SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

Technical Report 76-C1

THE INTERACTION BETWEEN A SOLID BODY AND VISCOUS FLUID BY MARKER-AND-CELL METHOD

(NASA-CR-146586) THE INTERACTION BETWEEN A SOLID BODY AND VISCOUS FLUID BY MARKER-AND-CELL METHOD (Old Dominion Univ. Research Foundation) 78 p HC $5.00 CSCL 20D
N76-20413
Unclas
G3/34 21421

By

Robert Y.K. Cheng

A Technical Report

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia

Under
Grant NGL 47-003-012

February 1976

SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

Technical Report 76-C1

THE INTERACTION BETWEEN A SOLID BODY AND VISCOUS FLUID BY MARKER-AND-CELL METHOD

By

Robert Y.K. Cheng

A Technical Report

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Grant NGL 47-003-012
Trafford J.W. Leland, Tehnical Monitor
Structures and Dynamics Division

Submitted by the
Old Dominion University Research Foundation
Norfolk, Virginia 23508

February 1976

TABLE OF CONTENTS

THE INTERACTION BETWEEN A SOLID BODY AND VISCOUS FLUID BY MARKER-AND-CELL METHOD

By

Robert Y.K. Cheng[1]

SUMMARY

A computational method for solving nonlinear problems relating to impact and penetration of a rigid body into a fluid-type medium is presented. The numerical technique, based on the Marker-and-Cell method, gives the pressure and velocity of the flow field. An important feature in this method is that the force and displacement of the rigid body interacting with the fluid during the impact and sinking phases are evaluated from the boundary stresses imposed by the fluid on the rigid body.

A sample problem of low velocity penetration of a rigid block into still water is solved by this method. The computed time histories of the acceleration, pressure, and displacement of the block show good agreement with experimental measurements. A sample problem of high velocity impact of a rigid block into soft clay is also presented. The constitutive relationships of the clay is represented as a very viscous non-Newtonian fluid.

---

1 Professor of Civil Engineering, School of Engineering, Old Dominion University, Norfolk, Virginia 23508.

INTRODUCTION

There is a need to develop an analytical method of studying the problem of aircraft landing impact on soil runways. Since deep ruts are created by the tires on unpaved runways, the drag forces on the landing gear can be large enough to endanger the safe operation of the aircraft.

Many studies have been performed (refs. 1, 2, 3, and 4) to study the soil-tire interactions at ground speeds associated with aircraft operations. The methods used in those studies are based on either empirical modeling techniques or on soil strengths obtained by static tests. Although the rut depth and drag forces will depend on many factors other than the soil strength, a rational method for predicting the landing-impact problem must take into account the dynamic soil properties, the interface-stress distribution between the tire and soil, and the large movement of soil masses.

As a first step in a program to provide an analytic method as an engineering tool for studying the landing-impact problem, a computational method is presented for solving the problem of large displacements of fluid-acting media when interacting with a solid body. An important feature in this method is that the force and displacement of the rigid body during the impact and sinking phases are derived from the fluid field.

LIST OF SYMBOLS

C	wave speed
D	divergence of fluid medium
E	energy
e	strain-rate
$\overline{\overline{e}}$	strain-rate tensor
f	force component contributed by a single cell

F	force component acting on rigid body
$\overline{F}$	force vector
$\overline{g}$	gravitational acceleration vector
g	gravitational acceleration
h	depth of fluid field measured from the base of rigid body
$\overline{\overline{I}}$	unity dyadic
L	length of fluid field
M	mass of rigid body
m	direction cosine of unit tangential vector
n	direction cosine of unit normal vector
p	hydrostatic pressure
$\overline{q}$	velocity vector
R	source term of Poisson's equation for calculating ϕ
t	time
u	velocity component in x-direction
v	velocity component in y-direction
x,y	cartesian coordinates (also used as subscripts)
Δt	time increment
$\Delta x, \Delta y$	width and height, respectively, of all cells
ε	strain
$\overline{\overline{\varepsilon}}$	strain tensor
σ	normal stress
$\overline{\overline{\sigma}}$	stress tensor
τ	shear stress
$\overline{\overline{\tau}}$	deviatoric stress tensor
∇	del operator
Δ	dilation

μ	modulus of rigidity
κ	bulk modulus
η	non-Newtonian viscosity
η_{app}	apparent viscosity
ν	kinematic viscosity
ϕ	ratio of pressure to mass density of fluid
ρ	mass density of fluid
λ	convergence criterion
λ_f	force tolerance for iterative scheme

Subscripts

11,22,22	diagonal elements of stress tensor
e	identifies trial value for rigid body
i,j	denotes x and y direction, respectively
o	initial impact velocity
w	identifies rigid body
a	externally applied pressure

Superscripts

"	spherical stress tensor
'	deviatoric stress tensor
n	time cycle
m	order of recycle
s	order of iteration

SOIL PROPERTIES AND CONSTITUTIVE RELATIONSHIPS

A valid solution in continuum mechanics requires that the solution satisfies the equations of conservation of mass and of energy, equations of motion, and the equation of state of the material. When the tire of an aircraft landing at relatively high speed strikes the free surface of the soil, a strain-rate which depends upon the landing velocity is imposed on the wheel and the soil. Assuming the deformation will be large, the momentum (or energy) of the aircraft will be dissipated by the soil in the form of elastic deformations and plastic flow. The elastic deformations can be expressed as a function of elastic stresses whereas the plastic flow can be expressed as functions of shear (or viscous) stresses with magnitudes depending on the rate of strain. The yield stress will be defined as the transition point from the elastic to the plastic state. Plastic flow will appear only when the stresses exceed a certain limit indicated by the yield stress. For materials exhibiting distinct yield points, there is no ambiguity in establishing this limit. The stress-strain behavior of most soils seldom indicate any distinct point that may be identified as a yield point. Therefore, in this paper the yield stress will be defined as the transition point from the linear stress-strain behavior to the non-linear stress-strain behavior. Below the yield point, the stress-strain relationship is not rate dependent.

The various phenomena which are generated by the landing impact of an aircraft may be treated as follows:

a. In the extremely "close-in region" in the neighborhood of the impact in which the stresses are very large, with resulting large displacements and velocities of the material, the medium will be in a state of plastic flow. The impact momentum is primarily dissipated by the shear resistance of the material. The phenomenon is of a deviatoric nature.

b. As the magnitude of stresses decreases from the zone of loading, there is a transition region in which the medium undergoes a transition from the plastic flow state to the elastic state. The momentum dissipation in this region is relatively small in comparison with the flow state. The stresses and displacements can be evaluated by the elastic constants of the "solid" material.

c. As the moving load moves away from the "close-in region", the applied load is released, and the flow state reverts to an elastic state. The elastic rebound is a rather common phenomenon observed upon unloading in many simple compression (or tension) tests. It must be noted that the material, after being subjected to plastic flow, reverts to a solid but with elastic and flow properties which may be drastically different from its virgin state.

Some of the physical variables of sandy and clayey soils influencing mass behavior are: air and water content, grain size distribution, and density. The environmental variables affecting the mass behavior are confining pressures, stress history, rate of loading, and duration of loading. The elastic and flow properties of soils are all influenced by these variables. The state in which the soil remains elastic is highly dependent upon the confining pressure.

The state of stress at a point will be described by a stress tensor $\bar{\bar{\sigma}}$, as:

$$\bar{\bar{\sigma}} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

The strain tensor $\bar{\bar{\varepsilon}}$ is given as:

$$\bar{\bar{\varepsilon}} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix}$$

The strain-rate tensor $\bar{\bar{e}}$ is related to strain tensor as:

$$\bar{\bar{e}} = \frac{d\bar{\bar{\varepsilon}}}{dt}$$

It is to be noted that the above relationship is valid only where the strains are small, so that higher order products of strains and rates of strains may be neglected. For a small incremental stress-strain relationship, as will be the case considered in this formulation, the relationship will be valid.

The velocity of a material particle at some point and at time t is given by the vector $\bar{q}$ which is completely described by time rate of change of the displacement functions. The strain rate tensor defined in terms of displacement functions is:

$$\bar{\bar{e}} = \frac{1}{2} (\nabla\bar{q} + \bar{q}\nabla)$$

The total deformation in a soil mass may be decomposed into (a) volumetric deformation caused by hydrostatic (or spherical) stress components, and (b) deviatoric deformation caused by deviatoric stress components. The stress, stain, and strain-rate tensors may be decomposed into hydrostatic and deviatoric parts given as:

$$\begin{aligned} \bar{\bar{\sigma}} &= \bar{\bar{\sigma}}'' + \bar{\bar{\sigma}}' \\ \bar{\bar{\varepsilon}} &= \bar{\bar{\varepsilon}}'' + \bar{\bar{\varepsilon}}' \\ \bar{\bar{e}} &= \bar{\bar{e}}'' + \bar{\bar{e}}' \end{aligned} \tag{1}$$

The deviatoric tensor has the property that the first invariant is zero. Generally the stress-strain behavior of a medium for the hydrostatic stress field is linear.

From small incremental stress-strain theory, the volumetric strain is the dilation Δ given by:

$$\Delta = \nabla \circ \bar{q} \quad .$$

The hydrostatic stress tensor is the mean of the normal stress components given by:

$$\bar{\bar{\sigma}} = \begin{bmatrix} -p & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & -p \end{bmatrix}$$

where

$$-p = \frac{1}{3}(\sigma_{11} + \sigma_{22} + \sigma_{33})$$

If the material is isotropic with a bulk modulus κ (not necessarily a constant), a constant rigidity modulus μ in the elastic state, and a flow parameter η in the plastic flow state, the constitutive equations are:

$$p = \kappa\Delta$$

$$\bar{\bar{\sigma}}' = 2\mu\bar{\bar{\varepsilon}}' \quad \text{(elastic region only)}$$

$$\bar{\bar{\tau}}' = 2\eta\bar{\bar{e}}' \quad \text{(plastic region only)}$$

In comparison with plastic flow, the shear deformation is relatively small in the elastic region. The material behavior in the elastic region may be associated with the conservative (recoverable) part whereas in the plastic region it is associated with the dissipative (non-recoverable) part, in which the flow takes place at constant volume. In large displacement problems, the deformation due to the conservative part will be small in comparison with the dissipative part. Henceforth, conservative-part deformation will be tentatively neglected. For a cohesive soil such as clay, the flow parameter η is non-Newtonian and it is a function of the confining pressure and strain-rate. A convenient method to express the constitutive relationship for a non-Newtonian material is the use of an apparent viscosity, η_{app}, (ref. 5) as shown in figure 1. The constitutive relationships for the Mississippi Buckshot clay of various water contents is given in reference 6.

GOVERNING EQUATIONS

The governing equations describing the motion of a continuous medium are:

$$\frac{d\rho}{dt} + \rho\nabla \cdot \overline{q} = 0 \qquad \text{Conservation of mass}$$

$$\rho\frac{d\overline{q}}{dt} = \rho\overline{g} + \nabla \cdot \overline{\overline{\sigma}} \qquad \text{Conservation of motion} \tag{3}$$

$$\rho\frac{dE}{dt} = \overline{\overline{\sigma}}:\overline{\overline{e}} - \nabla \cdot \overline{b} + c_h \qquad \text{Conservation of energy}$$

$$E = E(\rho,p) \qquad \text{Equation of state}$$

where $\overline{b}$ and c_h are the heat flux vector and radiative heat term respectively.

For compressible flow, the equation of state is necessary to complete the system of equations for solving the unknowns $\overline{q}$, ρ, p, and E. For incompressible flow, the governing equations are reduced to:

$$\nabla \cdot \overline{q} = 0$$

$$\rho \left(\frac{\partial}{\partial t} + \overline{q} \cdot \nabla\right)\overline{q} = \rho\overline{g} + \nabla \cdot \overline{\overline{\sigma}} \tag{4}$$

Imposing the stress strain relationship expressed in terms of displacement functions, the set of governing equations may be solved and the solution is given by these displacement functions in time derivative form.

For reasons given previously, large deformation is attributed to plastic flow so that volumetric deformation due to hydrostatic stresses will be neglected. The soil system is considered to flow under constant volume under the conditions of soil-tire interaction due to a moving wheel of an aircraft.

In the plastic flow state, the constitutive equation of the soil system is:

$$\bar{\bar{\sigma}} = \bar{\bar{\sigma}}'' + \bar{\bar{\tau}}' = -p\bar{\bar{I}} + 2\eta\bar{\bar{e}}'$$

The deviatoric strain-rate tensor expressed in terms of displacement functions is:

$$\bar{\bar{e}}' = \bar{\bar{e}} - \bar{\bar{e}}'' \equiv \frac{1}{2} (\nabla\bar{q} + \bar{q}\nabla) - \frac{1}{3} (\nabla \circ \bar{q})\bar{\bar{I}}$$

The constitutive equation becomes:

$$\bar{\bar{\sigma}} = -p\bar{\bar{I}} + \eta(\nabla\bar{q} + \bar{q}\nabla) - \frac{2}{3}\eta(\nabla \cdot \bar{q})\bar{\bar{I}} \tag{5}$$

Substituting the constitutive equation into Eq. (4), the equation of motion becomes:

$$\rho\left(\frac{\partial}{\partial t} + \bar{q} \cdot \nabla\right)\bar{q} = \rho\bar{g} - \nabla p + \eta\left[\frac{1}{3}\nabla (\nabla \circ \bar{q}) + \nabla^2\bar{q}\right] + (\nabla\eta) \cdot (\nabla\bar{q} + \bar{q}\nabla) - \frac{2}{3} (\nabla \cdot \bar{q}) (\nabla\eta) \tag{6}$$

Imposing the condition of incompressibility, the governing equations describing the motion of a medium are:

$$\begin{aligned} &\nabla \circ \bar{q} = 0 \\ &\rho\left(\frac{\partial}{\partial t} + \bar{q} \circ \nabla\right)\bar{q} = \rho\bar{g} - \nabla p + \eta\nabla^2\bar{q} + (\nabla\eta) \circ (\nabla\bar{q} + \bar{q}\nabla) \end{aligned} \tag{7}$$

where $\eta = \eta(p,e)$. For a special case in which the flow parameter η is treated as a constant, the equation of motion in Eq. (7) is the well known Navier-Stokes equation.

In view of the non-linear form of Eq. (7), the difficulties involved in obtaining an analytic form of solution satisfying Eq.

(7) and the imposed boundary conditions is formidable. By treating the soil as an incompressible and viscous material, the governing equations are similar to those in fluid dynamic problems.

The availability of large, high-speed computers and advanced numerical techniques (ref. 7), provides a powerful tool for solving complex non-linear boundary value problems in fluid dynamics. A computational method based on the Marker and Cell (MAC) technique (ref. 8) is used since the primitive variables of velocity components and pressure are solved directly, and the primitive variables are required to relate the interaction between the solid body and the fluid medium.

The Marker and Cell (MAC) method is a computational method with visual display for solving problems of time-dependent motions of a viscous, incompressible fluid with a free surface. Some recent workers using the MAC method are Donovan (ref. 9) and Viecelli (ref. 10). The MAC method requires that the external wall shapes be confined to the fixed rectangular cells of the Eulerian mesh. In this paper, the MAC method has been adapted and modified to handle the fluid-solid interaction problem involving the moving wall of the solid body. The restriction of the stationary wall boundary has been overcome by an iterative scheme involving the impulse-momentum principle and the translation of coordinates.

For a complete detailed description and discussion of the MAC method, the reader is urged to consult reference 8. Basically the method is a numerical technique for solving problems in incompressible hydrodynamics containing free surfaces. Using two spatial dimensions, the primary dependent variables are the pressure and the two velocity components. The variables are computed in each time step and the solution is advanced in small time increments. Using the Lagrangian approach to represent the fluid motion with time, massless "marker" particles are moved in each time step to new positions according to the velocity components in their vicinities. The marker particles serve to define the free surfaces and they do not enter into the solution of the Navier-Stokes equations.

As with most other numerical methods which work in small incremental form, the MAC method works with a small time cycle. The results of the field variables in each cycle act as initial conditions of the next time cycle, and the calculations are continued for as many cycles as the investigator wishes.

During the small, but finite, time cycle Δt, the flow parameter η may be treated as a constant within each cell, such that $\nabla \eta = 0$. Introducing

$$\phi = \frac{p}{\rho} \qquad \text{and} \qquad \nu = \frac{\eta}{\rho} \tag{8}$$

Eq. (7) describing the motions of an incompressible fluid in two-dimensional Cartesian coordinates are:

$$D = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{9}$$

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} u^2 + \frac{\partial}{\partial y} uv = g_x - \frac{\partial \phi}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{10}$$

$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x} uv + \frac{\partial}{\partial y} v^2 = g_y - \frac{\partial \phi}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{11}$$

The kinematic viscosity is considered to be a constant, but the numerical method can easily account for variable viscosity which may be treated as a function of pressure and velocity components.

Operating on Eq. (10) with $\frac{\partial}{\partial x}$ and Eq. (11) with $\frac{\partial}{\partial y}$ and allowing the interchange of space and time differentials, one obtains the Poisson equation:

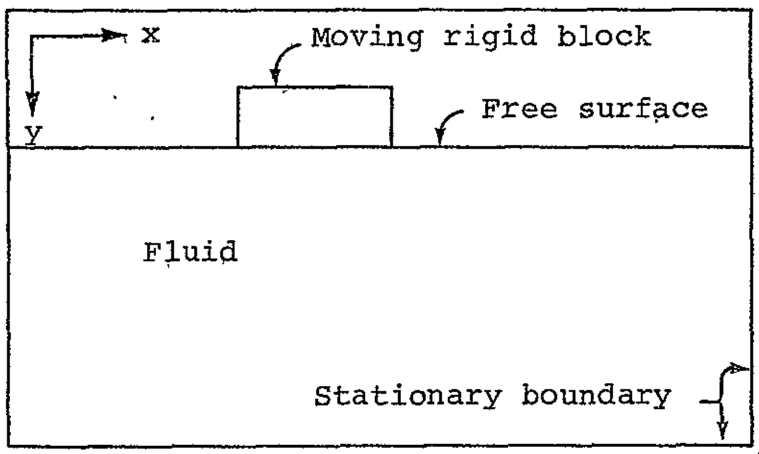
$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = -R \tag{12}$$

where

$$R = \frac{\partial^2 u^2}{\partial x^2} + 2\,\frac{\partial\ uv}{\partial x \partial y} + \frac{\partial^2 v^2}{\partial y^2} + \frac{\partial D}{\partial t} - \nu \quad \frac{\partial^2 D}{\partial x^2} + \frac{\partial^2 D}{\partial y^2}$$

Equations (9) through (12) constitute the basic equations from which a finite difference scheme is developed.

The boundary conditions required for the problem are indicated in sketch 1. The stationary boundaries are represented by no-slip

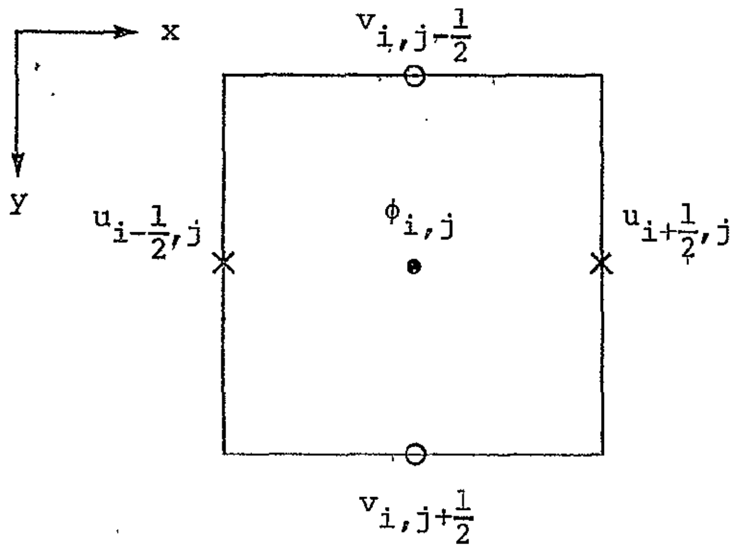


Sketch 1.

impervious boundaries, and they are placed at sufficiently large distances away from the moving boundary to simulate an infinitely large region. The moving rigid block is represented by a no-slip impervious moving boundary. For the symmetric case, the line of symmetry is represented by a free-slip, impervious boundary. At the free surface, the boundary condition requires that the tangential and normal stresses vanish.

DIFFERENCE EQUATIONS

The computational region is divided into rectangular cells as shown in figure 2. The positions of the variables on the rectangular cells are shown in sketch 2. The pressure parameter, ϕ,



Sketch 2.

is evaluated at the cell center. The field variables to be computed for each cell are $u(i+1/2, j)$, $v(i,j+1/2)$, and $\phi(i,j)$. The cell-centered values of u and v and corner values of uv are evaluated by simple averages. Representative examples are:

$$u_{i,j} = \frac{1}{2}\left(u_{i+1/2,j} + u_{i-1/2,j}\right)$$

$$(uv)_{i+1/2,\ j+1/2} = \frac{1}{4}\left(u_{i+1/2,j} + u_{i+1/2,\ j+1}\right) \cdot \left(v_{i,j+1/2} + v_{i+1,\ j+1/2}\right)$$

The time derivative quantities are approximated by forward differences of the first order

$$\left(\frac{\partial D}{\partial t}\right)_{i,j} = \frac{D^{n+1}_{i,j} - D^{n}_{i,j}}{\Delta t}$$

$$\left(\frac{\partial u}{\partial t}\right)_{i+1/2,j} = \frac{u^{n+1}_{i+1/2,j} - u^{n}_{i+1/2,j}}{\Delta t}$$

where superscript $n+1$ denotes the advance time step. All the space derivatives are approximated more accurately by a central difference scheme. Representative examples are:

$$\left(\frac{\partial u^2}{\partial x}\right)_{i,j} = \frac{1}{\Delta x} \left(u^2_{i+1,j} - u^2_{i,j}\right)$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{1}{\Delta x^2} \left(u_{i+1,j} + u_{i-1,j} - 2u_{i,j}\right)$$

$$\left(\frac{\partial^2 uv}{\partial x \partial y}\right)_{i,j} = \frac{1}{\Delta x \Delta y} \Big[(uv)_{i+1/2,j+1/2} + (uv)_{i-1/2,j-1/2}$$

$$(uv)_{i+1/2,j-1/2} - (uv)_{i-1/2,j+1/2}\Big]$$

In advancing the solution from time step n to $n+1$, the Poisson equation is first solved iteratively by the Liebmann's methods of successive correction; that is, to sweep along the positive x-direction and the positive y-direction from the origin. In representing Eq. (12) by finite difference form, the $\frac{\partial D}{\partial t}$ term should ideally vanish in order to satisfy the incompressibility condition. However, any iterative scheme will not solve Eq. (12) exactly so that the divergence D of each cell will not be zero. In order to compensate for this discrepancy, an auxiliary condition is imposed in reference (11) as

$$D^{n+1} = 0 \tag{13}$$

which ideally would result in a zero divergence in the advance time step $n+1$. The auxiliary condition generates a self-corrective scheme in the computational procedure, so that a relatively coarse convergence criterion can be used for solving the Poisson equation as time advances without the accumulation of error. Imposing the condition of Eq. (13) for cell (i,j),

$$\left(\frac{\partial D}{\partial t}\right)_{i,j} = \frac{D^{n+1}_{i,j} - D^{n}_{i,j}}{\Delta t} \simeq \frac{D^{n}_{i,j}}{\Delta t} \tag{14}$$

the finite difference form of Eq. (12) for the pressure parameter is

$$\phi^{n+1}_{i,j} = \frac{1}{Z}\left[\frac{1}{\Delta x^2}\left(\phi_{i+1,j} + \phi_{i-1,j}\right) + \frac{1}{\Delta y^2}\left(\phi_{i,j+1} + \phi_{i,j-1}\right) + R_{i,j}\right] \tag{15}$$

where

$$Z = 2\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right) \tag{16}$$

(cont'd.)

$$R_{i,j} = \frac{1}{\Delta x^2} \left[(u_{i+1,j})^2 + (u_{i-1,j})^2 - 2(u_{i,j})^2\right]$$

$$+ \frac{1}{\Delta y^2} \left[(v_{i,j+1})^2 + (v_{i,j-1})^2 - 2(v_{i,j})^2\right]$$

$$+ \frac{2}{\Delta x \Delta y} \left[(uv)_{i+1/2,j+1/2} + (uv)_{i-1/2,j-1/2}\right.$$

$$\left. - (uv)_{i+1/2,j-1/2} - (uv)_{i-1/2,j+1/2}\right] \qquad (16) \text{ (concl'd.)}$$

$$+ \frac{D_{i,j}}{\Delta t} - \nu \left[\frac{1}{\Delta x^2} (D_{i+1,j} + D_{i-1,j} - 2D_{i,j})\right.$$

$$\left. + \frac{1}{\Delta y^2} (D_{i,j+1} + D_{i,j-1} - 2D_{i,j})\right]$$

and

$$D_{i,j} = \frac{1}{\Delta x} (u_{i+1/2,j} - u_{i-1/2,j}) + \frac{1}{\Delta y} (v_{i,j+1/2} - v_{i,j-1/2}) \qquad (17)$$

After solving the pressure field, the new values of u and v for cell (i,j) are computed from the explicit finite difference form of Eqs. (2) and (3) evaluated at $i+1/2,j$ and $i,j+1/2$, respectively. The equations for u and v are:

$$u^{n+1}_{i+1/2,j} = u_{i+1/2,j} + \Delta t \left\{ \frac{1}{\Delta x} \left[(u_{i,j})^2 - (u_{i+1,j})^2\right]\right. \qquad (18)$$

$$+ \frac{1}{\Delta y} \left[(uv)_{i+1/2,j-1/2} - (uv)_{i+1/2,j+1/2}\right] \text{ (cont'd.)}$$

$$+ g_x + \frac{1}{\Delta x}\left(\phi^{n+1}_{i,j} - \phi^{n+1}_{i+1,j}\right) + \nu\left[\frac{1}{\Delta x^2}(u_{i+3/2,j}\right.$$

$$+ u_{i-1/2,j} - 2u_{i+1/2}) + \frac{1}{\Delta y^2}(u_{i+1/2,j+1} \tag{18}$$

(concl'd.)

$$\left.\left. + u_{i+1/2,j-1} - 2u_{i+1/2,j})\right]\right\}$$

$$v^{n+1}_{i,j+1/2} = v_{i,j+1/2} + \Delta t\left\{\frac{1}{\Delta y}\left[(v_{i,j})^2 - (v_{i,j+1})^2\right]\right.$$

$$+ \frac{1}{\Delta x}\left[(uv)_{i-1/2,j+1/2} - (uv)_{i+1/2,j+1/2}\right]$$

$$+ g_y + \frac{1}{\Delta y}\left(\phi^{n+1}_{i,j} - \phi^{n+1}_{i,j+1}\right) + \nu\left[\frac{1}{\Delta x^2}(v_{i+1,j+1/2}\right. \tag{19}$$

$$+ v_{i-1,j+1/2} - 2v_{i,j+1/2}) + \frac{1}{\Delta y^2}(v_{i,j+3/2}$$

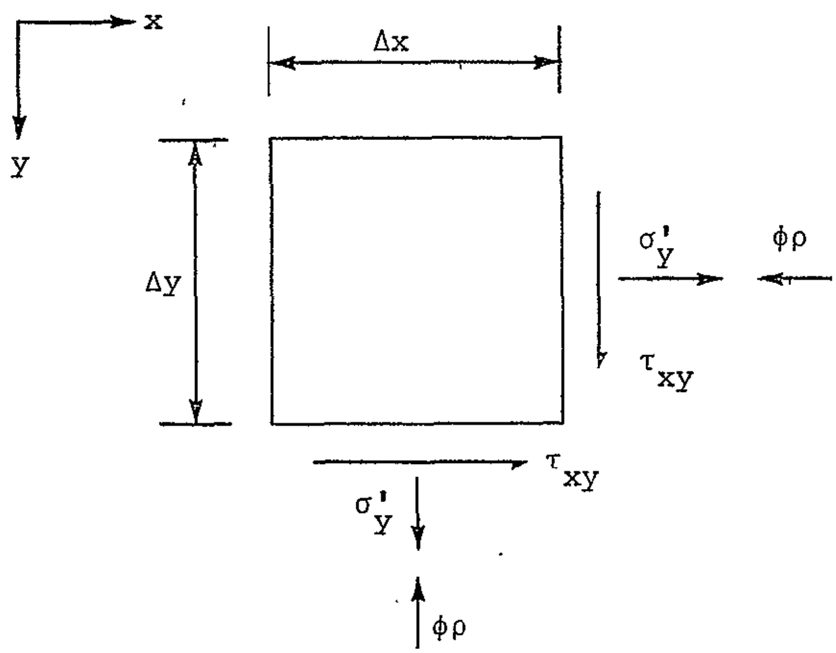
$$\left.\left. + v_{i,j-1/2} - 2v_{i,j+1/2})\right]\right\}$$

BOUNDARY CONDITIONS

Moving Boundary

As the rigid body penetrates into the fluid, the motion of the body introduces a nonsteady boundary condition. The position of the moving boundary (rigid body) depends on the forces exerted by the fluid. An iterative scheme involving the impulse-momentum principle and the translation of coordinates is adopted so that at the end of each time step, the Eulerian mesh is translated to match the boundaries of the moving rigid body.

