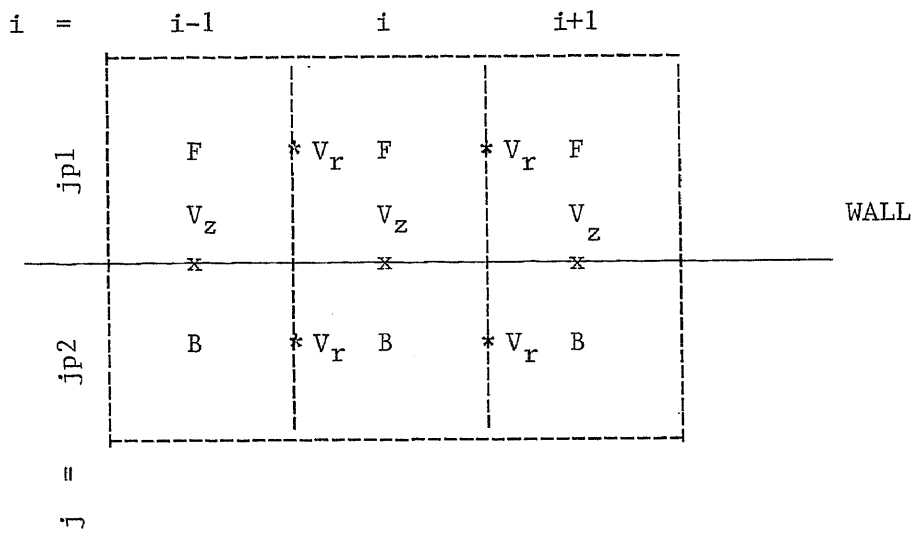


Figure 3.5

Boundary at the bottom wall

iii) At The Entrance (Figure 3.6)

An input wall or an inflow wall at the entrance allows the fluid to move into the system at a prescribed rate. The flow direction velocity field is a function of z and kept constant throughout the run. Marker particles are inserted through the wall to represent the incoming fluid.

a) $V_{r\ 1+\frac{1}{2},j} = f(z)$

b) $V_{z\ 1,j+\frac{1}{2}} = 0$

c) $P_{1,j} = P_{2,j} - (\eta)_{1+\frac{1}{2},j} \quad x$

$(\Delta r/\delta z^2)(V_{r\ 1+\frac{1}{2},j+1} + V_{r\ 1+\frac{1}{2},j-1} - 2V_{r\ 1+\frac{1}{2},j})$

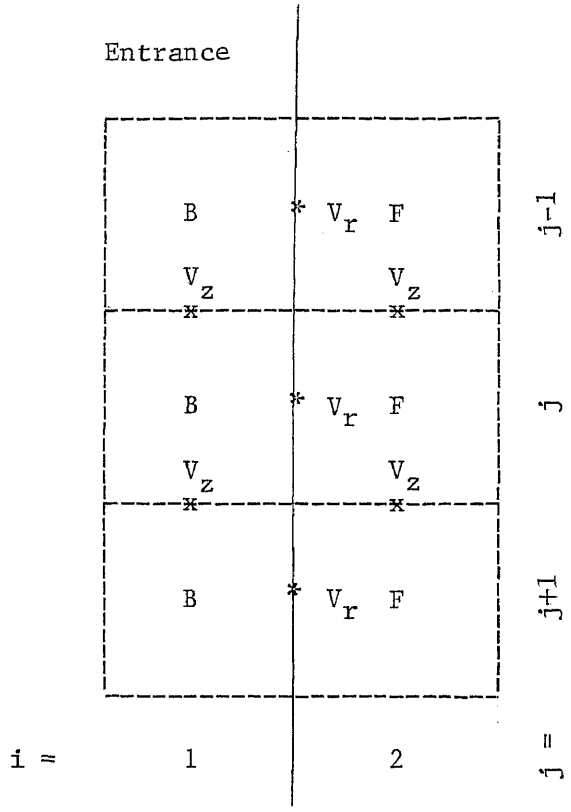


Figure 3.6

Boundary at the Entrance

iv) At The Right Side Wall (Figure 3.7)

Here the boundary condition is analoguous to top and bottom side wall where a no-slip rigid wall exists. Consequently,

a) $V_{r\ ip1+\frac{1}{2},j} = 0$

b) $V_{z\ ip2,j+\frac{1}{2}} = - V_{z\ ip1,j+\frac{1}{2}}$

c) $P_{ip2,j} = P_{ip1,j}$

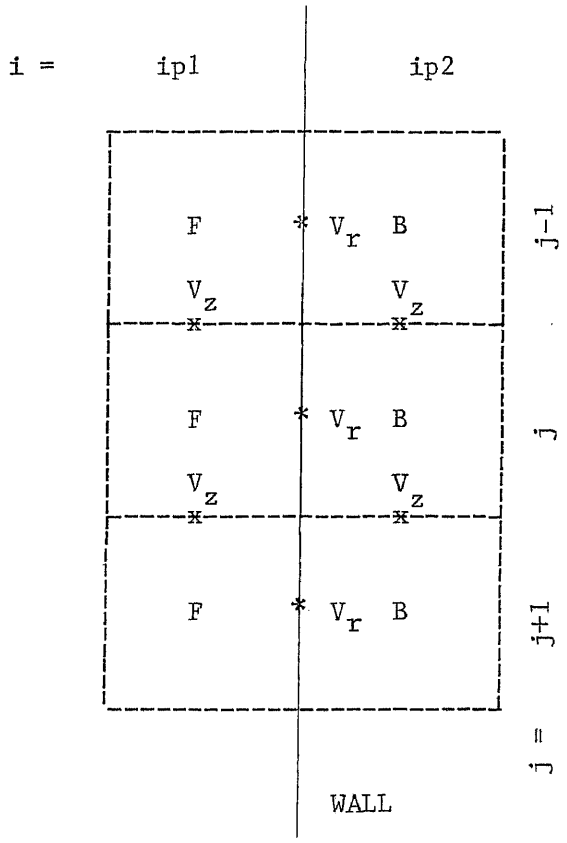


Figure 3.7

Boundary at the right side wall

v) Boundary Condition At The Free Surface

The free surface boundary conditions necessitates the condition that no momentum flux may pass through the free surface of the fluid In this connection, the stress tensor, $\tau_{i,j}$, is defined as the amount of i-th component momentum flowing per unit time through unit area normal to the j-th direction. To satisfy the above,free fluid surface condition may then be expressed by

$$\tau_{i,j}\ n_j = P_e \qquad \text{---------- (24)}$$

where, P_e is the external pressure applied to the fluid surface, assuming zero stress on the air side.

n_j is the unit normal to the surface

and, $\tau_{i,j}$, the stress tensor for an incompressible fluid, is given by :

$$\tau_{i,j} = -P\delta_{i,j} + \eta\left(\left(\partial V_j/\partial r_i\right)+\left(\partial V_i/\partial r_j\right)\right) \qquad \text{---------- (25)}$$

where δ_{ij} is the kronecker delta

$\delta_{ij} = 1$ if $i=j$

$\delta_{ij} = 0$ if $i\neq j$

and P is the internal pressure at the surface.

In the equation (24) above the external pressure term, P_e, consists of two pressures, one being atmospheric and the other due

to surface tension effects at the fluid surface. Thus,

$$P_e = P_a + \sigma (1/R_1 + 1/R_2) \quad \text{---------- (26)}$$

where, σ is the surface tension coefficient (stress/ unit length) and, R_1 and R_2 are the principal radii of curvature at the surface. It has been observed that, even with high surface tension coefficients, the surface tension is very small compared to the viscous stresses generated by the high viscous polymer fluid flow. Consequently, the external pressures P_e are represented by the atmospheric pressure P_a in this study.

The trace of the free fluid surface can be represented by

$$Z = S (r,t) \quad \text{---------- (27)}$$

Let n_r and n_z be the components of a unit outward vector, normal to the surface. Using equation (27) these components are given by :

$$n_r = -(1+ (\delta s/\delta r)^2)^{-\frac{1}{2}} \ (\partial s/\partial r) \quad \text{---------- (28)}$$

$$n_z = (1+ (\partial s/\partial r)^2)^{-\frac{1}{2}} \quad \text{---------- (29)}$$

Also, let m_r and m_z be the corresponding components of a unit tangential vectors so that

$$m_r = n_z \qquad \text{and} \qquad m_z = -n_r$$

Thus, within the fluid at its surface, the boundary condition equation (24) may be separated into two components, the tangential stress condition :

$$\eta\,(2n_r m_r(\partial V_r/\partial r) + (n_r m_z + n_z m_r)\,((\partial V_r/\partial z)+(\partial V_z/\partial r)) + 2n_z m_z(\partial V_z/\partial z) = 0 \qquad \text{---------- (30)}$$

and the normal stress condition :

$$P - 2\eta(n_r n_r\,(\partial V_r/\partial r) + n_r n_z\,((\partial V_r/\partial z)+(\partial V_z/\partial r)) + n_z n_z(\partial V_z/\partial z) = P_e \qquad \text{---------- (31)}$$

If the curvature is small, the $(\delta s/\delta r)^2$ term in equations (28) and (29) may be neglected and equations (30) and (31) may be approximated by :

$$\eta\,((\partial V_n/\partial m) + (\partial V_m/\partial n) = 0 \qquad \text{---------- (32)}$$

and

$$P - 2\eta\,(\partial V_n/\partial n) = P_e \qquad \text{---------- (33)}$$

where n refers to the local outward normal direction of the free surface and m to the tangential direction.

3.7 Application Of Normal Stress Condition

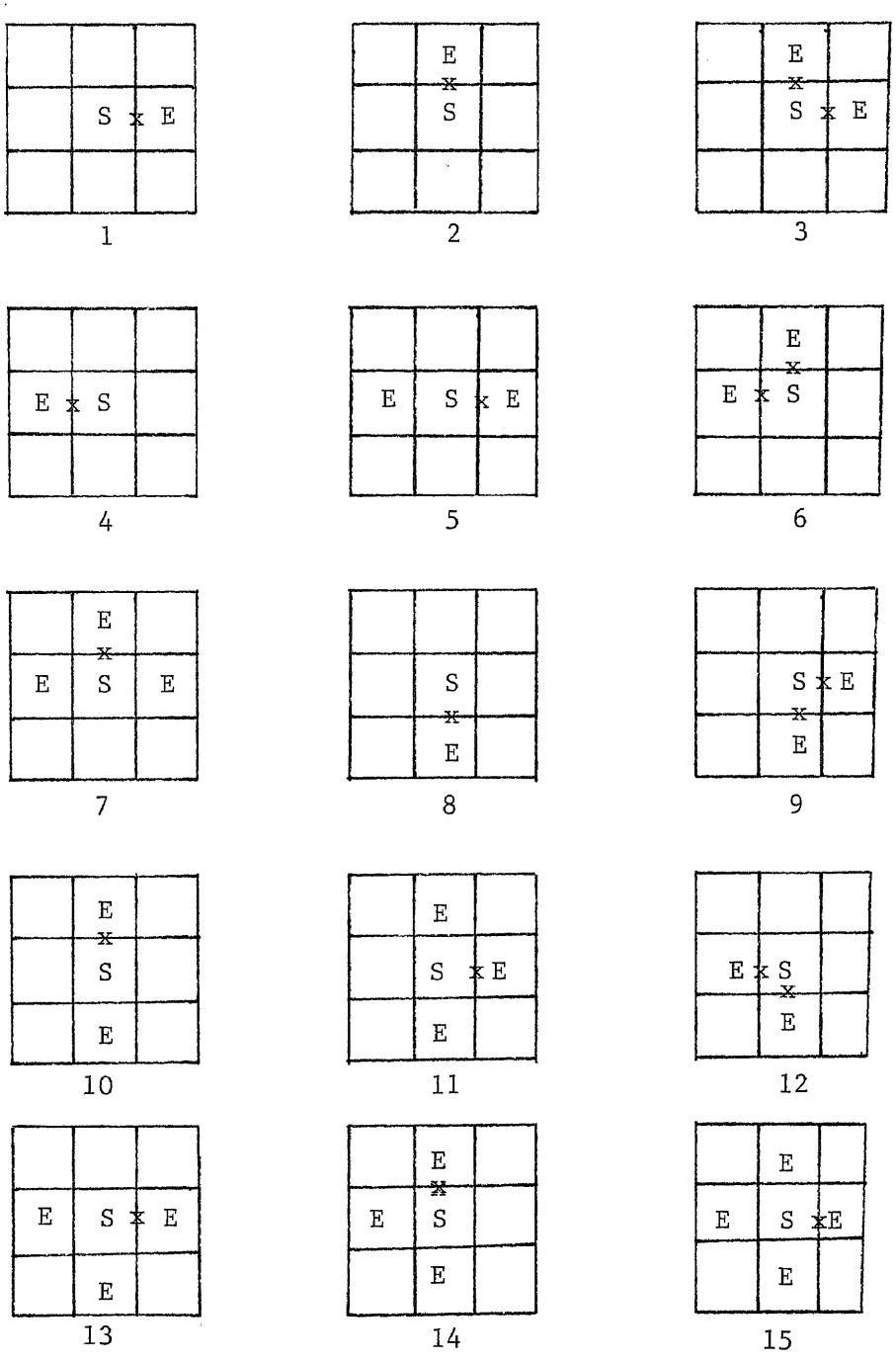
Equation (33) shows that the application of the normal stress condition requires the determination of both the surface slope and the location within the surface cell at which the internal and external normal stresses must balance. The exact surface slope may be determined at the expense of a considerable amount of calculation. Instead, an approximation, based on the cell flagging scheme in Fig.3.2 , is used.

Thus, if the cell above the surface cell is the only adjacent empty cell, then the surface is considered to be horizontal. If both the cell to the left and the one above the surface cell are empty, then the surface is considered to be oriented 45° to the horizontal. Similarly, orientations at 90° and 135°, as well as the equivalent cases with the surface exposed from below, may be also determined. The appropriate normal stress, which depends on the slope,is then derived from equation (33) . Fig.3.8 shows the fifteen possible arrangements of empty cells (E) about a surface cell (S).

By satisfying the continuity equation, given by eq.(14), for all free surface cells the surface velocities (indicated by ' x ' in Fig.3.8) are calculated. For one empty face (configurations 1, 2, 4 and 8) or two adjacent faces (3, 6, 9, 12) the

Figure 3.8

Fifteen Possible Arrangements of Empty Cells About a Surface Cell.



appropriate V_r and/or V_z may be calculated precisely. In the remaining cases (5, 7, 10, 11, 13, 14 and 15) more than one V_r and/or V_z are involved and the treatment can only be approximate. Here one velocity is adjusted to assure that at least $D = 0$ in the surface cell Accordingly, the appropriate forms of equation (14) , labelled (a to h) in equation (34) are used in the following arrangement for each of the fifteen configurations :

Configuration No. In Figure 3.8	Form of Equation (14) Used
1, 5, 11, 13, 15	a
2, 7, 10, 14	b
3	c and f
4	d
6	e and f
8	g
9	c and h
12	e and h

a) $V_{r\ i+\frac{1}{2},j} = ((rV_r)_{i-\frac{1}{2},j}/r_{i+\frac{1}{2},j}) - (r_{i,j}/r_{i+\frac{1}{2},j})(\delta r/\delta z)(V_{z\ i,j+\frac{1}{2}} - V_{z\ i,j-\frac{1}{2}})$

b) $V_{z\ i,j+\frac{1}{2}} = V_{z\ i,j-\frac{1}{2}} - (1/r_{i,j})(\delta z/\delta r)((rV_r)_{i+\frac{1}{2},j} - (rV_r)_{i-\frac{1}{2},j})$

c) $V_{r\ i+\frac{1}{2},j} = (r_{i-\frac{1}{2},j}/r_{i+\frac{1}{2},j})\ V_{r\ i-\frac{1}{2},j}$

d) $V_{r\ i-\frac{1}{2},j} = ((rV_r)_{i+\frac{1}{2},j}/r_{i-\frac{1}{2},j}) + (r_{i,j}/r_{i-\frac{1}{2},j})(\delta r/\delta z)(V_{z\ i,j+\frac{1}{2}} - V_{z\ i,j-\frac{1}{2}})$

e) $V_{r\ i-\frac{1}{2},j} = ((rV_r)_{i+\frac{1}{2},j}/r_{i-\frac{1}{2},j})$

f) $V_{z\ i,j+\frac{1}{2}} = V_{z\ i,j-\frac{1}{2}}$

g) $V_{z\ i,j-\frac{1}{2}} = V_{z\ i,j+\frac{1}{2}} + (1/r_{i,j})(\delta z/\delta r)((rV_r)_{i+\frac{1}{2},j} - (rV_r)_{i-\frac{1}{2},j})$

h) $V_{z\ i,j-\frac{1}{2}} = V_{z\ i,j+\frac{1}{2}}$

---------- (34)

Once the surface cell velocities on empty cell faces are known, the normal stress condition given by equation (33) may now be applied. The cell arrangements in Fig. 3.8 can be classified into three groups :

Group (i) : Surface cells with one empty side (1, 2, 4 and 8)
The free surface orientation is either horizontal (2 and 8) ,

$$P = 2\eta\,(\partial V_z/\partial z) + P_e \qquad \text{---------- (35)}$$

or vertical (1 and 4)

$$P = 2\eta\,(\partial V_r/\partial r) + P_e \qquad \text{---------- (36)}$$

Group (ii) : Surface cells with two diagonally adjacent empty cells (3, 6, 9 and 12)

The outward normal direction is assumed to be at 45° between the exposed sides of the cell. In this case the normal stress condition, given by equation (33) reduces to

$$P = \pm\,((\,V_r/\,z) + (\,V_z/\,r) + P_e \qquad \text{---------- (37)}$$

where the sign is chosen equal to the sign of $n_r n_z$.
For example, if the open sides are at $(i, j+\frac{1}{2})$ and $(i+\frac{1}{2}, j)$, then n_r and n_z are positive.

Group (iii) : Surface cells with three open sides or with two open sides that are opposite one another (5,7,10,11,13,14 and 15). The pressure for the cell is set equal to the external pressure P_e.

The velocity partial derivatives in the equations (35) through (37) can easily be approximated by local finite differences and these eight approximations are listed below :

Configuration No. in Fig. 3.8	Finite Difference Approximation For Surface Pressure $P_{i,j}$
1	$(2\eta/\delta r)((rV_r)_{i+\frac{1}{2},j} - (rV_r)_{i-\frac{1}{2},j}) + P_{e\ i,j}$
4	$(2\eta/\delta r)((rV_r)_{i-\frac{1}{2},j} - (rV_r)_{i+\frac{1}{2},j}) + P_{e\ i,j}$
2	$(2\eta/\delta z)(V_{z\ i,j+\frac{1}{2}} - V_{z\ i,j-\frac{1}{2}}) + P_{e\ i,j}$
8	$(2\eta/\delta z)(V_{z\ i,j-\frac{1}{2}} - V_{z\ i,j+\frac{1}{2}}) + P_{e\ i,j}$
3	$(\eta/2)((1/\ z)((rV_r)_{i+\frac{1}{2},j} - (rV_r)_{i+\frac{1}{2},j-1} + (rV_r)_{i-\frac{1}{2},j} - (rV_r)_{i-\frac{1}{2},j-1}) + (1/\delta r)(V_{z\ i,j+\frac{1}{2}} - V_{z\ i-1,j+\frac{1}{2}} + V_{z\ i,j-\frac{1}{2}} - V_{z\ i-1,j-\frac{1}{2}})) + P_{e\ i,j}$
6	$(\eta/2)((1/\delta z)((rV_r)_{i+\frac{1}{2},j} - (rV_r)_{i+\frac{1}{2},j-1} + (rV_r)_{i-\frac{1}{2},j} - (rV_r)_{i-\frac{1}{2},j-1}) + (1/\delta r)(V_{z\ i,j+\frac{1}{2}} - V_{z\ i+1,j+\frac{1}{2}} + V_{z\ i,j-\frac{1}{2}} - V_{z\ i+1,j-\frac{1}{2}})) + P_{e\ i,j}$
9	$(\eta/2)((1/\delta z)((rV_r)_{i+\frac{1}{2},j} - (rV_r)_{i+\frac{1}{2},j+1} + (rV_r)_{i-\frac{1}{2},j} + (rV_r)_{i-\frac{1}{2},j+1}) + (1/\delta r)(V_{z\ i,j+\frac{1}{2}} - V_{z\ i=1,j+\frac{1}{2}} + V_{z\ i,j-\frac{1}{2}} - V_{z\ i-1,j-\frac{1}{2}})) + P_{e\ i,j}$
12	$(\eta/2)((1/\delta z)(rV_r)_{i+\frac{1}{2},j} - (rV_r)_{i+\frac{1}{2},j+1} + (rV_r)_{i-\frac{1}{2},j} - (rV_r)_{i-\frac{1}{2},j+1}) + (1/\delta r)(V_{z\ i,j+\frac{1}{2}} - V_{z\ i+1,j+\frac{1}{2}} + V_{z\ i,j-\frac{1}{2}} - V_{z\ i+1,j-\frac{1}{2}})) + P_{e\ i,j}$

---------- (38)

3.8 Application Of Tangential Stress Condition

The tangential stress condition is given by the equation (32). The position and shape of the surface have to be known before this condition can be applied. From the arrangement of neighboring empty cells the surface slope can be roughly estimated, as was done in the application of normal stress condition, although these can be calculated from the location of marker particles. This estimation is sufficiently accurate for our purpose because of other inherent approximations in the MAC method itself. Consequently, for a two-dimensional surface, if the surface is nearly horizontal or vertical,which is the case when the surface cell is open at one end to the empty cell only, the tangential stress condition of equation (32) becomes :

$$\eta\,((\partial V_r/\partial z) + (\partial V_z/\partial r)) = 0 \qquad \text{---------- (39)}$$

If, however, the surface slope is at 45° to the horizontal, i.e. when there are two adjacent empty cells contiguous to the surface cell, the tangential stress condition reduces to

$$\eta\,((\partial V_r/\partial r) - (\partial V_z/\partial z)) = 0 \qquad \text{---------- (40)}$$

From these two forms of the tangential stress condition, the just outside tangential velocities may be calculated with the help of finite difference approximations. There are only four

basic arrangements of surface (S) to empty (E) cells that must be checked to satisfy equation (39). These are illustrated in Fig. 3.9 . The equation (40) can not be easily formulated in the same fashion. Instead, it is satisfied by making $(\partial V_r/\partial r)$ and $(\partial V_z/\partial z)$ each equal to zero.This was done in equation (34). The appropriate Finite Difference Approximations of the equation (39) are given below for each of the four arrangements shown in Fig. 3.9 .

a) $V_{z\ i+1,j+\frac{1}{2}} = V_{z\ i,j+\frac{1}{2}} - (1/r_{i,j})(\delta r/\delta z)((rV_r)_{i+\frac{1}{2},j+1} - (rV_r)_{i+\frac{1}{2},j})$

b) $V_{r\ i+\frac{1}{2},j+1} = ((rV_r)_{i+\frac{1}{2},j}/r_{i+\frac{1}{2},j+1}) - (r_{i,j}/r_{i+\frac{1}{2},j+1})(\delta z/\delta r)(V_{z\ i+1,j+\frac{1}{2}} - V_{z\ i,j+\frac{1}{2}})$

c) $V_{z\ i-1,j+\frac{1}{2}} = V_{z\ i,j+\frac{1}{2}} + (1/r_{i,j})(\delta r/\delta z)((rV_r)_{i-\frac{1}{2},j+1} - (rV_r)_{i-\frac{1}{2},j})$

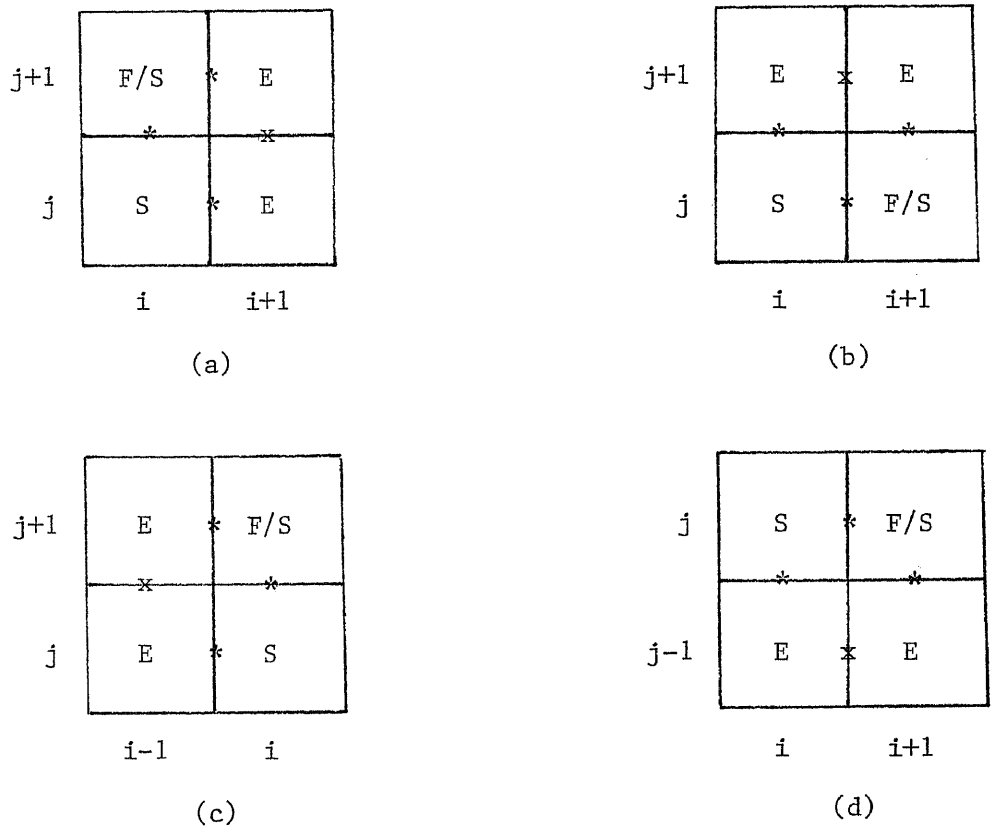
d) $V_{r\ i+\frac{1}{2},j-1} = ((rV_r)_{i+\frac{1}{2},j}/r_{i+\frac{1}{2},j-1}) + (r_{i,j}/r_{i+\frac{1}{2},j-1})(\delta z/\delta r)(V_{z\ i+1,j-\frac{1}{2}} - V_{z\ i,j-\frac{1}{2}})$

---------- (41)

Figure 3.9

Four Arrangements of Surface (S) to Empty (E) Cells For Tangential Stress Condition.