If the pressure and velocity components are known for cell (i,j) adjacent to the boundary, the stress components of the cell acting on the boundary by the fluid as shown in sketch 3 are:



Sketch 3.

The force components contributed by a single cell are:

$$
\begin{aligned}
f_x &= \sigma_x{}'\Delta y + \tau_{xy}\Delta x - \phi\rho\Delta y \\
f_y &= \sigma_y{}'\Delta x + \tau_{xy}\Delta y - \phi\rho\Delta x
\end{aligned}
\qquad (20)
$$

where

$$\sigma_x{}' = 2\eta \frac{\partial u}{\partial x}$$

$$\sigma_y{}' = 2\eta \frac{\partial v}{\partial y}$$

$$\tau_{xy} = \eta \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

The force components F_x and F_y experienced by the rigid body, treated as constant during one time step, are computed by integration of the forces of all the cells adjacent to the moving boundary. Equating the change in momentum of the rigid body between any time interval n and $n+1$, to the force acting on the body, one observes

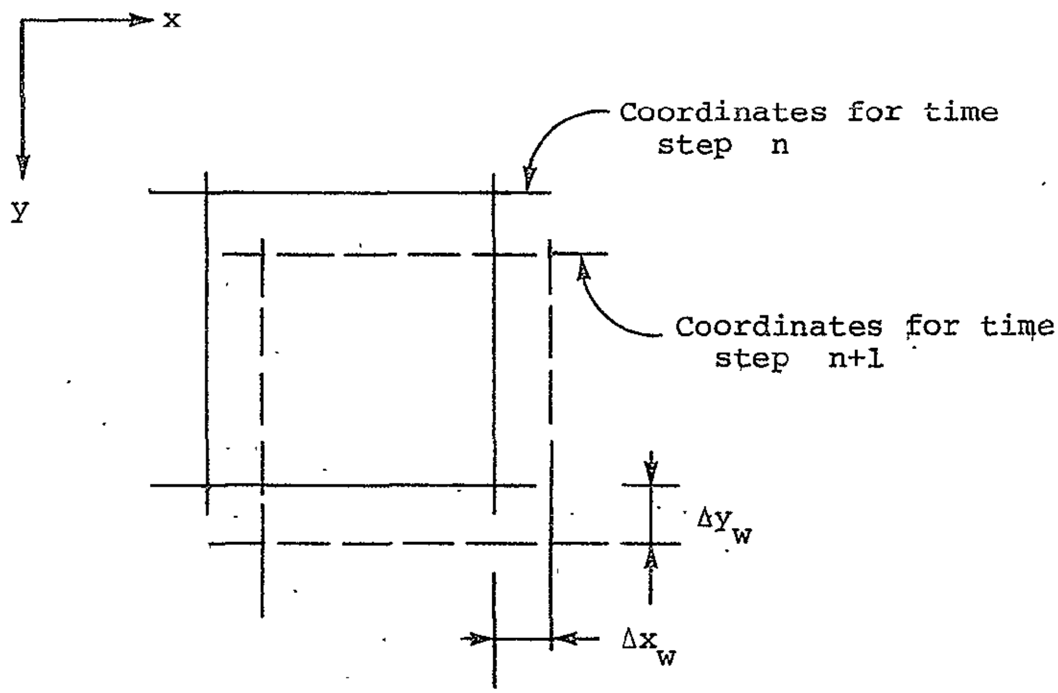
$$M \frac{d\bar{q}}{dt} = M\bar{g} + \bar{F} \tag{21}$$

The finite difference form of Eq. (21) for the velocity components of the rigid body are:

$$\begin{aligned} u_w^{n+1} &= u_w^n + \frac{F_x^{n+1}}{M} \Delta t \\ v_w^{n+1} &= v_w^n + \left(g_y + \frac{F_y^{n+1}}{M} \right) \Delta t \end{aligned} \tag{22}$$

with $g_x = 0$. The values of the force components F_x^{n+1} and F_y^{n+1} are required as boundary conditions at the beginning of each time cycle and must be found by an iterative scheme. The displacement of the body at the end of the time cycle, shown in sketch 4, is computed from the average velocities as:

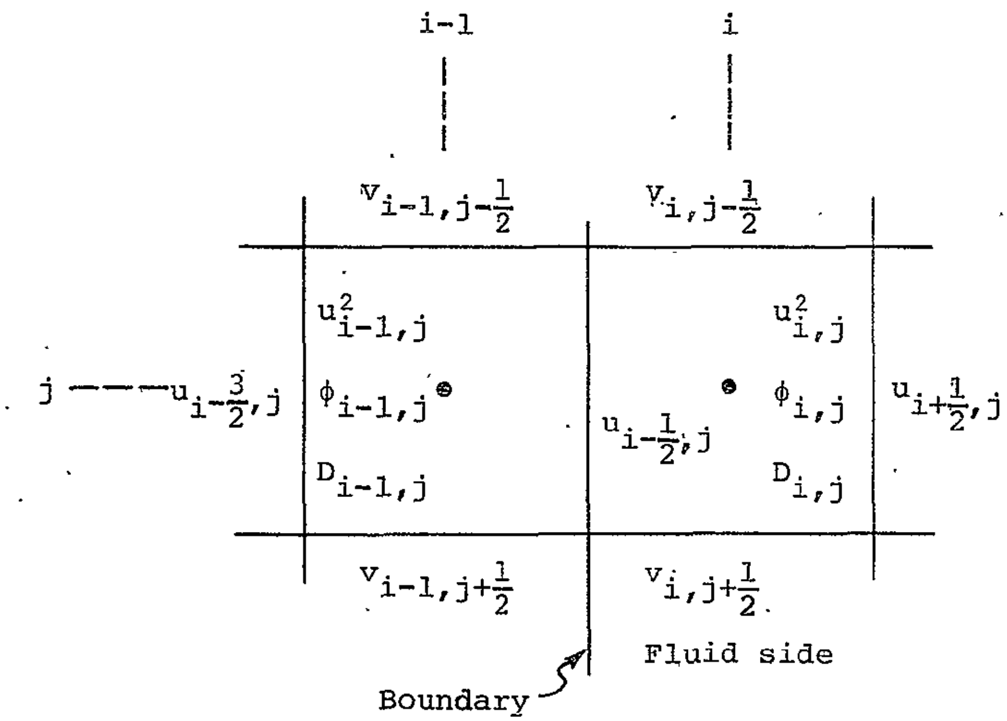
$$\begin{aligned} \Delta x_w &= \frac{u_w^{n+1} + u_w^n}{2} \Delta t \\ \Delta y_w &= \frac{v_w^{n+1} + v_w^n}{2} \Delta t \end{aligned} \tag{23}$$



Sketch 4.

Since the advanced time velocities u_w^{n+1} and v_w^{n+1} as expressed by Eq. (22) must also satisfy the momentum equations expressed by Eqs. (18) and (19), a judicious choice of trial values F_{xe}^{n+1} and F_{ye}^{n+1} representing F_x^{n+1} and F_y^{n+1} respectively are estimated at the beginning of the computation. The estimated advance time velocities of the body u_e^{n+1} and v_e^{n+1} can be immediately evaluated and the reference coordinates are then translated according to the magnitudes expressed by Eq. (23) using u_e^{n+1} and v_e^{n+1}.

In solving the finite difference Poisson equation, values of ϕ, u, and v outside the computing region are obtained from the appropriate mementum and continuity equation. Although the types of boundary conditions applied will depend on the boundary under consideration, the boundary conditions for the case of a boundary to the left of the fluid shown in sketch 5 will be presented. The conditions for other boundaries are analogous.



Sketch 5.

The conditions for the moving boundary are:

1) $v_{i-1,j+1/2} = 2v_w - v_{i,j+1/2}$

$v_{i-1,j-1/2} = 2v_w - v_{i,j-1/2}$

2) $u^2_{i-1,j} = u^2_{i,j}$

3) $(uv)_{i-1/2,j+1/2} = u_w \times v_w$

$(uv)_{i-1/2,j-1/2} = u_w \times v_w$

4) $\phi_{i-1,j} = \phi_{i,j} + \left(\frac{F_x}{M}\right) \Delta x + 2\nu(u_w - u_{i+1/2,j})/\Delta x$

5) $u_{i-1/2,j} = u_w$

6) $D_{i-1,j} = D_{i,j}$

The no-slip and free-slip boundary conditions are the same as given in reference 5.

Free Surface

The treatment of a free surface cell is adopted according to the procedure given by Hirt and Shannon, reference 12. The normal and tangential stress conditions at the free surface can be expressed as

$$\phi = \phi_a + 2\eta \left[n_x^2 \frac{\partial u}{\partial x} + n_x n_y \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + n_y^2 \frac{\partial v}{\partial y} \right]$$

$$2n_x m_x \frac{\partial u}{\partial x} + (n_x m_y + n_y m_x) \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + 2n_y m_y \frac{\partial v}{\partial y} = 0 \qquad (24)$$

where the direction cosines of the normal unit vector and the corresponding tangential unit vector relative to the surface are related as:

$$n_x = -m_y$$

and

$$n_y = m_x$$

The tangential stress condition is approximately satisfied by selecting the velocity components at the exterior edges of the surface cell so that the divergence of the cell vanishes. The normal stress condition can be treated accurately provided the orientation of the free surface is well defined. At present, the free surface defined by marker particles is approximated to lie either along the cell boundaries or along the diagonal of a cell. The normal unit vector in both of these cases is well defined.

COMPUTING METHOD

The method of this paper calculates a transient problem by working through a sequence of small time increments. The results of each time cycle are used to define the initial conditions for the next cycle. Each cycle itself is divided into the following phases:

1. Using a linear interpolation scheme, the trial values of the force components F_{xe}^{n+1} and F_{ye}^{n+1} are estimated. Fig. 3 shows that the correct choice will lie along the 45° line. Using the previous trial and computed values of the forces in the m and $m - 1$ recycles, the trial values for the next recycle will be chosen at the point of intersection. For the x-direction

$$F_{xe}^{n+1} = \frac{F_x^{m-1} F_{xe}^{m} - F_x^{m} F_{xe}^{m-1}}{F_{xe}^{m} - F_x^{m} - F_{xe}^{m-1} + F_x^{m-1}} \tag{25}$$

where F_{xe} is the estimated and F_x is the computed force in the m^{th} recycle. The equation for F_{ye}^{n+1} is analogous.

2. The estimated velocities of the moving boundary u_e^{n+1} and v_e^{n+1} computed from Eq. (22) are used in Eq. (23) for determining coordinate translations Δx_w and Δy_w.

3. The pressure $\phi_{i,j}^{m}$ for each cell is calculated for the new coordinates. Using the values of neighboring cells as shown in Fig. 4,

$$\phi_{i,j}^{m} = (1 - \Delta x_w)(1 - \Delta y_w)\ \phi_{i,j} + \Delta x_w \left(1 - \frac{\Delta y_w}{2}\right) \phi_{i+1,j} + \Delta y_w \left(1 - \frac{\Delta x_w}{2}\right) \phi_{i,j+1}$$

4. Eq. (15) is solved iteratively for the pressure $\phi_{i,j+1}^{m}$ by successive correction until the convergence criterion is satisfied as given in reference 8 as:

$$\left[\frac{\left|\phi^{s} - \phi^{s+1}\right|}{\left|\phi^{s}\right| + \left|\phi^{s+1}\right| + u_{i,j}^{2} + v_{i,j}^{2} + \left|g_y h\right| + \left|g_x L\right|}\right]_{max} < \lambda \qquad (26)$$

where s means the s^{th} iteration and λ is the convergence criterion which is a predetermined small positive number.

5. The new velocities of each cell are computed by Eqs. (18) and (19).

6. The forces F_x^{n+1} and F_y^{n+1} are computed by integrating the forces of all the cells adjacent to the moving boundary.

7. Steps (1) through (6) are recycled or repeated until the force tolerance is satisfied

$$\begin{aligned} \left|F_x^{n+1} - F_{xe}^{n+1}\right| &< \lambda_f \\ \left|F_y^{n+1} - F_{ye}^{n+1}\right| &< \lambda_f \end{aligned} \qquad (27)$$

where λ_f is the force tolerance.

8. The apparent viscosity, η_{app}, is computed for each cell using the cell centered velocity and the stress-strain-rate curve, (fig. 1).

9. The marker particles are moved to their new positions according to the same procedure given in reference 8. At the start of the calculation, the fluid configuration is represented by a uniform distribution of four particles per cell. The particles define the new free surface.

10. The cells are reflagged according to the new fluid configuration. The next time cycle can immediately begin.

A listing of the computer program is given in the Appendix.

STABILITY AND ACCURACY

All finite difference schemes expressed in explicit form for initial value problems are governed by some form of stability requirements. Roache (ref. 7) indicated that the existing mathematical theory for numerical solutions of nonlinear partial differential equations is still inadequate and that there are no rigorous stability analyses, error estimates, or convergence proofs. He wrote (ref. 7) "In computational fluid dynamics, it is still necessary to rely heavily on rigorous mathematical analysis of simpler, linearized, more or less related problems, and on heuristic reasoning, physical intuition, wind tunnel experience, and trial-and-error procedures."

The stability conditions recommended by MAC (ref. 8) are:

$$C \, \Delta t < \frac{2\Delta x \, \Delta y}{\Delta x + \Delta y} \tag{28a}$$

and

$$2\nu \, \Delta t < \frac{\Delta x^2 \, \Delta y^2}{\Delta x^2 + \Delta y^2} \tag{28b}$$

Hirt (ref. 13) in using a technique applicable to nonlinear equations with variable coefficients, further recommended the following "rule-of-thumb" for the MAC method:

$$\nu < \frac{1}{2} \Delta t \, \bar{u}^2 \tag{28c}$$

and

$$\nu < \frac{1}{2} \Delta x^2 \left(\frac{\partial \bar{u}}{\partial x}\right) \tag{28d}$$

where

$\bar{u}$ is the average maximum fluid speed

$\frac{\partial \bar{u}}{\partial x}$ is the average maximum velocity gradient in the direction of the flow.

It is necessary in any calculation to consider which one of these conditions is the more restrictive, and which will govern the size of Δt, the time increment per cycle. For calculations involving large values of viscosity, the diffusional stability condition, Eq. (28b) will be used as a criteria for choosing the time step, Δt.

The degree of accuracy desired for a given problem also dictates the size of the cells and time steps. In view of the iterative scheme adopted for the moving boundary-fluid interaction problem, experience indicated that the time step must be small enough to restrict the displacement of the rigid body Δx_w and Δy_w to one quarter the cell size. The degree of accuracy in choosing the convergence criterion λ used for the pressure iteration has a direct bearing on satisfying the incompressibility condition $D^{n+1}_{i,j} = 0$. The final result will be the force components F_x and F_y acting on the rigid body which is the prime feature of interest. Generally, if Eq. (15) is to be iterated to a high level of accuracy by imposing a very small value for λ, the computational time will be increased many fold for each time step, and also as λ becomes smaller, the force components approach an asymtotic value. Figure 5 shows the relationship, for two cases, of λ and the vertical accelerative force F_y (expressed in g units) for a vertical penetration problem at the end of the first time cycle. Taking g_{max} to be the asymtotic value, the gain in accuracy will be less than two percent for $\lambda < 2 \times 10^{-4}$ for case (a) and for $\lambda < 2 \times 10^{-5}$ for case (b) with force tolerance $\lambda_f = 1 \times 10^{-4}$. Using the computational method in an engineering application, it is necessary to establish the relationship between the convergence criterion and the feature of interest for each case to determine the magnitude of the convergence criterion for the computational experiment.

EXAMPLES OF THE CALCULATIONS AND COMPARISON WITH EXPERIMENTS

Example I. Low velocity penetration of rigid body into still fluid

In an attempt to provide experimental corroboration of the computational techniques, the impact of a rigid flat block on a still water surface was simulated as an example (ref. 14).

The apparatus shown in figure 6 was developed to experimentally observe and record the acceleration, pressure, and displacement of a block dropping in water. The water tank itself, a former wind tunnel test section, measures 1.23 m (4 ft) by 1.52 m (5 ft) by 1.23 m (4 ft) high, and can be sealed to provide for impact tests under reduced atmosphere. Viewing ports are provided (as shown) for lighting, camera access, and visual monitoring. For these tests, water was added to a depth of approximately 48 cm (19 in.) and colored with sea marker dye (sodium flouriscene) to provide better visual contrast.

The drop model shown in figure 7 was a rectangular structure fabricated from .95 cm (.375 in.) thick sheet plastic, and measured 7.6 cm (3 in.) wide by 25.4 cm (10 in.) high, and 61 cm (24 in.) in length. The length-to-width ratio was purposely made very high to better approximate the two-dimensional case used in the computer study. The minimum dropping weight of the model was 6.24 kg (13.75 lb), and provision was made for adding ballast in the form of lead shot so that the effects of added mass might be observed. The model was held at the desired height above the water surface by an electromagnet (shown in figure 7) which could be prepositioned and released from outside the tank, again to allow for future tests in reduced atmosphere. During a drop, the model was guided by teflon-coated steel rods to assure a flat impact on the water surface. Coil springs were installed on the bottom of the tank to absorb the shock loading resulting from extreme penetration at high dropping rates.

Model displacements were recorded by a high-speed camera (seen in the foreground of figure 6) operating at 500 frames/sec. High-speed color film was used with the necessary lighting provided by one quartz lamp inside the tank and two lamps on the outside aimed through viewing ports.

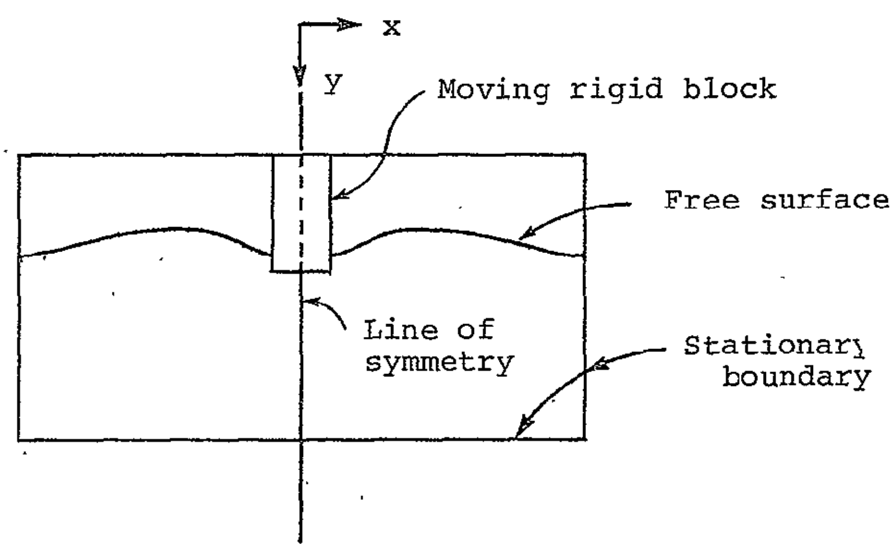
A reference probe (shown at the left of figure 7) could be adjusted vertically to pinpoint the still water surface, and vertical displacement of the model could be measured from the film by comparing the reference marks on the probe and on the face of the model. A time-code generator triggered an integral timing light in the camera so that time histories of displacement could be obtained using a film analyzer.

A strain-gage pressure transducer of 0 - 103 kPag (0 - 15 psig) range was installed inside the model with the sensing face flush with the bottom of the model. A ±12 g accelerometer was also installed inside the center bottom of the model oriented to sense vertical accelerations only, and outputs from these instruments were taken off the model and out through the top of the tank with trailing cables. Both outputs were amplified to increase sensitivity at low drop heights and were recorded on a direct-write oscillograph. The same time-code generator which triggered the camera timing light also provided a time base for the recorder.

Tests were conducted over a range of drop heights for two model weights. In every case, drop height was established with reference to the quiescent water level, and the drop was not initiated until the water surface was completely still. Camera orientation was checked periodically to insure parallel alinement with the surface. The camera and recorder were started about two seconds after impact.

Numerical Calculation

The arrangement of the computational domain is given in sketch 6.



Sketch 6.

Only one half of the field is necessary for a vertically symmetric case. The initial vertical impact velocity and the mass of the block were chosen from the experimental values. To test the computational method, the mass of the block was chosen small enough to allow the hydrostatic fluid pressure to push the block back towards the fluid surface.

Using a square grid, each cell had dimensions $\Delta x = \Delta y = .9524$ cm. The fluid depth and width were chosen to simulate an infinite medium. The fluid depth measured from the face of the block was represented by 24 cells (23 cm), and the width of the fluid for a half field was represented by 60 cells (57 cm). The height of the block was chosen to assure that at full penetration, the block would not be submerged in the fluid. Dimensions for a half-block were 4 cells (3.8 cm) wide × 34 cells (32.4 cm) long. The kinematic viscosity of the fluid was 0.1 cm^2/sec. The mass used for the two-dimensional case was calculated from the model weight per centimeter of length.

Comparison of Results

A comparison of computational and experimental results is summarized in table 1 and presented in figures 8(a), (b), and (c) to illustrate the effects of changes in drop height and in body mass. In figure 8(a), displacement, acceleration, and pressure time histories are shown for an experimental drop height of 1 cm using a model weight of 6.24 kg (13.75 lb). Analysis of the film used to measure displacement indicated the initial impact velocity, v_o, to be 33 cm/sec within an accuracy of ±5 percent. The computations were conducted for $v_o = 30$ cm/sec, with convergence criterion, $\lambda = 2 \times 10^{-4}$ and force tolerance $\lambda_f = 1 \times 10^{-4}$. A time increment of $\Delta t = .003$ sec was used for the first five cycles, and was reduced to .002 sec thereafter as the velocity of the model increased. Computations were terminated at $t = .411$ sec after the model had attained maximum penetration. As can be seen in figure 8(a), the agreement between experiment and computation is reasonably good for the water penetration phase. Calculations were also performed with the space grids, reduced by one-half with $\Delta x = \Delta y = .476$ cm and essentially the same results were obtained up to time $t = .22$ seconds after which the computational method became unstable.

The effects of increasing drop height to 2 cm are shown in figure 8(b) where much the same trends are noted as for the 1 cm case. In this instance, initial impact velocity for the experiment was found to be 47 cm/sec ± 5 percent, so for the computation $v_o = 50$ cm/sec was used. The convergence criterion, force tolerance, and time increments were the same as for figure 8(a), and computation was terminated at $t = .395$ sec.