(The surface cell in each case has the indices i,j and the velocity to be determined (x) is a function of three other velocities(*) .)



3.9 Stability And Accuracy

The solution of initial value problems by finite-difference approximations is almost always plagued by potential difficulties with numerical instability.There are unique numerical instabilities associated with each of the iteration sequences in MAC calculations, i.e., with the pressure iteration at a given time step and with the successive steps of time advancement. In the original MAC method [3] the condition necessary to insure stability through successive time steps was

$$\delta t < (1/2\,\eta)((\delta r^2\,\delta z^2)/(\delta r^2+\delta z^2)) \qquad (42)$$

However, this condition must be modified to suit the present non-Newtonian problem. This is done by setting the viscosity η to be at the zero shear rate for the least value of δt. For this particular high viscosity case, the time increment per cycle is quite small which necessitates a long consumption of computation time.

The obvious requirements, as far as the accuracy is concerned, include the necessity for cells fine enough to resolve the features of interest and for time steps small enough to prevent instability.However, it should be noted here that a precise statement concerning cellwise resolution appears impossible to give. In general, it may be stated that the cells must be so small that no field variable changes appreciably across any cell.

Experience with many different types of computing techniques has shown that one of the most important contributors to accuracy is the degree of conservation of the finite-difference equations. For incompressible flow calculations of the type discussed here, the primary quantities to conserve are mass and momentum. Mass conservation is assured if, at every cycle, $D = 0$ for every cell.

Momentum conservation is also strongly required for accuracy of calculations. Consequently, the terms in the transport part of the momentum equations of the form

$$V_r(\partial V_r/\partial r) + V_z(\partial V_r/\partial z)$$

are transformed (using the incompressibility condition) to

$$(1/r)(\partial(rV_r^2)/\partial r) + (\partial(V_rV_z)/\partial z)$$

The difference form of such an expression can then be written as pure differences, so that the flux of momentum out of one side of a cell exactly equals the flux into that same side of the adjacent cell.

CHAPTER IV

COMPUTATIONAL DETAILS

4.1 Sequency Of Computation

Based on the solution technique described in the last chapter,we are now in a position to set up a computer program to solve this particular problem. The computer logic flow diagram in Fig. 4.1 explains how this simulation of the filling process can be carried out. Each box in the flow diagram is now discussed in detail in the order in which it appears.

1) Input Information

   i) material parameters

      a) density ρ

      b) Rheological parameters n, m

   ii) Geometrical parameters

      a) dimension of gate

      b) dimension of mold

   iii) Process operating condition

      specify volumetric flow rate, Q

   iv) Specify inflow velocity profile

   v) Specify the number of particles per cell.

2) Reflagging

   Reflagging is performed by making,first, a sweep through all the particles to determine which cells contain particles and which cells do not. Then, all SUR cells that no longer contain particles

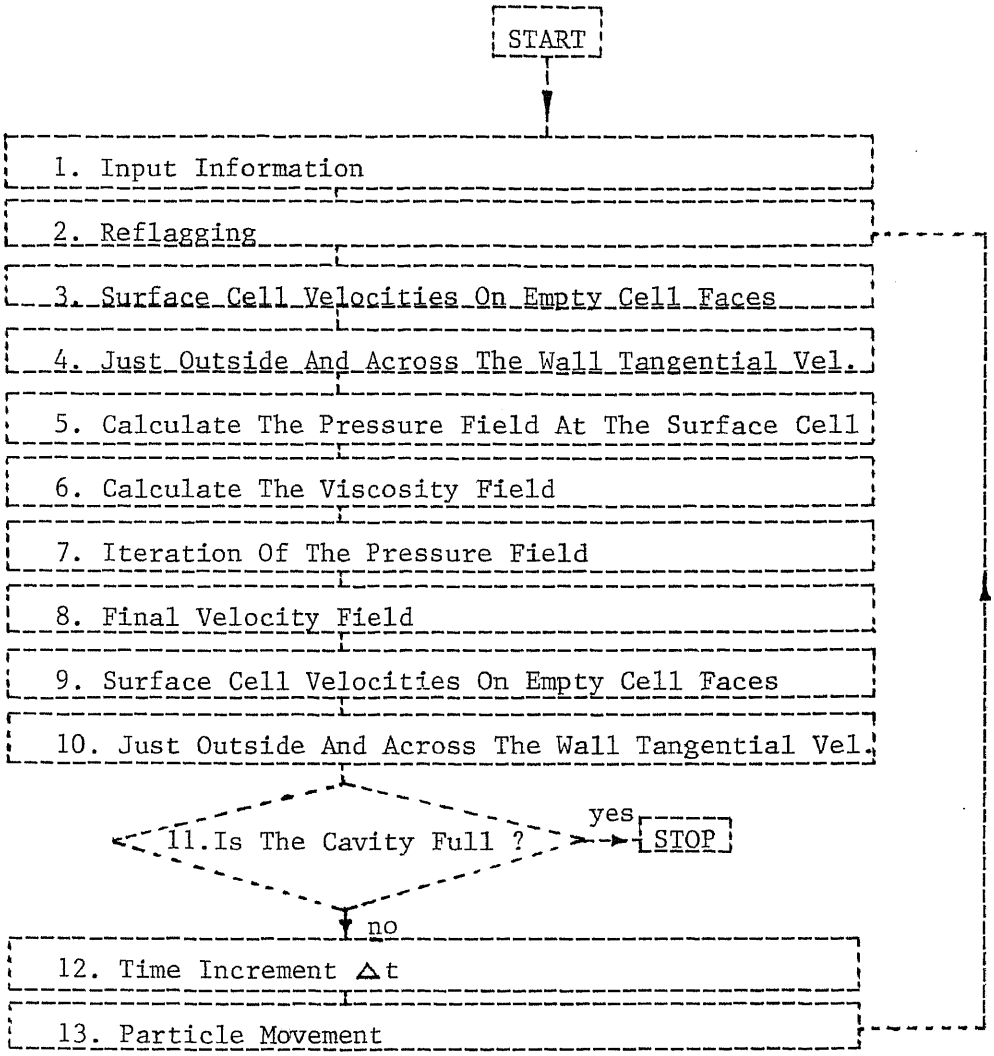


Figure 4.1

Computer Logic Flow Diagram

become EMP cells, and velocities are zeroed on any faces of such cells that are adjacent to other EMP cells. Next step is to make a check on the FUL cells. If FUL cell has any EMP neighbors, it becomes a SUR cell Finally, the SUR cells with no EMP neighbors become FUL cells.

3) SUR Cell Velocities On EMP Cell Faces

These velocities are forced to satisfy the equation of continuity (equation (34)).

4) Just-outside And Across-The-Wall Tangential Velocities

The velocities outside the free surface are adjusted to satisfy the tangential stress condition, equation (41). The wall boundary conditions are used to set the tangential velocities across the walls.

5) Calculate The Pressure Field At The Surface Cells

The pressure field is adjusted to satisfy the normal stress condition at the fluid surface, equation (38) .

6) Calculate The Viscosity Field

This is done with the help of equations (17) and (18).

7) Iteration Of The Pressure Field

The iteration of the equation (20) among all FUL cells leads to the proper pressure field for each cell.

8) Final Velocity Field

The pressure field obtained from can now be applied to equations (15) and (16) to calculate the final velocities field.

9) SUR Cell Velocities On EMP Cell Faces

Same as in step 3

10) Just-outside And Across-The-Wall Tangential Velocities

Same as in step 4

11) Is The Cavity Full ?

As long as there is any EMP cell remaining, the cycle is kept running. Otherwise the program can be stopped.

12) Time Increment δt

The size of time increment δt must consider the stability criterion and consequently equation (42) has to be checked.

13) Particle Movement

The marker particles are moved according to the velocity components in their vicinity as described in equations (22) and (23). After all the particles have been moved, particles are created at the inflow boundary at the prescribed rate.

The control is now passed back to the reflagging step to begin the next cycle.

4.2 Input Of Parameters In The Computer Program

A computer program in FORTRAN code, which is a modification of the original program written by Huang [6], has been used to find a numerical solution of the finite-difference equations developed in the preceding chapter. The rheological properties of glass-bead-filled polypropylene used by Huang [6] were originally investigated by Schmidt [8]. The same values for the parameters of the power law constitutive equation have been used in the present study :

	n	m
Glass-bead-filled polypropylene (ϕ = 0.05)	0.71	20000

The melt density is considered to be approximately equal to 0.735gm/cm^3 at a constant temperature of 240°C.

A listing of the computer program, which had to be modified for compatibility with New Jersey Institute Of Technology UNIVAC Computer, is given in Appendix A.

The input of geometrical parameters to the program includes the dimension of the mold and gate entrance. As discussed earlier, the mold under consideration is of cylindrical shape with a diameter of 1/4" and a length of 2.5" .The gate entrance to the mold is taken as 1/8" x 1/8".

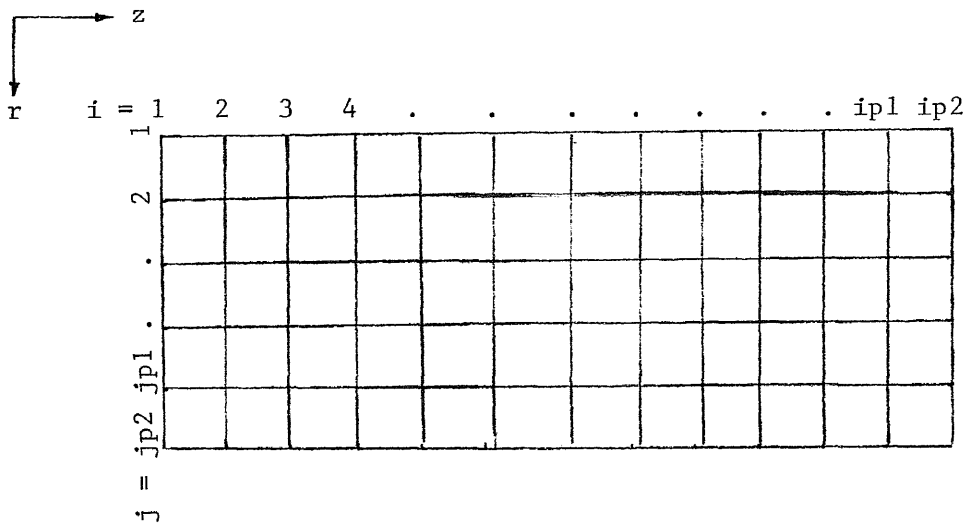
A constant volumetric flow rate of 2.21 in^3/min (0.603 cc/sec) is

used for the purpose of calculation.

Exactly one half of the flow field in r-z plane is considered for computation due to the presence of a line of symmetry along the center line of the cylindrical mold. Consequently, a 7 x 72 grid system, with a total number of cells equalling 504, is used for our purpose. Thus, the dimension of r-z plane becomes 1/8" x 2.5" (0.3175cm x 6.35cm) and the grid size equals δz = 0.0907 cm by δr = 0.03175 cm. An average of 10 particles per cell is maintained to represent the fluid flow and approximately 50 minutes of computation time is necessary for a complete run of the program.

Figure 4.2

Schematic Diagram of The 7 x 72 Grid System



ip2 = 72 ipl = 71 jp2 = 7 jpl = 6

Δr = 0.03175 cm Δz = 0.0907 cm

CHAPTER V

ANALYSIS OF COMPUTATIONAL RESULTS

In this chapter we will be trying to present and analyze the massive amount of data generated by the computer program in our simulation studies. Since the purpose of the present study is to visualize the flow phenomena as the glass-bead-filled polypropylene melt enters the cylindrical cavity, we will be particularly interested in the velocity and pressure distribution along the cavity as the flow front advances.

Srictly speaking, the flow in the specific mold is a three-dimensional flow. However, as mentioned earlier, because of the inherent mathematical difficulty in solving a three-dimensional flow we are compelled to use the two-dimensional flow model. Consequently, the velocity distributions can be plotted in the axial and transverse directions only.

A quantitative information of the flow details is provided by Figures 5.1 through 5.8. These plots are velocity profiles in the axial direction at different times. The value of z_d indicates the distance from the entrance of the mold.The velocity profiles show how the front surface moves with time and also the fact that the velocity decelerates in the flow direction. This is evident from the curves which show a gradual tendency of flattening out as we move further away from the entrance. The velocity at the wall is zero in each case and it

registers a pattern of sharp deceleration as we come near the wall.

Velocity Profile In The Axial Direction, V_z

For Various Position From The Entrance, z_d

Time = 0.107 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

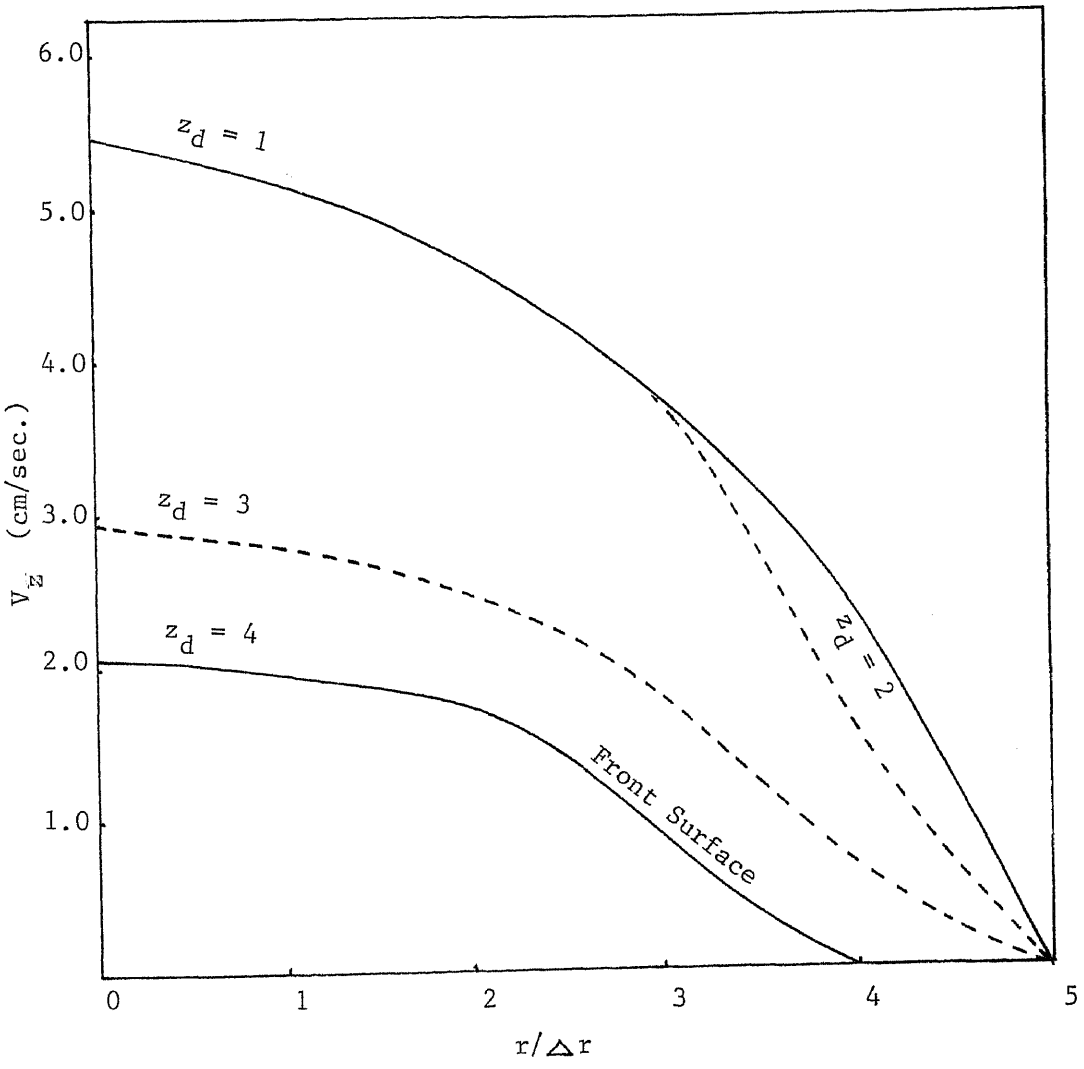


Figure 5.1

Velocity Profile In The Axial Direction, V_z

For Various Position From The Entrance, z_d

Time = 0.322 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

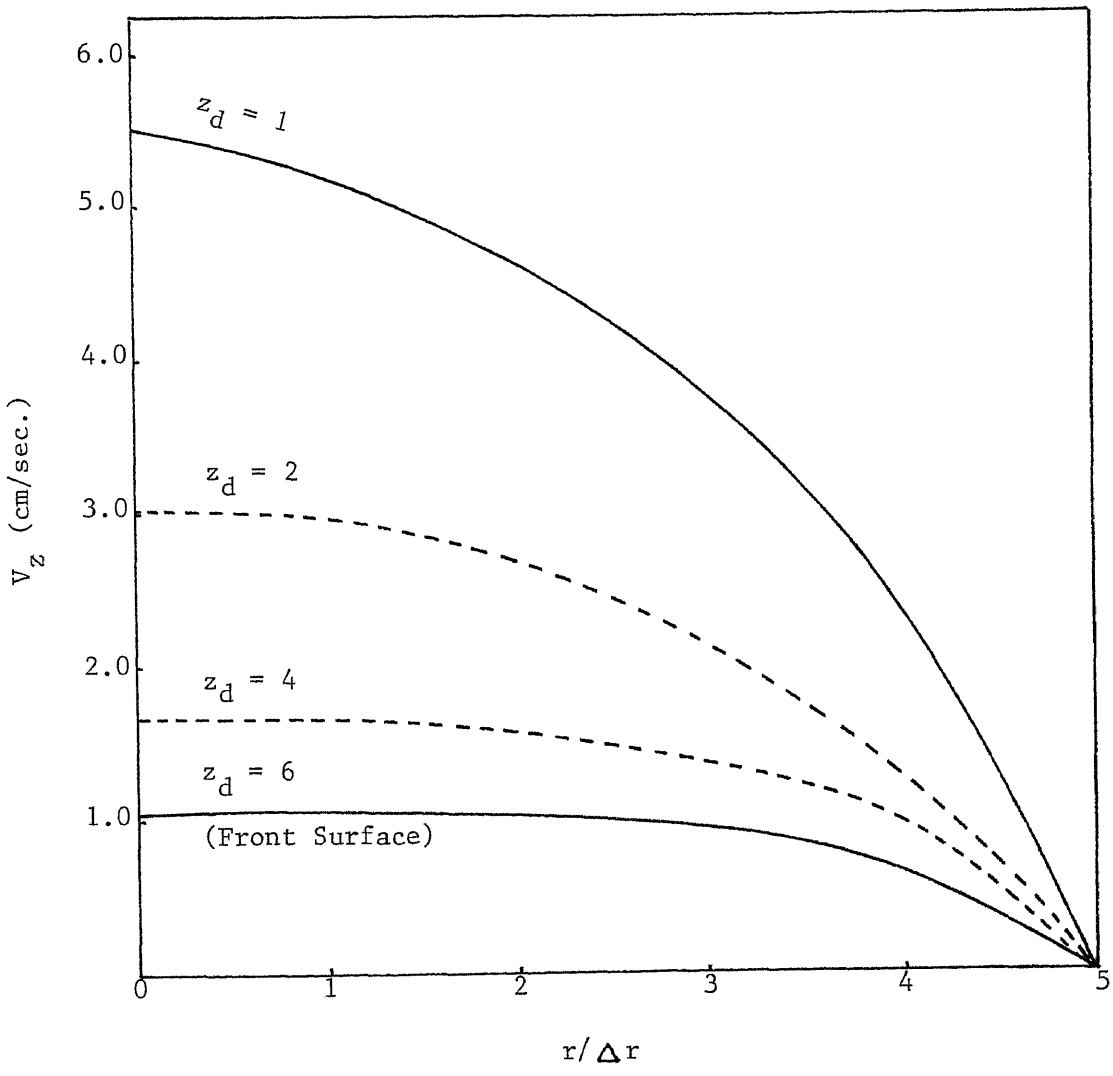


Figure 5.2

Velocity Profile In The Axial Direction, V_z

For Various Position From The Entrance, z_d

Time = 0.742 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

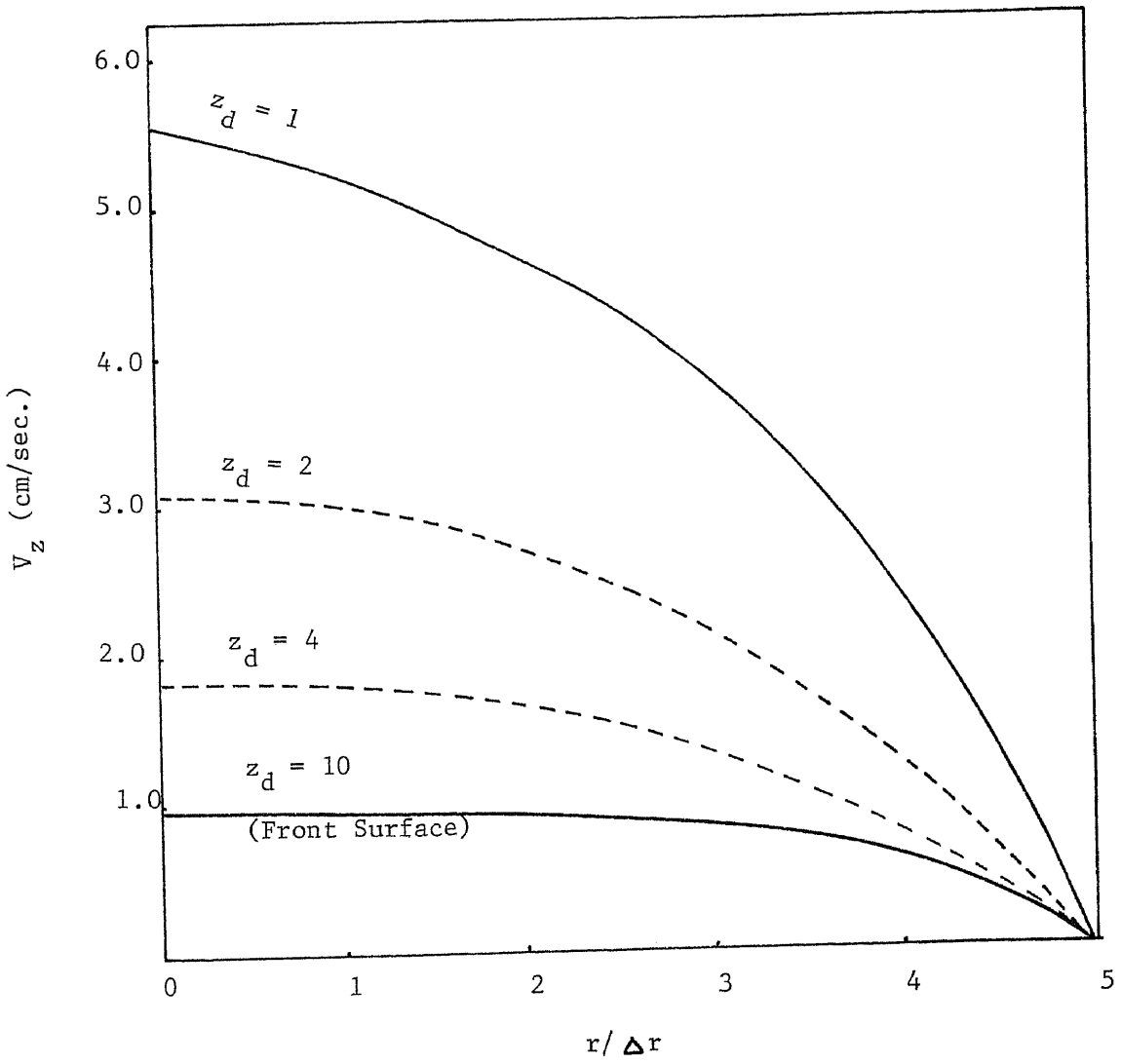


Figure 5.3

Velocity Profile In The Axial Direction, V_z

For Various Position From Entrance, z_d

Time = 1.393 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

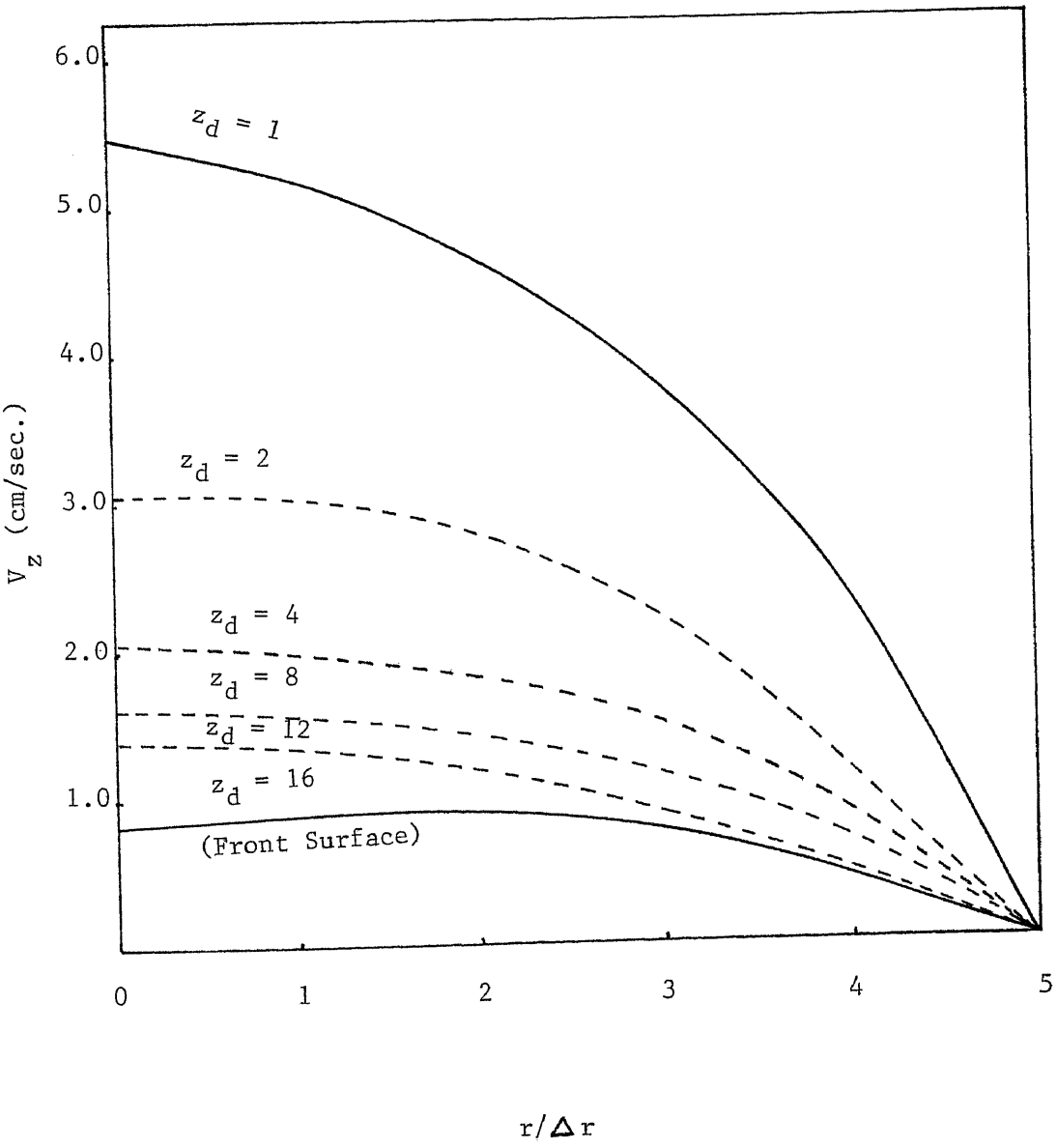


Figure 5.4

Velocity Profile In The Axial Direction, V_z

For Various Position From The Entrance, z_d

Time = 2.49 seconds $\quad z_d = z/\Delta z$

$\Delta z = 0.0907$ cm $\quad \Delta r = 0.03175$ cm

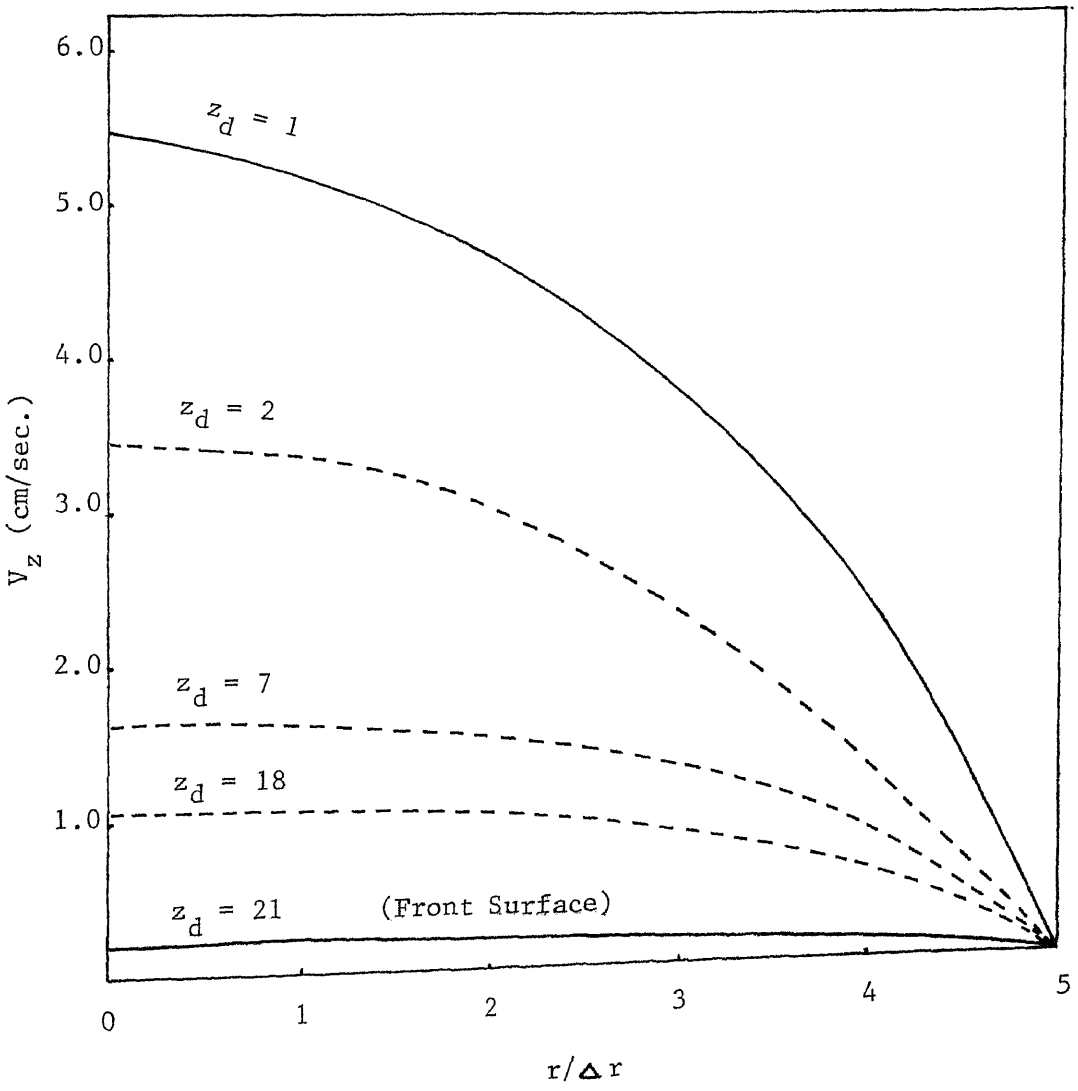


Figure 5.5

Velocity Profile In The Axial Direction, V_z

For Various Position From Entrance, z_d

Time = 3.797 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

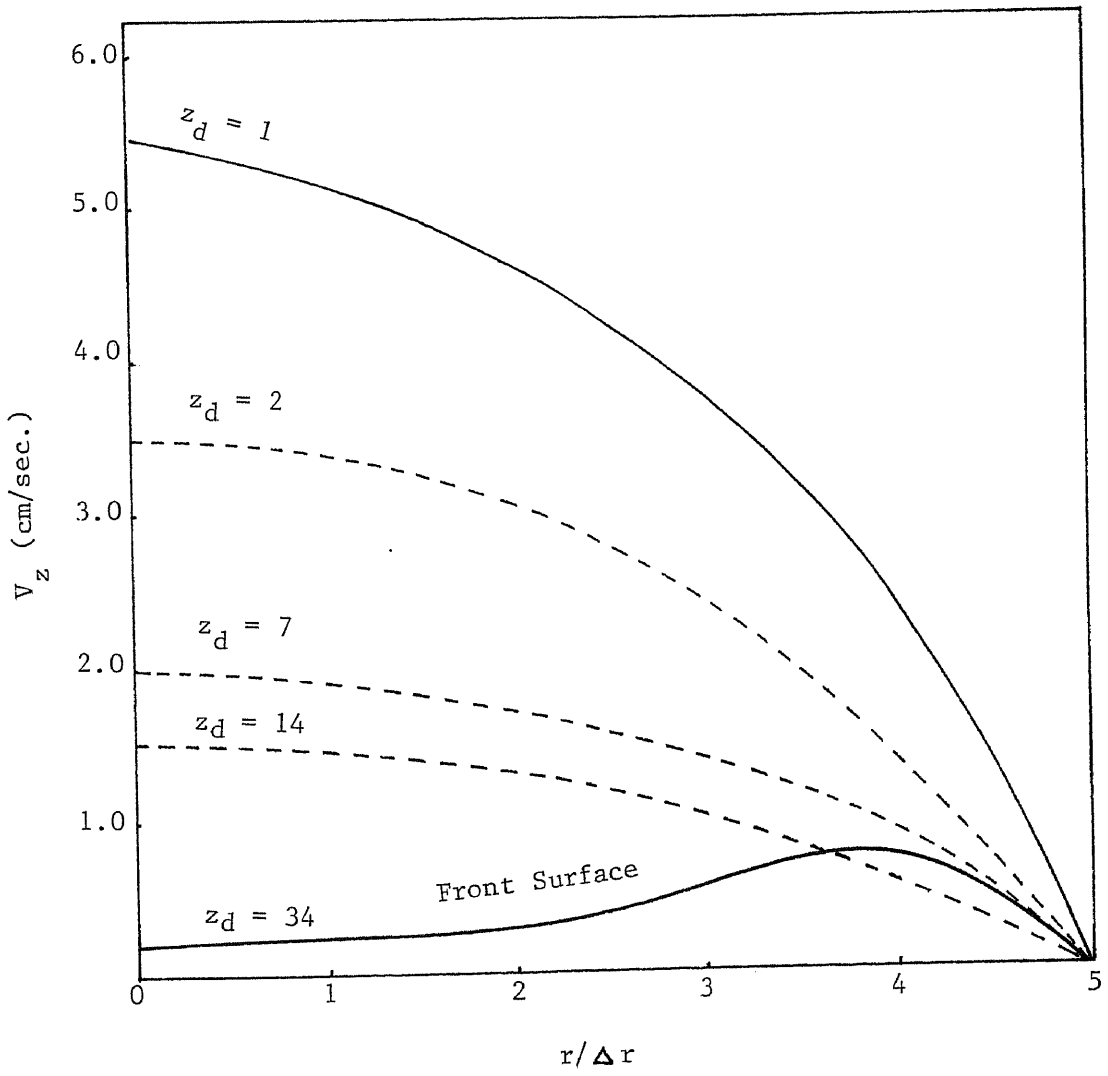


Figure 5.6

Velocity Profile In The Axial Direction, V_z

For Various Position From The Entrance, z_d

Time = 5.91 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

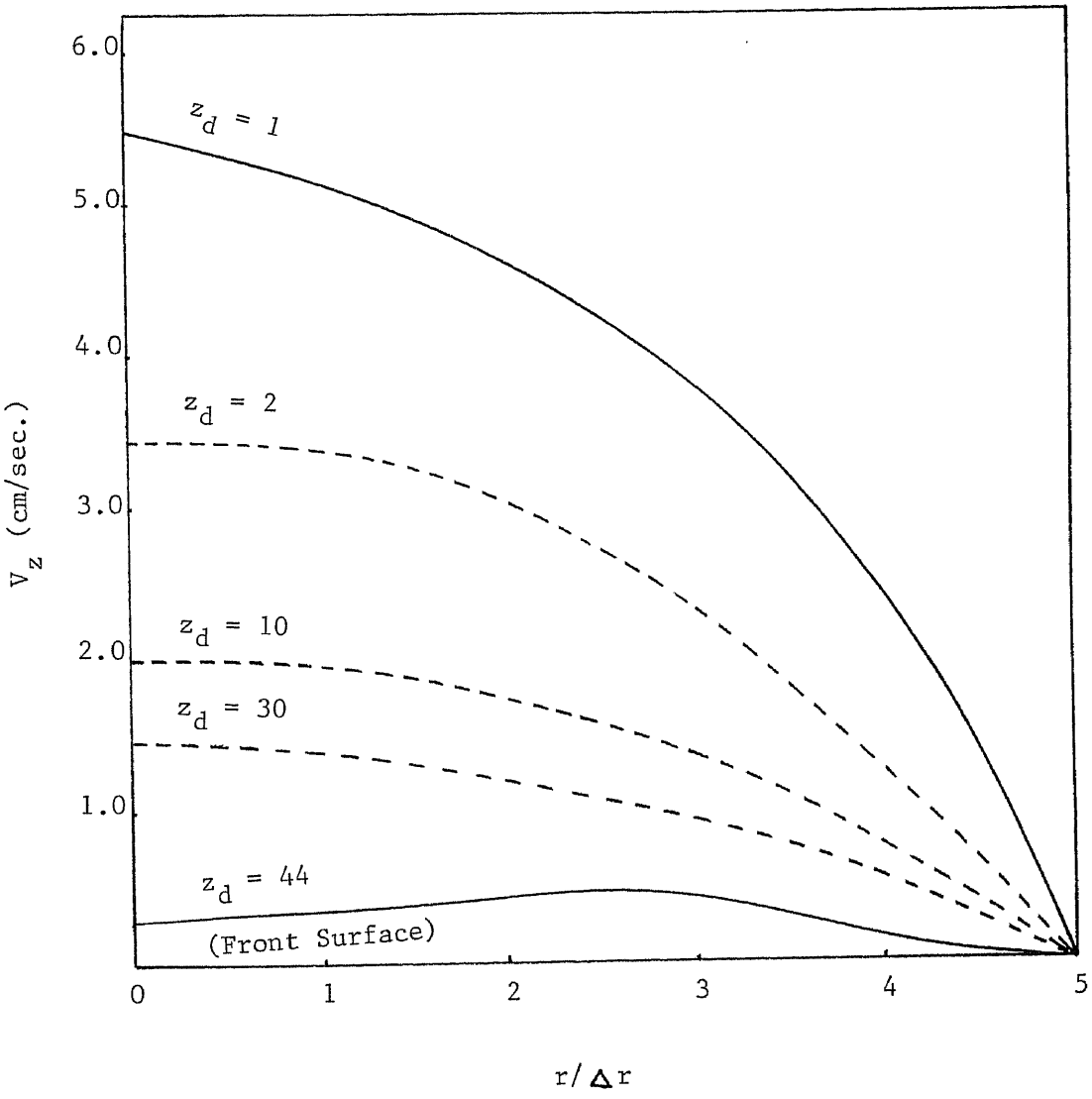


Figure 5.7

Velocity Profile In The Axial Direction, V_z

For Various Position From The Entrance, z_d

Time = 8.39 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

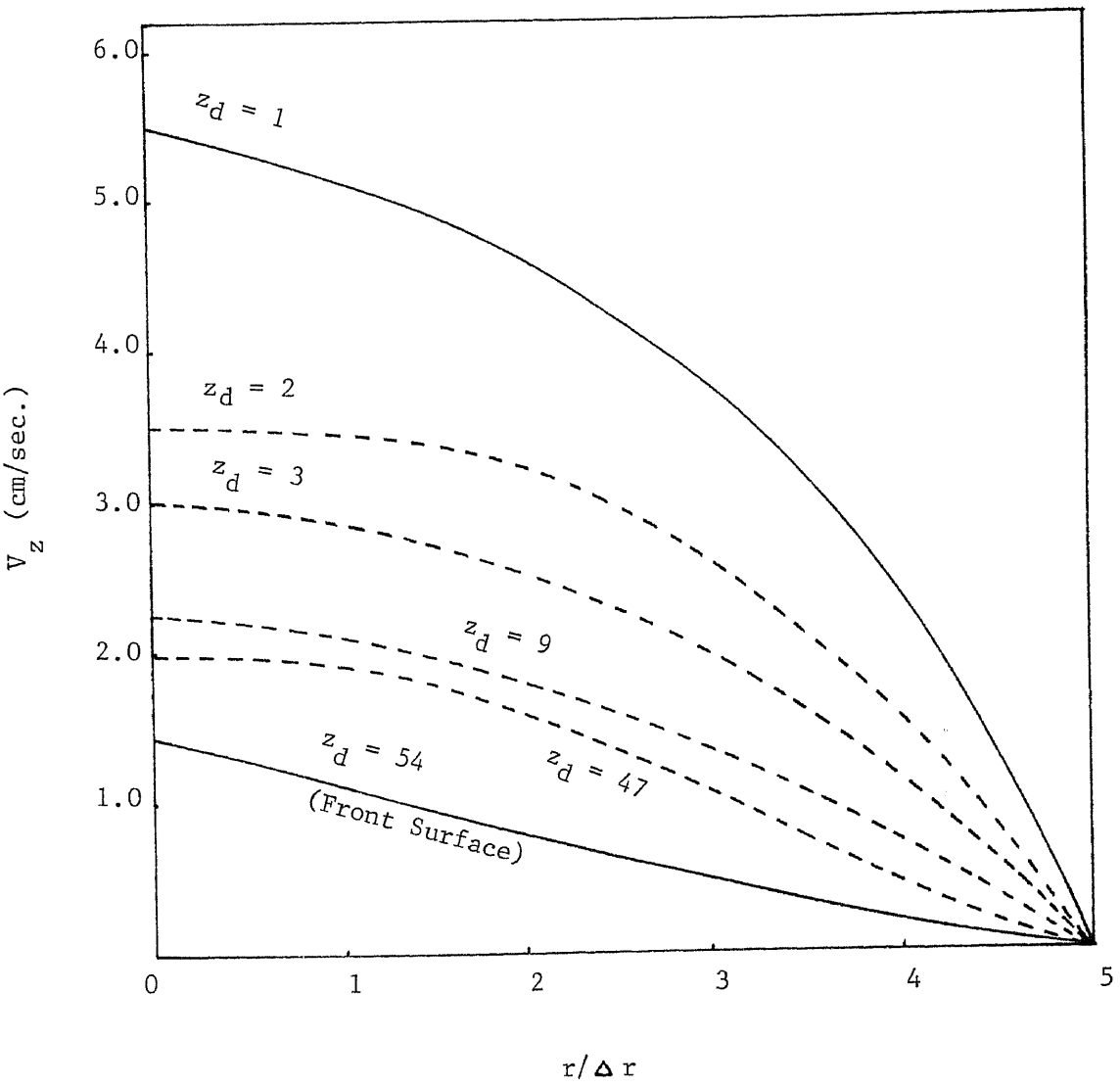


Figure 5.8

The transverse velocity profiles are plotted in Figures 5.9 through 5.16 .It can be observed from these diagrams that the velocity in the transverse direction is comparatively small in the region near the entrance and reaches a significant value as we move closer to the front surface region. This indicates an intensification of two-dimensional flow phenomena near the front region. The transverse velocity, as should be expected in this case, is observed to be zero both at the wall and the centerline. However, the velocity of the front surface at the centerline has a non-zero magnitude.