In figure 8(c), the mass of the model was increased to 153.5 cm, and the drop height was 1 cm. At this weight, the experimental initial impact velocity was found to be 30 cm/sec, and $v_o = 30$ cm/sec was also used in the computations. Again, the time increment, Δt, was .003 sec for the first five cycles, and was reduced to .002 sec thereafter. At time $t = .155$ sec, the displacement of the block exceeded one-quarter cell size, so the time increment was reduced 25 percent to .0015 sec, and computations were terminated at $t = .2745$ sec. Again, reasonable agreement is noted between experiment and computation. However, it is felt that the agreement is sufficiently good to validate the program and give confidence to the computational techniques employed.

As previously suggested, the computational method not only gives a solution for force and displacement of the block during impact and entry, but computes displacement, velocity, and pressure throughout the flow field. The computed behavior of the flow field cannot readily be examined experimentally, but the agreement between computed and experimental block behavior lends credence to the predicted fluid dynamics. As an example of how the computations may be used to examine in detail the evolution of the fluid dynamics during water entry, figures 9(a) and (b) presents particle and velocity plots for an initial block impact velocity, $v_o = 50$ cm/sec.

The fluid pressure developed under the block may also be of interest in certain applications. Figure 10 shows how the excess hydrostatic pressure developed along the centerline of block varies with fluid depth as measured from the bottom of the block as it contacts the free surface. Two initial impact velocities are shown, $v_o = 30$ cm/sec and $v_o = 50$ cm/sec, and the figure indicates that the size of the fluid field originally chosen was adequate to simulate an infinite medium, since the excess hydrostatic pressure is insignificantly small.

Example II. High velocity impact of a rigid block into a very viscous non-Newtonian fluid.

The arrangement of the computational domain is given in sketch 1. This numerical example simulated the case of an aircraft wheel landing on soil runway with a locked-wheel braking condition and zero lift. The dimensions and mass of the rigid block represented a two-dimensional equivalent of the case given in the figure 21 of reference 2. The stress and strain-rate curve for the viscous non-Newtonian fluid, simulated a soft clay soil with water-content of 33 percent (fig. 11d of ref. 6) as shown in figure 1.

Using a square grid, each cell had dimensions $\Delta x = \Delta y = 2$ cm. The fluid depth measured from the bottom face of the block was represented by 10 cells (20 cm) and the width of the fluid was represented by 29 cells (58 cm). The height of the block was represented by 4 cells (8 cm) and the width represented by 7, 9 and 11 cells (14, 18 and 22 cm) to simulate various contact length. The mass per centimeter of length was 90,700 gm. $\Delta t = 1 \times 10^{-5}$ sec was chosen using the diffusional stability condition and the maximum flow parameter, η_{app}, obtained from the initial secant slope of the stress and strain-rate curve. This small time step will impose a severe restriction on the practical use of the method because of the large amount of computer time required for a given problem in which the time history of the force and displacements of the rigid block would be long. Only short calculations were made for this example, for $t = 0$ to

$t = 1 \times 10^{-4}$ sec, using the same convergence criterion and force tolerance of example I.

Figure 11 shows the time history for the horizontal and vertical forces experienced by the block for various contact lengths. In all cases the initial horizontal impact velocity, $u_o = 500$ cm per second, vertical velocity $v_o = 0$ and the gravitational force are instantly applied. The horizontal force, developed by the no-slip boundary condition, is proportional to the contact length. The magnitude of the horizontal force decreased gradually with time, whereas the magnitude of the vertical force oscillated upwards. The horizontal and vertical displacements would be very small. Figure 12 indicates the displacements at the end of $t = 1 \times 10^{-4}$ sec. As expected, the horizontal and vertical displacements decreased as the contact length increased.

The variation of forces with initial horizontal impact velocity was calculated for case with contact length = 14 cm. Figure 13 shows the effects of increasing initial horizontal impact velocity for the time interval between $t = 3 \times 10^{-5}$ sec and $t = 1 \times 10^{-4}$ sec. As expected, the vertical force remained constant as the horizontal velocity increased. The horizontal force continued to increase as the horizontal velocity increased. Since the sinkage of the block (vertical displacement) was extremely small, the horizontal force experienced by the block was contributed mainly by the viscous tractive force and resistance associated with the inertia effect or the "bow-wave" did not materialize. The viscous tractive force increases linearly with velocity.

CONCLUDING REMARKS

The viability of the computational method for studying the low velocity penetration of a flat rigid block on a fluid medium has been demonstrated to agree well with experimentation. The method is capable of providing the complete pressure and velocity fields, and the visual display of the history of fluid configuration.

Some limitations inherent in the method must be mentioned. First, only non-turbulent flows are considered in the model. At high speeds of impact, the turbulent effects may not be ignored and some implementation for turbulent simulation may have to be incorporated in the method. Second, if the main feature of interest is centered at the short time impact phase, the choice of time increment may require some computational experimentation. This limitation will not be critical if the main feature of interest is centered at the maximum penetration of the impact body. In the comparative test cases by the numerical calculation and by experiment, the viscous effect is relatively insignificant since the viscosity of water is small, and the forces acting on the body are contributed predominantly by the pressure.

The computational method has not been tested for the case of fluid-solid interaction involving a highly viscous non-Newtonian medium. The apparatus to provide experimental corroboration of the computational technique for the case of example II would be complex and no experimental study was made for this case.

REFERENCES

1. Crenshaw, B. M., Butterworth, C. K., and Truesdale, W. B.: Aircraft Landing Gear Dynamic Loads from Operation on Clay and Sandy Soil. AFFDL-TR-69-51, U. S. Air Force, February 1971.

2. Leland, T. J. W., and Smith E. G.: Aircraft Tire Behavior During High-speed Operations in Soil. NASA TN D-6813, August 1972.

3. Luming, H.: Analytical Aircraft Landing Gear-soil Interaction Phase III. Technical Report AFFDL-TR-70-142, U. S. Air Force, May 1971.

4. Crenshaw, B. M.: Development of an Analytical Technique to Predict Aircraft Landing Gear/Soil Interation, AFFDL-TR-74-115 I and II, U. S. Air Force, January 1975.

5. Maxwell, B.: The Dynamic Behavior and Stress Relaxation of Polymer Melts, Polymer Engineering and Science, 8, No. 4, 1968.

6. Cheng, R. Y. K.: The Effect of Shearing Strain-rate on the Ultimate Shearing Resistance of Clay, ODU School of Engineering Technical Report 75-C1, March 1975.

7. Roache, P. J.: Computational Fluid Dynamics, Hermosa Publisher, 1972.

8. Welch, J. E., Harlow, F. H., Shannon, J. P., and Daly, B. J.: The MAC Method, Los Alamos Scientific Laboratory Report, LA-3425, 1966.

9. Donavon, L. F.: A Numerical Solution of Unsteady Flow in a Two-Dimensional Square Cavity, AIAA Journal, 8, No. 3, March 1970, pp 524-529.

10. Viecelli, J. A.: A Computing Method for Incompressible Flows Bounded by Moving Walls, Journal of Computational Physics, 8, 1971, pp. 119-143.

11. Hirt, C. W., and Harlow, F. H.: A General Corrective Procedure for the Numerical Solution of Initial-Value Problems, Journal of Computational Physics, 2, 1967, pp. 114-119.

12. Hirt, C. W., and Shannon, J. P.: Free-Surface Stress Conditions for Incompressible-Flow Calculations, Journal of Computational Physics, 2, 1968, pp. 403-411.

13. Hirt, C. W.: Heuristic Stability Theory for Finite-Difference Equations, Journal of Computational Physics, 2, 1968, pp. 339-355.

14. Cheng, R. Y. K. and Leland, T. J. W.: Numerical Solution for Low Velocity Penetration of Rigid Body into Still Fluid, Numerical Methods in Fluid Dynamics, Ed. Brebbia, C. A., and Connor, J. J., Pentech Press, 1974, pp. 272-288.