Transverse Velocity Profile, V_r

For Various Position From The Entrance, z_d

Time = 0.107 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

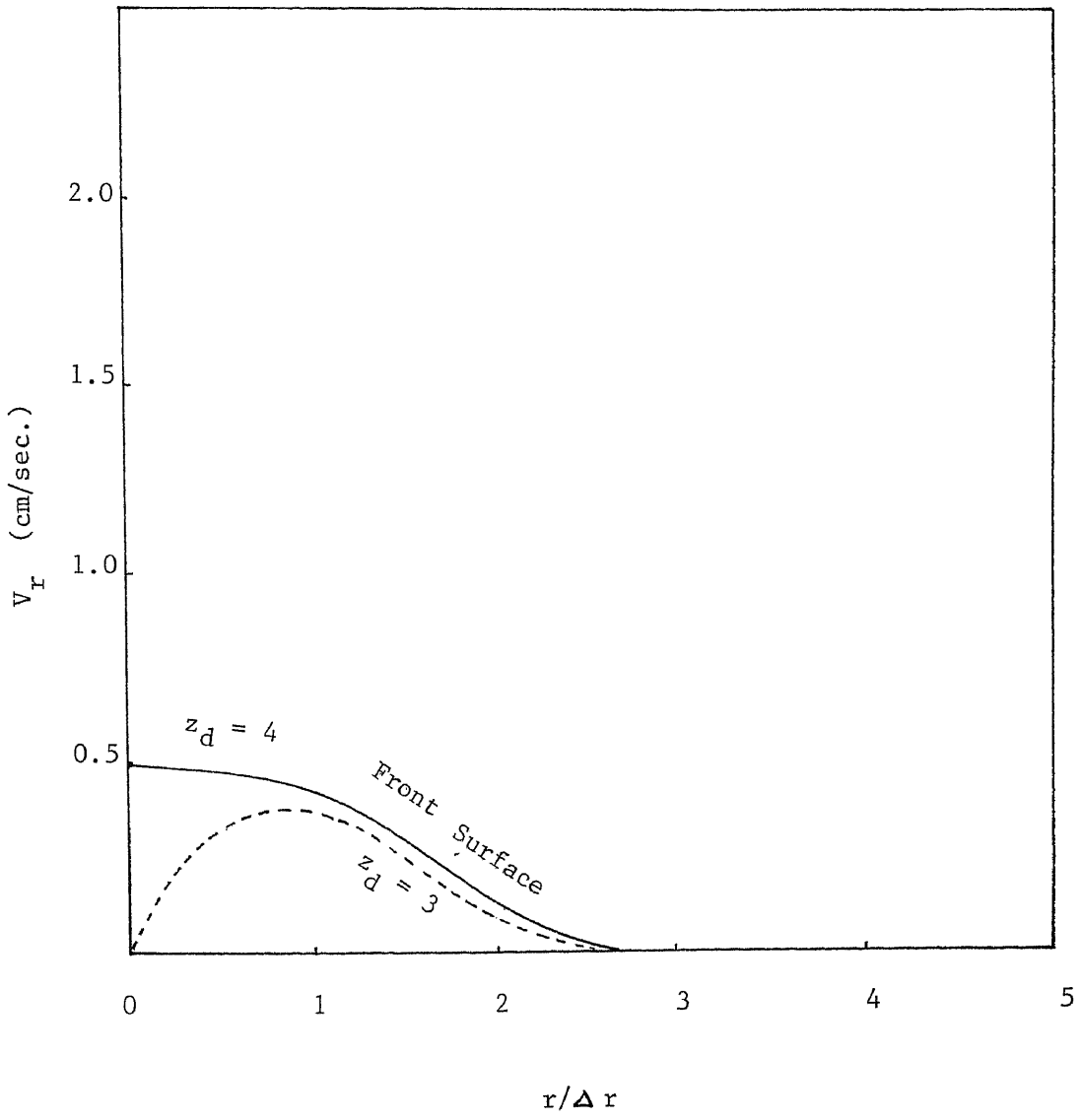


Figure 5.9

Transverse Velocity Profile, V_r

For Various Position From The Entrance, z_d

Time = 0.322 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

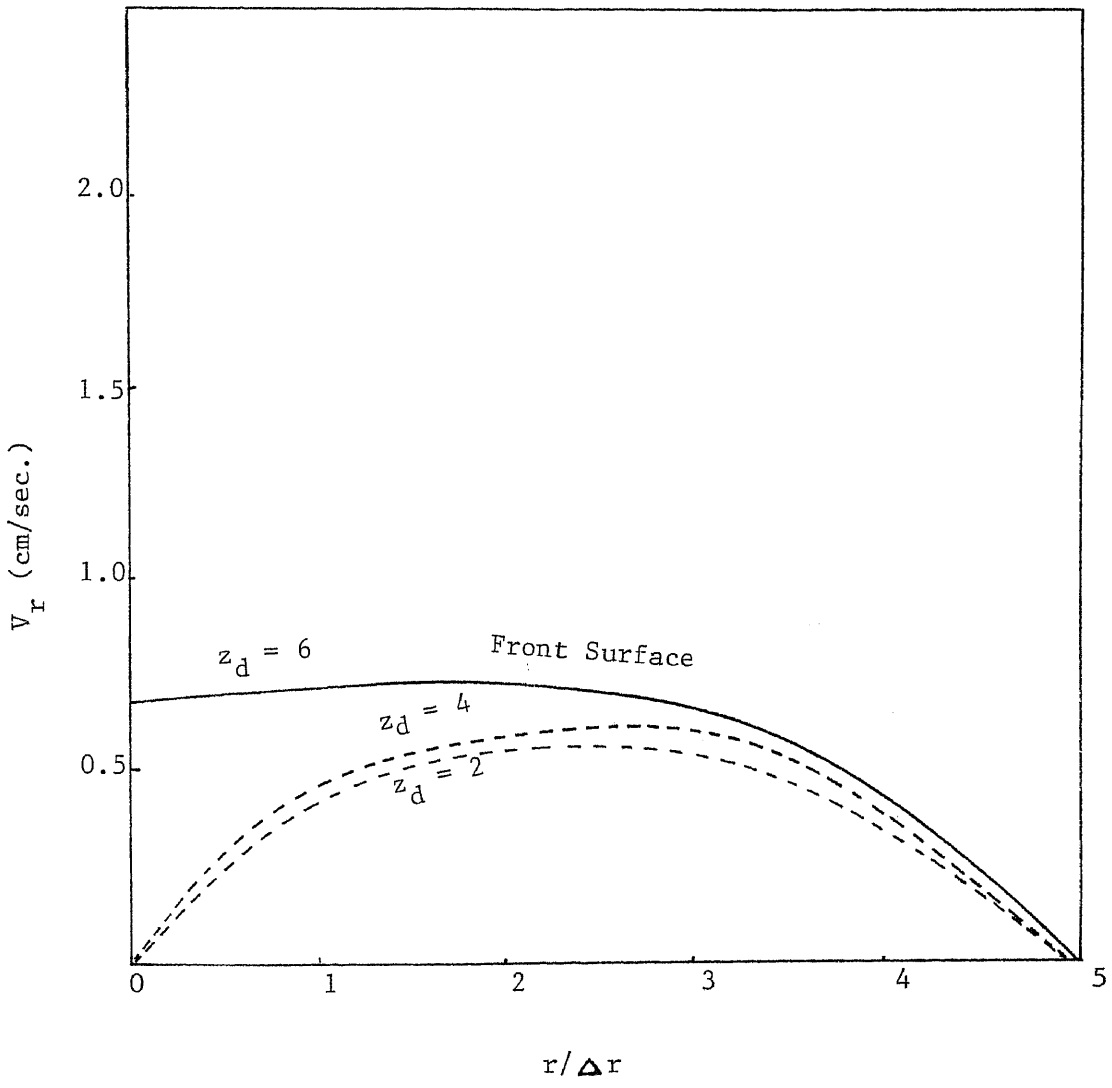


Figure 5.10

Transverse Velocity Profile, V_r

For Various Position From The Entrance, z_d

Time = 0.742 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

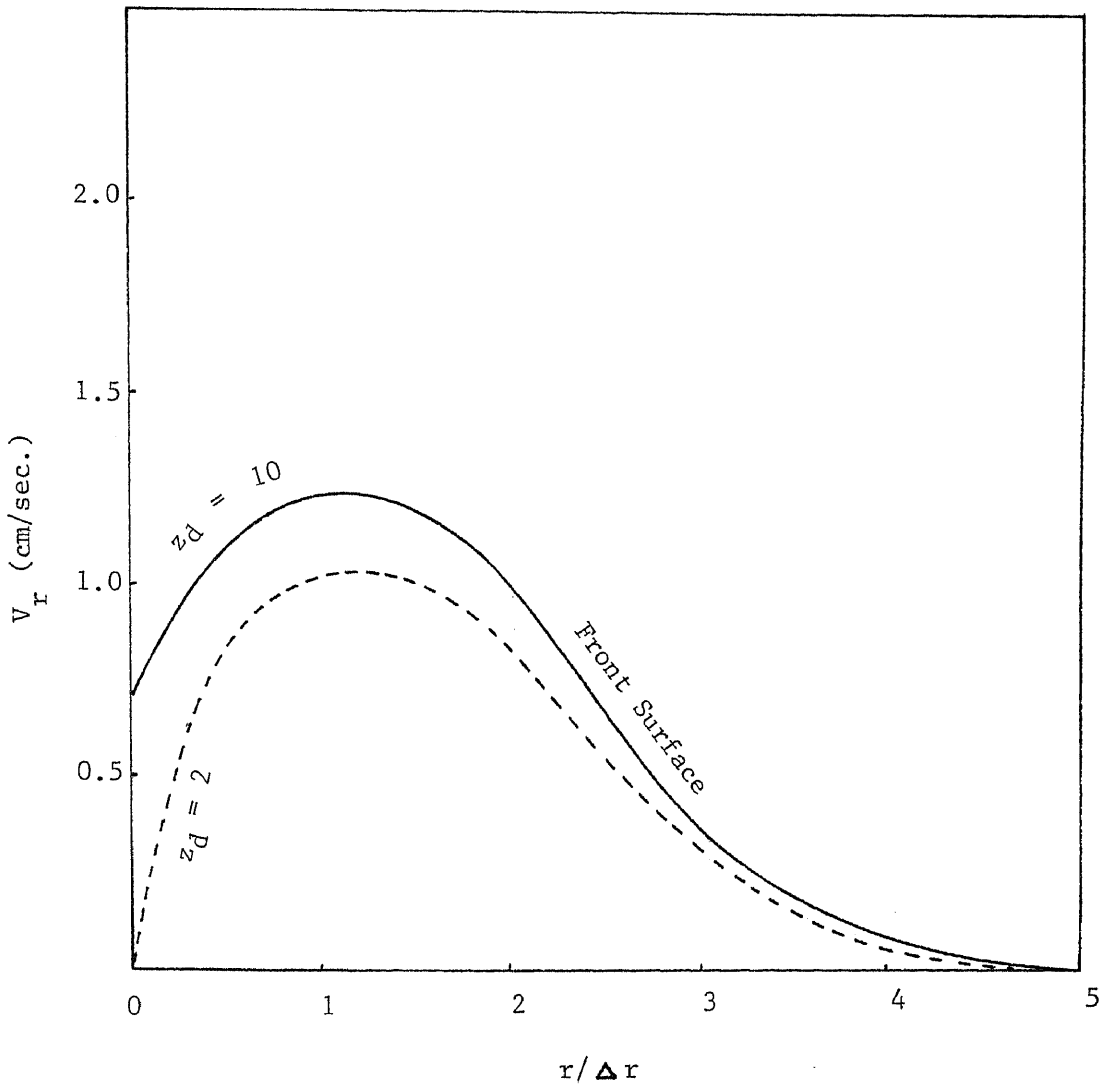


Figure 5.11

Transverse Velocity Profile, V_r

For Various Positions From The Entrance, z_d

Time = 1.393 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

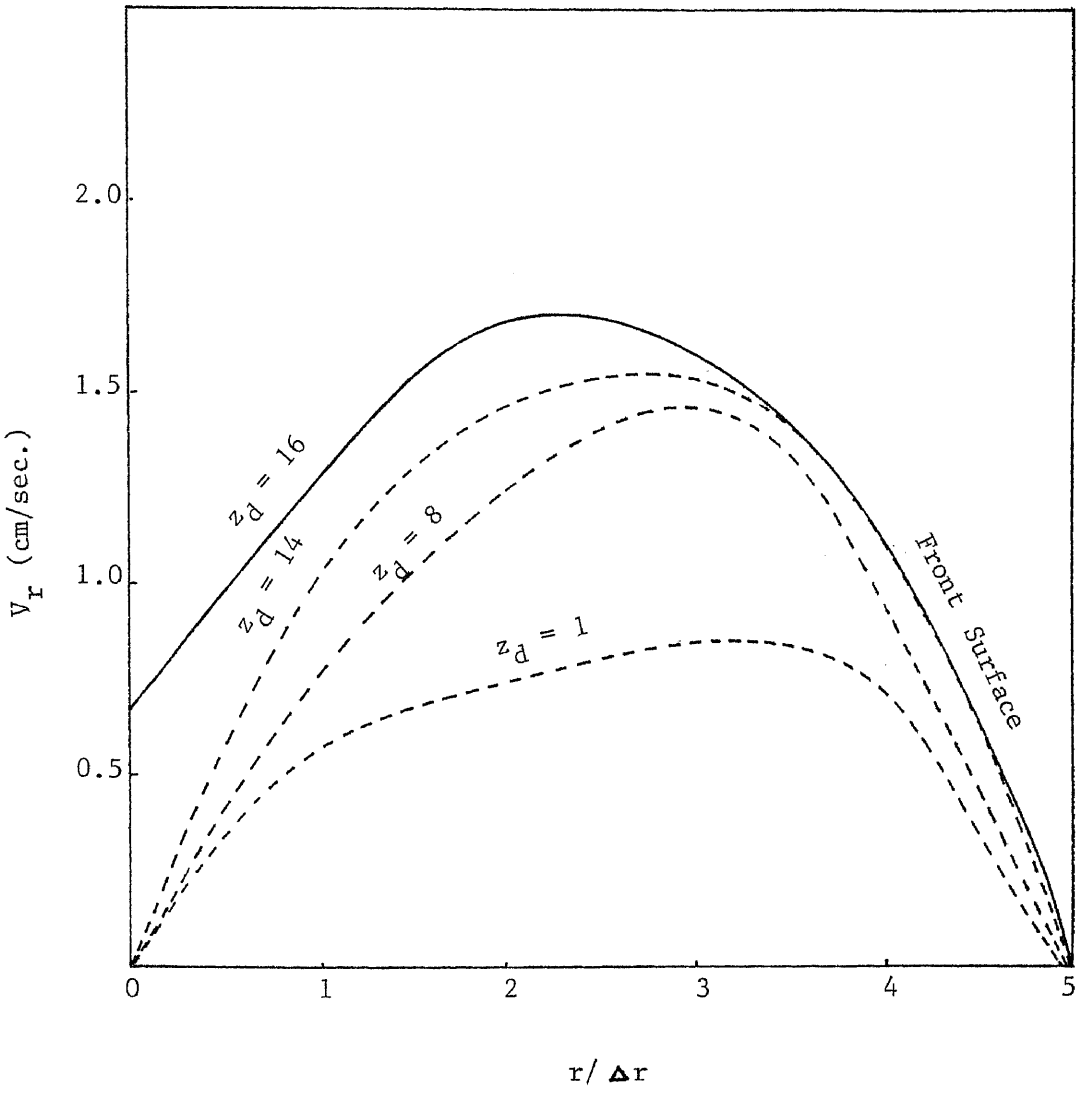


Figure 5.12

Transverse Velocity Profile, V_r

For Various Positions From The Entrance, z_d

Time = 2.489 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

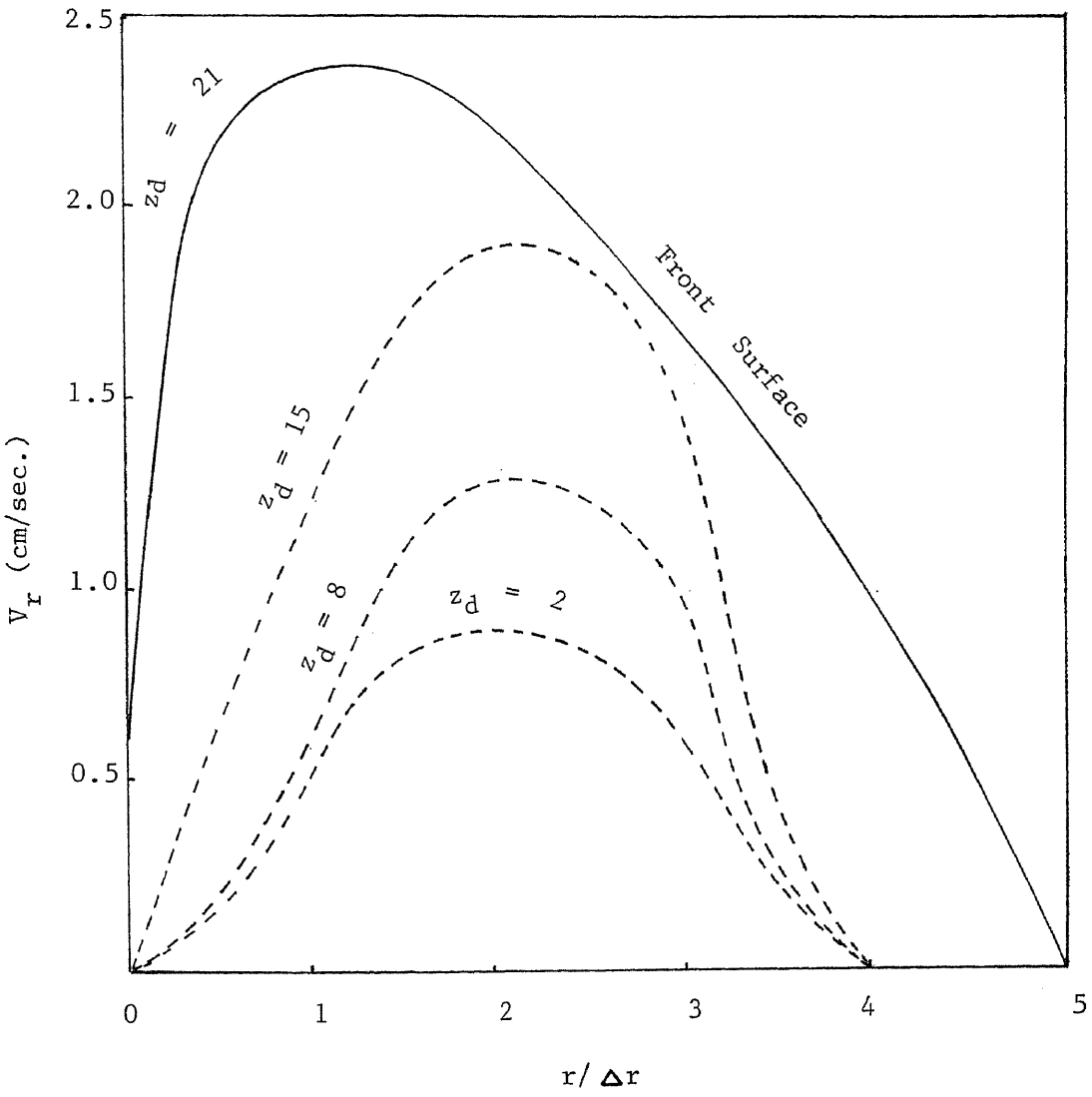


Figure 5.13

Transverse Velocity Profile, V_r

For Various Positions From The Entrance, z_d

Time = 3.797 seconds $\quad z_d = z/\Delta z$

Δz = 0.0907 cm $\quad \Delta r$ = 0.03175 cm

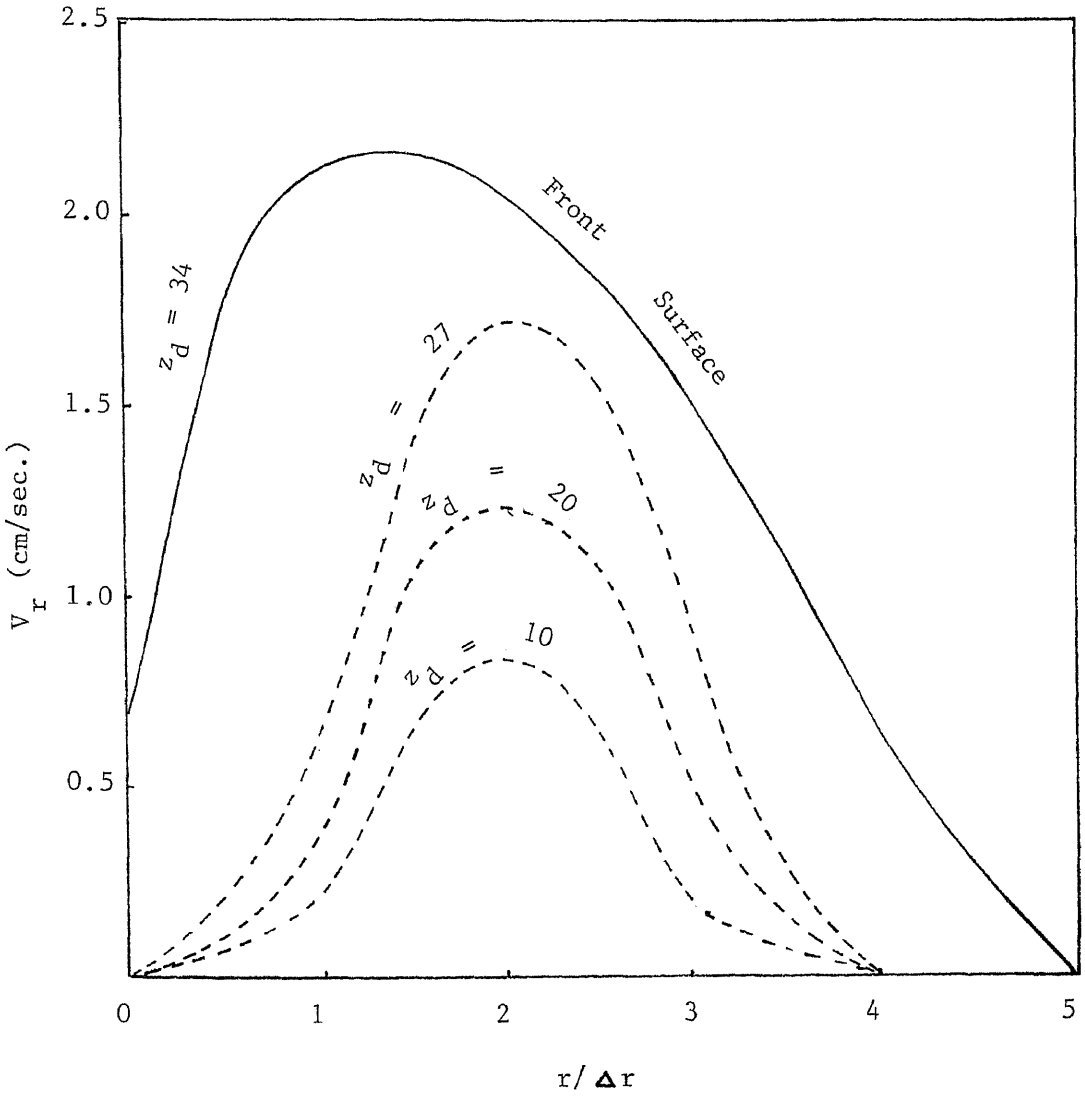


Figure 5.14

Transverse Velocity Profile, V_r

For Various Positions From The Entrance, z_d

Time = 5.909 seconds $\quad z_d = z/\Delta z$

$\Delta z = 0.0907$ cm $\quad \Delta r = 0.03175$ cm

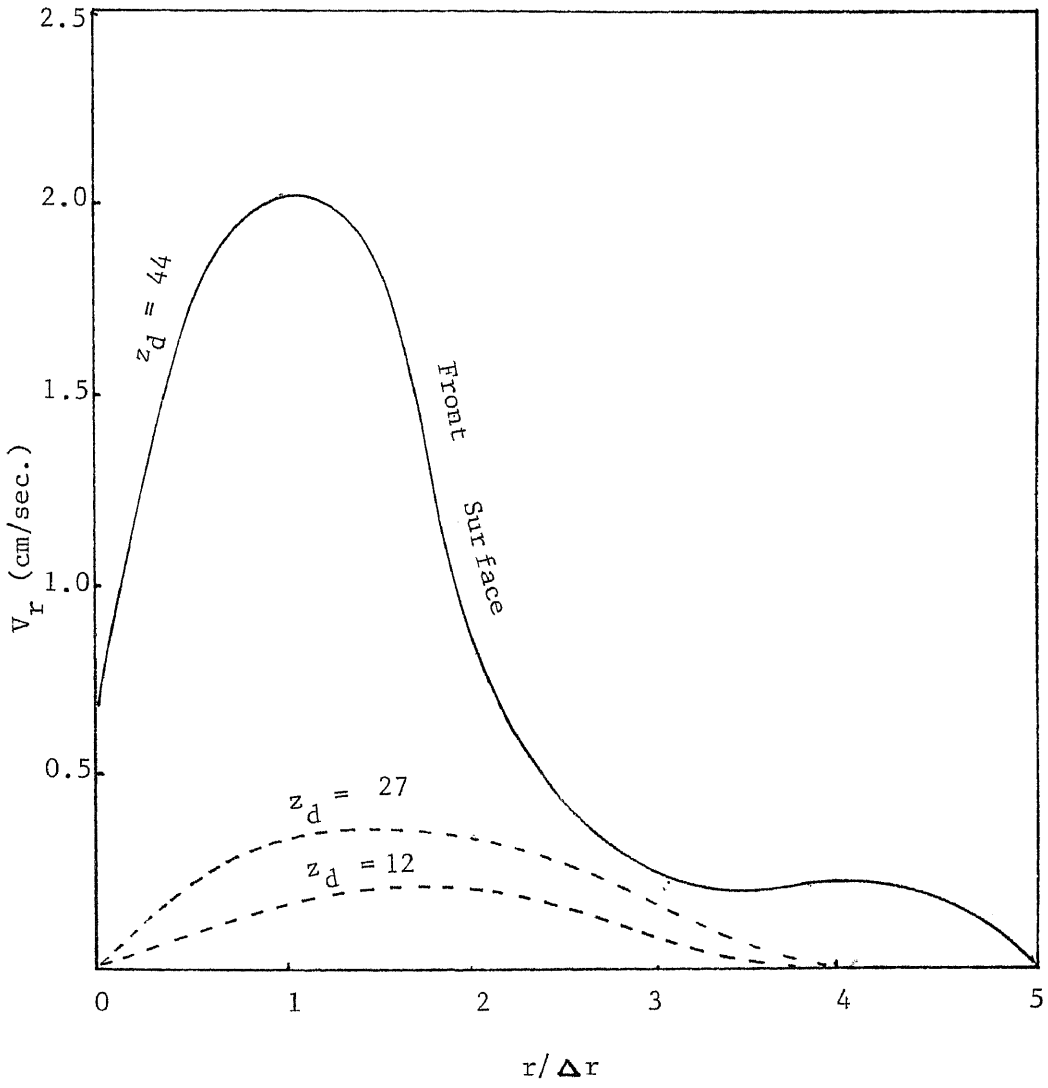


Figure 5.15

Transverse Velocity Profile, V_r

For Various Positions From The Entrance, z_d

Time = 8.39 seconds $\quad z_d = z/\Delta z$

$\Delta z = 0.0907$ cm $\quad \Delta r = 0.03175$ cm

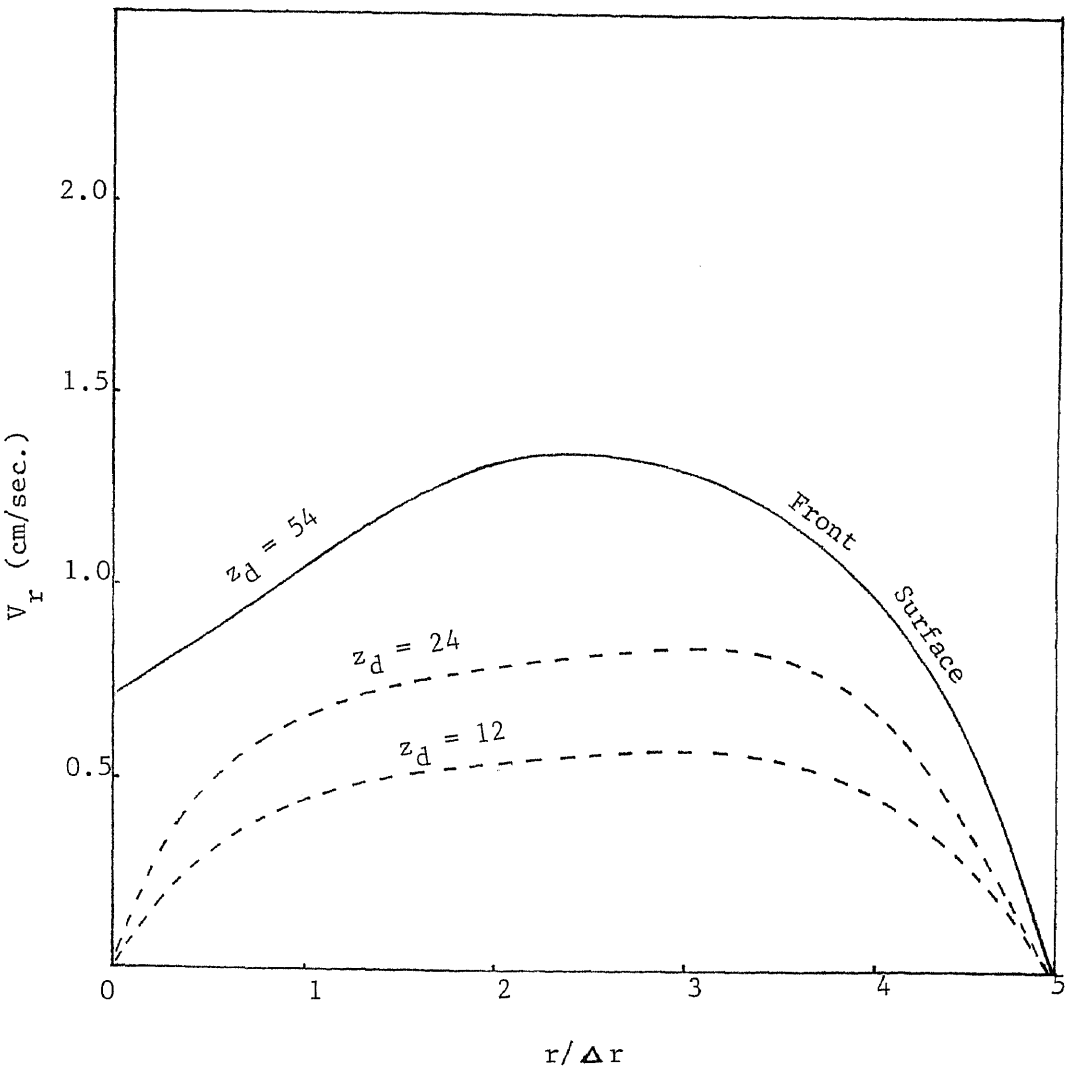


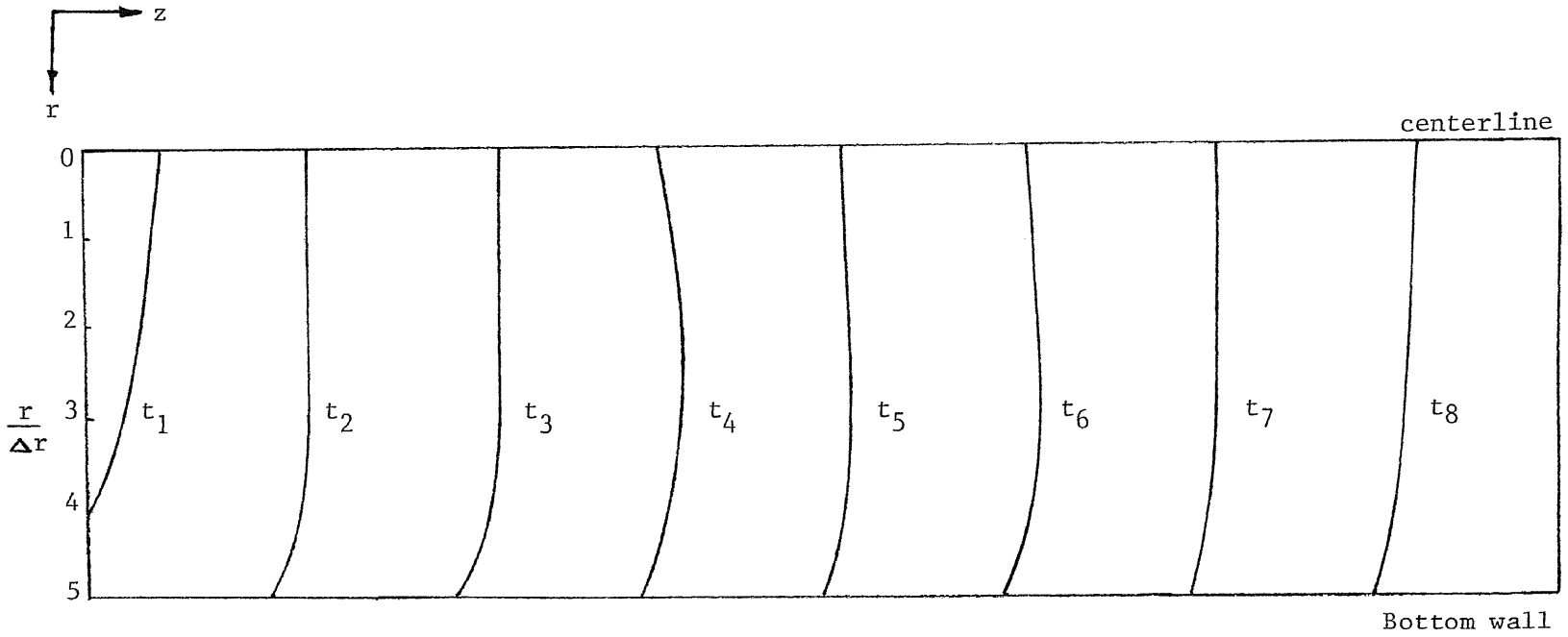
Figure 5.16

Based on the velocity profiles plotted earlier, we are now in a position to draw the flow front progression in r-z plane at various times. The plot, as shown in Fig. 5.17 , is drawn over half of the symmetrical mold cavity. The values of $r/\triangle r$ equal to 0 and 5 represent the centerline and bottom wall respectively.

Except for the initial time of 0.107 seconds, the shape of the flow front becomes flatter and flatter as the time progresses. In each case it reaches a maximum value at the centerline and falls off to zero as we approach the bottom wall.

Figure 5.17

Flow Front Progression in r-z plane at various times.



t_1 = 0.107 sec t_2 = 0.322 sec t_3 = 0.742 sec t_4 = 1.393 sec

t_5 = 2.489 sec t_6 = 3.797 sec t_7 = 5.909 sec t_8 = 8.39 sec

Figures 5.18 through 5.25 show the pressure distribution across the mold at different times. A dimensionless form is utilized by taking the ratio of local and maximum pressure in each case. From the figures it is observed that the pressure falls off to zero at the wall,as was expected in the case of velocity profiles discussed earlier. We can also observe from the distribution plots that, at each indicated time, over half of the filled cavity has reached a pressure of 50% of the maximum pressure or higher.

Finally, we can conclude this investigation by saying that such simulation of the mold filling process with Marker And Cell numerical technique is indeed a powerful tool for the detailed study of the velocity profiles and pressure distribution in two dimensional isothermal problems. Although such simulation works have already been reported in the literature, the present study purports to be an extension of the earlier work done by Kamal and Kenig[1] (using semi-circular mold geometries) and Huang[6] who used a flow region between two rigid parallel plate boundaries.