APPENDIX

```
C        SOIL-TIRE INTERACTION
C
C
C
C
C
C    *** DEFINITIONS OF VARIABLES
C
C        ALPHAU=X-VELOCITY FRACTION OF WHEEL AT TIME STEP N+1 TO STEP N
C        ALPHAV=Y-VELOCITY FRACTION OF WHEEL AT TIME STEP N+1 TO STEP N
C        CONV=CONVERGE CRITERIA OF PRESSURE ITERATION
C        D(I,J)=DISCREPANCY TERM (DIVERGENCE)
C        DATA(I)=ARRAY FOR TAPE STORAGE OF DATA ON TAPES 8 AND 13
C        DATB(I)=ARRAY FOR TAPE STORAGE OF DATA ON TAPE 10
C        DMAX= MAXIMUM DISCREPANCY (EXCLUDING FIRST CYCLE)
C        DT=TIME STEP BETWEEN CYCLE
C        DTCP=TIME BETWEEN CELL PRINTS
C        DTPP=TIME BETWEEN PARTICLE PRINTS
C        DTSP=TIME BETWEEN STRESS PRINTS
C        DX=CELL WIDTH
C        DXK=INITIAL PARTICLE SPACE IN X-DIRECTION
C        DXSF=SHIFTING DISTANCE OF WHEEL IN X-DIRECTION
C        DY=CELL HEIGHT
C        DYK=INITIAL PARTICLE SPACE IN Y-DIRECTION
C        DYSF=SHIFTING DISTANCE OF WHEEL IN Y-DIRECTION
C        ERRORU=PERCENTAGE DIFFERENCE BETWEEN ESTIMATE AND
C        CALCULATED FINAL HORIZONTAL WHEEL FORCES
C        ERROPV=PERCENTAGE DIFFERENCE BETWEEN ESTIMATED AND
C        CALCULATED FINAL VERTICAL WHEEL FORCES
C        ETA(I,J)=SOIL FLOW PARAMETER OF EACH CELL
C        ETAO=SOIL FLOW PARAMETER TREATED AS CONSTANT
C        F(I,J)=1.0=RND(STATIONARY RIGID BOUNDARY CELL)
C        F(I,J)=2.0=FULL (FULL CELL)
C        F(I,J)=3.0=SUR  (FREE SURFACE CELL)
C        F(I,J)=4.0=EMP  (EMPTY CELL)
C        F(I,J)=5.0=RNOW (WHEEL CELL)
C        F(I,J)=6=SYM (VERTICAL LINE OF SYMMETRY AT LEFT BOUNDARY)

C        FC(I,J)=0.0=SECOND STRESS INVARIANT LESS THAN FLOW YIELD
C        FC(I,J)=1.0=SECOND STRESS INVARIANT GREATER THAN FLOW YIELD
C        FC(I,J)=2=FORMERLY EMPTY CELL FLAGGED SUR WHICH NEEDS AN ETA
C        FE(I,J)=1.0=COR (CORNER CELL OF WHEEL)
C        FE(I,J)=2.0=RB  (CELL DIRECTLY ADJACENT TO RND AND RNOW)
C        FK(I,J)=VARIABLE-YIELD CRITERIA
C        FKO=CONSTANT-YIELD CRITERIA
C        FS= INITIAL VALUE OF J FOR FREE SURFACE CELLS
C        FX,FY= TRIALSHIFT VALUE
C        GAMAE= STRAIN OFFSET FOR ELASTIC RANGE
C        GX=GRAVITY IN X-DIRECTION
C        GY=GRAVITY IN Y-DIRECTION
C        IBAR=NO. OF CELL IN X-DIRECTION
C        IC= ITERATION COUNT (NUMBER OF PRESSURE DENSITY ITERATIONS)
C        IDMAX,JDMAX= CELL CONTAINING DMAX
C        IFSYM=0=GENERAL IMPACT PROBLEM
C        IFSYM=1=PROBLEM SYMMETRIC ABOUT VERTICAL C/L OF GEOMETRY
C        ITER=DUMMY ITERATION COUNT OF FC LOOP
C        IWHEEL=0=AUXILLARY GEOMETRY PROBLEM
C        IWHEEL=1=WHEEL GEOMETRY PROBLEM
C        JBAR=NO. OF CELL IN Y-DIRECTION
C        KBAR=TOTAL NUMBER OF PARTICLES
C        KC(I,J)=NUMBER OF PARTICLES IN CELL I,J
C        MHAX=MAX. NO. OF POINT TO BE INTERPOLATED=1
C        MNCVS=MAX. NO. OF CURVES
C        MNPTS=MAX. NO. OF POINT IN THE INDEPENDENT VARIABLE ARRAY
C        MS=NO. OF DESIRED POINTS TO BE INTERPOLATED=1
C        NCS= NUMBER OF CYCLES STORED ON TAPE
C        NCVS=ACTUAL NO. OF CURVE
C        NIF=ENDING I VALUE FOR ON LINE PRINT
C        NIS=STARTING I VALUE FOR ON LINE PRINT
C        NJF=ENDING J VALUE FOR ON LINE PRINT
C        NJS=STARTING J VALUE FOR ON LINE PRINT
C        NOR=NUMBER OF RE-CYCLES INCURRED
C        NS=ACTUAL NO. OF POINT IN THE INDEPENDENT VARIABLE ARRAY
C        NSP=NUMBER OF CYCLES OF STRESSES STORED ON TAPE
C        NX=INITIAL NO. OF PARTICLE IN X-DIRECTION
C        NY=INITIAL NO. OF PARTICLE IN Y-DIRECTION
C        PHI(I,J)= CELL PRESSURE DIVIDED BY (CONSTANT) DENSITY
C        PHIO(I,J)= INITIAL VALUE BEFORE CONVERGENCE ITERATIONS
C        R(I,J)= SOURCE TERM FOR PRESSURE CALCULATIONS
C        RHOS= MASS DENSITY OF MEDIUM (GM/(CM**3))
C        RW=WHEEL RADIUS
```

```
C         SDXSF= ACCUMULATED MOVEMENT OF WHEEL IN HORIZONTAL DIRECTION
C         SDYSF= ACCUMULATED MOVEMENT OF WHEEL IN VERTICAL DIRECTION
C         SFX=TOTAL FORCES IN X-DIRECTION
C         SFXG=ESTIMATE OF FINAL HORIZONTAL WHEEL FORCES
C         SFY=TOTAL FORCES IN Y-DIRECTION
C         SFYG=ESTIMATE OF FINAL VERTICAL WHEEL FORCES
C         SIGMAX(I,J)= NORMAL STRESS IN X-DIRECTION
C         SIGMAY(I,J)= NORMAL STRESS IN Y-DIRECTION
C         SIGMA1(I,J)=MAXIMUM PRICIPLE STRESS OF EACH CELL
C         STNMAX=MAXIMUM STRAIN ENCOUNTERED DURING CYCLE
C         STRAN= STRAIN (1./SEC) DATA POINTS
C         STRES=STRESS (DYNES/(CM**2)) DATA POINTS
C         SUMR= SUMMATION OF ABSOLUTE VALUE OF R(I,J)
C         T= TIME
C         TAPE(8)-  OUTPUT STORAGE OF VELOCITIES
C         TAPE(9)-  OUTPUT STORAGE OF STRESSES
C         TAPE(10)- OUTPUT STORAGE OF PARTICLE COORDINATES
C         TAPE(11)- TEMPORARY STORAGE OF VELOCITIES OF PREVIOUS TIME STEP
C         TAPE(13)- OUTPUT STORAGE OF PHI,ETA,F,FF
C         TAPE(14)- TEMPORARY STORAGE OF PARTICLE COORDINATES
C         TAPE(15)- TEMPORARY STORAGE OF PARTICLE COORDINATES
C         TAPE(16)- TEMPORARY STORAGE OF VELOCITIES FOR RESTARTING
C         TAUXY(I,J)= SHEAR STRESS ON X-Y PLANE
C         TCP= TIME FOR CELL PRINT
C         TDS=TIME OF DATA STORAGE
C         THETA(I,J)=ANGLE BETWEEN POSITIVE X-DIRECTION AND SIGMA1(I,J)
C         TL=TIME TO QUIT
C         TPP= TIME FOR PARTICLE PRINT
C         TSP= TIME FOR STRESS PRINT
C         U(I,J)=X-VELOCITY OF CELL
C         UG=ESTIMATE OF FINAL HORIZONTAL WHEEL VELOCITY
C         UN1(I,J)=X-VELOCITY AT TIME STEP N+1
C         UW=X-VELOCITY OF WHEEL AT TIME STEP N
C         V(I,J)=Y-VELOCITY OF CELL
C         VG=ESTIMATE OF FINAL VERTICAL WHEEL VELOCITY
C         VN1(I,J)=Y-VELOCITY AT TIME STEP N+1
C         VW=Y-VELOCITY OF WHEEL AT TIME STEP N
C         WM=MASS OF WHEEL PER UNIT WIDTH
C         WR=SQUARE OF DISTANCE FROM ANY POINT TO WHEEL CENTER
C         WX=X-COORDINATE OF WHEEL CENTER AT TIME STEP N
C         WY=Y-COORDINATE OF WHEEL CENTER AT TIME STEP N
C         X(I)=X-COORDINATE OF CELL CENTER
C         XK(K)=X-COORDINATE OF PARTICLE K

C         XO=INITIAL X-COORDINATE OF FIRST PARTICLE
C         XPL(I)=X-COORDINATE OF CELL RIGHT BOUNDARY
C         XR=X-COORDINATE OF RIGHT BOUNDARY
C         XSEPAR=COUNTER FOR NO. OF GETTING COLUMN OF PARTICLES
C         Y(J)=Y-COORDINATE OF CELL CENTER
C         YK(K)=Y-COORDINATE OF PARTICLE K
C         YO=INITIAL Y-COORDINATE OF FIRST PARTICLE
C         YPL(J)=Y-COORDINATE OF CELL LOWER BOUNDARY
C         YSEPAR=COUNTER FOR NO. OF GETTING ROW OF PARTICLES
C         YT=Y-COORDINATE OF LOWER BOUNDARY
C
      DIMENSION U(31,17),V(31,17),PHI(31,17),R(31,17),F(31,17),
     *FF(31,17),KC(31,17),ETA(31,17),EK(31,17),FC(31,17),
     *SIGMAX(31,17),SIGMAY(31,17),TAUXY(31,17),XK(1020),YK(1020),
     *X(31),Y(17),XPL(31),YPL(17),NAME(10),STRAN(56),STRES(56,1),
     *STRAIN(1),STRESS(1,1),DFLY(56,1),H(56),S2(56,1),S3(56,1),
     *TXK(1020),TYK(1020),P(31,17),DATA(39),TU(1020),TV(1020),DATB(30)
      EQUIVALENCE (XK,TU),(YK,TV)
      INTEGER F,FF,FC,FS
C
C         REGION 1)  GENERAL SETUP
C
 4000 CONTINUE
      READ (5,1) (NAME(I),I=1,8)
      IF(EOF,5)9999,4001
 4001 CONTINUE
      READ (5,2)IBAR,JBAR,NIS,NIF,NJS,NJF
      READ (5,3) ETAO,FKO,GX,GY,XR,YT,RW,WX,WY,WM,UW,VW,NX,NY,XO,YO,DXK,
     *DYK,DX,DY,DT,TCP,TPP,DTSP,ALPHAU,ALPHAV,CONV,GAMAF,REPS,IWHEEL,
     *IFSYM,FS,ITMAX,NI
      READ(5,10)NMPTS,NMCVS,MMAX,NS,NCVS,MS,IW,ITG,EPS
      READ(5,20) (STRAN(I),STRES(I) ,I=1,NS)
      READ(5,25) T,TL,NCS,NSP,NPP
    1 FORMAT(8A3)
 0002 FORMAT(6I5)
    3 FORMAT(5E12.4/6E12.4/2I12,4E12.4/6E12.4/5E12.4,4I4,F4.0)
 0004 FORMAT(1H1//1X,8A8,//,6X,2HI=,I12,6X,2HJ=,I12,3X,5HETAO=,E12.4,4X,
     *4HFKO=,E12.4,5X,3HGX=,E12.4,5X,3HGY=,E12.4,/,5X,3HXR=,E12.4,5X,
     *3HYT=,E12.4,5X,3HRW=,E12.4,5X,3HWX=,E12.4,5X,3HWY=,E12.4,5X,3HWM=,
     *E12.4,/,5X,3HUW=,E12.4,5X,3HVW=,E12.4,5X,3HNX=,I12,5X,3HNY=,I12,
     *5X,3HXO=,E12.4,5X,3HYO=,E12.4,/,4X,4HDXK=,E12.4,4X,4HDYK=,E12.4,
     *5X,3HDX=,E12.4,5X,3HDY=,E12.4,5X,3HDT=,E12.4,3X,5HDTCP=,E12.4,/,
     *3X,5HDTPP=,E12.4,3X,5HDTSP=,E12.4,1X,7HALPHAU=,E12.4,1X,7HALPHAV=,
```

```
     *E12.4,3X,5HCONV=,F12.4,2X,6HGAMAF=,F12.4,/,3X,5HRHOS=,E12.4,6X,
     *2HT=,E12.4,5X,3HTL=,E12.4,4X,4HNCS=,I12,4X,4HNSP=,I12,
     *4X,4HNPP=,I12,/,1X,7HIWHEEL=,I12,2X,6HISYM=,I12,5X,3HFS=,I12,
     *2X,6HIMAX=,I12,5X,3HWL=,E12.4,///)
   5 FORMAT(/////,5X,13H TIME STEP T=,E12.4,2X,31H TOTAL NUMBER OF
     *PARTICLE=,I12)
0006 FORMAT(//,2X,29HCELL HAS GONE THRU TWO STAGES,/,2X,
     *19HELEMENT AFFECTED IS,/ ,2X,2HI=,I2,2X,2HJ=,I2,2X,2HF=,I2,
     *2X,3HFF=,I2,2X,3HKC=,I2,/)
0007 FORMAT(2X,2HI=,I2,2X,2HJ=,I2,2X,7HR(I,J)=,E12.4,2X,7HD(I,J)=,
     *E12.4)
0008 FORMAT(///,2X,23HR(I,J) CHECK AT TIME T=,E12.4,/)
0009 FORMAT(/,2X,8HCYCLE T=,E12.4,2X,9HBEGAN AT ,F10.2,/)
0010 FORMAT(  4X,3H T=,F12.4,4X,7H    IC=,I12,4X,7H  CONV=,E12.4,4X,
     *7H  SUMR=,E12.4)
0011 FORMAT(  4X,3H T=,E12.4,4X,7H   SFX=,E12.4,4X,7H   SFY=,E12.4)
0012 FORMAT(//,2X,23HCOMPUTATION ENDED AT TIME T=,F10.2,/)
0013 FORMAT(  4X,3H T=,F12.4,4X,7HERRORU=,F12.8,4X,7HERRORV=,F12.8,
     *4X,7H   NOR=,I12)
0014 FORMAT(  2X,43HRE-CYCLE TO IMPROVE VELOCITY ESTIMATE AT T=,F12.4,
     */ ,2X,4HUWN=,E16.8,5X,4HVWN=,E16.8,6X,3HUG=,E16.8,6X,3HVG=,E16.8,
     */ ,2X,4HSFX=,E16.8,5X,4HSFY=,E16.8,2X,7HERRORU=,E16.8,2X,
     *7HERRORV=,E16.8,2X,3HIC=,I10)
0015 FORMAT(/,2X,35HHORIZONTAL WHEEL VELOCITY ZERO AT T=,E12.4,/,
     *2X,15HCALCULATED SFX=,E12.4)
0016 FORMAT(4X,23HADJUSTED TIME VARIABLES,//,4X,3HDT=,E12.4,2X,5HDTCP=,
     *E12.4,2X,5HDTPP=,E12.4,2X,5HDTSP=,E12.4,/)
0017 FORMAT(/,4X,26HDT IS TOO LARGE AT TIME T=,E12.4,/)
0018 FORMAT(  4X,3H T=,E12.4,4X,7H SDXSF=,E12.4,4X,7H SDYSF=,E12.4)
  19 FORMAT(3I5,E 5.6)
  20 FORMAT(2E12.4)
0021 FORMAT(1X,5HSFXG=,E16.8,4X,5HSFYG=,E16.8,/)
0022 FORMAT(  4X,3H T=,E12.4,4X,7H  DXSF=,E12.4,4X,7H  DYSF=,E12.4)
0023 FORMAT(//,3X,4HI   J,2X,4HX(I),1X,4HY(J),2X,6HU(I,J),6X,6HV(I,J),
     *6X,6HP(I,J),5X,8HPHI(I,J),6X,6HR(I,J),4X,11HSIGMAX(I,J),1X,
     *11HSIGMAY(I,J),1X,10HTAUXY(I,J),2X,10HF FE FC KC,/)
  24 FORMAT(1H ,2I4 ,9E12.4,I6)
0025 FORMAT(  4X,3H T=,E12.4,4X,7H   NSP=,I12)
0026 FORMAT(  4X,3H T=,E12.4,4X,7H   NPP=,I12)
0027 FORMAT(/,2X,5HSYM AT T=,E12.4,2X,7HF(I,J)=,I2,2X,8HKC(I,J)=,I2,
     *2X,8HF(IM,J)=,I2,2X,9HKC(IM,J)=,I2)
0028 FORMAT(/,2X,20HFREE-SURFACE CELL I=,I2,3H J=,I2,1X,29HHAS NO EMPTY
     * CELL BOUNDING IT)

0029 FORMAT(/)
0030 FORMAT(  4X,3H T=,E12.4,4X,7H    UW=,E12.4,4X,7H    VW=,E12.4,
     *4X,7HSTNMAX=,E12.4)
  31 FORMAT(1H ,2I4 ,9E12.4,E8.1)
0032 FORMAT(  4X,3H T=,E12.4,4X,7HXSFPAR=,E12.4,4X,7HYSFPAR=,E12.4,
     *4X,7H  KBAR=,I12)
0033 FORMAT(  4X,3H T=,E12.4,4X,7HALPHAU=,E12.4,4X,7HALPHAV=,E12.4,
     *4X,7HCONMAX=,E12.4)
  34 FORMAT(4X,3H T=,E12.4,4X,7H  DMAX=,E12.4,4X,7H IDMAX=I12,
     *4X,7H JDMAX=,I12)
  51 FORMAT(7H ERROPI)
0052 FORMAT(/,1X,7HTRACE B,/)
0053 FORMAT(/,1X,7HTRACE C,/)
0054 FORMAT(/,1X,7HTRACE A,/)
0056 FORMAT(2F10.0,3I5)
0057 FORMAT(/,1X,52HDATA TO BE READ IS NOT AT PROPER TIME,JOB TERMINATE
     *D,/)
0058 FORMAT(2X,2HI=,I3,2X,2HJ=,I3,2X,2HF=,I2,2X,3HFF=,I2)
0059 FORMAT(2X,2HI=,I3,2X,2HJ=,I3,2X,4HTXK=,E12.4,2X,4HTYK=,E12.4,2X,
     *2HF=,I2,2X,3HFE=,I2)
0060 FORMAT(//,2X,44HPARTICLE HAS MOVED INTO WHEEL,JOB TERMINATED,//,
     *2X,19HELEMENT AFFECTED IS,/)
4002 FORMAT(1H ,2I3,1X,2F5.1,8E12.4,1X,I2,1X,I2,1X,I2,1X,I2)
4027 FORMAT(2X,5HNPTS=,I5,2X,6HMNCVS=,I5,2X,5HMMAX=,I5,2X,3HNS=,I5,2X,
     *5HNCVS=,I5,2X,3HMS=,I5,2X,3HIW=,I5,2X,4HITG=,I5,2X,4HEPS=,E15.6,//
     *)
4028 FORMAT(/,1X,28HDATA POINTS FOR SPLINE CURVE,//,2X,3HNO.,2X,9HSTRAI
     *N(I),5X,9HSTRESS(I),/)
4029 FORMAT(1X,I3,2E12.4)
4030 FORMAT(  4X,3H T=,E12.4,4X,7H   NCS=,I12)
4031 FORMAT(/,1X,55HOUTPUT OF CELL VARIABLES ABOVE PRODUCED FOR TIME ST
     *EP T=,E12.4,/)
4032 FORMAT(1X,32HINPUT DATA FOR SPLINE SUBROUTINE,/)
C
C          INITIALIZE COUNTERS AND ESTABLISH SOME EQUATION CONSTANTS
C
      TCP=T+DTCP
      TPP=T+DTPP
      TSP=T+DTSP
      DXS=DX*DX
      DYS=DY*DY
      Z=2.0*(1.0/DXS+1.0/DYS)
      GYYI=ABS(GY*YI)
```

```
      GXXR=ABS(GX*XR)
      NOR=0
      STNMAX=0.
      UWO=UW
      VWO=VW
      IBAR1=IBAR-1
      IBAR2=IBAR-2
      JBAR1=JBAR-1
      JBAR2=JBAR-2
      IJBA22=IBAR2*JBAR2
      NIJ=IJBAR2/1020
      NNIJ=IJBAR2-NIJ*1020
      IJBAR=IBAR*JBAR
      NIJN=IJBAR/1020
      NNIJN=IJBAR-NIJN*1020
      NIJN1=NIJN+1
C
C          INITIALIZE VARIABLES TO START PROBLEM
C
  100 DO 101 J=1,JBAR
      DO 101 I=1,IBAR
      D(I,J)=0.
      SIGMAX(I,J)=0.
      SIGMAY(I,J)=0.
      TAUXY(I,J)=0.
      U(I,J)=0.
      V(I,J)=0.
      PHI(I,J)=0.
      R(I,J)=0.0
      FK(I,J)=0.
      F(I,J)=-
      FF(I,J)=0
      FC(I,J)=0
      KC(I,J)=0
      ETA(I,J)=0.
  101 CONTINUE
C
C          SET-UP BND CELLS
C
      IF(NCS.NE.0) GO TO 161
      DO 102 I=2,IBAR1
      F(I,1)=1
 0102 F(I,JBAR)=1

      DO 103 J=2,JBAR1
      IF(IFSYM.NE.1) GO TO 116
      F(1,J)=6
      GO TO 103
 0116 F(1,J)=1
 0103 F(IBAR,J)=1
 0161 CONTINUE
C
C       SETUP COORDINATE OF EACH CELL
C
  105 XTE=0.
      DO 106 I=1,IBAR
      X(I)=XTE-0.5*DX
      XPL(I)=XTE
      XTE=XTE+DX
  106 CONTINUE
      YTE=0.0
      DO 107 J=1,JBAR
      Y(J)=YTE-0.5*DY
      YPL(J)=YTE
      YTE=YTE+DY
  107 CONTINUE
C
C          SETUP FULL CELL AND COORDINATES OF EACH PARTICLE
C
      IF(NCS.NE.0) GO TO 162
  110 K=1
      N=0
      SFX=0.
      SFY=0.
      SFXG=0.
      SFYG=0.
      SDXSF=0.
      SDYSF=0.
      XSFPAR=0.
      YSFPAR=0.
      ERRCPU=0.
      ERRCPV=0.
      DXSF=0.
      DYSF=0.
      CONMAX=0.
      IC=0
      SUMP=0.
```

```
      KBAR=NX*NY
      MPAR=KBAR/1020
      MPRR=KBAR-MPAR*1020
      MPAR1=MPAR+1
      XP=X0
      DO 111 ICX=1,NX
      YP=Y0
      DO 112 ICY=1,NY
      N=N+1
      TXK(N)=XP
      TYK(N)=YP
      I=TXK(N)/DX+2.0
      J=TYK(N)/DY+2.0
      F(I,J)=2
      ETA(I,J)=ETA0
      KC(I,J)=KC(I,J)+1
      IF(N.LT.1020) GO TO 115
      WRITE(14) (TXK(N),TYK(N),N=1,1020)
      N=0
  115 K=K+1
      YP=YP+DYK
  112 CONTINUE
      XP=XP+DXK
  111 CONTINUE
      KBAR=K-1
      WRITE(14) (TXK(N),TYK(N),N=1,MPRR)
      REWIND 14
 0162 CONTINUE
C
C          SET UP SNOW AND OR ADJACENT TO SNOW CELLS
C
      IF(NCS.NE.2) GO TO 113
C
C          AUXILLARY GEOMETRY SET-UP
C
      IF(IWHEEL.EQ.1) GO TO 132
      IF(IFSYM.NE.1) GO TO 134
      DO 135 I=2,6
      DO 135 J=3,41
      F(I,J)=5
      U(I,J)=UW
 0135 V(I,J)=VW
      DO 136 J=3,41

  136 FE(7,J)=2
      DO 137 I=2,6
      FE(I,2)=2
  137 FE(I,42)=2
      FE(6,3)=1
      FE(6,41)=1
      GO TO 113
 0134 CONTINUE
      DO 133 I=10,16
      DO 133 J=3,6
      F(I,J)=5
      U(I,J)=UW
      U(I-1,J)=UW
      V(I,J)=VW
 0133 CONTINUE
      DO 128 J=3,6
      FE(9,J)=2
      FE(17,J)=2
 0128 CONTINUE
      FE(10,3)=1
      FE(10,6)=1
      FE(16,3)=1
      FE(16,6)=1
      DO 129 I=10,16
      FE(I,2)=2
      FE(I,7)=2
 0129 CONTINUE
      GO TO 113
 0132 CONTINUE
C
C          WHEEL GEOMETRY SET-UP
C
      M=WX/DX+2.0
      MM=(WX+RW)/DX+2.0
      N=WY/DY+2.0
      NN=(WY+RW*COS(3.1416/4.0))/DY+2.0
      IM=M
      DO 126 J=N,NN
      DO 121 I=M,MM
      IM=IM-1
      WR=(X(I)-WX)**2+(Y(J)-WY)**2
      IF(WR.GT.(RW**2)) GO TO 121
      F(I,J)=5
```

```
      U(I,J)=UW
      V(I,J)=VW
      IF(IFSYM.EQ.1) GO TO 138
      F(IM,J)=5
      U(IM-1,J)=UW
      U(IM,J)=UW
      V(IM,J)=VW
0138  CONTINUE
      IF((WR+2.*DX*(X(I)-WX)+DX**2).LE.(RW**2)) GO TO 121
      IF((WR+2.*DX*(X(I)-WX)+DX**2-2.*DY*(Y(J)-WY)+DY**2).LE.(RW**2))
     *GO TO 122
      FE(I+1,J)=2
      IF(IFSYM.EQ.1) GO TO 121
      FE(IM-1,J)=2
      GO TO 121
0122  FE(I+1,J)=2
      FE(I+1,J-1)=1
      IF(IFSYM.EQ.1) GO TO 121
      FE(IM-1,J)=2
      FE(IM-1,J-1)=1
 121  CONTINUE
      IM=M
 126  CONTINUE
      MM=(WX+RW*COS(3.1416/4.0))/DX+3.0                                 0
      N=NN-1
      NN=(WY+RW)/DY+2.0
      IM=IM-1
      DO 127 I=M,MM
      DO 124 J=N,NN
      WR=(X(I)-WX)**2+(Y(J)-WY)**2
      IF(WR.GT.(RW**2)) GO TO 124
      F(I,J)=5
      U(I,J)=UW
      V(I,J)=VW
      IF(IFSYM.EQ.1) GO TO 139
      F(IM,J)=5
      U(IM-1,J)=UW
      U(IM,J)=UW
      V(IM,J)=VW
0139  CONTINUE
      IF((WR+2.0*DY*(Y(J)-WY)+DY**2).LE.(RW**2)) GO TO 124
      IF((WR-2.0*DX*(X(I)-WX)+DX**2+2.0*DY*(Y(J)-WY)+DY**2).LE.(RW**2))
     *GO TO 125

      FE(I,J+1)=2
      IF(IFSYM.EQ.1) GO TO 124
      FE(IM,J+1)=2
      GO TO 124
0125  FE(I,J+1)=2
      FE(I-1,J+1)=1
      IF(IFSYM.EQ.1) GO TO 124
      FE(IM,J+1)=2
      FE(IM+1,J+1)=1
 124  CONTINUE
      IM=IM-1
 127  CONTINUE
0113  CONTINUE
C
C          SET-UP OB ADJACENT TO BND CELLS
C
      IF(NCS.NE.0) GO TO 130
      DO 141 I=2,IBAR1
      IF(F(I,2).EQ.5) GO TO 143
      FE(I,2)=2
0143  IF(F(I,JBAR1).EQ.5) GO TO 141
      FE(I,JBAR1)=2
0141  CONTINUE
      DO 142 J=2,JBAR1
      IF(F(2,J).EQ.5) GO TO 144
      FE(2,J)=2
0144  IF(F(IBAR1,J).EQ.5) GO TO 142
      FE(IBAR1,J)=2
0142  CONTINUE
0130  CONTINUE
C
C          SET UP SUR CELL AT INITIAL CONDITIONS
C
      IF(NCS.NE.0) GO TO 163
      J=FS
      DO 131 I=2,IBAR1
      IF(FE(I,J).EQ.2) GO TO 131
      F(I,J)=3
 131  CONTINUE
      IF(IFSYM.EQ.1) GO TO 140
      F(2,J)=3
      U(2,J)=0.0
      V(2,J)=0.0
```

```
 0140 CONTINUE
      F(IBAR1,J)=3
      U(IBAR1,J)=0.0
      V(IBAR1,J)=0.0
 0163 CONTINUE
C
C          ESTABLISH INITIAL HYDROSTATIC PRESSURE
C
      IF(NCS.NE.0) GO TO 114
      N=FS+1
      DO 109 J=N,JBAR1
      DO 109 I=2,IBAR1
      PHI(I,J)=(Y(J)-(WY+RW+.5*DY))*GY
 0109 CONTINUE
 0114 CONTINUE
C
C     READ U(IJ),V(IJ) FROM TAPE( 8)AND STORE THESE ON MAIN STORAGE
C
      IF(NCS.EQ.0) GO TO 157
 0104 READ(8)DATA,U,V
      IF(EOF,8)104,108
 0108 CONTINUE
      TDS=DATA(1)
      KBAR=DATA(2)
      SDXSF=DATA(3)
      SDYSF=DATA(4)
      XSFPAR=DATA(5)
      YSFPAR=DATA(6)
      SFX=DATA(7)
      SFY=DATA(8)
      VW=DATA(9)
      UW=DATA(10)
      ALPHAU=DATA(11)
      ALPHAV=DATA(12)
      ODTCP=DATA(13)
      ODTSP=DATA(14)
      ODTPP=DATA(15)
      DXSF=DATA(25)
      DYSF=DATA(26)
      ERRORU=DATA(27)
      ERRORV=DATA(28)
      IC=DATA(29)
      SUMR=DATA(30)

      SFXG=DATA(31)
      SFYG=DATA(32)
      IF(ABS(TDS-T).LT.(ODTCP/2.)) GO TO 904
      WRITE(6,57)
      GO TO 901
 0904 READ(8)
      IF(EOF,8)155,904
C
C     READ FINAL SET DATA OF PHI(IJ),F(IJ),FE(IJ),AND ETA(IJ) FROM
C     TAPE(13) AND STORE THESE ON MAIN STORAGE
C
 0155 READ(13)DATA,PHI,ETA,F,FE
      IF(EOF,13)155,905
 0905 TDS=DATA(1)
      ODTCP=DATA(13)
      IF(ABS(TDS-T).LT.(ODTCP/2.)) GO TO 903
      WRITE(6,57)
      GO TO 901
 0903 READ(13)
      IF(EOF,13)156,903
C
C          ESTABLISH COUNTERS FOR PARTICLE COORD. CALCULATIONS
C
 0156 READ(14)DATB
      IF(EOF,14)156,158
 0158 CONTINUE
      TDS=DATB(1)
      KBAR=DATB(2)
      ODTPP=DATB(14)
      IF(ABS(TDS-T).LT.(ODTPP/2.)) GO TO 902
      WRITE(6,57)
      GO TO 901
 0902 MPAR=KBAR/1020
      MPRP=KBAR-MPAR*1020
      MPAR1=MPAR+1
C
C          PRINT INPUT DATA FOR VERIFICATION
C
 0157 WRITE (6,4) (NAME(I),I=1,8),IBAR,JBAR,ETAO,FKO,GX,GY,XR,YT,RW,WX,
     *WY,WM,UW,VW,NX,NY,XO,YO,DXK,DYK,DX,DY,DT,DTCP,DTPP,DTSP,ALPHAU,
     *ALPHAV,CONV,GAMAE,RHOS,T,TL,NCS,NSP,NPP,IWHEEL,IFSYM,FS,ITMAX,WI
      WRITE(6,4032)
      WRITE(6,4027)MNPTS,MNCVS,MPAX,NS,NCVS,MS,IN,ITG,EPS
```

```
      WRITE(6,4028)
      WRITE(6,4029)(I,STRAN(I),STRES(I),I=1,NS)
C
C        STORE INITIAL VALUES ON TAPES
C
      IF(NCS.NE.0) GO TO 171
      WRITE(6,29)
      ASSIGN 170 TO KRET
      TPP=0.
      GO TO 836
 0170 ASSIGN 171 TO KRET
      TCP=0.
      GO TO 802
 0171 CONTINUE
C
C        REGION 30   R(IJ) CALCULATION
C
  300 T=T+DT
      CALL SECOND(BTIME)
      WRITE(6,9)T,BTIME
C
C        CALCULATE D FOR COMPUTING THE SOURCE TERM
C
      DO4034 J=2,JBAR1
      DO4034 I=2,IBAR1
      D(I,J)=0.
      IF(F(I,J).NE.2.AND.F(I,J).NE.3) GO TO 4034
      D(I,J)=((U(I,J)-U(I-1,J))/DX)+((V(I,J)-V(I,J-1))/DY)
 4034 CONTINUE
C
C        CALCULATION OF THE SOURCE TERM
C
      SUMR=0.
      DO 301 J=2,JBAR1
      DO 301 I=2,IBAR1
      R(I,J)=0.
      IF(F(I,J).NE.2) GO TO 301
      US=(U(I,J)+U(I-1,J))**2
      VS=(V(I,J)+V(I,J-1))**2
      UVBR=(U(I,J+1)+U(I,J))*(V(I+1,J)+V(I,J))
      UVTL=(U(I-1,J)+U(I-1,J-1))*(V(I,J-1)+V(I-1,J-1))
      UVTR=(U(I,J)+U(I,J-1))*(V(I+1,J-1)+V(I,J-1))
      UVBL=(U(I-1,J+1)+U(I-1,J))*(V(I,J)+V(I-1,J))

      DLEFT=D(I-1,J)
      DRIGHT=D(I+1,J)
      DABOVE=D(I,J-1)
      DBELOW=D(I,J+1)
      IF(FE(I,J).NE.2) GO TO 319
      IF(F(I-1,J).EQ.1.OR.F(I-1,J).EQ.5.OR.F(I-1,J).EQ.6) GO TO 302
      ULS=(U(I-1,J)+U(I-2,J))**2
      GO TO 303
 0302 ULS=US
      DLEFT=D(I,J)
      IF(F(I-1,J).EQ.5) GO TO 304
      UVBL=0.
      UVTL=0.
      GO TO 303
 0304 UVBL=UW*VW*4.
      UVTL=UW*VW*4.
      IF(FE(I-1,J).NE.1) GO TO 303
      IF(F(I-1,J-1).NE.5) GO TO 305
      DBELOW=DBELOW+(U(I-1,J+1)-UW)/DX
      GO TO 303
 0305 DABOVE=DABOVE+(U(I-1,J-1)-UW)/DX
 0303 IF(F(I+1,J).EQ.1.OR.F(I+1,J).EQ.5) GO TO 306
      URS=(U(I+1,J)+U(I,J))**2
      GO TO 307
 0306 URS=US
      DRIGHT=D(I,J)
      IF(F(I+1,J).EQ.5) GO TO 308
      UVBR=0.
      UVTR=0.
      GO TO 307
 0308 UVBR=UW*VW*4.
      UVTR=UW*VW*4.
      IF(FE(I+1,J).NE.1) GO TO 307
      IF(F(I+1,J-1).NE.5) GO TO 309
      DBELOW=DBELOW+(UW-U(I,J+1))/DX
      GO TO 307
 0309 DABOVE=DABOVE+(UW-U(I,J-1))/DX
 0307 IF(F(I,J-1).EQ.1.OR.F(I,J-1).EQ.5) GO TO 310
      VAS=(V(I,J-1)+V(I,J-2))**2
      GO TO 311
 0310 VAS=VS
      DABOVE=D(I,J)
      IF(F(I,J-1).EQ.5) GO TO 312
```

```
      UVTL=0.
      UVTR=0.
      GO TO 311
 0312 UVTL=UW*VW*4.
      UVTR=UW*VW*4.
      IF(FF(I,J-1).NE.1) GO TO 311
      IF(F(I-1,J-1).NE.5) GO TO 313
      DRIGHT=DRIGHT+(V(I+1,J-1)-VW)/DY
      GO TO 311
 0313 DLEFT=DLEFT+(V(I-1,J-1)-VW)/DY
 0311 IF(F(I,J+1).EQ.1.OR.F(I,J+1).EQ.5) GO TO 314
      VBS=(V(I,J+1)+V(I,J))**2
      GO TO 315
 0314 VBS=VS
      DBELOW=D(I,J)
      IF(F(I,J+1).EQ.5) GO TO 316
      UVBL=0.
      UVBR=0.
      GO TO 315
 0316 UVBL=UW*VW*4.
      UVBR=UW*VW*4.
      IF(FF(I,J+1).NE.1) GO TO 315
      IF(F(I-1,J+1).NE.5) GO TO 318
      DRIGHT=DRIGHT+(VW-V(I+1,J))/DY
      GO TO 315
 0318 DLEFT=DLEFT+(VW-V(I-1,J))/DY
      GO TO 315
 0319 IF(FF(I-1,J-1).NE.1) GO TO 320
      UVTL=UW*VW*4.
      DABOVE=DABOVE+(U(I-1,J-1)+U(I-1,J)-(2.*UW))/DX
      DLEFT=DLEFT+(V(I-1,J-1)+V(I,J-1)-(2.*VW))/DY
      GO TO 323
 0320 IF(FF(I-1,J+1).NE.1) GO TO 321
      UVBL=UW*VW*4.
      DLEFT=DLEFT+((2.*VW)-V(I,J)-V(I-1,J))/DY
      DBELOW=DBELOW+(U(I-1,J+1)+U(I-1,J)-(2.*UW))/DX
      GO TO 323
 0321 IF(FF(I+1,J-1).NE.1) GO TO 322
      UVTR=UW*VW*4.
      DABOVE=DABOVE+((2.*UW)-U(I,J)-U(I,J-1))/DX
      DRIGHT=DRIGHT+(V(I+1,J-1)+V(I,J-1)-(2.*VW))/DY
      GO TO 323
 0322 IF(FF(I+1,J+1).NE.1) GO TO 323

      UVBR=UW*VW*4.
      DRIGHT=DRIGHT+((2.*VW)-V(I,J)-V(I+1,J))/DY
      DBELOW=DBELOW+((2.*UW)-U(I,J)-U(I,J+1))/DX
 0323 URS=(U(I+1,J)+U(I,J))**2
      ULS=(U(I-1,J)+U(I-2,J))**2
      VBS=(V(I,J+1)+V(I,J))**2
      VAS=(V(I,J-1)+V(I,J-2))**2
 0315 R(I,J)=(URS+ULS-(2.*US))/(4.*DXS)
     *+(VBS+VAS-(2.*VS))/(4.*DYS)
     *+(UVBR+UVTL-UVTR-UVBL)/(2.*DX*DY)-D(I,J)/DT
     *-ETA(I,J)*(DRIGHT+DLEFT-(2.*D(I,J)))/DXS
     *-ETA(I,J)*(DBELOW+DABOVE-(2.*D(I,J)))/DYS
      SUMR=SUMR+ABS(R(I,J))
 0301 CONTINUE
C
C        REGION 40-A    FREE-SURFACE PRESSURE CORRECTION
C
      DO 451 J=2,JBAR1
      DO 451 I=2,IBAR1
      IF(F(I,J).NE.3) GO TO 451
      PHI(I,J)=0.
      IF(F(I,J-1).NE.4) GO TO 452
      IF(F(I-1,J).NE.4) GO TO 453
      IF(F(I+1,J).EQ.4.OR.F(I,J+1).EQ.4) GO TO 451
      PHI(I,J)=0.5*ETA(I,J)*((U(I,J+1)+U(I-1,J+1)-U(I,J)-U(I-1,J))/DY
     *+(V(I+1,J)+V(I+1,J-1)-V(I,J)-V(I,J-1))/DX)
      GO TO 451
 0453 IF(F(I+1,J).NE.4) GO TO 454
      IF(F(I,J+1).EQ.4) GO TO 451
      PHI(I,J)=-.5*ETA(I,J)*((U(I,J+1)+U(I-1,J+1)-U(I,J)-U(I-1,J))/DY
     *+(V(I,J)+V(I,J-1)-V(I-1,J)-V(I-1,J-1))/DX)
      GO TO 451
 0454 IF(F(I,J+1).EQ.4) GO TO 451
      PHI(I,J)=2.0*ETA(I,J)*(V(I,J)-V(I,J-1))/DY
      GO TO 451
 0452 IF(F(I+1,J).NE.4) GO TO 455
      IF(F(I,J+1).NE.4) GO TO 456
      IF(F(I-1,J).EQ.4) GO TO 451
      PHI(I,J)=0.5*ETA(I,J)*((U(I,J)+U(I-1,J)-U(I,J-1)-U(I-1,J-1))/DY
     *+(V(I,J)+V(I,J-1)-V(I-1,J)-V(I-1,J-1))/DX)
      GO TO 451
 0456 IF(F(I-1,J).EQ.4) GO TO 451
      PHI(I,J)=2.0*ETA(I,J)*(U(I,J)-U(I-1,J))/DX
```

```
      GO TO 451
0455  IF(F(I,J+1).NE.4) GO TO 457
      IF(F(I-1,J).L ).4) GO TO 458
      PHI(I,J)=2.0*ETA(I,J)*(V(I,J)-V(I,J-1))/DY
      GO TO 451
0458  PHI(I,J)=-.5*ETA(I,J)*((U(I,J)+U(I-1,J)-U(I,J-1)-U(I-1,J-1))/DY
     *+(V(I+1,J)+V(I+1,J-1)-V(I,J)-V(I,J-1))/DX)
      GO TO 451
0457  IF(F(I-1,J).NE.4) GO TO 451
      PHI(I,J)=2.0*ETA(I,J)*(U(I,J)-U(I-1,J))/DX
0451  CONTINUE
C
C        STORE VARIABLES FOR RE-CYCLING
C
      WRITE(16)U,V,ETA,PHI,R
      GO TO 4038
C
C        RE-ESTABLISH VARIABLES FOR RE-CYCLE
C
4036  READ(16)U,V,ETA,PHI,R
      IF(EOF,16)4036,4038
4038  REWIND 16
C
C        CALCULATE ESTIMATE OF FINAL WHEEL VELOCITY
C
      UG=UW+(GX+(SFXG/WM))*DT
      VG=VW+(GY+(SFYG/WM))*DT
      FX=(((UG+UW)/2.)*DT)/DX
      FY=(((VG+VW)/2.)*DT)/DY
C
C        CELL VARIABLES AFTER SHIFTING COODINATES
C
      M=1
      N=1
      IF(FX.GE.0.0) GO TO 417
      M=-1
0417  IF(FY.GE.0.0) GO TO 418
      N=-1
0418  FX=ABS(FX)
      FY=ABS(FY)
      DO 420 J=2,JBAR1
      DO 420 I=2,IBAR1
      IF(F(I,J).NE.2.AND.F(I,J).NE.3) GO TO 420

      IF(FE(I,J).NE.2) GO TO 421
      IF(F(I+1,J).NE.1) GO TO 422
      PHI(I+1,J)=PHI(I,J)+GX*DX+2.*ETA(I,J)*U(I-1,J)/DX
      GO TO 423
0422  IF(F(I+1,J).NE.5) GO TO 423
      PHI(I+1,J)=PHI(I,J)-(SFYG/WM)*DX+2.*ETA(I,J)*(U(I-1,J)-UW)/DX
0423  IF(F(I-1,J).NE.1) GO TO 424
      PHI(I-1,J)=PHI(I,J)-GX*DX-2.*ETA(I,J)*U(I,J)/DX
      GO TO 425
0424  IF(F(I-1,J).NE.5) GO TO 429
      PHI(I-1,J)=PHI(I,J)+(SFXG/WM)*DX+2.*ETA(I,J)*(UW-U(I,J))/DX
      GO TO 425
0429  IF(F(I-1,J).NE.6) GO TO 425
      PHI(I-1,J)=PHI(I,J)-GX*DX
0425  IF(F(I,J+1).NE.1) GO TO 426
      PHI(I,J+1)=PHI(I,J)+GY*DY+2.*ETA(I,J)*V(I,J-1)/DY
      GO TO 427
0426  IF(F(I,J+1).NE.5) GO TO 427
      PHI(I,J+1)=PHI(I,J)-(SFYG/WM)*DY+2.*ETA(I,J)*(V(I,J-1)-VW)/DY