Pressure Distribution At Various Positions From The Entrance

Time = 0.107 seconds

$z_d = z/\Delta z$

$\Delta z = 0.0907$ cm

$\Delta r = 0.03175$ cm

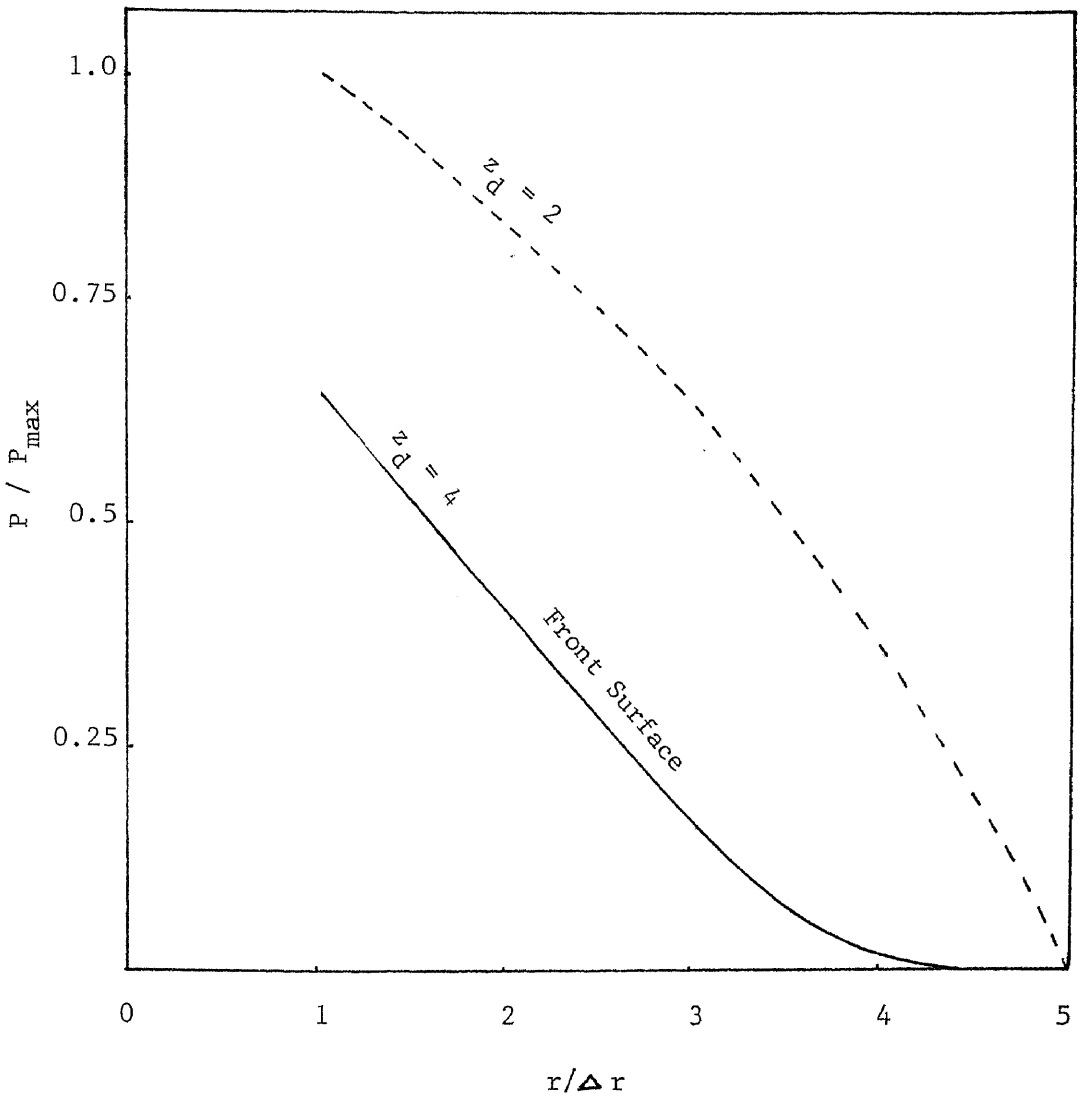


Figure 5.18

Pressure Distribution At Various Positions From The Entrance

Time = 0.322 seconds $z_d = z/\Delta z$

Δz = 0.0907 cm Δr = 0.03175 cm

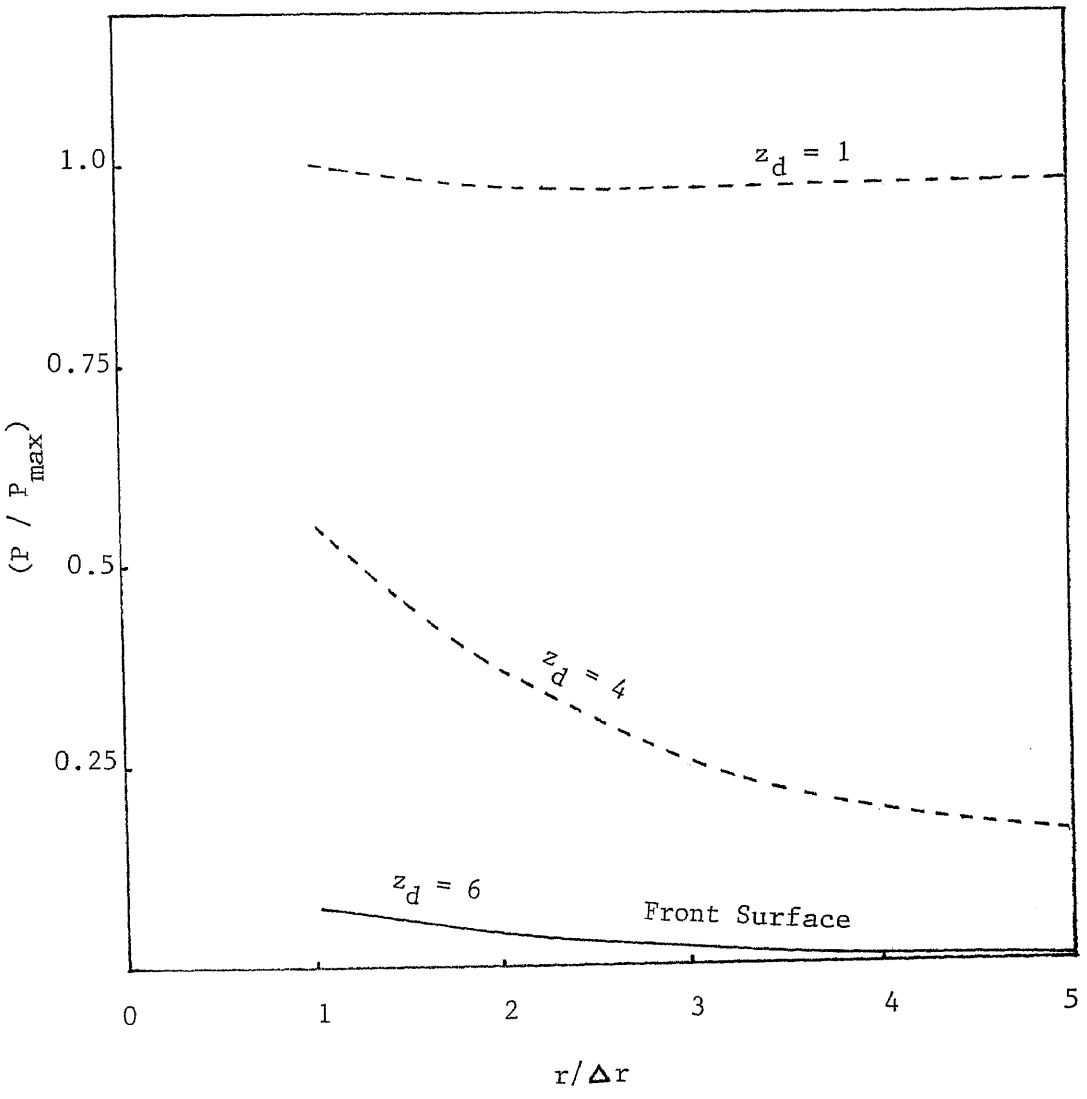


Figure 5.19

Pressure Distribution At Various Positions From The Entrance

Time = 0.742 seconds

$z_d = z/\Delta z$

$\Delta z = 0.0907$ cm

$\Delta r = 0.03175$ cm

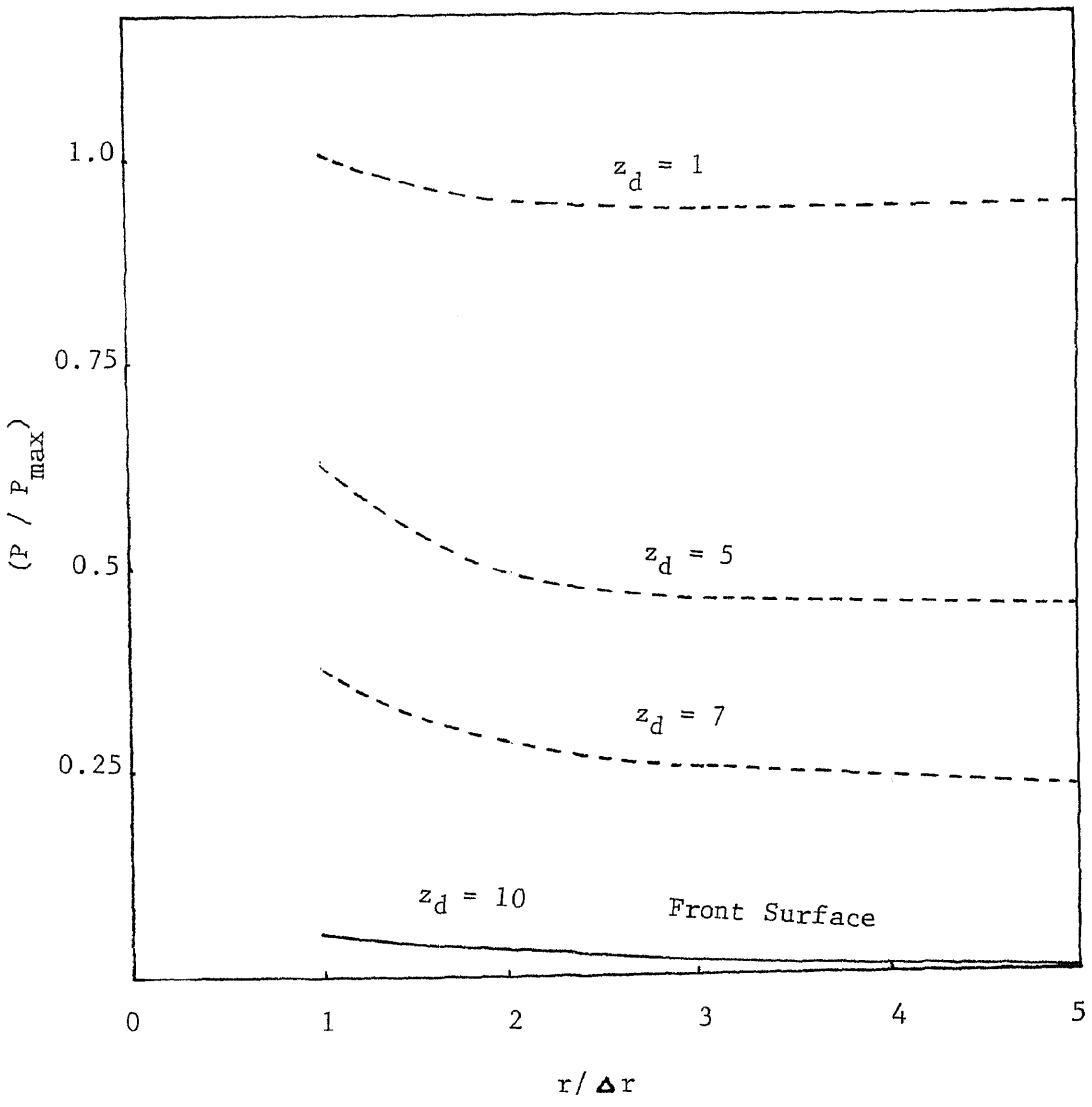


Figure 5.20

Pressure Distribution At Various Positions From The Entrance

Time = 1.393 seconds

Δz = 0.0907 cm

$z_d = z/\Delta z$

Δr = 0.03175 cm

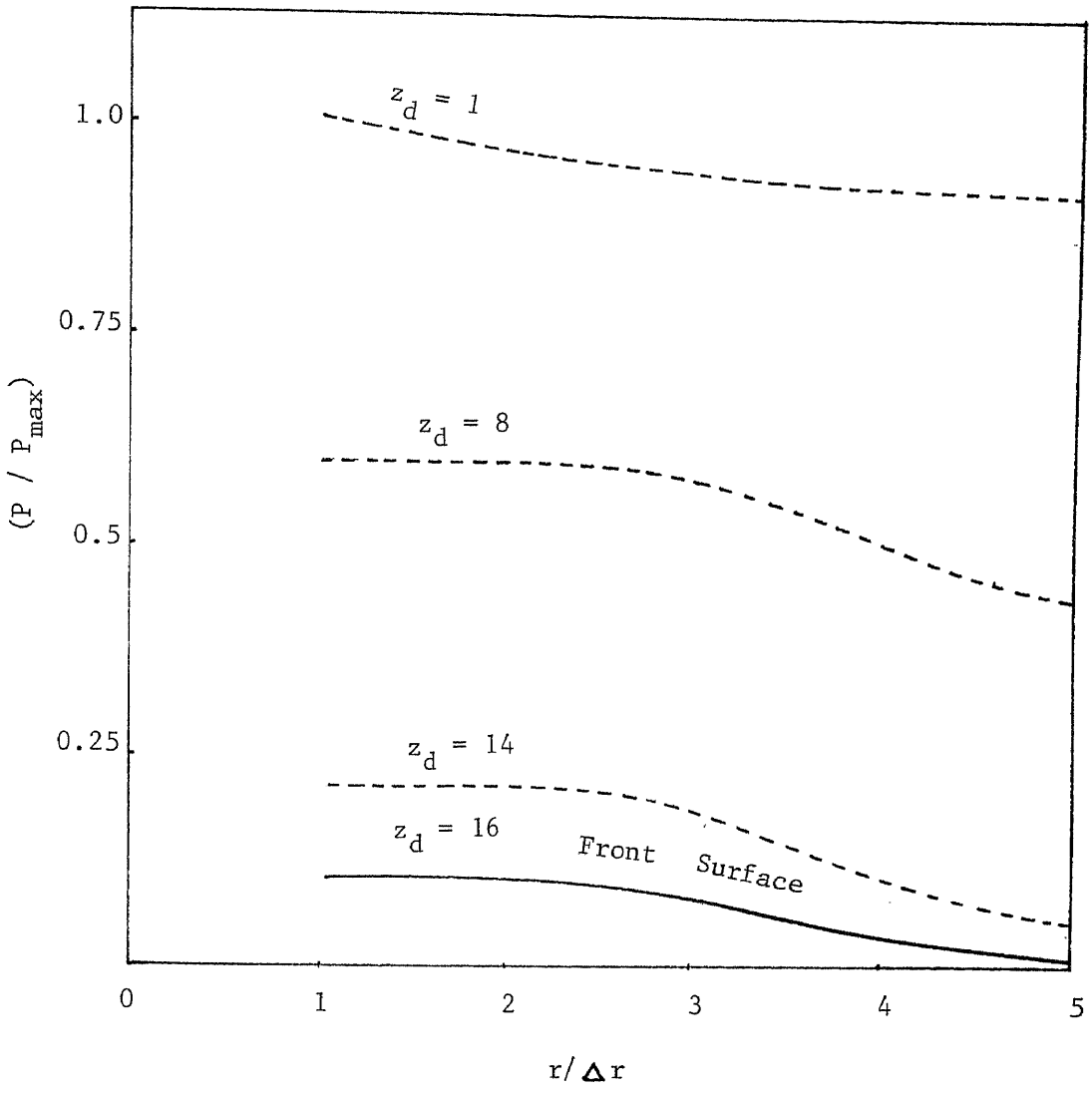


Figure 5.21

Pressure Distribution At Various Positions From The Entrance

Time = 2.489 seconds

$z_d = z/\Delta z$

$\Delta z = 0.0907$ cm

$\Delta r = 0.03175$ cm

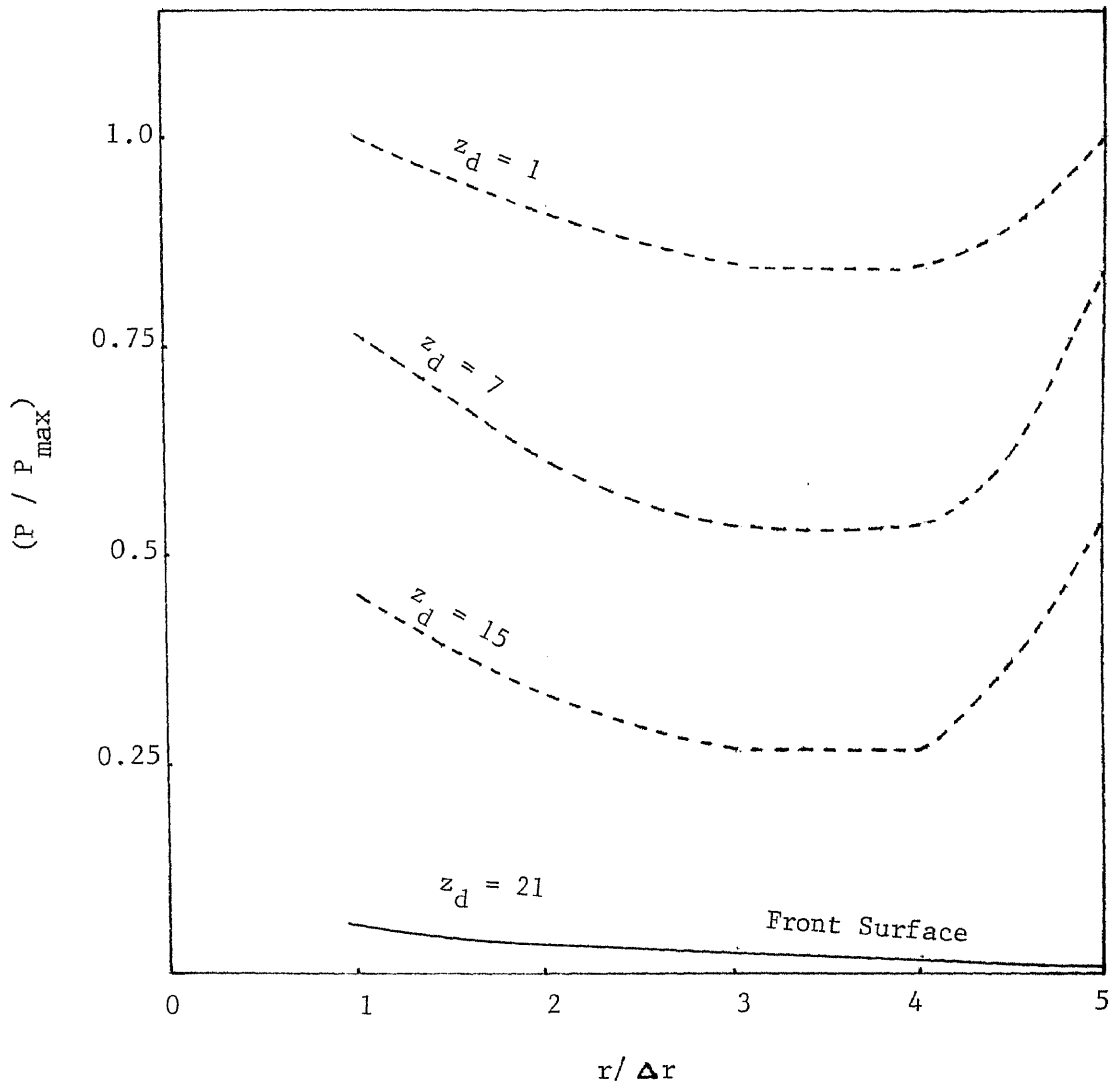


Figure 5.22

Pressure Distribution At Various Positions From The Entrance

Time = 3.797 seconds

Δz = 0.0907 cm

$z_d = z/\Delta z$

Δr = 0.03175 cm

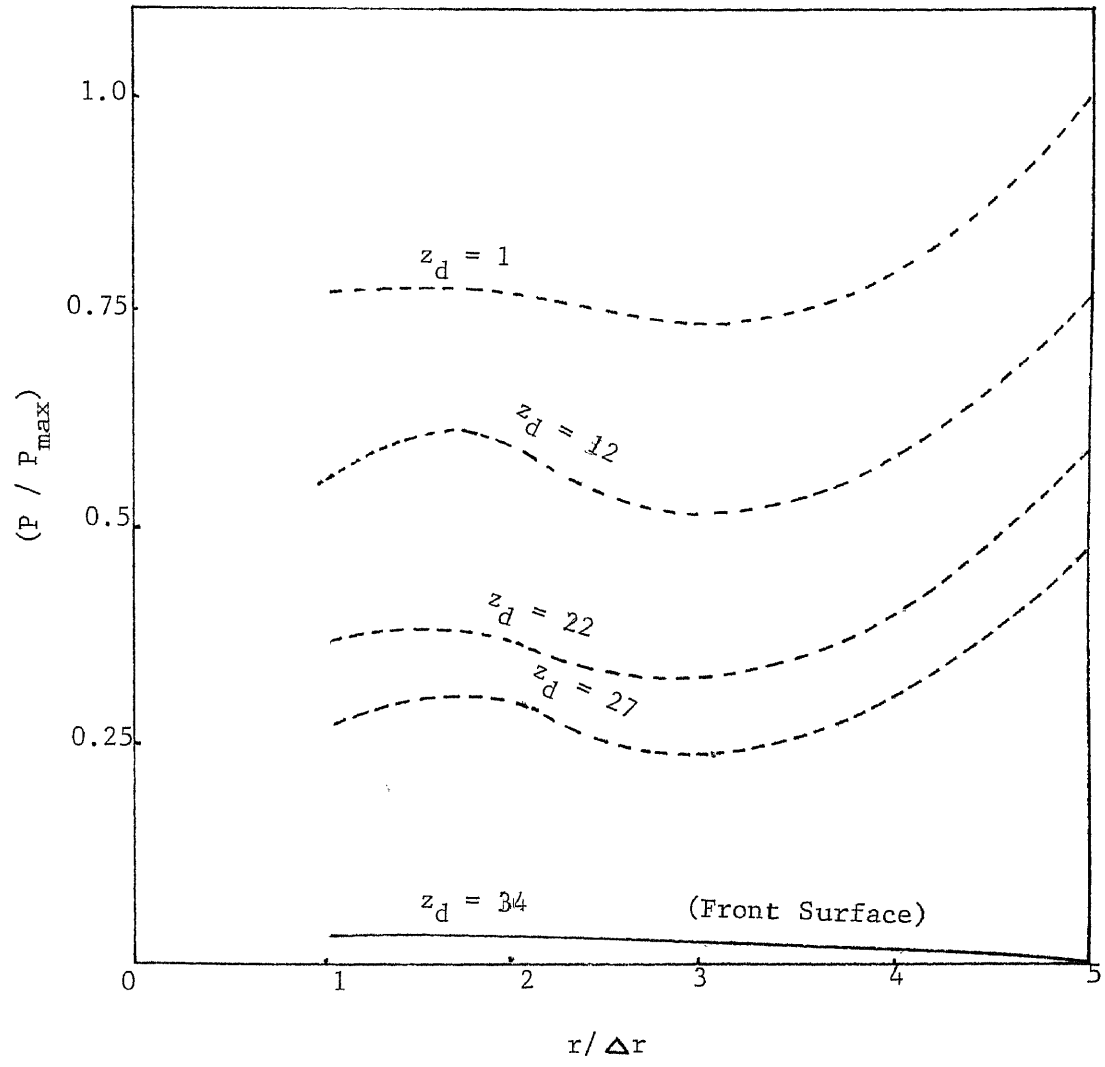


Figure 5.23

Pressure Distribution At Various Positions From The Entrance

Time = 5.909 seconds

$z_d = z/\Delta z$

$\Delta z = 0.0907$ cm

$\Delta r = 0.03175$ cm

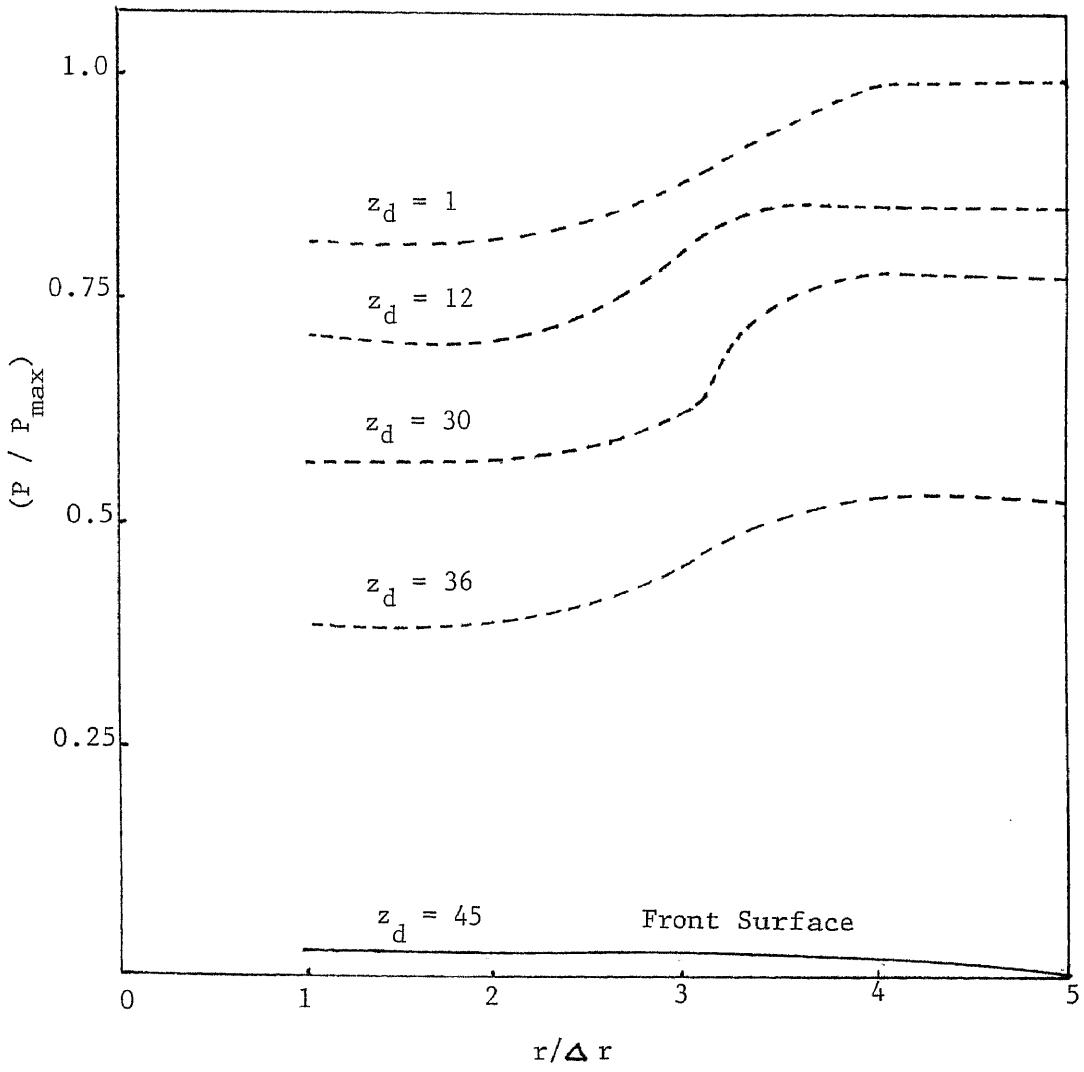


Figure 5.24

Pressure Distribution At Various Positions From The Entrance

Time = 8.39 seconds

$z_d = z/\Delta z$

$\Delta z = 0.0907$ cm

$\Delta r = 0.03175$ cm

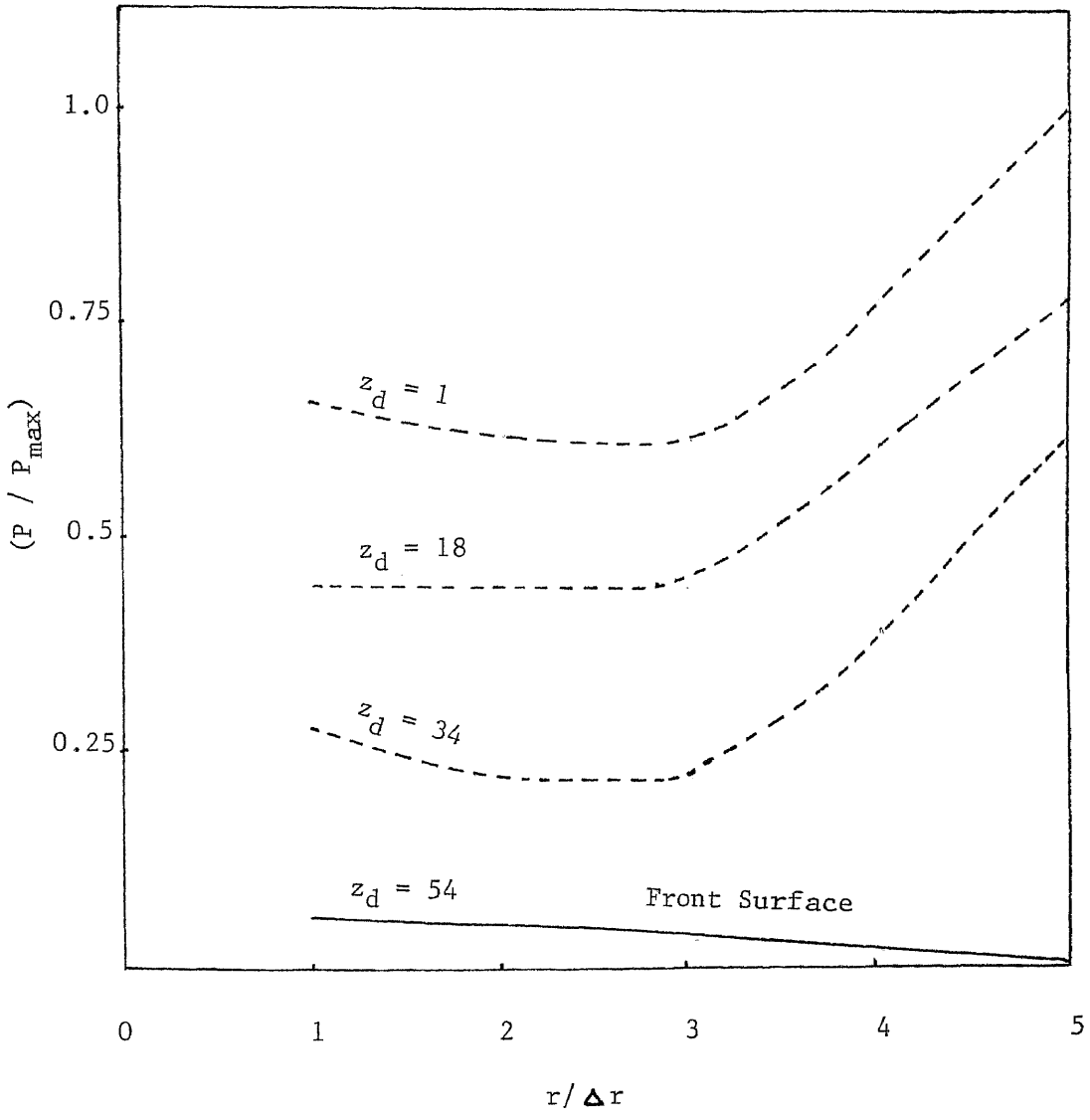


Figure 5.25

Appendix A

A Listing of The Computer Program

```
C Program For Simulation of Cavity Filling in Injection Molding
      DIMENSION F(72,7),V(72,7),UNEW(72,7),U(72,7)
     %,THETA(72,7),S(72,7),KF(72,7),VNEW(72,7)
      DIMENSION RIP(72),RRI(72),RMORP(72),R(72),RRP(72)
      DIMENSION BCLT(7),BCRT(7),ROS(7),RPORM(72)
      DIMENSION XP(8000),VISS(7)
      VI1=0
      IFULL=0
      INTEGER CYCLE
C**************************************************
C**************************************************
C**************************************************
C     Input Data
C     Flag
C**************************************************
C**************************************************
C**************************************************
      WRITE(6,3037)
 3037 FORMAT(2X,'Remark : Parabolic Front Vel',/2X,'ALP=0.9
     %',/2X,'VI1=U(I,2)   ITWO=I')
      BND=1.0
      FUL=2.0
      SUR=3.0
      EMP=4.0
C**************************************************
C**************************************************
C**************************************************
C     Geometric Parameters
C**************************************************
C**************************************************
C**************************************************
      PI=3.1416
      IBAR=70
      JBAR=5
      RDIS=(2.5)*2.54
      H=0.125*2.54
      ZDIS=H/2
      WIDTH=RDIS
      GH=0.125*2.54
      GH2=GH/2
      ZB=PI*WIDTH/8
      DR=RDIS/IBAR
      DZ=ZDIS/JBAR
      HWID=WIDTH/2
      ITT=0
      L1=0
      L2=JBAR
      L3=0
      L4=0
      L5=IBAR
```

```
      L6=IBAR
      L7=0
      H=2*DZ*JBAR
      RO=DR*IBAR
      RI=DR*L2
C*************************************************************
C*************************************************************
C*************************************************************
C     Program Control
C*************************************************************
C*************************************************************
C*************************************************************
      PC=0.0
      ALP=0.9
      EPS=0.05
      ITYPE=1
      NCLCO=1
      NSPA=20
      NXB=2
      NYB=2
      ITTC=50
      DTC=10
      VI1C=1.0
      SXE=0.1*DR
      SXE=VI1C*DZ
C*************************************************************
C*************************************************************
C*************************************************************
C     Boundary  Conditions
C*************************************************************
C*************************************************************
C*************************************************************
      BCB=1.0
      BCR=-1.0
      BCT=-1.0
      BCL=-1.0
C*************************************************************
C*************************************************************
C*************************************************************
C     Material Parameters
C*************************************************************
C*************************************************************
C*************************************************************
      GR=0
      GZ=0
      VIS=35000
      DT1=(DR**2*DZ**2)/(4*VIS*(DR**2+DZ**2))
      DT=DT1
      RHO=0.735
      CP=0.65
```

```
      RHOCP=RHO*CP
      THCON=2.3E-04
      HTC=5.0/360
      HTCA=HTC/10
      GASC=1.987E-03
      ACE=6.0
      AA=1.85E-00
      EN=1.0+(ALOG10(9000.)-ALOG10(32000.))/(ALOG10(8.)-ALOG1
     10(0.1))
      EM=20000/(1.0**(EN-1))
C**************************************************
C**************************************************
C**************************************************
C     Operating Parameters
C**************************************************
C**************************************************
C**************************************************
      TIME=0
      QFLOW=(2.21/60)*2.54*2.54*2.54
      VMAXR=((3*EN+1)/(EN+1))*QFLOW/(PI*(0.1875*2.54)**2)
      TFILL=2.5*2.5*0.125*60/2.21
      QIN=H*RO/TFILL
      UL=QIN/H
      ULI=QFLOW/(PI*GH2*H)
      ULSR=ULI
      UR=0
      TIN=190
      TW=30
      TA=30
      DPDX=-(EM/RHO)*(ULI*(1/EN+2)/((H/2)**(1/EN+1)))**EN
      WRITE(6,3035) DR,DZ,DT1,UL,H,RI,RO,QIN,TFILL,RHO,VIS,
     %CP,THCON,HTC,ACE,AA,TIN,TA,TW,EN,EM
 3035 FORMAT(2X,'DR=',F12.9,/2X,'DZ=',F12.9,/2X,'DT1=',F12.9,/2X
     %,'UL=',F12.3,/2X,'H=',F12.5,/2X,'RI=',F12.5,/2X,'RO=',F12.5
     %,/2X,'QIN=',F12.5,/2X,'TFILL=',F12.5,/2X,'RHO=',F12.5,/
     %2X,'VIS=',F12.5,/2X,'CP=',F12.5,/2X,'THCON=',F12.9,/2X
     %,'HTC=',F12.9,/2X,'ACE=',F12.5,/2X,'AA=',F12.9,/2X,
     %'TIN=',F12.5,/2X,'TA=',F12.5,/2X,'TW=',F12.5,/2X,
     %'EN=',F12.5,/2X,'EM=',F12.5)
C***Parameters Listing **************************************
      IP2=IBAR+2
      JP2=JBAR+2
      IP1=IBAR+1
      JP1=JBAR+1
      IALL=IP2*JP2
      IF(PC.EQ.0) BCL=1.0
      X=PC
      Y=DR*(1.0-PC)
      RIP(1)=PC
      RRP(1)=PC
```

```
      R(1)=RIP(1)-0.5*Y
      DO 29 I=2,IP2
      RIP(I)=X+Y
      RRP(I)=1.0/RIP(I)
      R(I)=RIP(I)-0.5*Y
      RRI(I)=1./R(I)
      Z=4.0*R(I)
      RMORP(I)=(Z-Y)/(Z+Y)
      RPORM(I)=1.0/RMORP(I)
   29 X=X+Y
      RDR=1.0/DR
      RDR2=RDR*RDR
      RDZ=1.0/DZ
      RDZ2=RDZ*RDZ
      DRODZ=DR*RDZ
      DZODR=DZ*RDR
      RDRDZ=RDR*RDZ
      W=(1.0+ALP)/(2.0*(RDR2+RDZ2))
      DTODR=DT*RDR
      DTODZ=DT*RDZ
      XR=IBAR*DR
      YT=JBAR*DZ
      XL=0
      YB=0
      CYCLE=0
C**********************************************************
C**********************************************************
C**********************************************************
C     Initial Cell Flag
C**********************************************************
C**********************************************************
C**********************************************************
      DO 37 J=2,JP1
      DO 37 I=2,IP1
      F(I,J)=EMP
      IF(I.EQ.2) F(I-1,J)=BND
      IF(J.EQ.2) F(I,J-1)=BND
      IF(I.EQ.IP1) F(I+1,J)=BND
      IF(J.EQ.JP1) F(I,J+1)=BND
   37 CONTINUE
      DO 38 J=1,JP2
      BCLT(J)=BCL
      ROS(J)=1.0
   38 BCRT(J)=BCR
      NP=0
      IST=1
      XDIS=SXE/DR
      YDIS=1.0/NYB
      YFIR=0
      NIN=NYB*(L2-L1)
```

```
      COLS=0.0
C*************************************************************
C*************************************************************
C*************************************************************
C     Inflows Velocity
C*************************************************************
C*************************************************************
C*************************************************************
      DO 110 J=1,JP2
      IF (J.GE.(L1+2).AND.J.LE.(L2+1)) GO TO 107
  106 IF (J.GE.(L3+2).AND.J.LE.(L4+1)) GO TO 109
      GO TO 110
  107 U(1,J)=((2*EN+1)/(EN+1))*ULI*(1-((J-1.5)*DZ/(H/2))**(1/E
     1N+1))
      UNEW(1,J)=U(1,J)
      BCLT(J)=-1
      GO TO 106
  109 U(IP1,J)=UR
      BCRT(J)=-1
      IF(UR.NE.0) GO TO 110
      ROS(J)=0
      BCRT(J)=1
  110 CONTINUE
      UMAX=((2*EN+1)/(EN+1))*ULI*(1-((1.5-1.5)*DZ/(H/2))**(1/E
     1N+1))
      U(1,1)=U(1,2)
      U(1,JP2)=-U(1,JP1)
      DO 1110 J=2,JP1
      JPLUS=J+1
      JMINUS=J-1
      SRA=(U(1,JPLUS)-U(1,JMINUS))/(2*DZ)
      VISS(J)=EM*((SRA)**2)**((EN-1)/2)
 1110 WRITE(6,1111) SRA,J,VISS(J)
 1111 FORMAT(2X,'SR=',E11.3,2X,'VIS(',I1,')=',E11.3)
C*************************************************************
C*************************************************************
C*************************************************************
C     Reflessin=
C*************************************************************
C*************************************************************
C*************************************************************
      ICELL=1
 1250 CONTINUE
      IF(ICELL.EQ.1) GO TO 250
      DT=DR/(DTC*ULSR)
      TIME=TIME+DT
      GO TO 1251
  250 DO 255 J=2,JP1
      DO 255 I=2,IP1
  255 KF(I,J)=0
```

```
      NPT=0
      K=1
  260 I=XP(K)+2
      J=XP(K+1)+2
      KF(I,J)=KF(I,J)+1
      K=K+2
      NPT=NPT+1
      IF(NPT.LT.NP) GO TO 260
      DO 265 J=2,JP1
      DO 265 I=2,IP1
    5 IF (F(I,J).NE.SUR) GO TO 265
      IF (KF(I,J).NE.0) GO TO 265
      F(I,J)=EMP
      IF (F(I+1,J).EQ.EMP) U(I,J)=0
      IF (F(I-1,J).EQ.EMP) U(I-1,J)=0
      IF (F(I,J+1).EQ.EMP) V(I,J)=0
      IF (F(I,J-1).EQ.EMP) V(I,J-1)=0
  265 CONTINUE
      DO 270 J=2,JP1
      DO 270 I=2,IP1
      IF(KF(I,J).EQ.0) GO TO 267
      IF (F(I,J).NE.FUL) GO TO 267
      IF(F(I+1,J).EQ.EMP.OR.F(I-1,J).EQ.EMP.OR.F(I,J+1).EQ.EM