0427  IF(F(I,J-1).NE.1) GO TO 428
      PHI(I,J-1)=PHI(I,J)-GY*DY-2.*ETA(I,J)*V(I,J)/DY
      GO TO 421
0428  IF(F(I,J-1).NE.5) GO TO 421
      PHI(I,J-1)=PHI(I,J)+(SFYG/WM)*DY+2.*ETA(I,J)*(VW-V(I,J))/DY
0421  PHI(I,J)=(1.-FX)*(1.-FY)*PHI(I,J)+FX*(1.-FY/2.)*PHI(I+M,J)
     *+FY*(1.-FX/2.)*PHI(I,J+N)
0420  CONTINUE
C
C        REGION 40  PRESSURE DENSITY CALCULATIONS
C
 400  IC=0
 401  CONMAX=0.0
      DO 402 ITER=1,10
      DO 403 J=2,JBAR1
      DO 403 I=2,IBAR1
      IF(F(I,J).NE.2) GO TO 403
      IF(FF(I,J).NE.2) GO TO 405
      IF(F(I+1,J).NE.1) GO TO 407
      PHI(I+1,J)=PHI(I,J)+GX*DX+2.*ETA(I,J)*U(I-1,J)/DX
      GO TO 406
0407  IF(F(I+1,J).NE.5) GO TO 406
      PHI(I+1,J)=PHI(I,J)-(SFXG/WM)*DX+2.*ETA(I,J)*(U(I-1,J)-UW)/DX
0406  IF(F(I-1,J).NE.1) GO TO 409
```

```
      PHI(I-1,J)=PHI(I,J)-GX*DX-2.*ETA(I,J)*U(I,J)/DX
      GO TO 408
 0409 IF(F(I-1,J).NE.5) GO TO 415
      PHI(I-1,J)=PHI(I,J)+(SFXG/WM)*DX+2.*ETA(I,J)*(UW-U(I,J))/DX
      GO TO 408
 0415 IF(F(I-1,J).NE.6) GO TO 408
      PHI(I-1,J)=PHI(I,J)-GX*DX
 0408 IF(F(I,J+1).NE.1) GO TO 411
      PHI(I,J+1)=PHI(I,J)+GY*DY+2.*ETA(I,J)*V(I,J-1)/DY
      GO TO 410
 0411 IF(F(I,J+1).NE.5) GO TO 410
      PHI(I,J+1)=PHI(I,J)-(SFYG/WM)*DY+2.*ETA(I,J)*(V(I,J-1)-VW)/DY
 0410 IF(F(I,J-1).NE.1) GO TO 413
      PHI(I,J-1)=PHI(I,J)-GY*DY-2.*ETA(I,J)*V(I,J)/DY
      GO TO 405
 0413 IF(F(I,J-1).NE.5) GO TO 405
      PHI(I,J-1)=PHI(I,J)+(SFYG/WM)*DY+2.*ETA(I,J)*(VW-V(I,J))/DY
  405 PHINEW=(PHI(I+1,J)+PHI(I-1,J))/(Z*DXS)
     *+(PHI(I,J+1)+PHI(I,J-1))/(Z*DYS)+R(I,J)/Z
      IF(ITEN.NE.10) GO TO 412
      CONIJ=ABS(PHINEW-PHI(I,J))/(ABS(PHINEW)+ABS(PHI(I,J))
     *+0.25*(U(I,J)+U(I-1,J))**2+0.25*(V(I,J)+V(I,J-1))**2+GYYT+GXXR)
      IF(CONIJ.LT.CONMAX) GO TO 412
      CONMAX=CONIJ
  412 PHI(I,J)=PHINEW
  403 CONTINUE
  402 IC=IC+1
      IF(NOR.LE.2) GO TO 414
      IF(IC.EQ.NIC)500,401
 0414 IF(CONMAX.GT.CONV) GO TO 401
C
C        REGION 50   VELOCITY CALCULATION
C
 0500 N=0
      ICOUNT=0
      DO 511 J=2,JBAR1
      DO 511 I=2,IBAR1
      N=N+1
      IF(F(I,J).NE.5) GO TO 503
      TU(N)=UG
      TV(N)=VG
      PHI(I,J)=0.
      GO TO 532

 0503 CONTINUE
      IF(F(I,J).NE.2.AND.F(I,J).NE.3) GO TO 529
      UABOVE=U(I,J-1)
      UBELOW=U(I,J+1)
      VT=V(I,J-1)
      VB=V(I,J)
      VTR=V(I+1,J-1)
      VBR=V(I+1,J)
      IF(F(I+1,J).EQ.4) GO TO 512
      IF(F(I+1,J).EQ.1) GO TO 513
      IF(F(I+1,J).EQ.5) GO TO 501
      IF(F(I+1,J+1).EQ.1.OR.F(I,J+1).EQ.1) GO TO 514
      IF(F(I+1,J+1).EQ.5.OR.F(I,J+1).EQ.5) GO TO 5140
      IF(F(I+1,J+1).NE.4.OR.F(I,J+1).NE.4) GO TO 516
      UBELOW=U(I,J)
      GO TO 516
  512 TU(N)=U(I,J)+GX*DT
      GO TO 517
 0513 TU(N)=0.
      GO TO 517
 0501 TU(N)=UG
      GO TO 517
  514 UBELOW=-U(I,J)
      VB=0.
      VBR=0.
      GO TO 516
 5140 UBELOW=2.*UW-U(I,J)
      VB=VW
      VBR=VW
 0516 IF(F(I+1,J-1).EQ.1.OR.F(I,J-1).EQ.1) GO TO 518
      IF(F(I+1,J-1).EQ.5.OR.F(I,J-1).EQ.5) GO TO 5180
      IF(F(I+1,J-1).NE.4.OR.F(I,J-1).NE.4) GO TO 520
      UABOVE=U(I,J)
      GO TO 520
  518 UABOVE=-U(I,J)
      VT=0.
      VTR=0.
      GO TO 520
 5180 UABOVE=2.*UW-U(I,J)
      VT=VW
      VTR=VW
  520 SUM1=ETA(I,J)*((U(I+1,J)+U(I-1,J)-2.0*U(I,J))/DXS+
     *(UBELOW+UABOVE-2.0*U(I,J))/DYS)+(PHI(I,J)-PHI(I+1,J))/DX+GX
```

```
      SUM2=(UABOVE+U(I,J))*(VTR+VT)/4.-(UBELOW+U(I,J))*(VBR+VB)/4.
  527 TU(N)=U(I,J)+DT*(SUM1+SUM2/DY+0.25*(U(I,J)+U(I-1,J))**2/DX
     *-0.25*(U(I+1,J)+U(I,J))**2/DX)
  517 VLEFT=V(I-1,J)
      VRIGHT=V(I+1,J)
      UR=U(I,J)
      UBR=U(I,J+1)
      UL=U(I-1,J)
      UBL=U(I-1,J+1)
      IF(F(I,J+1).EQ.4) GO TO 521
      IF(F(I,J+1).EQ.1) GO TO 522
      IF(F(I,J+1).EQ.5) GO TO 502
      IF(F(I+1,J+1).EQ.1.OR.F(I+1,J).EQ.1) GO TO 523
      IF(F(I+1,J+1).EQ.5.OR.F(I+1,J).EQ.5) GO TO 5230
      IF(F(I+1,J+1).NE.4.OR.F(I+1,J).NE.4) GO TO 524
      VRIGHT=V(I,J)
      GO TO 524
 5230 VRIGHT=2.*VW-V(I,J)
      UR=UW
      UBR=UW
      GO TO 524
  521 TV(N)=V(I,J)+GY*DT
      GO TO 532
 0522 TV(N)=0.
      GO TO 532
 0502 TV(N)=VG
      GO TO 532
  523 VRIGHT=-V(I,J)
      UR=0.
      UBR=0.
 0524 IF(F(I-1,J+1).EQ.1.OR.F(I-1,J).EQ.1) GO TO 525
      IF(F(I-1,J+1).EQ.5.OR.F(I-1,J).EQ.5) GO TO 5250
      IF(F(I-1,J).EQ.5) GO TO 507
      IF(F(I-1,J+1).NE.4.OR.F(I-1,J).NE.4) GO TO 526
      VLEFT=V(I,J)
      GO TO 526
 5250 VLEFT=2.*VW-V(I,J)
      UL=UW
      UBL=UW
      GO TO 526
  525 VLEFT=-V(I,J)
      UL=0.
      UBL=0.

      GO TO 526
 0507 VLEFT=V(I,J)
      UL=0.
      UBL=0.
  526 SUM1=ETA(I,J)*((V(I,J+1)+V(I,J-1)-2.0*V(I,J))/DYS
     *+(VRIGHT+VLEFT-2.0*V(I,J))/DXS)+(PHI(I,J)-PHI(I,J+1))/DY+GY
      SUM2=(UL+UBL)*(V(I,J)+VLEFT)/4.-(UR+UBR)*(V(I,J)+VRIGHT)/4.
  528 TV(N)=V(I,J)+DT*(SUM1+SUM2/DX+0.25*(V(I,J)+V(I,J-1))**2/DY
     *-0.25*(V(I,J)+V(I,J+1))**2/DY)
      GO TO 532
 0529 IF(F(I+1,J).NE.5) GO TO 504
      TU(N)=UG
      GO TO 505
 0504 TU(N)=U(I,J)
 0505 IF(F(I,J+1).NE.5) GO TO 506
      TV(N)=VG
      GO TO 532
 0506 TV(N)=V(I,J)
  532 IF(N.LT.1020)GO TO 511
      WRITE(11) (TU(N),TV(N),N=1,1020)
      ICOUNT=ICOUNT+1
      N=0
  511 CONTINUE
      WRITE(11) (TU(N),TV(N),N=1,NNIJ)
      REWIND 11
      GO TO 540
  540 N=0
      M=0
      DO 541 J=2,JBAR1
      DO 541 I=2,IBAR1
      N=N+1
      IF(N.EQ.1) GO TO 542
  544 U(I,J)=TU(N)
      V(I,J)=TV(N)
      GO TO 599
  542 IF(M.EQ.ICOUNT) GO TO 543
      READ(11) (TU(N),TV(N),N=1,1020)
      M=M+1
      GO TO 544
  543 READ(11) (TU(N),TV(N),N=1,NNIJ)
      GO TO 544
  599 IF(N.LT.1020) GO TO 541
      N=0
```

```
  541 CONTINUE                                                        0
      REWIND 11
C
C         FREE-SURFACE VELOCITY CORRECTIONS
C
      ASSIGN 599 TO KR-T
 0545 DO 546 J=2,JBAR1
      DO 546 I=2,IBAR1
      IF(F(I,J).NE.3) GO TO 546
      IF(F(I,J-1).NE.4) GO TO 547
      IF(F(I-1,J).NE.4) GO TO 548
      IF(F(I+1,J).EQ.4) GO TO 549
      IF(F(I,J+1).EQ.4) GO TO 550
      U(I-1,J)=U(I,J)
      V(I,J-1)=V(I,J)
      GO TO 546
 0548 IF(F(I+1,J).NE.4) GO TO 551
      IF(F(I,J+1).EQ.4) GO TO 553
      U(I,J)=U(I-1,J)
      V(I,J-1)=V(I,J)
      GO TO 546
 0551 IF(F(I,J+1).EQ.4) GO TO 552
      V(I,J-1)=V(I,J)+DY*(U(I,J)-U(I-1,J))/DX
      GO TO 546
 0552 V(I,J)=.5*DY*(U(I-1,J)-U(I,J))/DX
      V(I,J-1)=-V(I,J)
      GO TO 546
 0553 U(I,J)=U(I-1,J)
      V(I,J)=(V(I,J)+V(I,J-1))/2.
      V(I,J-1)=V(I,J)
      GO TO 546
 0549 IF(F(I,J+1).EQ.4) GO TO 554
      V(I,J-1)=V(I,J)
      U(I,J)=(U(I,J)+U(I-1,J))/2.
      U(I-1,J)=U(I,J)
      GO TO 546
 0554 U(I,J)=(U(I,J)+U(I-1,J))/2.
      U(I-1,J)=U(I,J)
      V(I,J)=(V(I,J)+V(I,J-1))/2.
      V(I,J-1)=V(I,J)
      GO TO 546
 0550 U(I-1,J)=U(I,J)
      V(I,J)=(V(I,J)+V(I,J-1))/2.

      V(I,J-1)=V(I,J)
      GO TO 546
 0547 IF(F(I+1,J).NE.4) GO TO 555
      IF(F(I,J+1).NE.4) GO TO 556
      IF(F(I-1,J).EQ.4) GO TO 557
      U(I,J)=U(I-1,J)
      V(I,J)=V(I,J-1)
      GO TO 546
 0557 V(I,J)=V(I,J-1)
      U(I,J)=(U(I,J)+U(I-1,J))/2.
      U(I-1,J)=U(I,J)
      GO TO 546
 0556 IF(F(I-1,J).EQ.4) GO TO 558
      U(I,J)=U(I-1,J)-DX*(V(I,J)-V(I,J-1))/DY
      GO TO 546
 0558 U(I,J)=.5*DX*(V(I,J-1)-V(I,J))/DY
      U(I-1,J)=-U(I,J)
      GO TO 546
 0555 IF(F(I-1,J).NE.4) GO TO 559
      IF(F(I,J+1).EQ.4) GO TO 560
      U(I-1,J)=U(I,J)+DX*(V(I,J)-V(I,J-1))/DY
      GO TO 546
 0560 U(I-1,J)=U(I,J)
      V(I,J)=V(I,J-1)
      GO TO 546
 0559 IF(F(I,J+1).NE.4) GO TO 561
      V(I,J)=V(I,J-1)-DY*(U(I,J)-U(I-1,J))/DX
      GO TO 546
 0561 WRITE(6,28)I,J
      GO TO 901
 0546 CONTINUE
C
C         IMPOSE RIGID WALL (NO-SLIP) BOUNDARY CONDITIONS
C
  570 DO 571 I=2,IBAR1
      IF(F(I,J).EQ.3) GO TO 572
      V(I,1)=0.0
      U(I,1)=-U(I,2)
      GO TO 573
  572 V(I,1)=V(I,2)
      U(I,1)=U(I,2)
  573 V(I,JBAR)=0.0
      U(I,JBAR)=-U(I,JBAR1)
```

```
  571 CONTINUE
      DO 574 J=1,JBAR
      U(1,J)=0.0
      IF(IESYM.NE.1) GO TO 575
      V(1,J)=V(2,J)
      GO TO 576
 0575 V(1,J)=-V(2,J)
 0576 U(IBAR,J)=0.
      V(IBAR,J)=-V(IBAR1,J)
  574 CONTINUE
      GO TO KRET
 0699 CONTINUE
C
C          COMPUTE FORCES BETWEEN WHEEL AND SOIL
C
      SFX=0.
      SFY=0.
      ASSIGN 653 TO KRET
      DO 651 J=2,JBAR1
      DO 651 I=2,IBAR1
      IF(FE(I,J).NE.2) GO TO 651
      IF(F(I+1,J).NE.5.AND.F(I-1,J).NE.5.AND.F(I,J+1).NE.5.
     *AND.F(I,J-1).NE.5) GO TO 651
      SAVE=ETA(I,J)
      KEEP=FC(I,J)
      IF(F(I,J).NE.2.AND.F(I,J).NE.3) 651,611
 0653 SUM1=2.*DY*ETA(I,J)*(U(I,J)-U(I-1,J))/DX
      IF(SUM1.LT.0.0) GO TO 654
      SUM1=0.
 0654 SUM2=2.*DX*ETA(I,J)*(V(I,J)-V(I,J-1))/DY
      IF(SUM2.LT.0.0) GO TO 655
      SUM2=0.
 0655 IF(PHI(I,J).LT.0.0) GO TO 656
      SUM3=PHI(I,J)*RHOS*DX
      SUM4=PHI(I,J)*RHOS*DY
      GO TO 657
 0656 SUM3=0.
      SUM4=0.
 0657 SUM5=TAUXY(I,J)*DX
      SUM6=TAUXY(I,J)*DY
      ETA(I,J)=SAVE
      FC(I,J)=KEEP
      IF(F(I,J+1).NE.5) GO TO 660
      SFX=SFX-SUM5
      SFY=SFY-SUM2+SUM3
 0660 IF(F(I,J-1).NE.5) GO TO 658
      SFX=SFX+SUM5
      SFY=SFY+SUM2-SUM3
 0658 IF(F(I+1,J).NE.5) GO TO 659
      SFX=SFX-SUM1+SUM4
      SFY=SFY-SUM6
 0659 IF(F(I-1,J).NE.5) GO TO 651
      SFX=SFX+SUM1-SUM4
      SFY=SFY+SUM6
 0651 CONTINUE
      IF(IESYM.NE.1) GO TO 661
      SFX=0.
      SFY=2.*SFY
 0661 CONTINUE
C
C          REGION 80A- CHECK ESTIMATE OF WHEEL VELOCITY AGAINST
C          CALCULATED VALUE, ADJUST ESTIMATE AND RE-CYCLE IF NECESSARY
C          COMPUTE NEW WHEEL VELOCITY BY IMPULSE-MOMENTUM PRICIPLE
C
      UWN=UW+(GX+SFX/WM)*DT
      VWN=VW+(GY+SFY/WM)*DT
      IF(IESYM.EQ.1) GO TO 803
      IF(SFX.NE.0.0) GO TO 810
      IF(SFXG.EQ.0.0) GO TO 822
      NOR=0
      SFXC=0.
      SFYG=0.
      GO TO 4036
 0822 ERRORU=0.
      GO TO 803
 0810 ERRORU=((SFX-SFXG)/SFX)*100.
      IF(ABS(ERRORU).LT.0.0001) GO TO 800
      IF(NOR.GE.1) GO TO 805
 0800 SFX1=SFX
      SFXG1=SFXG
      SFXG=(SFX+SFXG)/2.
      GO TO 303
 0805 X1=SFXG
      SFXG=((SFX1*SFXG)-(SFX*SFXG1))/(SFXG-SFXG1+SFX1-SFX)
      SFX1=SFX
      SFXG1=X1
```

```
 0803 IF(SFY.NE.0.0) GO TO 819
      IF(SFYG.EQ.0.0) GO TO 825
      NOR=0
      SFXG=0.
      SFYG=0.
      GO TO 4036
 0825 ERRORV=0.
      GO TO 807
 0819 ERRORV=((SFY-SFYG)/SFY)*100.
      IF(ABS(ERRORV).LT.0.0001) GO TO 806
      IF(NOR.GE.1) GO TO 808
 0806 SFY1=SFY
      SFYG1=SFYG
      SFYG=(SFY+SFYG)/2.
      GO TO 807
 0808 Y1=SFYG
      SFYG=((SFY1*SFYG)-(SFY*SFYG1))/(SFYG-SFYG1+SFY1-SFY)
      SFY1=SFY
      SFYG1=Y1
 0807 IF(ABS(ERRORU).LT.0.0001.AND.ABS(ERRORV).LT.0.0001) GO TO 804
      IF(ABS(TCP-T).GT.(DT/2.)) GO TO 809
      WRITE(6,14)T,UWN,VWN,UG,VG,SFX,SFY,ERRORU,ERRORV,IC
      WRITE(6,21)SFXG,SFYG
 0809 IF(NOR.GE.10) GO TO 804
      NOR=NOR+1
      IF(NOR.NE.3) GO TO 4036
      NIC=IC
      GO TO 4036
 0804 CONTINUE
C
C         REGION 80- CALCULATE MOVEMENT OF WHEEL AND TEST FOR SHIFT
C         GREATER THAN ALLOWED
C         RE-CYCLE WITH REDUCED DT IF NECESSARY
C
      DXSF=((UW+UG)/2.)*DT
      DYSF=((VW+VG)/2.)*DT
      ISF=(4.*DXSF)/DX
      JSF=(4.*DYSF)/DY
      IF(ISF.EQ.0.AND.JSF.EQ.0) GO TO 857
      WRITE(6,17)T
      T=T-DT/4.
      DT=.75*DT
      DTCP=.75*DTCP

      DTPP=.75*DTPP
      DTSP=.75*DTSP
      TCP=T
      TPP=T
      TSP=T
      NOR=0
      SFXG=0.
      SFYG=0.
      WRITE(6,16)DT,DTCP,DTPP,DTSP
      GO TO 4036
  857 CONTINUE
      IF(UW.EQ.0.0) GO TO 6870
      ALPHAU=UG/UW
 6870 IF(VW.EQ.0.0) GO TO 6868
      ALPHAV=VG/VW
 6868 CONTINUE
      UW=UG
      VW=VG
      SFXG=SFX
      SFYG=SFY
      SDXSF=SDXSF+DXSF
      SDYSF=SDYSF+DYSF
C
C         REGION 60   COMPUTE STRESS TENSOR OF EACH CELL
C                     FIND PRINCIPAL STRESSES AND DIRECTION
C                     TEST YIELD CRITERIA
C
      MM=1
      STNMAX=0.
      ASSIGN 615 TO KRET
 0600 DO 601 J=2,JBAR1
      DO 601 I=2,IBAR1
      GO TO (611,612,613) MM                                    C
 0611 FC(I,J)=0
      TAUXY(I,J)=0.0                                            C
      IF(F(I,J).NE.2.AND.F(I,J).NE.3) GO TO 601
      URP=(U(I,J+1)+U(I,J))/2.
      ULP=(U(I-1,J+1)+U(I-1,J))/2.
      UAP=(U(I,J)+U(I,J-1))/2.
      UAL=(U(I-1,J)+U(I-1,J-1))/2.
      VLA=(V(I,J-1)+V(I-1,J-1))/2.
      VLB=(V(I,J)+V(I-1,J))/2.
      VRA=(V(I+1,J-1)+V(I,J-1))/2.
```

```
      VRB=(V(I+1,J)+V(I,J))/2.
      IF(FE(I,J).NE.2) GO TO 616
      IF(F(I+1,J).NE.1) GO TO 618
      UBR=0.
      UAR=0.
      VRA=0.
      VRB=0.
      GO TO 617
0618  IF(F(I+1,J).NE.5) GO TO 617
      UBR=UG
      UAR=UG
      VRA=VG
      VRB=VG
0617  IF(F(I-1,J).NE.1.OR.F(I-1,J).NE.6) GO TO 620
      UAL=0.
      UBL=0.
      VLA=0.
      VLB=0.
      GO TO 619
0620  IF(F(I-1,J).NE.5) GO TO 619
      UAL=UG
      UBL=UG
      VLA=VG
      VLB=VG
0619  IF(F(I,J+1).NE.1) GO TO 622
      UBR=0.
      UBL=0.
      VRB=0.
      VLB=0.
      GO TO 621
0622  IF(F(I,J+1).NE.5) GO TO 621
      UBR=UG
      UBL=UG
      VRB=VG
      VLB=VG
0621  IF(F(I,J-1).NE.1) GO TO 624
      UAR=0.
      UAL=0.
      VRA=0.
      VLA=0.
      GO TO 623
0624  IF(F(I,J-1).NE.5) GO TO 623
      UAR=UG

      UAL=UG
      VRA=VG
      VLA=VG
      GO TO 623
0616  IF(FE(I+1,J-1).NE.1) GO TO 627
      UAR=UG
      VRA=VG
0627  IF(FE(I-1,J-1).NE.1) GO TO 625
      UAL=UG
      VLA=VG
0625  IF(FE(I+1,J+1).NE.1) GO TO 626
      UBR=UG
      VRB=VG
0626  IF(FF(I-1,J+1).NE.1) GO TO 623
      UBL=UG
      VLB=VG
0623  IF(F(I,J).NE.3) GO TO 628
      SUM1=0.
      SUM2=0.
      IF(F(I,J-1).NE.4) GO TO 604
      IF(F(I-1,J).NE.4) GO TO 605
      IF(F(I+1,J).EQ.4.OR.F(I,J+1).EQ.4) GO TO 603
      SUM1=.5*(U(I,J+1)+U(I-1,J+1)-U(I,J)-U(I-1,J))/DY
      SUM2=.5*(V(I+1,J)+V(I+1,J-1)-V(I,J)-V(I,J-1))/DX
      GO TO 603
0605  IF(F(I+1,J).NE.4) GO TO 606
      IF(F(I,J+1).EQ.4) GO TO 603
      SUM1=.5*(U(I,J+1)+U(I-1,J+1)-U(I,J)-U(I-1,J))/DY
      SUM2=.5*(V(I,J)+V(I,J-1)-V(I-1,J)-V(I-1,J-1))/DX
      GO TO 603
0606  IF(F(I,J+1).EQ.4) GO TO 603
      SUM1=.5*(U(I,J+1)+U(I-1,J+1)-U(I,J)-U(I-1,J))/DY
      SUM2=(VRA-VLA+VRB-VLB)/(2.*DX)
      GO TO 603
0604  IF(F(I+1,J).NE.4) GO TO 607
      IF(F(I,J+1).NE.4) GO TO 608
      IF(F(I-1,J).EQ.4) GO TO 603
      SUM1=.5*(U(I,J)+U(I-1,J)-U(I,J-1)-U(I-1,J-1))/DY
      SUM2=.5*(V(I,J)+V(I,J-1)-V(I-1,J)-V(I-1,J-1))/DX
      GO TO 603
0608  IF(F(I-1,J).EQ.4) GO TO 603
      SUM1=(UBR-UAR+UBL-UAL)/(2.*DY)
      SUM2=.5*(V(I,J)+V(I,J-1)-V(I-1,J)-V(I-1,J-1))/DX
```

```
      GO TO 603
 0607 IF(F(I,J+1).NE.4) GO TO 609
      IF(F(I-1,J).EQ.4) GO TO 610
      SUM1=.5*(U(I,J)+U(I-1,J)-U(I,J-1)-U(I-1,J-1))/DY
      SUM2=(VRA-VLA+VRB-VLB)/(2.*DX)
      GO TO 603
 0610 SUM1=.5*(U(I,J)+U(I-1,J)-U(I,J-1)-U(I-1,J-1))/DY
      SUM2=.5*(V(I+1,J)+V(I+1,J-1)-V(I,J)-V(I,J-1))/DX
      GO TO 603
 0609 IF(F(I-1,J).NE.4) GO TO 603
      SUM1=(UBR-UAR+UBL-UAL)/(2.*DY)
      SUM2=.5*(V(I+1,J)+V(I+1,J-1)-V(I,J)-V(I,J-1))/DX
      GO TO 603
 0628 SUM1=(UBR-UAR+UBL-UAL)/(2.*DY)
      SUM2=(VRA-VLA+VRB-VLB)/(2.*DX)
 0603 SUM3=2.*(ETA(I,J)**2)*((((U(I,J)-U(I-1,J))/DX)**2)
     *+(((V(I,J)-V(I,J-1))/DY)**2)+.25*((SUM1+SUM2)**2))
      STRN=0.5*(SUM1+SUM2)
      STRAIN(1)=ABS(STRN)
      IF(STRAIN(1).LT.STNMAX) GO TO 629
      STNMAX=STRAIN(1)
 0629 CONTINUE
      ETA(I,J)=ETA0
      TAUXY(I,J)=2.0*ETA(I,J)*STRN
      FC(I,J)=0
      GO TO KRET
  615 FK(I,J)=FK0
      IF(SUM3.LT.(FK(I,J)**2))GO TO 601
      IF(STRAIN(1).LE.GAMAE) GO TO 601
      CALL SPLINE (MNPTS,MNCVS,MMAX,NS,NCVS,MS,STRAN,STRES,STRAIN,EPS,
     *PROXIV,STRESS,SS1,SS2,S2,S3,DELY,H,IW,ITG)
      ETA(I,J)=0.5*STRESS(1,1)/STRAIN(1)
      IF(STRN.GE.0.0) GO TO 680
      TAUXY(I,J)=-STRESS(1,1)
      GO TO 681
  680 TAUXY(I,J)=STRESS(1,1)
  681 IW=IW+1
      FC(I,J)=1
      GO TO 601
 0612 IF(F(I,J).NE.2.AND.F(I,J).NE.3) GO TO 675
      SIGMAX(I,J)=2.0*ETA(I,J)*(U(I,J)-U(I-1,J))/DX
      GO TO 601
 0675 ETA(I,J)=0.

      SIGMAX(I,J)=0.0
      GO TO 601
 0613 IF(F(I,J).NE.2.AND.F(I,J).NE.3) GO TO 677
      SIGMAY(I,J)=2.0*ETA(I,J)*(V(I,J)-V(I,J-1))/DY
      GO TO 601
 0677 ETA(I,J)=0.
      SIGMAY(I,J)=0.0
  601 CONTINUE
C
C     STORE STRESSES ON TAPE
C
      IF(ABS(TSP-T).GT.(DT/2.)) GO TO 673
      GO TO(670,671,672)MM
 0670 WRITE(9)T
      WRITE(9)TAUXY
      GO TO 673
 0671 WRITE(9)SIGMAX
      GO TO 673
 0672 WRITE(9)SIGMAY
      END FILE 9
      TSP=TSP+DTSP
      NSP=NSP+1
      WRITE(6,25)T,NSP
 0673 CONTINUE
      MM=MM+1
      IF(MM.LE.3) GO TO 600
C
C        CALCULATE THE CELL DISCREPANCIES
C
      IDMAX=0
      JDMAX=0
      DMAX=0.
      DO 4035 J=2,JBAR1
      DO 4035 I=2,IBAR1
      D(I,J)=0
      ADMAX=0.
      IF(F(I,J).NE.2.AND.F(I,J).NE.3) GO TO 4035
      D(I,J)=((U(I,J)-U(I-1,J))/DX)+((V(I,J)-V(I,J-1))/DY)
      ADMAX=D(I,J)
      IF(D(I,J).LT.0.0) ADMAX=-D(I,J)
      IF(DMAX.GE.ADMAX) GO TO 4035
      DMAX=ADMAX
      IDMAX=I
```

```
      JDMAX=J
 4035 CONTINUE
C
C          TEST FOR TIME TO PRINT CELL VARIABLES ON LINE
C
      IF(ABS(TCP-T).GT.(DT/2.)) GO TO 720
      WRITE(6,23)
      DO 4033 J=MJS,NJF
      DO 4033 I=NIS,NII
      WRITE(6,4002)I,J,X(I),Y(J),U(I,J),V(I,J),D(I,J),PHI(I,J),R(I,J),
     *SIGMAX(I,J),SIGMAY(I,J),TAUXY(I,J),F(I,J),FE(I,J),FC(I,J),KC(I,J)
 4033 CONTINUE
      WRITE(6,4031)T
 0720 CONTINUE
C
C          REGION 70  PARTICLE MOVEMENT
C
      NN=1020
      DO 721 M=1,MPARL
      IF(M.LE.MPAR) GO TO 722
      NN=MPRR
 0722 READ(14)(TXK(N),TYK(N),N=1,NN)
      DO 701 N=1,NN
      XK(N)=TXK(N)
      YK(N)=TYK(N)
      I=XK(N)/DX+2.0
      J=YK(N)/DY+2.0
      FI=I
      FJ=J
      FX=XK(N)/DX+2.0-FI
      FY=YK(N)/DY+2.0-FJ
      IF(F(I,J).NE.2.AND.F(I,J).NE.3) GO TO 701
      SX=(X(I)-XK(N))/DX
      U1=U(I-1,J)
      U2=U(I,J)
      IF(FY.LT.0.5) GO TO 703
      SY=(YPL(J)-YK(N))/DY
      U3=U(I-1,J+1)
      U4=U(I,J+1)
      IF(FE(I-1,J).NE.1) GO TO 704
      U3=UW
      U1=UW
      GO TO 705
 0704 IF(FE(I-1,J+1).NE.1) GO TO 705
      U3=U1
 0705 IF(FE(I+1,J).NE.1) GO TO 706
      U4=UW
      U2=UW
      GO TO 707
 0706 IF(FE(I+1,J+1).NE.1) GO TO 707
      U4=U2
      GO TO 707
 0703 SY=(YK(N)-YPL(J-1))/DY
      U3=U(I-1,J-1)
      U4=U(I,J-1)
      IF(FE(I-1,J).NE.1) GO TO 708
      U3=UW
      U1=UW
      GO TO 709
 0708 IF(FE(I-1,J-1).NE.1) GO TO 709
      U3=U1
 0709 IF(FE(I+1,J).NE.1) GO TO 710
      U4=UW
      U2=UW
      GO TO 707
 0710 IF(FE(I+1,J-1).NE.1) GO TO 707
      U4=U2
  707 UK=(0.5+SX)*(0.5+SY)*U1+(0.5-SX)*(0.5+SY)*U2
     *+(0.5+SX)*(0.5-SY)*U3+(0.5-SX)*(0.5-SY)*U4
      XK(N)=XK(N)+UK*DT
      SY=(Y(J)-YK(N))/DY
      V1=V(I,J-1)
      V2=V(I,J)
      IF(FX.LT.0.5) GO TO 711
      SX=(XPL(I)-XK(N))/DX
      V3=V(I+1,J-1)
      V4=V(I+1,J)
      IF(FE(I,J-1).NE.1) GO TO 712
      V3=VW
      V1=VW
      GO TO 713
 0712 IF(FE(I+1,J-1).NE.1) GO TO 713
      V3=V1
 0713 IF(FE(I,J+1).NE.1) GO TO 714
      V4=VW
      V2=VW
```

```
      GO TO 715
 0714 IF(FF(I+1,J+1).NE.1) GO TO 715
      V4=V2
      GO TO 715
  711 SX=(XK(N)-XPL(I-1))/DX
      V3=V(I-1,J-1)
      V4=V(I-1,J)
      IF(FF(I,J-1).NE.1) GO TO 716
      V3=VW
      V1=VW
      GO TO 717
 0716 IF(FF(I-1,J-1).NE.1) GO TO 717
      V3=V1
 0717 IF(FF(I,J+1).NE.1) GO TO 718
      V4=VW
      V2=VW
      GO TO 715
 0718 IF(FF(I-1,J+1).NE.1) GO TO 715
      V4=V2
  715 VK=(0.5+SX)*(0.5+SY)*V1+(0.5+SX)*(0.5-SY)*V2
     *+(0.5-SX)*(0.5+SY)*V3+(0.5-SX)*(0.5-SY)*V4
      YK(N)=YK(N)+VK*DT
  701 CONTINUE
      WRITE(15) (XK(N),YK(N),N=1,NN)
  721 CONTINUE
      REWIND 14
      REWIND 15
C
C         COMPUTE COORDINATES OF PARTICLES AFTER SHIFTING
C
      DO 847 J=1,JBAR
      DO 847 I=1,IBAR
      KC(I,J)=0
 0847 CONTINUE
      KPLUM=0
      KPLM=0
      NN=1020
      DO 841 M=1,MPAR1
      IF(M.LE.MPAR) GO TO 842
      NN=MPPR
 0842 READ(15)(XK(N),YK(N),N=1,NN)
      DO 811 N=1,NN
      IF((XK(N)-DXSF).LT.0.0.OR.(YK(N)-DYSF).GT.YT) GO TO 811

      KPLUM=KPLUM+1
      KPLM=KPLM+1
      TXK(KPLUM)=XK(N)-DXSF
      TYK(KPLUM)=YK(N)-DYSF
      I=TXK(KPLUM)/DX+2.0
      J=TYK(KPLUM)/DY+2.0
      IF(F(I,J).NE.5) GO TO 844
      II=(TXK(KPLUM)+DXSF)/DX+2.
      IF(F(II,J).EQ.5) GO TO 845
      TXK(KPLUM)=TXK(KPLUM)+DXSF
      I=II
      GO TO 844
 0845 JJ=(TYK(KPLUM)+DYSF)/DY+2.
      IF(F(I,JJ).EQ.5) GO TO 846
      TYK(KPLUM)=TYK(KPLUM)+DYSF
      J=JJ
      GO TO 844
 0846 WRITE(6,60)
      WRITE(6,59)I,J,TXK(KPLUM),TYK(KPLUM),F(I,J),FE(I,J)
      GO TO 901
 0844 CONTINUE
      KC(I,J)=KC(I,J)+1
      IF(KPLUM.EQ.1020) GO TO 812
      GO TO 811
  812 WRITE(14) (TXK(N),TYK(N),N=1,1020)
      KPLUM=0
  811 CONTINUE
  841 CONTINUE
  813 N=KPLUM
C
      K=KPLM
C     CONTROL THE DISPLACEMENT OF SHIFTING COORDINATES LESS THAN
C     HALF CELL WIDTH EACH TIME
C
      CNX=NX
      CNY=NY
      XKNX=XO+(CNX-1.0)*DXK
      YKNY=YO+(CNY-1.0)*DYK
      SMX=XR+SDXSF
      SMY=YT+SDYSF
      M=(SMX-XKNX)/DXK-XSFPAR
      IF(M)828,816,826
 0828 XSFPAR=XSFPAR-1.
```

```
      GO TO 816
 0826 CONTINUE
      YP=YO-SDYSF
      XP=XO+(CNX+XSFPAR)*DXK-SDXSF
      DO 814 ICY=1,NY
      N=N+1
      TXK(N)=XP
      TYK(N)=YP
      I=TXK(N)/DX+2.0
      J=TYK(N)/DY+2.0
      KC(I,J)=KC(I,J)+1
      IF(N.LT.1020) GO TO 815
      WRITE(14) (TXK(N),TYK(N),N=1,1020)
      N=0
  815 K=K+1
      YP=YP+DYK
  814 CONTINUE
      XSFPAR=XSFPAR+1.0
  816 NN=(SMY-YKNY)/DYK-YSFPAR
      IF(NN)829,8000,827
 0829 YSFPAR=YSFPAR-1.
      GO TO 8000
 0827 CONTINUE
      YP=YO+(CNY+YSFPAR)*DYK-SDYSF
      XP=XO+XSFPAR*DXK-SDXSF
      DO 818 ICX=1,NX
      N=N+1
      TXK(N)=XP
      TYK(N)=YP
      I=TXK(N)/DX+2.0
      J=TYK(N)/DY+2.0
      KC(I,J)=KC(I,J)+1
      IF(N.LT.1020) GO TO 817
      WRITE(14) (TXK(N),TYK(N),N=1,1020)
      N=0
  817 K=K+1
      XP=XP+DXK
  818 CONTINUE
      YSFPAR=YSFPAR+1.0
 8000 KBAR=K
      MPAR=KBAR/1020
      MPRR=KBAR-MPAR*1020
      MPAR1=MPAR+1

      WRITE(14) (TXK(N),TYK(N),N=1,MPRR)
      REWIND 14
      REWIND 15
C
C         STORE PARTICLE COORDINATES ON TAPE 10
C
      IF(ABS(TPP-T).GT.(DT/2.)) GO TO 834
      ASSIGN 834 TO KRET
 0836 CONTINUE
      DATB(1)=T
      DATB(2)=KBAR
      DATB(3)=IBAR
      DATB(4)=JBAR
      DATB(5)=XR
      DATB(6)=YT
      DATB(7)=WX
      DATB(8)=WY
      DATB(9)=RW
      DATB(10)=UW
      DATB(11)=VW
      DATB(12)=DX
      DATB(13)=DY
      DATB(14)=DTPP
      DATB(15)=IWHEEL
      DATB(16)=IFSYM
      DATB(17)=ETAO
      DATB(18)=UWO
      DATB(19)=VWO
      DATB(20)=WM
      DATB(21)=RHOS
      DATB(22)=SDXSF
      DATB(23)=SDYSF
      WRITE(10)DATB
      NN=1020
      DO 833 M=1,MPAR1
      IF(M.LE.MPAR) GO TO 835
      NN=MPRR
 0835 READ(14)(TXK(N),TYK(N),N=1,NN)
      WRITE(10)(TXK(N),TYK(N),N=1,NN)
 0833 CONTINUE
      END FILE 10
      TPP=TPP+DTPP
      NPP=NPP+1
```

```
      WRITE(6,26)T,N2P
      REWIND 14
      GO TO KRET
 0834 CONTINUE
C
C         REGION 90- REFLAG CELLS
C
 0820 DO 821 J=2,JBAR1
      DO 821 I=2,IBAR1
      IF(F(I,J).NE.3) GO TO 850
      IF(KC(I,J).NE.0) GO TO 821
      F(I,J)=4
      PHI(I,J)=0.
      IF(F(I+1,J).NE.4) GO TO 830
      U(I,J)=0.
 0830 IF(F(I-1,J).NE.4) GO TO 831
      U(I-1,J)=0.
 0831 IF(F(I,J+1).NE.4) GO TO 832
      V(I,J)=0.
 0832 IF(F(I,J-1).NE.4) GO TO 821
      V(I,J-1)=0.
      GO TO 821
 0850 IF(F(I,J).NE.4) GO TO 821
      PHI(I,J)=0.
      IF(KC(I,J).EQ.0) GO TO 821
      F(I,J)=3
      ETA(I,J)=ETA0
      FE(I,J)=2
 0821 CONTINUE
      DO 823 J=2,JBAR1
      DO 823 I=2,IBAR1
      IF(KC(I,J).EQ.0) GO TO 823
      IF(F(I,J).NE.2) GO TO 851
      IF(F(I+1,J).NE.4.AND.F(I-1,J).NE.4.AND.F(I,J+1).NE.4.AND.
     *F(I,J-1).NE.4) GO TO 823
      F(I,J)=3
      PHI(I,J)=0.
      GO TO 823
  851 IF(F(I,J).NE.3) GO TO 823
      SUMPHI=0.
      N=0
      IF(F(I+1,J).EQ.4.OR.F(I-1,J).EQ.4.OR.F(I,J+1).EQ.4.OR.
     *F(I,J-1).EQ.4) GO TO 823

      IF(F(I+1,J).NE.2) GO TO 852
      SUMPHI=SUMPHI+PHI(I+1,J)
      N=N+1
 0852 IF(F(I-1,J).NE.2) GO TO 853
      SUMPHI=SUMPHI+PHI(I-1,J)
      N=N+1
 0853 IF(F(I,J+1).NE.2) GO TO 854
      SUMPHI=SUMPHI+PHI(I,J+1)
      N=N+1
 0854 IF(F(I,J-1).NE.2) GO TO 855
      SUMPHI=SUMPHI+PHI(I,J-1)
      N=N+1
 0855 F(I,J)=2
      FPN=N
      IF(N.EQ.0) GO TO 856
      PHI(I,J)=SUMPHI/FPN
      GO TO 823
 0856 IF(FE(I,J).NE.2) GO TO 823
      WRITE(6,6)I,J,F(I,J),FE(I,J),KC(I,J)
 0823 CONTINUE
C
C         RE-IMPOSE BOUNDARY CONDITIONS TO FREE SURFACE
C
      ASSIGN 880 TO KRET
      GO TO 545
 0880 CONTINUE
C
C         ESTABLISH ETA FOR FORMERLY EMPTY CELLS
C
      ASSIGN 824 TO KRET
      DO 824 J=2,JBAR1
      DO 824 I=2,IBAR1
      IF(FE(I,J).NE.2) 824,611
 0824 CONTINUE
C
C         TEST FOR TIME TO PRINT CELL VARIABLES
C
      ASSIGN 801 TO KRET
 0802 CONTINUE
      WRITE(6,11) T,SFX,SFY
      WRITE(6,18) T,SDXSF,SDYSF
      WRITE(6,22)T,DXSF,DYSF
      WRITE(6,30)T,UM,VM,STNMAX
```

```
      WRITE(6,33)T,ALPHAU,ALPHAV,CONMAX
      WRITE(6,34)T,DMAX,IDMAX,JDMAX
      WRITE(6,13)T,ERRORU,ERRORV,NUR
      WRITE(6,32)T,XSFPAR,YSFPAR,KBAR
      WRITE(6,10)T,IC,CONV,SUMR
      NUR=0
      IF(ABS(TCP-T).GT.(DT/2.)) GO TO 801
      DATA(1)=T
      DATA(2)=KBAR
      DATA(3)=SDXSF
      DATA(4)=SDYSF
      DATA(5)=XSFPAR
      DATA(6)=YSFPAR
      DATA(7)=SFX
      DATA(8)=SFY
      DATA(9)=VW
      DATA(10)=UW
      DATA(11)=ALPHAU
      DATA(12)=ALPHAV
      DATA(13)=DTCP
      DATA(14)=DTSP
      DATA(15)=DTPP
      DATA(16)=IBAR
      DATA(17)=JBAR
      DATA(18)=XR
      DATA(19)=YT
      DATA(20)=WX
      DATA(21)=WY
      DATA(22)=RW
      DATA(23)=DX
      DATA(24)=DY
      DATA(25)=DXSF
      DATA(26)=DYSF
      DATA(27)=ERRORU
      DATA(28)=ERRORV
      DATA(29)=IC
      DATA(30)=SUMR
      DATA(31)=SFXG
      DATA(32)=SFYG
      DATA(33)=INHFEL
      DATA(34)=IFSYM
      DATA(35)=ETAO
      DATA(36)=UWO

      DATA(37)=VWO
      DATA(38)=WM
      DATA(39)=RHOS
      WRITE(8)DATA,U,V
      END FILE 8
      WRITE(13)DATA,PHI,ETA,F,FE
      END FILE 13
      TCP=TCP+DTCP
      NCS=NCS+1
      WRITE(6,4030)T,NCS
      GO TO KRET
 0801 CONTINUE
C
C          END PROBLEM
C
      IF((TL-T).GT.(DT/2.)) GO TO 300
 0901 CALL SECOND(BTIME)
      WRITE(6,12)BTIME
      GO TO 4000
 9909 CONTINUE
      STOP
      END
```