     1P.OR.F(I,J-1).EQ.EMP) F(I,J)=SUR
      GO TO 270
  267 IF(F(I+1,J).EQ.EMP.OR.F(I-1,J).EQ.EMP.OR.F(I,J+1).EQ.
     %EMP.OR.F(I,J-1).EQ.EMP) GO TO 270
      F(I,J)=FUL
  270 CONTINUE
      ITWO=2
      IPMINU=IP1-JP2
      DO 2270  I=2,IPMINU
      IF(F(I+JP2,JP1).EQ.SUR) ITWO=2
      IF(F(I,JP1).EQ.SUR) IFSW=I
      IF (F(I,2).EQ.EMP.AND.F(I,3).EQ.SUR) F(I,2)=SUR
      IF(F(I,2).EQ.SUR.AND.F(I-1,2).EQ.SUR) F(I-1,2)=FUL
 2270 CONTINUE
      DO 2280 I=2,IP1
      J=JP1
 2281 IF(F(I,J).EQ.SUR.AND.F(I,J-1).EQ.EMP) GO TO 2282
      GO TO 2283
 2282 F(I,J-1)=SUR
      F(I-1,J-1)=FUL
 2283 IF(J.EQ.3) GO TO 2280
      J=J-1
      GO TO 2281
 2280 CONTINUE
      ASSIGN 280 TO KRET
      GO TO 650
C
```

```
C
C
C     Theta Calculation ********************************
C
C
C
  280 IF(CYCLE.NE.0) GO TO 300
  300 CONTINUE
      DO 320 J=2,JP1
      DO 320 I=3,IP1
      IF(F(I,J).NE.SUR) GO TO 315
      SNB=KF(I,J)
      N=0
      IF(F(I+1,J).EQ.EMP) N=N+1
      IF(F(I,J+1).EQ.EMP) N=N+2
      IF(F(I-1,J).EQ.EMP) N=N+4
      IF(F(I,J-1).EQ.EMP) N=N+8
      GO TO (305,310,306,308,320,309,320,312,313,320,320,314,320,
     1320,320), N
  305 I1=I-1
      GO TO 307
  308 I1=I+1
  307 FNB=KF(I1,J)
      IF(VISS(J).EQ.0) VISS(J)=VIS
      THETA(I,J)=2*(VISS(J)/RHO)*(U(I-1,J)-U(I-2,J))/DR
      GO TO 316
  310 J1=J-1
      GO TO 317
  312 J1=J+1
  317 FNB=KF(I,J1)
      IF(VISS(J).EQ.0)  VISS(J)=VIS
      THETA(I,J)=2*(VISS(J)/RHO)*(V(I-1,J)-V(I-1,J-1))/DZ
      GO TO 316
  306 I1=I-1
      J1=J-1
      GO TO 311
  309 I1=I+1
      J1=J-1
      GO TO 311
  314 I1=I+1
      J1=J+1
      GO TO 311
  313 I1=I-1
      J1=J+1
  311 FRCN=0.5
      IF(F(I1,J).EQ.SUR.OR.F(I,J1).EQ.SUR) FRCN=1.0
      FNB=FRCN*(KF(I1,J)+KF(I,J1))
      IF(VISS(J).EQ.0) VISS(J) = VIS
      THETA(I,J)=0.5*(VISS(J)/RHO)*(RDZ*(U(I1-1,J)-U(I1-1,J1)+U(I-1,J)
     %-U(I-1,J1))+RDR*(V(I-1,J)+V(I-1,J1)-V(I1-1,J)-V(I1-1,J1)))
```

```
  316 CONTINUE
C       IF(ETA.LT.0.6667) ETA=0.6667
C       IF(ETA.GT.2.0) ETA=2.0
        IF(THETA(I,J).LT.0.0) THETA(I,J)=-THETA(I,J)
  315 CONTINUE
  320 CONTINUE
        PPP=0
        J=JP1
 1322 I=IP1
 1321 IF(F(I,J).NE.SUR) GO TO 1320
        IF(THETA(I,J).EQ.0) THETA(I,J)=THETA(I,J+1)
        THETA(I,J)=THETA(I,J)+PPP
        PPP=THETA(I,J)
 1320 I=I-1
        IF(I.GT.1) GO TO 1321
        J=J-1
        IF(J.GT.1) GO TO 1322
        ASSIGN 370 TO KRET
C
C
C
C***  Just Outside And Across The Wall Tangential Velocities
C
C
C
C***  Just Outside Tangential Velocity
C
C
C
        DO 1330 I=2,IP1
        IF(F(I,2) .EQ. SUR.AND.IFSW.LT.JP2) VI1=V(I,2)
        IF(F(I,2) .EQ.SUR) VI1C=0.5*SXE*(V(I,2)/(U(1,2)-UL))/DZ
        IF(VI1C.EQ.0) VI1C=SXE/DZ
 1330 CONTINUE
  330 DO 340 J=2,JP1
        DO 340 I=ITWO,IP1
        IF(F(I,J) .NE.SUR) GO TO 340
        IF(F(I+1,J).EQ.EMP.AND.F(I+1,J+1).EQ.EMP.AND.F(I,J+1).
       %LT.EMP) V(I+1,J)=V(I,J)-DRODZ*(U(I,J+1)-U(I,J))
        IF(F(I,J+1).EQ.EMP.AND.F(I+1,J+1).EQ.EMP.AND.F(I+1,J).
       %LT.EMP) U(I,J+1)=U(I,J)-DZODR*(V(I+1,J)-V(I,J))
        IF(F(I-1,J).EQ.EMP.AND.F(I-1,J+1).EQ.EMP.AND.F(I,J+1).
       %LT.EMP) V(I-1,J)=V(I,J)+DRODZ*(U(I-1,J+1)-U(I-1,J))
        IF(F(I,J-1).EQ.EMP.AND.F(I+1,J-1).EQ.EMP.AND.F(I+1,J)
       %.LT.EMP) U(I,J-1)=U(I,J)+DZODR*(V(I+1,J-1)-V(I,J-1))
  340 CONTINUE
C
C
C
C*** Across The Wall Tangential Velocity
```

```
C
C
C
      DO 350 I=1,IBAR
      V(I,1)=0
      IF(F(I,2).EQ.SUR) V(I,1)=VI1
      V(I,JP1)=0
      IF(F(I,JP1-1).EQ.SUR) VJP1S=V(I,JP1-1)
      IF(F(I,JP1).EQ.SUR) V(I,JP1)=V(I,JP1-1)
      U(I,1)=U(I,2)
      U(I,JP2)=-U(I,JP1)
  350 CONTINUE
      DO 360 J=2,JBAR
      U(IP1,J)=0
      V(1,J)=0
      V(2,J)=V(2,J)/2.
  360 V(IP2,J)=-V(IP1,J)
      GO TO KRET ,(280,370,640,630)
  370 CONTINUE
      IF(CYCLE.EQ.0) GO TO 415
      DT=DT1
      TIME=TIME+DT
  415 CONTINUE
      CYCLE=CYCLE+1
      IF(ITWO.EQ.2) GO TO 1500
      ITWO1=ITWO-1
      DO 1501 I=2,ITWO1
      DO 1501 J=2,JP1
      UNEW(I,J)=UNEW(1,J)
 1501 VNEW(I,J)=0
 1500 CONTINUE
C
C
C
C*****Momentum Eqn ******************************
C
C
C
  500 DO 1520 J=2,JP1
      DO 1520 I=ITWO,IP1
      IF(F(I,J).GE.SUR) GO TO 1522
      VX2=U(I,J)*U(I-1,J)
      IF(I.EQ.IP1) VX2IP=0
      IF(I.NE.IP1) VX2IP=U(I+1,J)*U(I,J)
      IF(I.EQ.ITWO) VX2IM=U(I-1,J)**2
      IF(I.NE.ITWO) VX2IM=U(I-1,J)*U(I-2,J)
      VY2=V(I,J)*V(I,J-1)
      IF(J.EQ.JP1) VY2JP=0
      IF(J.NE.JP1) VY2JP=V(I,J+1)*V(I,J)
      IF(J.EQ.2) VY2JM=0
```

```
      IF(J.NE.2) VY2JM=V(I,J-1)*V(I,J-2)
      VXYPP=0.25*(U(I,J+1)+U(I,J))*(V(I+1,J)+V(I,J))
      VXYMM=0.25*(U(I-1,J)+U(I-1,J-1))*(V(I,J-1)+V(I-1,J-1))
      VXYPM=0.25*(U(I,J)+U(I,J-1))*(V(I+1,J-1)+V(I,J-1))
      VXYMP=0.25*(U(I-1,J+1)+U(I-1,J))*(V(I,J)+V(I-1,J))
      IF(J.EQ.2) VXYMM=0
      IF(J.EQ.2) VXYPM=0
      IF(J.EQ.JP1) VXYPP=0
      IF(J.EQ.JP1) VXYMP=0
      IF(I.EQ.ITWO) VXYMP=0
      IF(I.EQ.ITWO) VXYMM=0
      QIJ=(R(I+1)*VX2IP+R(I-1)*VX2IM-2*R(I)*VX2)/(R(I)*DR**2)
     %+(VY2JP+VY2JM-2*VY2)/DZ**2+2*(RIP(I)*VXYPP+RIP(I-1)
     %*VXYMM-RIP(I)*VXYPM-RIP(I-1)*VXYMP)/(R(I)*DR*DZ)
      DIJ=(RIP(I)*U(I,J)-RIP(I-1)*U(I-1,J))/(R(I)*DR)
     %+(V(I,J)-V(I,J-1))/DZ
      DVXX1=(U(I,J)-U(I-1,J))/DR
      IF(I.FQ.ITWO) DVXX2=0
      IF(I.NE.ITWO) DVXX2=(U(I-1,J)-U(I-2,J))/DR
      IF(J.EQ.JP1) DVXX3=0
      IF(J.NE.JP1) DVXX3=((U(I,J+1)+U(I,J))-(U(I-1,J+1)+U(I-1
     %,J)))/(2*DR)
      DVXX4=((U(I,J)+U(I,J-1))-(U(I-1,J)+U(I-1,J-1)))/(2*DR)
      DVXY1=(U(I,J+1)-U(I,J-1))/(DZ*2)
      DVXY2=(U(I-1,J+1)-U(I-1,J-1))/(DZ*2)
      DVXY3=((U(I,J+1)+U(I-1,J+1))-(U(I,J)+U(I-1,J)))/(2*DZ)
      IF(J.EQ.JP1) DVXY3=0.5*(U(I,J)+U(I-1,J))/(DZ/2)
      DVXY4=((U(I,J)+U(I-1,J))-(U(I,J-1)+U(I-1,J-1)))/(2*DZ)
      DVYX1=((V(I+1,J)+V(I+1,J-1))-(V(I,J)+V(I,J-1)))/(2*DR)
      IF(I.EQ.ITWO) DVYX2=0
      IF(I.NE.ITWO) DVYX2=((V(I,J)+V(I,J-1))-(V(I-1,J)+V(I-1,
     %J-1)))/(2*DR)
      DVYX3=(V(I,J)-V(I-1,J))/DR
      IF(I.EQ.ITWO) DVYX3=V(I,J)/(DR/2)
      DVYX4=(V(I,J-1)-V(I-1,J-1))/DR
      IF(I.EQ.ITWO) DVYX4=V(I,J-1)/(DR/2)
      DVYY1=((V(I+1,J)+V(I,J))-(V(I+1,J-1)+V(I,J-1)))/(2*DZ)
      IF(I.EQ.ITWO) DVYY2=0
      IF(I.NE.ITWO) DVYY2=((V(I,J)+V(I-1,J))-(V(I,J-1)+V(I-1,
     %J-1)))/(2*DZ)
      IF(J.EQ.JP1) DVYY3=(V(I,J)-V(I,J-1))/DZ
      IF(J.NE.JP1) DVYY3=(V(I,J+1)-V(I,J-1))/(2*DZ)
      IF(J.EQ.2) DVYY4=0
      IF(J.NE.2) DVYY4=(V(I,J)-V(I,J-2))/(2*DZ)
      RV1=2*(U(I,J)/RIP(I))**2
      IF(I.NE.2) RV2=2*(U(I-1,J)/RIP(I-1))**2
      IF(I.EQ.2) RV2=2*(U(I-1,J)/R(I))**2
      RV3=2*((U(I,J)+U(I,J+1)+U(I-1,J)+U(I-1,J+1))/(4*R(I)))**2
      RV4=2*((U(I,J)+U(I,J-1)+U(I-1,J)+U(I-1,J-1))/(4*R(I)))**2
      VIS1=EM*(2*DVXX1**2+(DVXY1+DVYX1)**2+RV1+2*DVYY1**2)**((EN-
```

```
     %1)/2)
      VIS2=EM*(2*DVXX2**2+(DVXY2+DVYX2)**2+RV2+2*DVYY2**2)**((EN
     %-1)/2)
      VIS3=EM*(2*DVXX3**2+(DVXY3+DVYX3)**2+RV3+2*DVYY3**2)**((EN-
     %1)/2)
      VIS4=EM*(2*DVXX4**2+(DVXY4+DVYX4)**2+RV4+2*DVYY4**2)
     %**((EN-1)/2)
      IF(F(I+1,J).EQ.SUR) VISS(J)=(VIS1+VIS2+VIS3+VIS4
     %)/4
      DVIX1=(VIS1-VIS2)/DR
      DVIX2=(VIS1-VIS2)/DR
      DVIX3=(VIS1-VIS3)/(DR/2)
      DVIX4=(VIS1-VIS4)/(DR/2)
      DVIY1=(VIS3-VIS1)/(DZ/2)
      DVIY2=(VIS3-VIS2)/(DZ/2)
      DVIY3=(VIS3-VIS4)/DZ
      DVIY4=(VIS3-VIS4)/DZ
      IF(I.NE.IP1) D2VXX1=(U(I+1,J)+U(I-1,J)-2*U(I,J))/DR**2
      IF(I.EQ.IP1) D2VXX1=(U(I,J)+U(I-2,J)-2*U(I-1,J))/DR**2
      IF(I.EQ.ITWO) D2VXX2=0
      IF(I.NE.ITWO) D2VXX2=(U(I,J)+U(I-2,J)-2*U(I-1,J))/DR**2
      D2VXY1=(U(I,J+1)+U(I,J-1)-2*U(I,J))/DZ**2
      D2VXY2=(U(I-1,J+1)+U(I-1,J-1)-2*U(I-1,J))/DZ**2
      D2VYX3=(V(I+1,J)+V(I-1,J)-2*V(I,J))/DR**2
      D2VYX4=(V(I+1,J-1)+V(I-1,J-1)-2*V(I,J-1))/DR**2
      IF(J.EQ.JP1) D2VYY3=(V(I,J)+V(I,J-2)-2*V(I,J-1))/DZ**2
      IF(J.NE.JP1) D2VYY3=(V(I,J+1)+V(I,J-1)-2*V(I,J))/DZ**2
      IF(J.EQ.2) D2VYY4=(V(I,J+1)+V(I,J-1)-2*V(I,J))/DZ**2
      IF(J.NE.2) D2VYY4=(V(I,J)+V(I,J-2)-2*V(I,J-1))/DZ**2
      RB1=(U(I+1,J)-U(I-1,J))/(2*DR*RIP(I))-U(I,J)/RIP(I)**2
      IF(I.NE.2) RB2=(U(I,J)-U(I-2,J))/(2*DR*RIP(I-1))-U(I-1,J)/
     %RIP(I-1)**2
      IF(I.EQ.2) RB2=(U(I,J)-U(I-1,J))/(DR*R(I))-U(I-1,J)/
     %R(I)**2
      RB3=(V(I,J)-V(I-1,J))/(R(I)*DR)
      RB4=(V(I,J-1)-V(I-1,J-1))/(R(I)*DR)
      B1=(VIS1*(D2VXX1+RB1+D2VXY1)+2*DVXX1*DVIX1+(DVXY1+DVYX1)*D
     %VIY1)/RHO
      B2=(VIS2*(D2VXX2+RB2+D2VXY2)+2*DVXX2*DVIX2+(DVXY2+DVYX2)*DV
     %IY2)/RHO
      B3=(VIS3*(D2VYX3+RB3+D2VYY3)+2*DVYY3*DVIY3+(DVXY3+DVYX3)*DV
     %IX3)/RHO
      B4=(VIS4*(D2VYX4+RB4+D2VYY4)+2*DVYY4*DVIY4+(DVXY4+DVYX4)*D
     %VIX4)/RHO
      S(I,J)=OIJ-DIJ/DT-((RIP(I)*B1-RIP(I-1)*B2)/(R(I)*DR)
     %+(B3-B4)/DZ)
1522  CONTINUE
1520  CONTINUE
      ITER=0
      IND=1
```

```
C
C
C
C*** Pressure Field Calculation **********************
C
C
C
      DO 1550 J=2,JP1
      I=IP1
 1549 IF(THETA(I,J).LT.THETA(I+1,J)) THETA(I,J)=THETA(I+1,J)
     %-DPDX*DR
      IF(I.EQ.2) GO TO 1550
      I=I-1
      GO TO 1549
 1550 CONTINUE
  550 DO 555 I=2,IP1
      THETA(I,1)=THETA(I,2)
  555 THETA(I,JP2)=THETA(I,JP1)
      DO 560 J=2,JP1
      VISE=EM*(((U(ITWO-1,J+1)-U(ITWO-1,J-1))/(2*DZ))**2
     %)**((EN-1)/2)
      THETA(ITWO-1,J)=THETA(ITWO,J)-DPDX*DR
  560 THETA(IP2,J)=THETA(IP1,J)
      IF(IND.EQ.0) GO TO 600
      IND=0
      ITER=ITER+1
      DO 570 J=2,JP1
      DO 570 I=ITWO,IP1
  571 IF(F(I,J).NE.FUL) GO TO 570
      PSIT=THETA(I,J+1)
      PSIR=THETA(I+1,J)
      PSIB=THETA(I,J-1)
      PSIL=THETA(I-1,J)
      X=W*((RIP(I)*PSIR+RIP(I-1)*PSIL)/(R(I)*DR**2)+(PSIT+PSIB)/DZ**2+
     %S(I,J))-ALP*THETA(I,J)
      Y=ABS(X)-ABS(THETA(I,J))
      Z=ABS(X)+ABS(THETA(I,J))
      THETA(I,J)=X
      IF(Z.EQ.0) GO TO 570
      IF(ABS(Y/Z).GT.EPS) IND=1
  570 CONTINUE
      GO TO 550
  600 DO 1551 J=2,JP1
      I=IP1
 1552 IF(THETA(I,J).LT.THETA(I+1,J)) THETA(I,J)=THETA(I+1,J)-
     %DPDX*DR
      IF(I.EQ.2) GO TO 1551
      I=I-1
      GO TO 1552
 1551 CONTINUE
```

```
C
C
C
C**** Final Velocity ***************************************
C
C
C
      DO 620 J=2,JP1
      DO 620 I=ITWO,IP1
      IF(I.EQ.IP1.OR.F(I,J).GE.EMP) GO TO 620
      DVXX1=(U(I,J)-U(I-1,J))/DR