5

```
      SUBROUTINE SPLINE (MNPTS,MNCVS,MMAX,N,NCVS,M,X,Y,T,EPS,PROXIN,SS,SS1
     1,SS2,S2,S3,DELY,H,IW,IIG)
      DIMENSION X(MNPTS),Y(MNPTS,MNCVS),T(MMAX),DELY(MNPTS,MNCVS),
     1S2(MNPTS,MNCVS),S3(MNPTS,MNCVS),SS1(MNCVS,MMAX),SS(MNCVS,MMAX) ,
     AH(MNPTS)         ,SS2(MNCVS,MMAX),PROXIN(MNCVS),
     2         B(60),DELSQY(60),H2(60),C(60)
      EPSLN=EPS
      IF(IW)15,2,15
    2 N1=N-1
      IW=2
      DO 101 K=1,NCVS
    3 DO 51 I=1,N1
      H(I)=X(I+1)-X(I)
      JJ=I+1
   51 DELY(I,K)=(Y(JJ,K)-Y(I,K))/ H(I)
    4 DO 52 I=2,N1
      II=I-1
      H2(I)=H(II)   +H(I)
      DELSQY(I)=(DELY(I,K)-DELY(II,K))/ H2(I)
      B(I)=.5*H(II)   /H2(I)
      S2(I,K)=2.*DELSQY(I)
   52 C(I)=3.* DELSQY(I)
      S2(1,K)=0.0
      S2(N,K)=0.0
      OMEGA=1.0717968
    5 ETA=0.
    6 DO 10 I=2,N1
      IB=I-1
      IF=I+1
    7 W=(C(I)-B(I)*S2(IB,K)-(.5-B(I))*S2(IF,K)-S2(I,K))*OMEGA
    8 IF (ABS(W)-ETA) 10,10,9
    9 ETA=ABS(W)
   10 S2(I,K)=S2(I,K)+W
   13 IF (ETA-EPSLN) 14,5,5
   14 DO 53 I=1,N1
      II=I+1
   53 S3(I,K)=(S2(II,K)-S2(I,K))/ H(I)
  101 CONTINUE
   15 CONTINUE
  104 J=0
  105 J=J+1
   16 I=1

   54 IF (T(J)-X(1)) 58,17,55
   55 IF (T(J)-X(N)) 57,59,58
   56 IF (T(J)-X(I)) 60,17,57
   57 I=I+1
      GO TO 56
   58 PRINT 44, J
   44 FORMAT (I4,24HTH ARGUMENT OUT OF RANGE)
      GO TO 61
   59 I=N
   60 CONTINUE
      IW=-1
      I=I-1
   17 DO 110 K=1,NCVS
      HT1=T(J)-X(I)
      II=I+1
      HT2=T(J)-X(II)
      PROD=HT1 * HT2
      SS2(K,J)=S2(I,K)+HT1*S3(I,K)
      DELSQS=(S2(I,K)+S2(II,K)+SS2(K,J))/6.
      SS(K,J)=Y(I,K)+HT1*DELY(I,K)+PROD*DELSQS
      SS1(K,J)=DELY(I,K)+(HT1+HT2)*DELSQS+PROD*S3(I,K)/6.0
  110 CONTINUE
   61 CONTINUE
      IF(J.LT.M)GO TO 105
      IF(IIG.GT.0)RETURN
      DO 120 K=1,NCVS
   20 PROXIN(K)=0.0
      DO 62 I=1,N1
      II=I+1
   62 PROXIN(K)=PROXIN(K)+.5*H(I)   *(Y(I,K)+Y(II,K))-H(I)   **3*(
     1S2(I,K)+S3(II,K))/24.
  120 CONTINUE
      RETURN
      END
```

TABLE I

SUMMARY OF RESULTS

Quantity	Comp.	Exp.	Comp.	Exp.	Comp.	Exp.
Equivalent mass of block, gm	102.2	102.2	102.2	102.2	153	153
Initial impact velocity, cm/sec	30	33	50	47	30	30
Peak acceleration, g	1.76	1.52	2.84	2.28	1.24	1.12
Peak pressure, kN/m^2	2.9	6.3	4.7	9.2	3.0	6.5
Maximum penetration, cm	27.11	23.07	28.16	24.46	--	--
Time at maximum penetration, sec	.395	.390	.375	.373	--	--

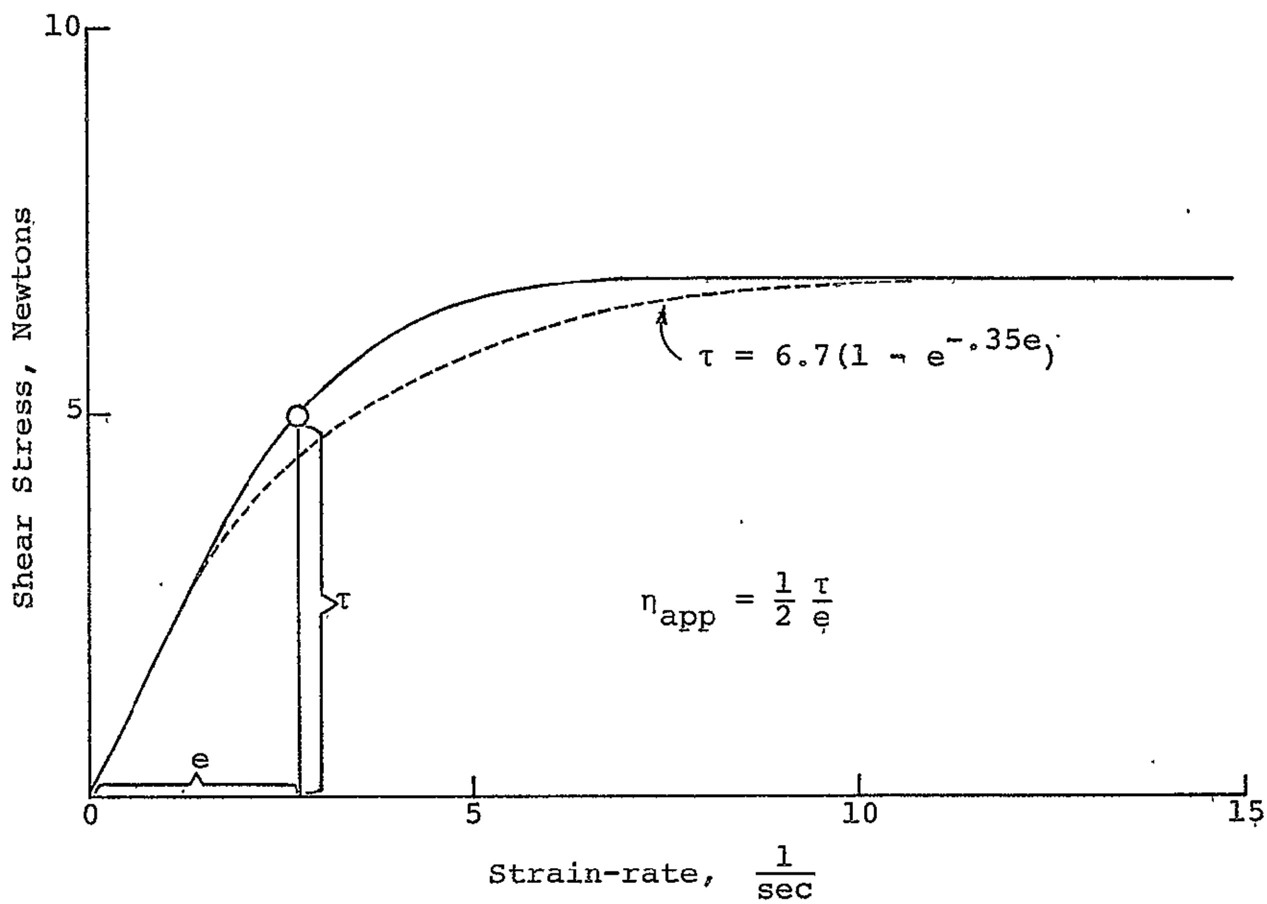
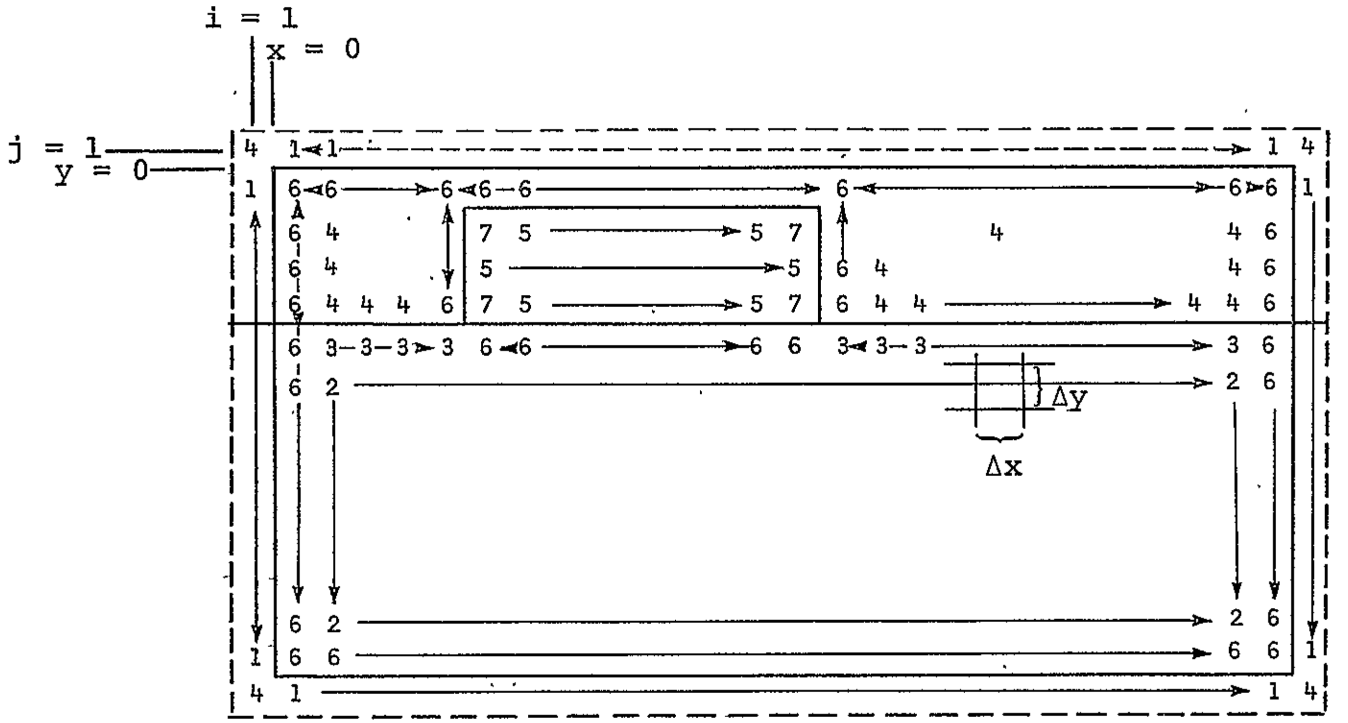


Figure 1. Soil constitutive relationship.



1 = No-slip Boundary

2 = Full Cells

3 = Free-surface Cells

4 = Empty Cells

5 = Moving Boundary

6 = Cell Adjacent to Boundary

7 = Corner Cells

Figure 2. Position for different types of cells.

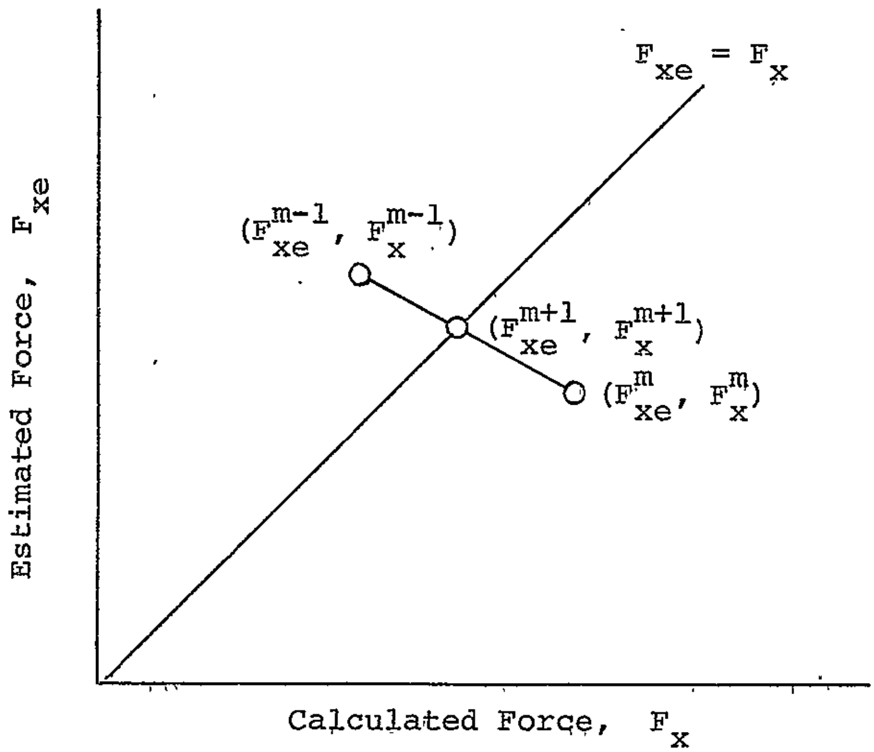


Figure 3. Linear scheme for estimating F_{xe}^{n+1}.

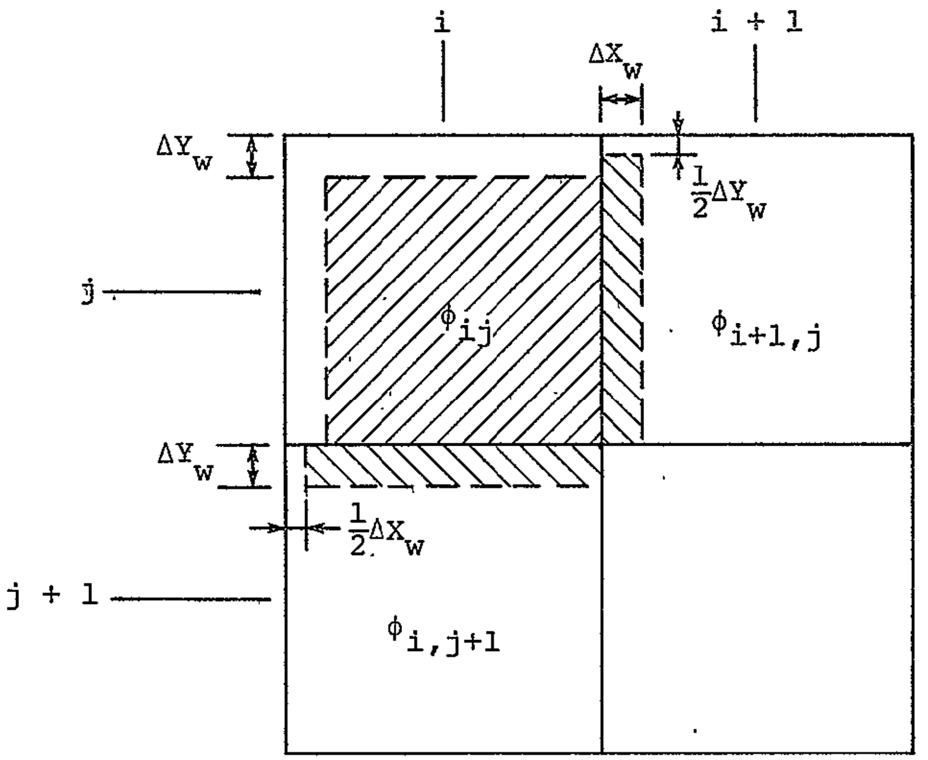


Figure 4. Areas used in shifting of pressure field for a positive ΔX_w and ΔY_w.

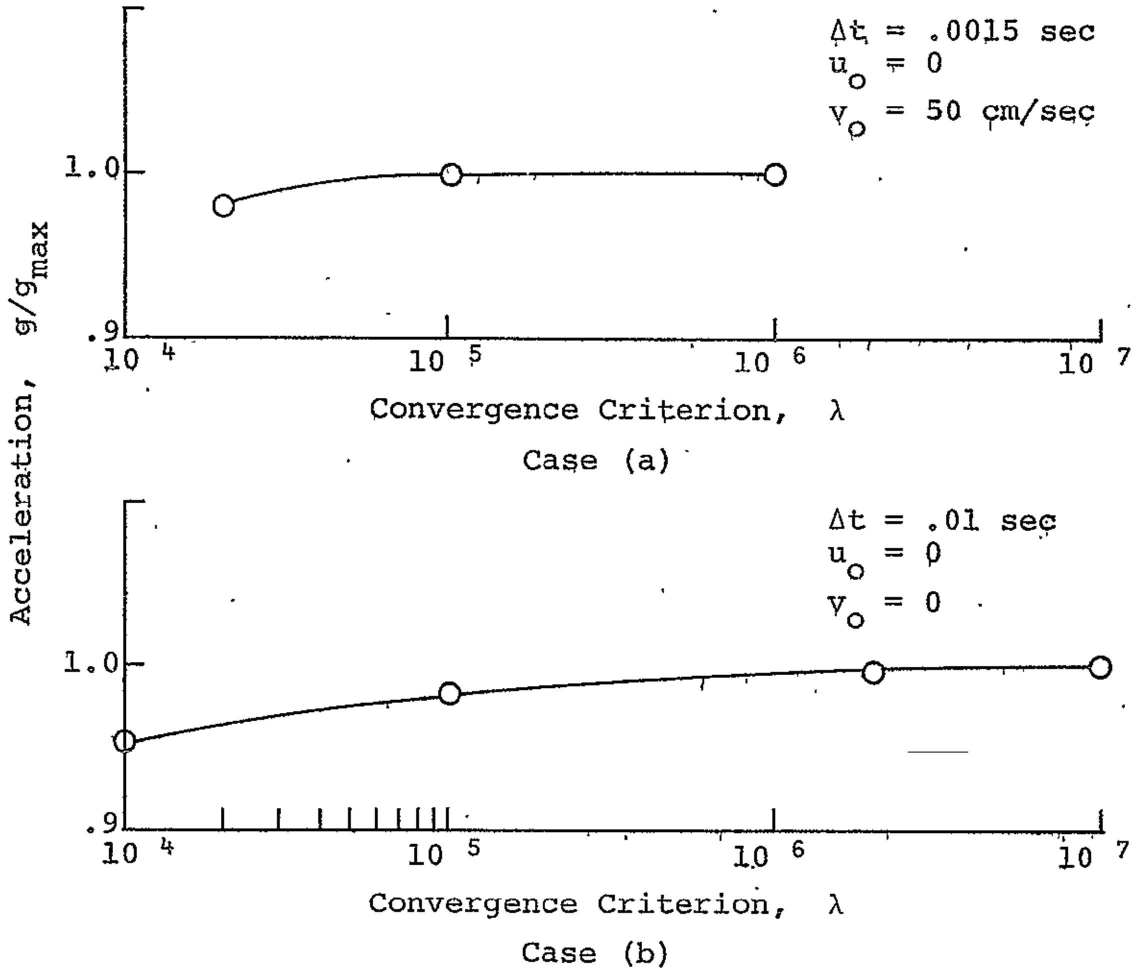


Figure 5. Accuracy of vertical force on rigid block.

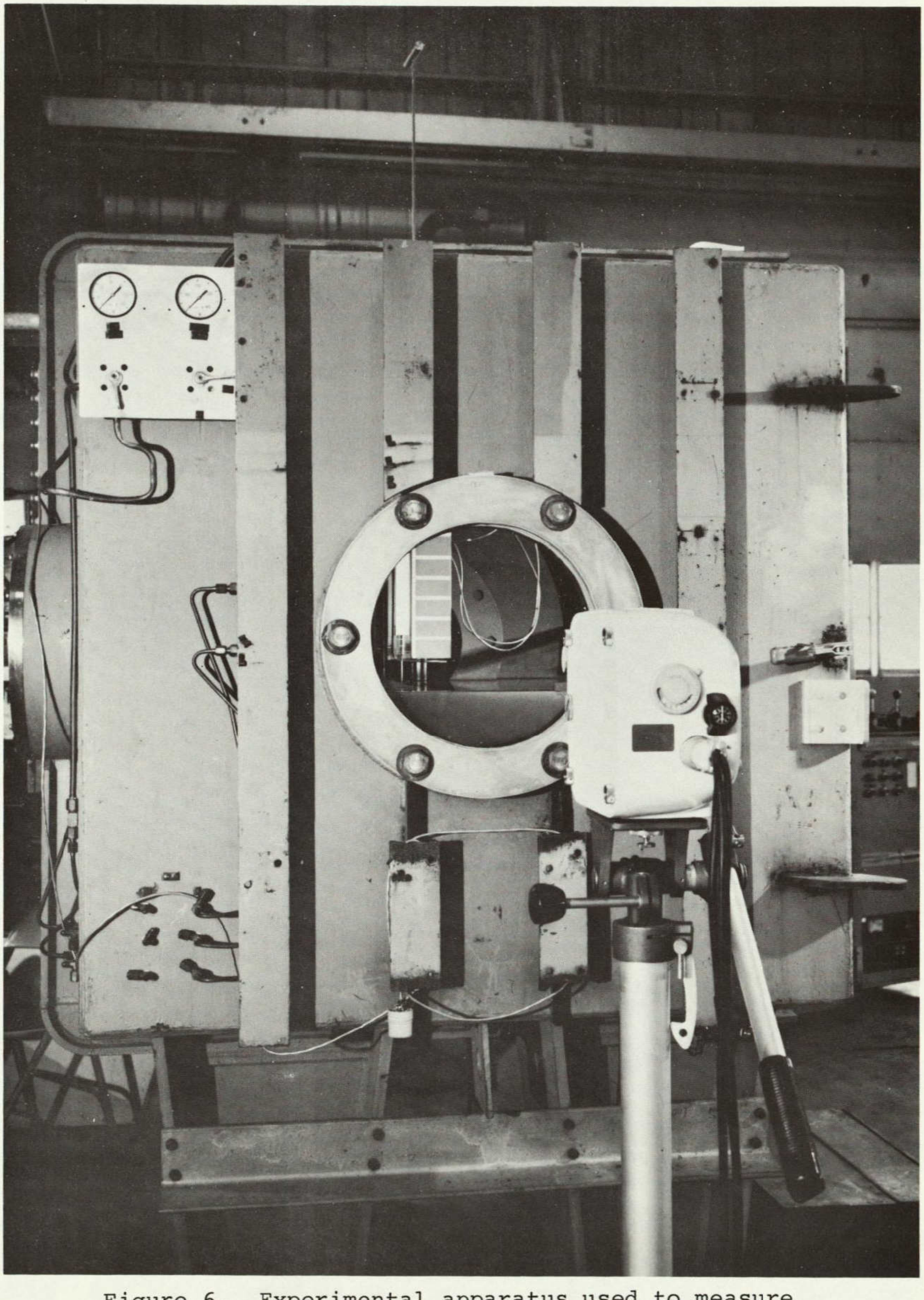


Figure 6. Experimental apparatus used to measure displacement, acceleration, and pressure.

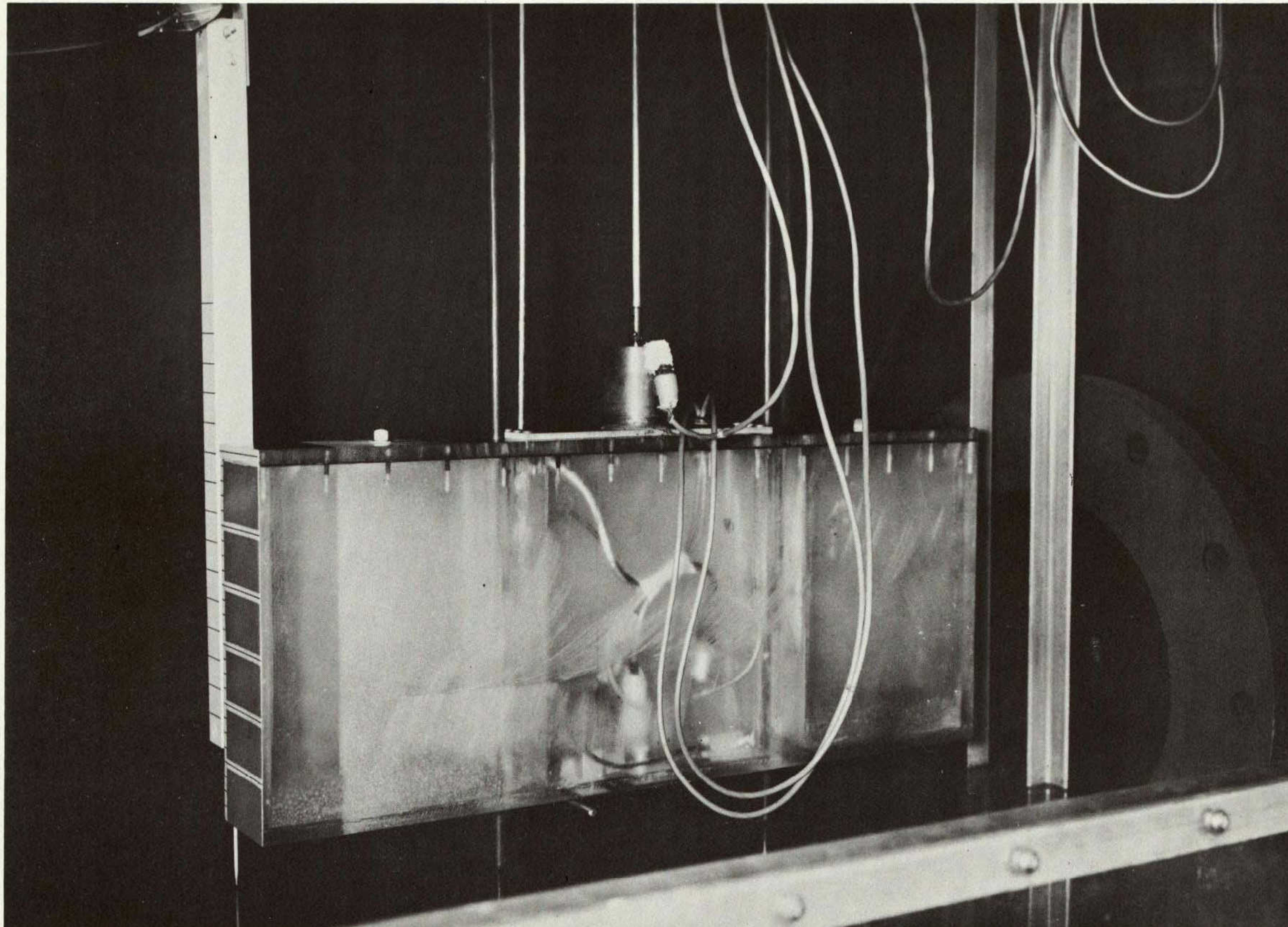
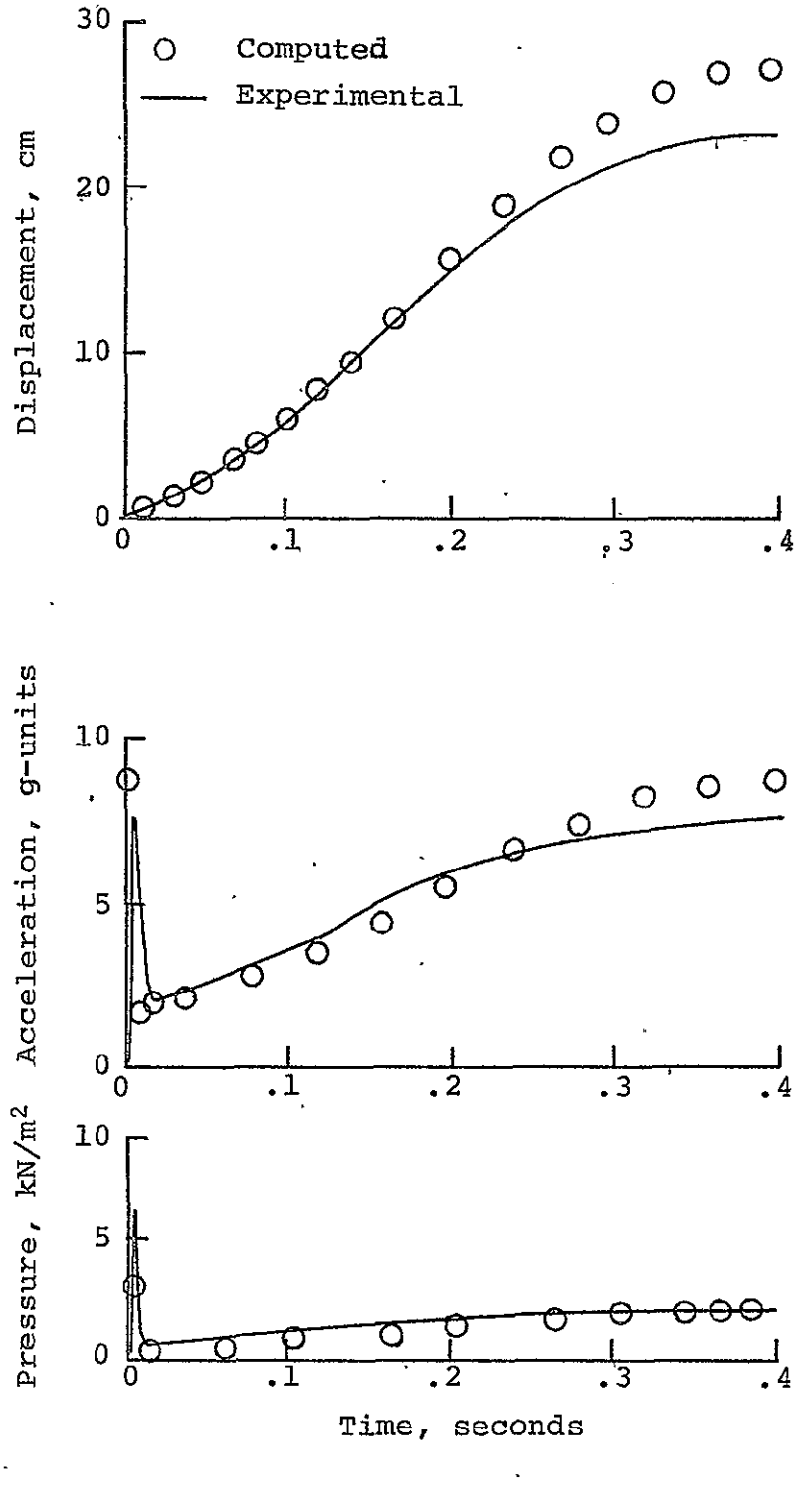
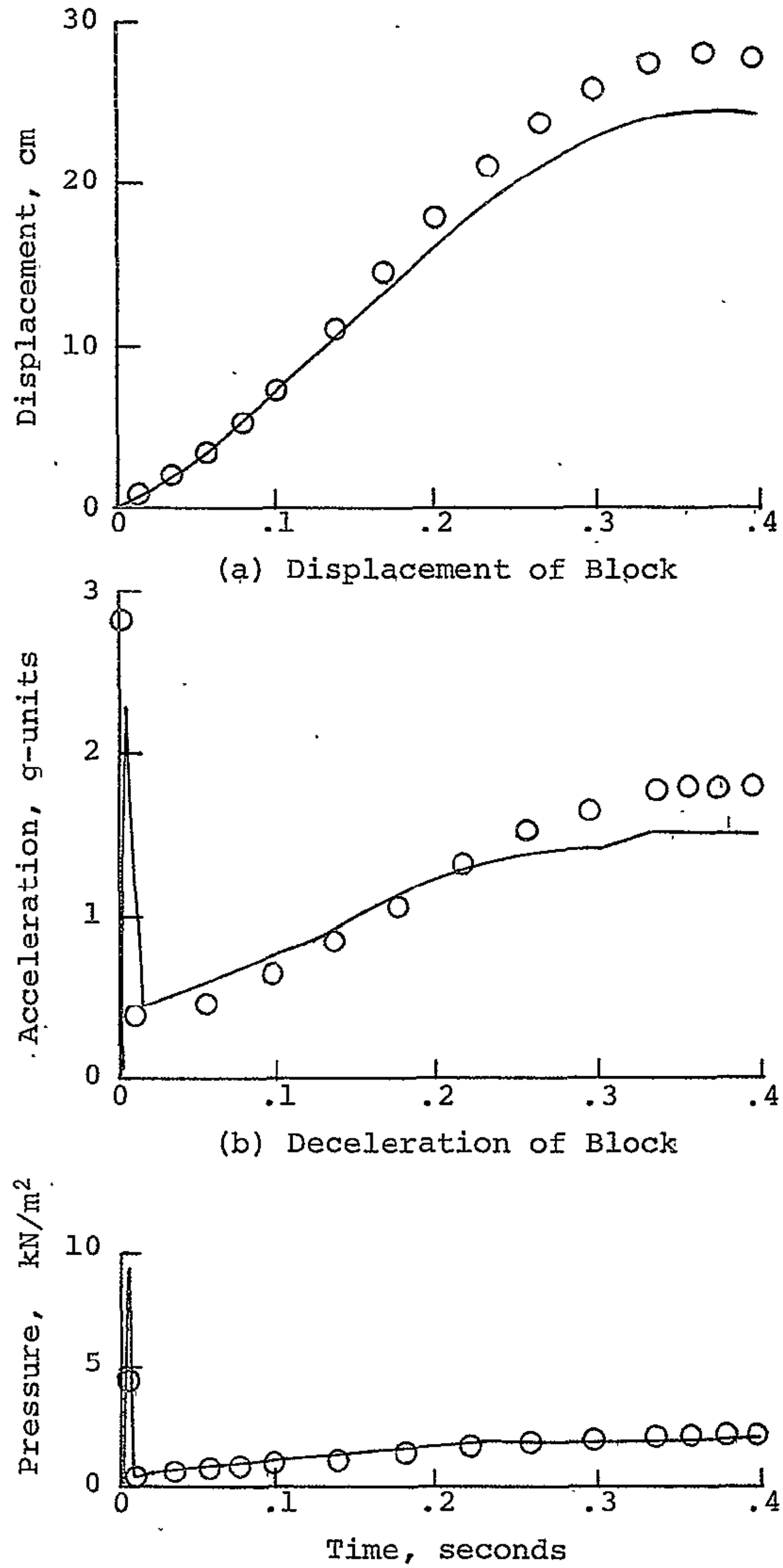