      IF(I.EQ.ITWO) DVXX2=0
      IF(I.NE.ITWO) DVXX2=(U(I-1,J)-U(I-2,J))/DR
      IF(J.EQ.JP1) DVXX3=0
      IF(J.NE.JP1) DVXX3=((U(I,J+1)+U(I,J))-(U(I-1,J+1)+U(I-1,J)))/
     %(2*DR)
      DVXX4=((U(I,J)+U(I,J-1))-(U(I-1,J)+U(I-1,J-1)))/(2*DR)
      DVXY1=(U(I,J+1)-U(I,J-1))/(DZ*2)
      DVXY2=(U(I-1,J+1)-U(I-1,J-1))/(DZ*2)
      DVXY3=((U(I,J+1)+U(I-1,J+1))-(U(I,J)+U(I-1,J)))/(2*DZ)
      IF(J.EQ.JP1) DVXY3=0.5*(U(I,J)+U(I-1,J))/(DZ/2)
      DVXY4=((U(I,J)+U(I-1,J))-(U(I,J-1)+U(I-1,J-1)))/(2*DZ)
      DVYX1=((V(I+1,J)+V(I+1,J-1))-(V(I,J)+V(I,J-1)))/(2*DR)
      IF(I.EQ.ITWO) DVYX2=0
      IF(I.NE.ITWO) DVYX2=((V(I,J)+V(I,J-1))-(V(I-1,J)+V(I-1,J-1)
     %))/(2*DR)
      DVYX3=(V(I,J)-V(I-1,J))/DR
      IF(I.EQ.ITWO) DVYX3=V(I,J)/(DR/2)
      DVYX4=(V(I,J-1)-V(I-1,J-1))/DR
      IF(I.EQ.ITWO) DVYX4=V(I,J-1)/(DR/2)
      DVYY1=((V(I+1,J)+V(I,J))-(V(I+1,J-1)+V(I,J-1)))/(2*DZ)
      IF(I.EQ.ITWO) DVYY2=2
      IF(I.NE.ITWO) DVYY2=((V(I,J)+V(I-1,J))-(V(I,J-1)+V(I-1,J-1)
     %))/(2*DZ)
      IF(J.EQ.JP1) DVYY3=(V(I,J)-V(I,J-1))/DZ
      IF(J.NE.JP1) DVYY3=(V(I,J+1)-V(I,J-1))/(2*DZ)
      IF(J.EQ.2) DVYY4=0
      IF(J.NE.2) DVYY4=(V(I,J)-V(I,J-2))/(2*DZ)
      RV1=2*(U(I,J)/RIP(I))**2
      IF(I.NE.2) RV2=2*(U(I-1,J)/RIP(I-1))**2
      IF(I.EQ.2) RV2=2*(U(I-1,J)/R(I))**2
      RV3=2*((U(I,J)+U(I,J+1)+U(I-1,J)+U(I-1,J+1))/(4*R(I)))**2
      RV4=2*((U(I,J)+U(I,J-1)+U(I-1,J)+U(I-1,J-1))/(4*R(I)))**2
      VIS1=EM*(2*DVXX1**2+(DVXY1+DVYX1)**2+RV1+2*DVYY1**2)**((EN-1)/2)
      VIS2=EM*(2*DVXX2**2+(DVXY2+DVYX2)**2+RV2+2*DVYY2**2)**((EN-1)/2)
      VIS3=EM*(2*DVXX3**2+(DVXY3+DVYX3)**2+RV3+2*DVYY3**2)**((EN-1)/2)
      VIS4=EM*(2*DVXX4**2+(DVXY4+DVYX4)**2+RV4+2*DVYY4**2)**((EN-1)/2)
      DVIX1=(VIS1-VIS2)/DR
      DVIY1=(VIS3-VIS1)/(DZ/2)
      DVIY3=(VIS3-VIS4)/DZ
```

```
      DVIX3=(VIS3-VIS2)/(DR/2)
      IF(F(I+1,J).GE.EMP) GO TO 610
      RB1=(U(I+1,J)-U(I-1,J))/(2*DR*RIP(I))-U(I,J)/RIP(I)**2
      RB3=(V(I,J)-V(I-1,J))/(R(I)*DR)
      UNEW(I,J)=U(I,J)+DT*(-(R(I+1)*U(I+1,J)*U(I,J)-R(I)*U(I,J)*U(I-1,J)
     %)/(DR*RIP(I))
     %-0.25*((U(I,J+1)+U(I,J))*(V(I+1,J)+V(I,J))-(U(I,J)+U(I,J-1))*
     %(V(I+1,J-1)+V(I,J-1)))/DZ-(THETA(I+1,J)-THETA(I,J))/DR
     %+(VIS1/RHO)*((U(I+1,J)+U(I-1,J)-2*U(I,J))/DR**2+RB1
     %+(U(I,J+1)+U(I,J-1)-2*U(I,J))/DZ**2)+2*DVXX1*DVIX1+(DVXY1+DVYX1)
     %*DVIY1)
      IF(I.EQ.ITWO) UNEW(I,J)=UNEW(1,J)
  610 IF(J.EQ.JP1 .OR.F(I,J+1).EQ.EMP) GO TO 620
      VNEW(I,J)=V(I,J)+DT*(-(V(I,J+1)*V(I,J)-V(I,J)*V(I,J-1))/DZ
     %-0.25*(RIP(I)*(U(I,J+1)+U(I,J))*(V(I+1,J)+V(I,J))-RIP(I-1)*(
     %U(I-1,J+1)+U(I-1,J))*(V(I,J)+V(I-1,J)))/(R(I)*DR)-(THETA(I,J+1)-
     %THETA(I,J))/DZ+(VIS3/RHO)*((V(I,J+1)+V(I,J-1)-2*V(I,J))/DZ**2
     %+RB3+(V(I+1,J)+V(I-1,J)-2*V(I,J))/DR**2)+2*DVYY3*DVIY3+(DVXY3+
     %DVYX3)*DVIX3)
      IF(VNEW(I,J).LT.0) VNEW(I,J)=VNEW(I-1,J)
  620 CONTINUE
      IF(ITWO.EQ.2) GO TO 1600
      DO 1620 I=2,ITWO1
      DO 1620 J=2,JP1
      UNEW(I,J)=UNEW(1,J)
 1620 VNEW(I,J)=0
 1600 CONTINUE
      DO 1521 J=2,JP1
      DO 1521 I=2,IP1
      U(I,J)=UNEW(I,J)
 1521 V(I,J)=VNEW(I,J)
      DO 1331   I=2,IP1
      IF(F(I,2) .EQ.SUR.AND.IFSW.LT.JP2) VI1=V(I,2)
 1331 CONTINUE
      I=ITWO-1
 1297 IF(I.EQ.1)  GO TO 1298
      DO 1296 J=2,JP1
 1296 THETA (I,J)=THETA(I+1,J)-DPDX*DR
      I=I-1
      GO TO 1297
 1298 CONTINUE
      DO 1621 I=2,IP1
      JP1M=JP1-1
      DO 1621 J=2,JP1M
 1621 IF(U(I,J).LT.U(I,J+1)) U(I,J)=U(I-1,J)
      ASSIGN 630 TO KRET
      GO TO 650
  630 ASSIGN 640 TO KRET
      GO TO 330
 640  GO TO 1251
```

```
C
C
C
C*** Calculate SUR Cell Velocity on EMP Cell Faces!*********
C
C
C
 650  DO 670 J=2,JP1
      DO 670 I=ITWO,IP1
      IF(F(I,J).NE.SUR) GO TO 670
      N=0
      IF(F(I+1,J) .EQ.EMP) N=N+1
      IF(F(I,J+1) .EQ.EMP) N=N+2
      IF(F(I-1,J) .EQ.EMP ) N=N+4
      IF(F(I,J-1).EQ.EMP) N=N+8
      GO TO (651,652,653,654,651,655,652,657,658,652,651,659,651,6
     %52,651),N
  651 IF(R(I).GT.GH2) ULS=OFLOW/(PI*R(I)*H)
      IF(R(I).LE.GH2) ULS=OFLOW/(PI*GH2*H)
      IF(J.EQ.2) ULSP=ULS
      UMAXS=((2*EN+1)/(EN+1))*ULS
      ROP=(RIP(1)-DR)/RIP(I)
      IF(I.EQ.2) ROP=RMORP(I)
      U(I,J)=ROP*U(I-1,J)*(UMAX/(UMAX+VI1))+ULS*(VI1/(UMAX+VI1))
      GO TO 670
  652 V(I,J)=V(I,J-1)-DZODR*RRI(I)*(U(I,J)*RIP(I)-U(I-1,J)*RIP(I-1))
      GO TO 670
  653 IF(R(I).GT.GH2) ULS=OFLOW/(PI*R(I)*H)
      IF(R(I).LE.GH2) ULS=OFLOW/(PI*GH2*H)
      ROP=(RIP(I)-DR)/RIP(I)
      IF(I.EQ.2) ROP=RMORP(I)
      U(I,J)=ROP*U(I-1,J)*(UMAX/(UMAX+VI1))+ULS*(VI1/(UMAX+VI1))
      GO TO 656
  654 U(I-1,J)=(U(I,J)*RIP(I)+R(I)*DRODZ*(V(I,J)-V(I,J-1)))*RRP(I-1)
      GO TO 670
  655 U(I-1,J)=U(I,J)*RPORM(I)
  656 V(I,J)=V(I,J-1)-0.25*DZ*(U(I,J)+U(I-1,J))*RRI(I)*(1.0-PC)
      GO TO 670
  657 V(I,J-1)=V(I,J)+DZODR*RRI(I)*(U(I,J)*RIP(I)-U(I-1,J)*RIP(I-1))
      GO TO 670
  658 U(I,J)=U(I-1,J)*RMORP(I)
      GO TO 660
  659 U(I-1,J)=U(I,J)*RPORM(I)
  660 V(I,J-1)=V(I,J)+0.25*DZ*RRI(I)*(U(I,J)+U(I-1,J))*(1.0-PC)
  670 CONTINUE
      GO TO KRET,(280,370,640,630)
 1251 CONTINUE
C
C
C
```

```
C**** Energy Equation **********************************************
C
C
C
      IF(ITT.EQ.ITTC) GO TO 1270
      GO TO 1263
 1270 ITT=0
      WRITE(6,1269) ITWO
1269  FORMAT(/,2X,'ITWO=',I5)
      WRITE(6,1254) TIME
 1254 FORMAT(5X,'TIME=',F9.5,//,2X,'FLAG')
      DO 1252 J=2,JP1
 1252 WRITE(6,1253) (F(I,J),I=2,36)
      DO 1152 J=2,JP1
1152  WRITE(6,1253) (F(I,J) ,I=37,IP1)
 1253 FORMAT(2X,35F2.0)
      WRITE(6,1285)
1285  FORMAT(2X,'SRES= (CM)',2X,'NPC=   ',2X,'XS=  ',4X
     %,'X=   ',5X,'Y=')
      NN=1
      MCONT=0
      NPC=NN
      SRES=0.0
 1401 NPCKN=2*NPC-1
      NPCKN1=NPCKN+1
      IF(XP(NPCKN1).EQ.0) GO TO 1405
      RXS=XP(NPCKN)*DR
      XPX=RXS
      WRITE(6,1280) SRES,NPC,XPX,XP(NPCKN),XP(NPCKN1)
 1405 CONTINUE
      IF(MCONT.EQ.0) NPC=NPC+NIN
      IF(MCONT.NE.0) NPC=NPC+1
      IF(NPC.GT.NPN) GO TO 1402
      SRES=SRES+SXE*VMAXR/UMAX
      MCONT=MCONT+1
      IF(MCONT.EQ.NSPA) MCONT=0
      GO TO 1401
 1402 CONTINUE
      DO 1282 NN=1,NYR
      IF(NN.EQ.1) GO TO 1282
      NCC=0
      NPC=NN
      SRES=0
 1281 NPCKN=2*NCC*(NIN+NSPA-1)+2*NN-1
      NPCKN1=NPCKN+1
      RXS=XP(NPCKN)*DR
      XPX=RXS
      WRITE(6,1280) SRES,NPC,XPX,XP(NPCKN),XP(NPCKN1)
 1280 FORMAT(2X,F8.4,2X,I5,2X,F8.2,2X,F8.2,2X,F8.2)
      NPC=NPC+NIN+NSPA-1
```

```
      NCC=NCC+1
      SRES=SRES+SXE*(UMAXR/UMAX)*NSPA
      IF(NPC.LT.NPN) GO TO 1281
 1282 CONTINUE
      WRITE(6,1255)
1255  FORMAT(//,2X,'UXR')
      DO 1256 I=1,72
1256  WRITE(6,1257) (U(I,J),J=1,7)
1257  FORMAT(2X,10E10.2)
      WRITE(6,1258)
1258  FORMAT(//,2X,'VYZ')
      DO 1259 I=1,72
1259  WRITE(6,1257) (V(I,J),J=1,7)
      WRITE(6,1260)
1260  FORMAT(//,2X,'PRESSURE PSI')
      DO 2260 I=2,IP1
      DO 2260 J=2,JP1
2260  THETA(I,J)=THETA(I,J)/68947
      DO 1261 I=1,72
1261  WRITE(6,1257) (THETA(I,J),J=1,7)
      DO 2261 I=2,IP1
      DO 2261 J=2,JP1
2261  THETA(I,J)=THETA(I,J)*68947
      GO TO 1263
 1263 CONTINUE
      IF(IFULL.EQ.1) GO TO 9999
      ITT=ITT+1
      IFULL=1
      DO 1274 J=2,JP1
      DO 1274 I=2,IP1
      IF(F(I,J).EQ.EMP) IFULL=0
 1274 CONTINUE
      IF(IFULL.EQ.0) GO TO 1271
      WRITE(6,1272)
 1272 FORMAT(//,2X,'FULL')
      GO TO 1270
 1271 CONTINUE
      IST=IST+1
C
C
C
C**** Move, Pack, In-ut Particles **************************
C
C
C
  700 NPT=0
      NPN=0
      K=1
      KN=1
      ICELL=0
```

```
  710 IF(NPT.GE.NP) GO TO 735
      ID=XP(K)+2
      HPX=ID-1.-XP(K)
      HMX=1.-HPX
      JR=XP(K+1)+1.5
      HPY=JR-0.5-XP(K+1)
      HMY=1.-HPY
      UVX=UMAX+VI1
      UDR1=U(ID,JR+1)
      UDR=U(ID,JR)
      UK=HPX*HMY*U(ID-1,JR+1)+HMX*HMY*UDR1+HPX*HPY*U(ID-1,JR)+HMX
     %*HPY*UDR
      IR=XP(K)+1.5
      HPX=IR-0.5-XP(K)
      HMX=1.0-HPX
      JD=XP(K+1)+2
      HPY=JD-1.0-XP(K+1)
      HMY=1.0-HPY
      VK=HPX*HMY*V(IR,JD)+HMX*HMY*V(IR+1,JD)+HPX*HPY*V(IR,JD-1)+HM
     %X*HPY*V(IR+1,JD-1)
      IF(XP(K+1).NE.0) GO TO 1710
      IF(K.EQ.1) GO TO 1710
      NIN2=NIN*2
      SJBAR=JBAR-1
      IF(XP(K-1).GE.SJBAR) KNIN=K-NIN2
      IF(XP(K-1).LT.SJBAR) KNIN=K-2
      IXPK=XP(K)+2
      IF(ITWO.GT.2) GO TO 1721
      DD=(UMAX**2+VI1**2)**0.5
      RRR=XP(K)*DR
      UMAXS=((2*EN+1)/(EN+1))*ULSR
      IF(UMAXS.GE.UMAX) UMAXS=UMAX
      DD=0.5*(UMAX+UMAXS)+VI1
      DD=(UMAXS**2+VI1**2)**0.5
 1721 UK=DD
      VK=0
      STEST=XP(KN)+(DT/DR)*UK
      IF(STEST.LT.XP(KNIN)) GO TO 1711
      XP(KN+1)=0.5*XP(KNIN+1)
      XP(KN)=XP(KNIN)
      GO TO 1713
 1710 CONTINUE
      IF(F(ID,JD).EQ.SUR) UK=U(ID,JD)
 1711 CONTINUE
      DPAR=NIN/(NIN+NSPA-1)
      RNNN=K/(2*(NIN+NSPA-1))
      NNN=K/(2*(NIN+NSPA-1))
      DRNNN=RNNN-NNN
      IF(DRNNN.LT.DPAR) GO TO 1714
      JW=XP(K+1)+2
```

```
      IW=XP(K)+2
      IF(JW.EQ.JP1) VK=VJP1S
 1714 CONTINUE
      XP(KN)=XP(K)+UK*DT/DP
      XP(KN+1)=XP(K+1)+VK*DT/DZ
 1713 CONTINUE
      IF(XP(KN+1).GT.JBAR) XP(KN+1)=JBAR-0.00001
      IF(XP(KN).GT.IBAR) XP(KN)=IBAR-0.00001
      I=XP(KN)+2
      J=XP(KN+1)+2
      IF(I.LT.2.OR.I.GT.IP1) GO TO 730
      IF(J.LT.2. OR.J.GT.JP1) GO TO 730
      IF(ITYPE.EQ.0) GO TO 715
      IF(XP(KN) .GE.IBAR) GO TO 720
  715 KN=KN+2
      NPN=NPN+1
  720 IF(F(I,J).EQ.EMP) GO TO 740
  730 K=K+2
      NPT=NPT+1
      GO TO 710
C
C
C
C***  Input Particles ***********************************
C
C
C
  735 NP=NPN
      IF(ITYPE.EQ.0) GO TO 1250
  740 X=UMAX*RDR*(TIME+DT)-XDIS*COLS
      IF(X.LT.0) GO TO 1250
      COLS=COLS+1
      Y=YFIR
      IF(NPN.GE.NIN) GO TO 1750
      NPN=NP+NIN
      GO TO 750
 1750 IF(NCLCO.NE.NSPA) NPN=NP+1
      IF(NCLCO.EQ.NSPA) NPN = NP+NIN
      IF(NCLCO.EQ.NSPA) NCLCO=0
      NCLCO=NCLCO+1
  750 XP(KN)=X
      XP(KN+1)=Y
      KN=KN+2
      I=X+2
      J=Y+2
      NP=NP+1
      Y=Y+YDIS
      IF(F(I,J).NE.EMP)  GO TO 755
      F(I,J)=SUR
      ICELL=1
```

```
      U(I,J)=U(I,J)
  755 IF(NP.LT.NPN) GO TO 750
      GO TO 740
  760 F(I,J)=SUR
      ICELL=1
      U(I,J)=U(I-1,J)
      V(I,J)=V(I-1,J)
      V(I,J-1)=V(I-1,J-1)
      IF(I.GT.3) V(I-1,J)=V(I-2,J)
      THETA(I,J)=THETA(I-1,J)
      IF(I.GE.3) THETA(I-1,J)=THETA(I-2,J)
      GO TO 730
 9999 STOP
      END
```

BIBLIOGRAPHY

1. Kamal,M. R., and Kenig,S., Polym.Eng.Sci.,12,294(1972)
2. Manzione, Louis T., Polym.Eng.Sci.,18,1234(1981)
3. Welch, J. E.,Harlow, F.H., Shannon, J.P., and Daly,B.J
   " THE MAC METHOD : A Computing Technique for Solving Viscous, Incompressible, Transient Fluid-Flow Problems Involving Free Surfaces " Report LA-3425,Los Alamos Scientific Laboratory of University of California, 1966
4. Spencer,R.S., and Gilmore,G.D., Mod. Plastics,28,97(1950)
5. Kamal, M.R., and Lafleur,P.G.,Polym. Eng. Sci.,17,1066(1982)
6. Huang, C.F., Ph.D.Thesis,Stevens Institute of Technology, Dept. of Chemical Engineering (1978)
7. Amsden, A.A.and Harlow,F.H., U.S.Govt.Report(NTISLA-4370)(1970)
8. Schmidt,L.R., Polym. Eng. Sci., 14, 797(1974)
9. Denson,C.D., Trans.Soc. Rheol.16 : 2, 377(1972)
10. Hirt,C.W., : Heuristic Stability Theory for Finite-Difference Equations, J. Computa. Phys., 2, 339(1968)
11. Carnahan, B., Luther, H.A. and Wilkes,J.O., Applied Numerical Methods, New York, Wiley & Sons Inc.