Figure 7. Impact block.



(a) v_o = 30 cm/sec, mass of block = 102.2 gm

Figure 8. Comparison of computational and experimental time histories of rigid block impacting on still water.

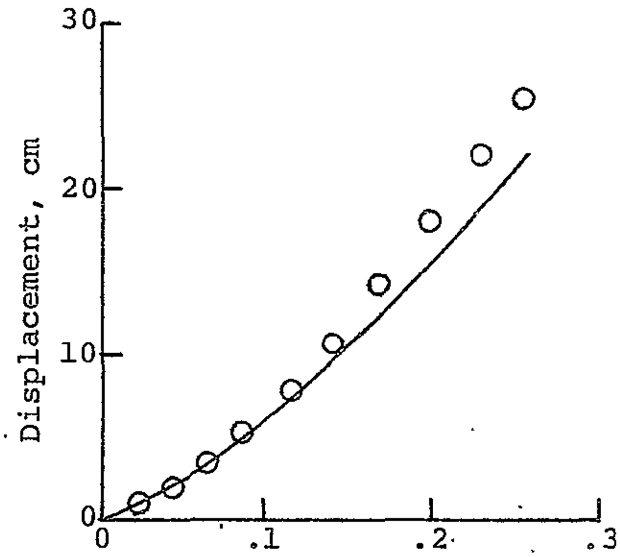


(a) Displacement of Block

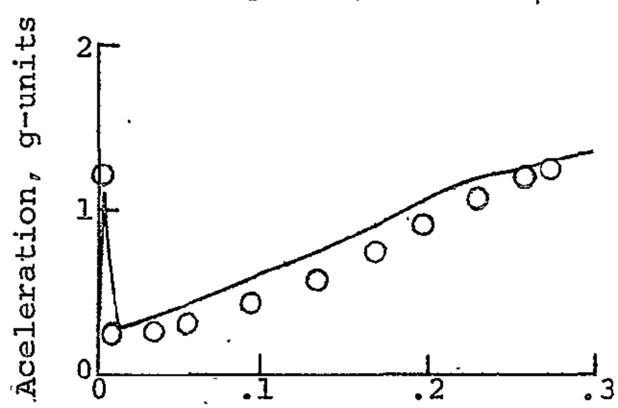
(b) Deceleration of Block

(b) v_o = 50 cm/sec, mass of block = 102.2 gm

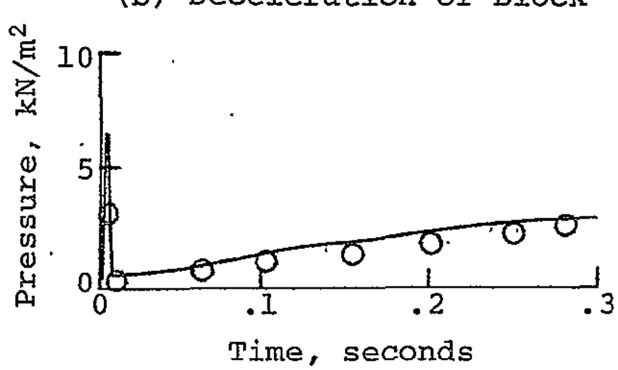
Figure 8. Continued.



(a) Displacement of Block

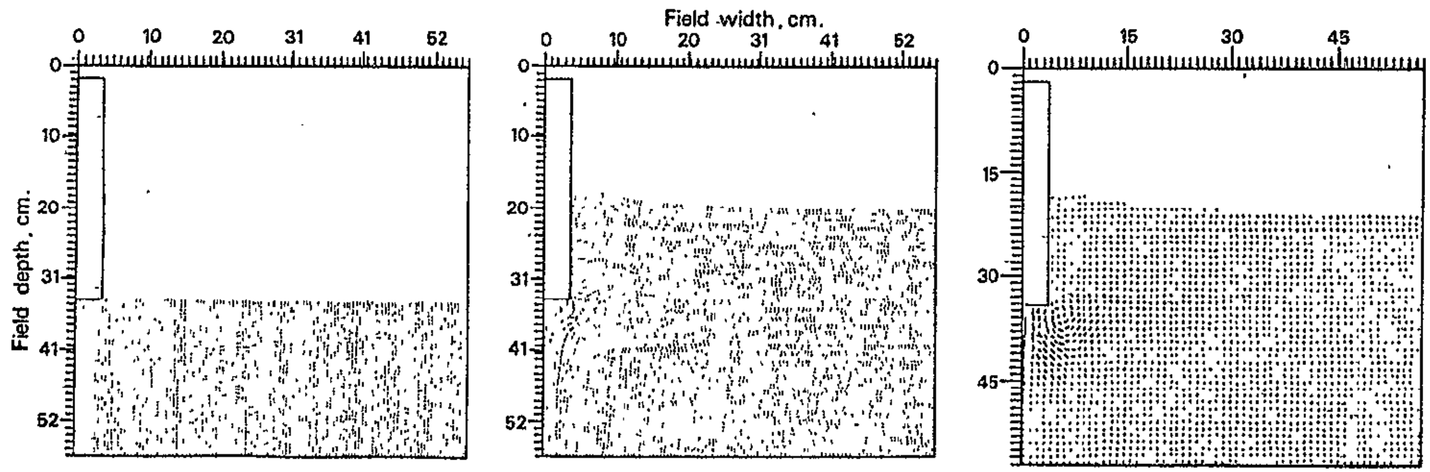


(b) Deceleration of Block



(c) v_o = 30 cm/sec, mass of block = 153 gm

Figure 8. Concluded.



(a) Particle configuration, t = 0 sec.   (b) Particle configuration, t = .155 sec.   (c) Velocity vector configuration, t = .155 sec.

Computational Parameters

$\Delta x = \Delta y = .9524$ cm.	Initial fluid depth = 24 cells (23cm)
No. of cells = 62 x 62	Width of fluid (half field) = 60cells (57cm)
V_o = 50 cm/sec.	Size of half block = 4 x 34 cells (3.8 x 32.4 cm)
Mass = 102.2 gm.	Fluid kinematic viscosity = .01cm^2/sec.

Figure 9. Fluid dynamics evolution for typical computational study.

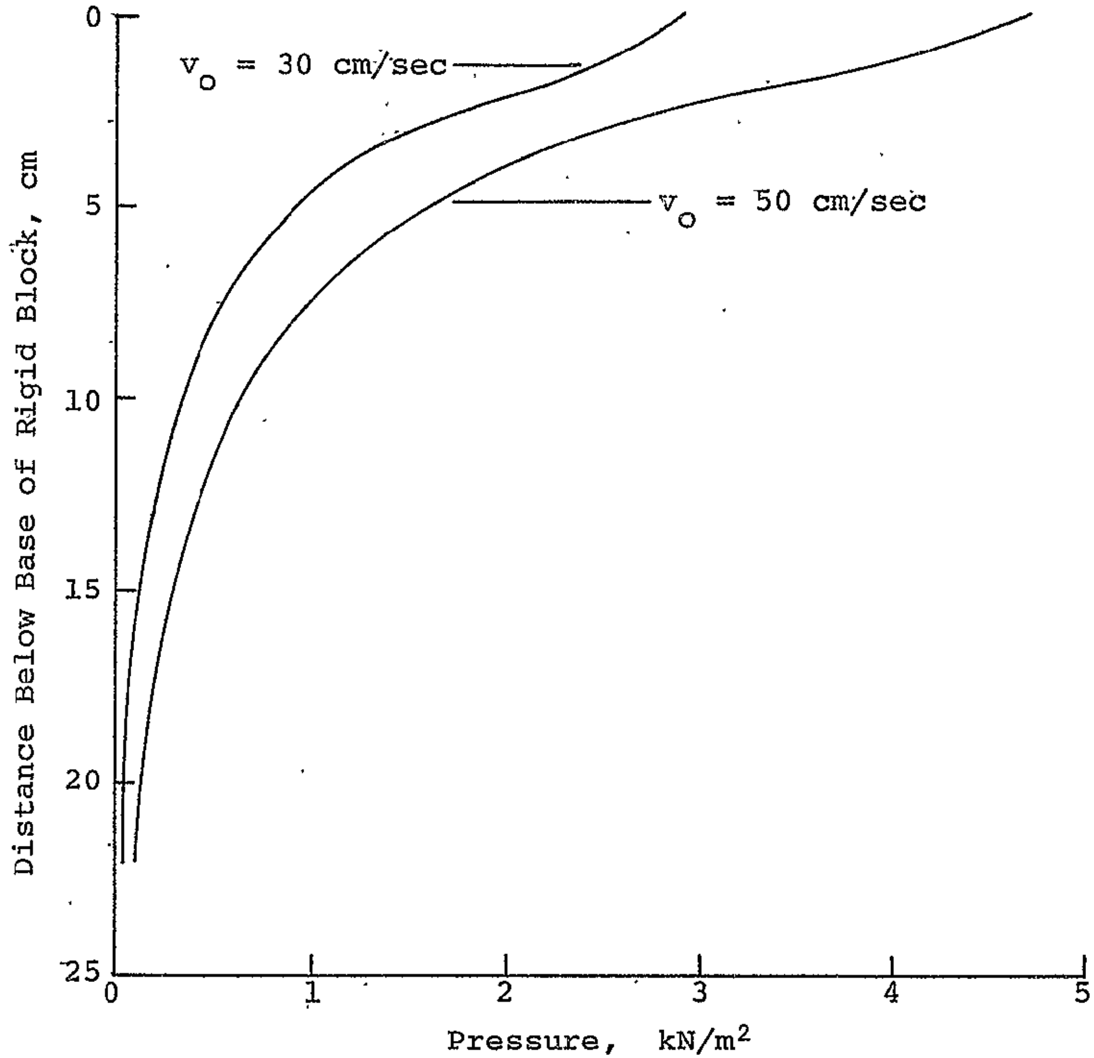


Figure 10. Computed excess hydrostatic pressure at center of base of rigid block at time $t = .003$ seconds after impact. Mass = 102.2 gm

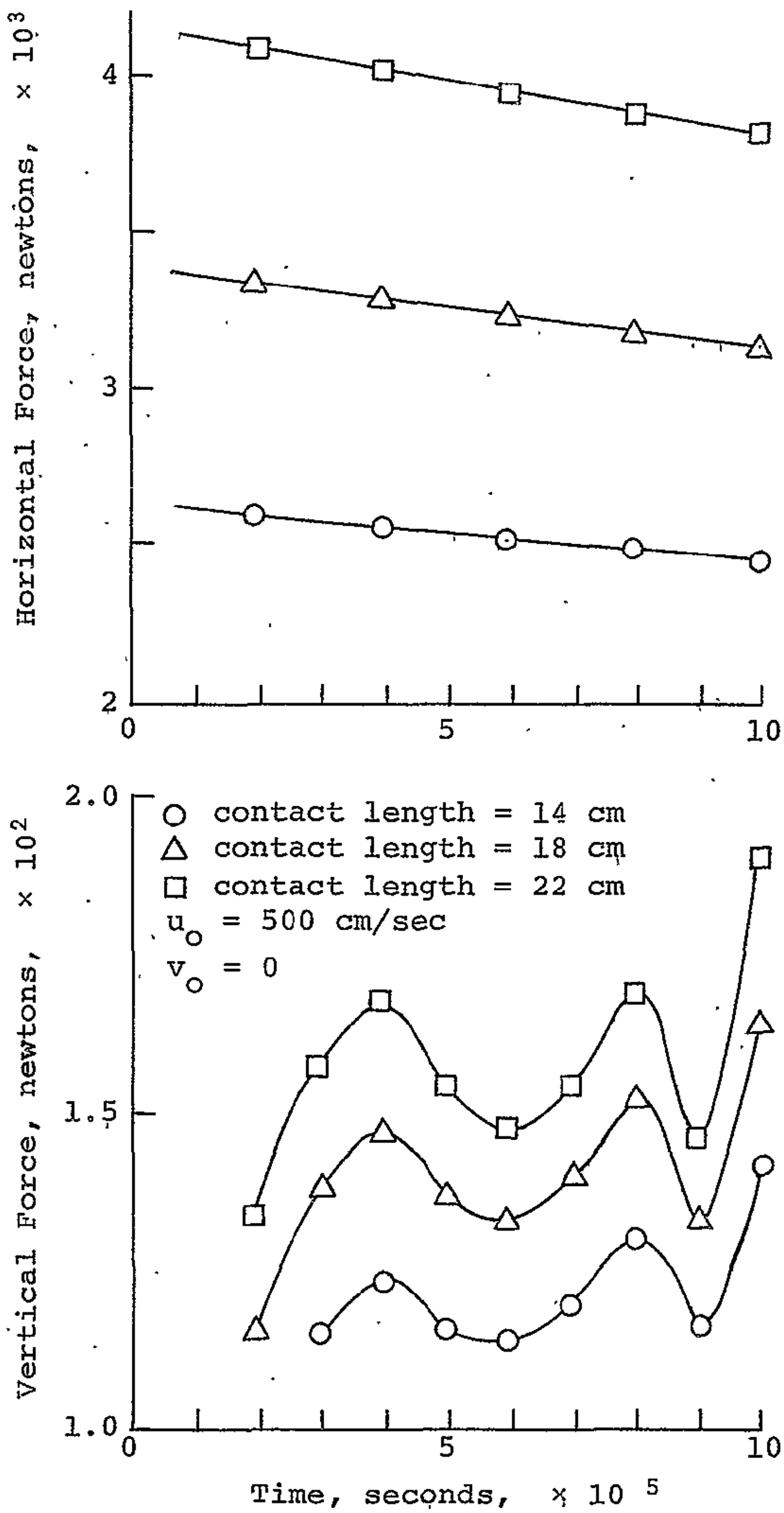


Figure 11. Time history of forces on rigid block of various contact length.

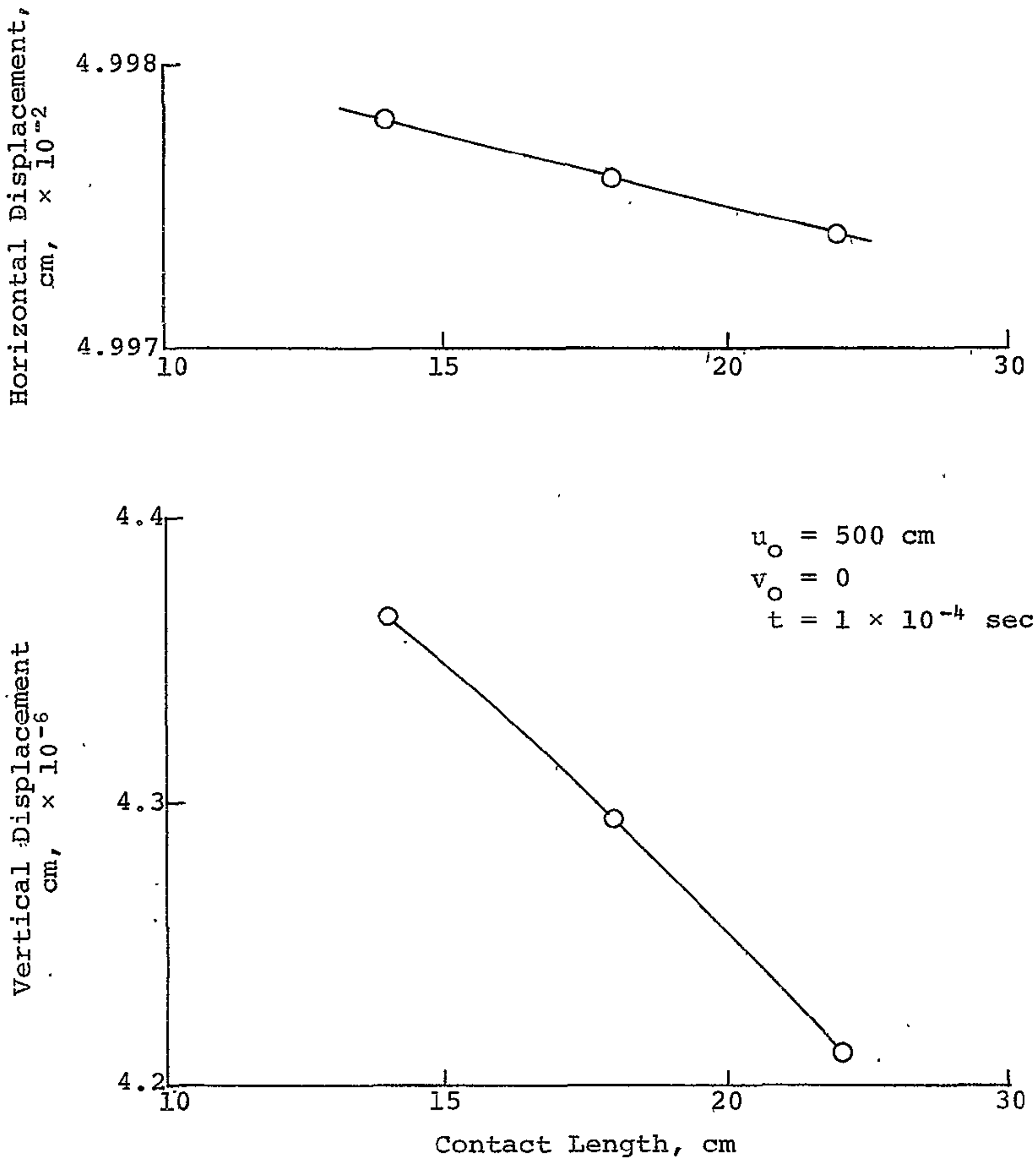


Figure 12. Effect of contact length on displacements.

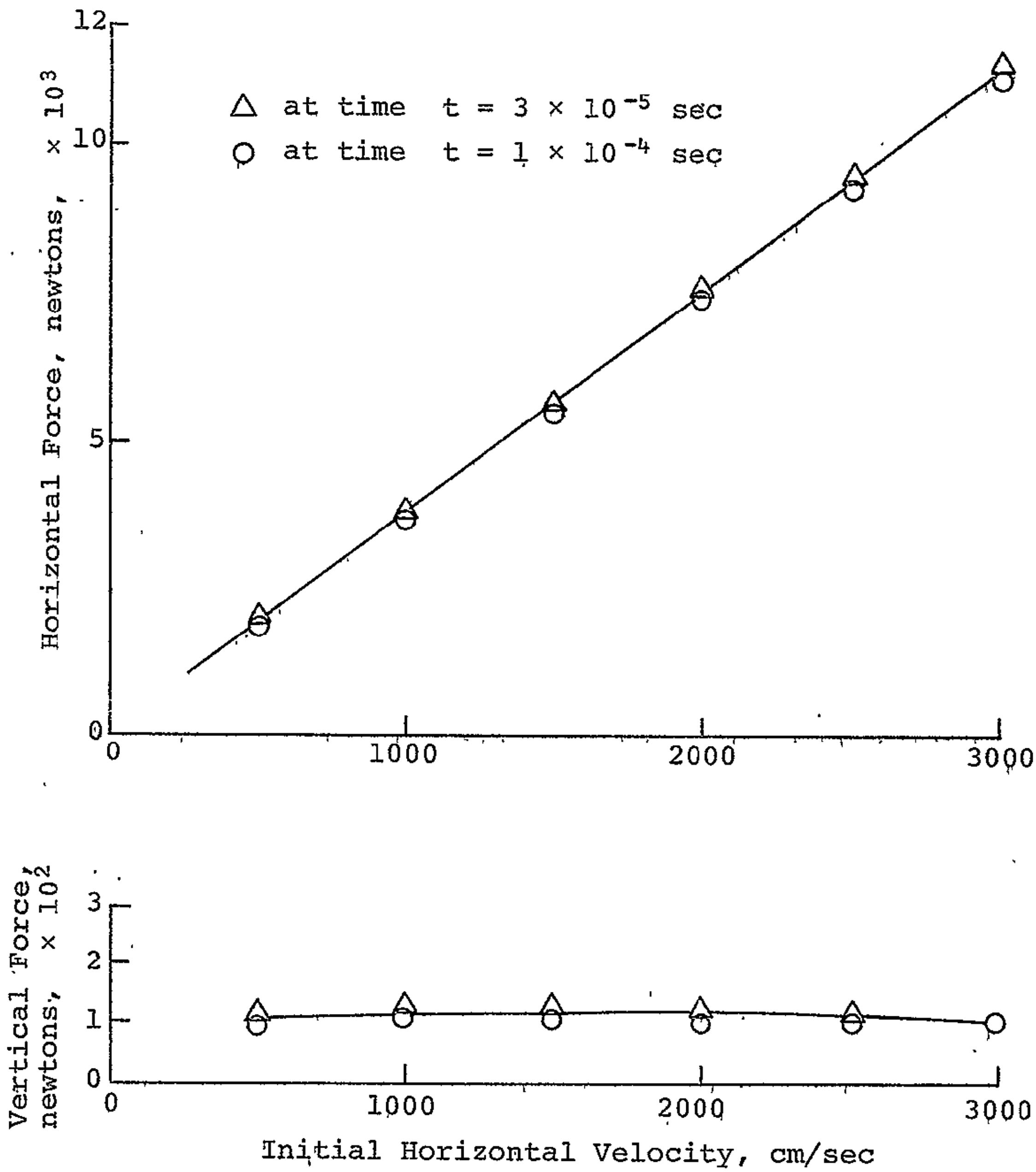


Figure 13. Effect of initial horizontal block velocity on forces.