NUREG/CR-0607
UCRL-52620

COUPLED FLUID-STRUCTURE METHOD FOR PRESSURE SUPPRESSION ANALYSIS

W. H. McMaster
D. M. Norris, Jr.
G. L. Goudreau
D. F. Quiñones
E. Y. Gong
B. Moran
N. A. Macken

Prepared for
U. S. Nuclear Regulatory Commission
by
Lawrence Livermore Laboratory

Available from
National Technical Information Service
Springfield, Virginia 22161.

NUREG/CR-0607
UCRL-52620
R4

COUPLED FLUID-STRUCTURE METHOD FOR PRESSURE SUPPRESSION ANALYSIS

W. H. McMaster
D. M. Norris, Jr.
G. L. Goudreau
D. F. Quiñones
E. Y. Gong
B. Moran

N. A. Macken
(Swarthmore College, Swarthmore, PA 19081)

Manuscript Submitted: February 1, 1979
Date Published: May 1979

Prepared for

Division of Reactor Safety Research
U.S. Nuclear Regulatory Commission
Washington, D. C. 20555
Under Interagency Agreement DOE 40-550-75
NRC FIN No. A 0116
by
Lawrence Livermore Laboratory
Livermore, CA 94550

operated by University of California

for the U.S. Department of Energy

FOREWORD

In January 1976, we began a program to improve the analysis of boiling water reactor pressure suppression systems. Our early work used the CHAMP computer code to model the fluid and the structure. This code, developed over the past four years at the Lawrence Livermore Laboratory for weapon-effects applications, had features that were applicable for blowdown suppression analysis.

Experience gained during this effort showed that this computer code performed well in simulating the blowdown suppression phenomenology, but that it was inefficient for the low Mach-number flows that characterize this behavior. The explicit Eulerian differencing used to solve the hydrodynamic equations of motion was expensive due to the near incompressibility of water at these low pressures. The detailed description of the wave motion proved to be unnecessary for our application. We replaced the CHAMP code in January 1978 with the semi-implicit incompressible PELE-IC code developed specifically for this application.

This report presents the progress made in the development and application of the PELE-IC code.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore Laboratory under Contract W-7405-Eng-48. The work was sponsored by the Nuclear Regulatory Commission, Office of Nuclear Regulatory Research and is designated Task 189:A0116-6. Robert L. Cudlin is the Nuclear Regulatory Commission Program Manager. We are pleased to acknowledge the assistance of Dr. Sang-Wook Kang of this Laboratory.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SYMBOLS

Symbols used in more than one section are defined below.

$\bar{D}$	modulus for shell theory
$\bar{\bar{D}}$	modulus for shell theory
E	Young's modulus
g	gravitational acceleration
G	shear modulus
h	shell thickness
H	water height
$\bar{K}$	effective stiffness matrix
$\bar{M}$	mass matrix
n	cycle number
p	ratio of fluid pressure to constant fluid density, $p = p/\rho$
P	pressure
R	bubble, plate or pipe radius; cell width ratio
r	radial distance
s	arc length or distance
t	time
u	x velocity or radial velocity
v	y velocity or axial velocity
x,y,z	rectangular coordinates
$\delta x, \delta y$	cell width and height
δP	pressure difference
δt	time step
θ	void fraction
ν	kinematic viscosity, Poisson's ratio
ξ	coordinate symmetry 0 = plane, 1 = cylindrical
ρ	fluid density

ABSTRACT

This report discusses an effective method of fluid-structure coupling for application in blowdown suppression analysis in boiling-water reactors. A computer program couples an incompressible, Eulerian fluid algorithm with a finite element shell algorithm. The fluid algorithm provides a trial velocity field using the Navier-Stokes equations that, at each time step, is corrected iteratively to satisfy the incompressibility condition and fluid-structure interface compatibility. The code is verified by comparing calculations of classical problems and experiments. Computer simulations of air blowdown and steam chugging in pool-suppression geometries are also presented.

INTRODUCTION AND SUMMARY

The pressure suppression system of a boiling-water reactor condenses steam flowing into the containment vessel during a loss-of-coolant accident and during normal overpressure excursions. Steam condensation is required to limit overpressure rise in the secondary containment structure. Three designs are available in the United States. The system now in operation is the Mark I design shown in Fig. 1.

The essential component of the Mark I design is a toroidal shell partially filled with water. During an accident, air and then steam are forced down large pipes from the secondary containment dry-well structure and into the water. The initial transient causes large impulsive loads to be transmitted through the water to the toroidal shell structure. Later, pulsating loads act on the downcomer pipe, ringheader, and shell structure when large steam bubbles condense and collapse. The sudden rise of the suppression pool surface generates uploads in addition to downloads caused by the early air-steam impulse.

This report describes a method for calculating the loads and the structural response resulting from blowdown suppression. The method uses a computer code that couples an incompressible Eulerian fluid algorithm with a finite element shell algorithm. The fluid physics is based on the SOLA algorithm.[1] It provides a trial velocity field using the Navier-Stokes equations that is corrected iteratively, at each time step, to satisfy the incompressibility condition and fluid-structure interface compatibility. The air blowdown loads are obtained by pressure prescriptions applied in the downcomer pipes. The oscillatory steam loads are described by simple condensation models and by upper-bound pressure excursions in the steam bubble taken from experiments.

These fluid and fluid-structure coupling algorithms are verified with calculations of solved problems from the classical literature and from recent air and steam blowdown experiments. Some simple models of oscillatory steam

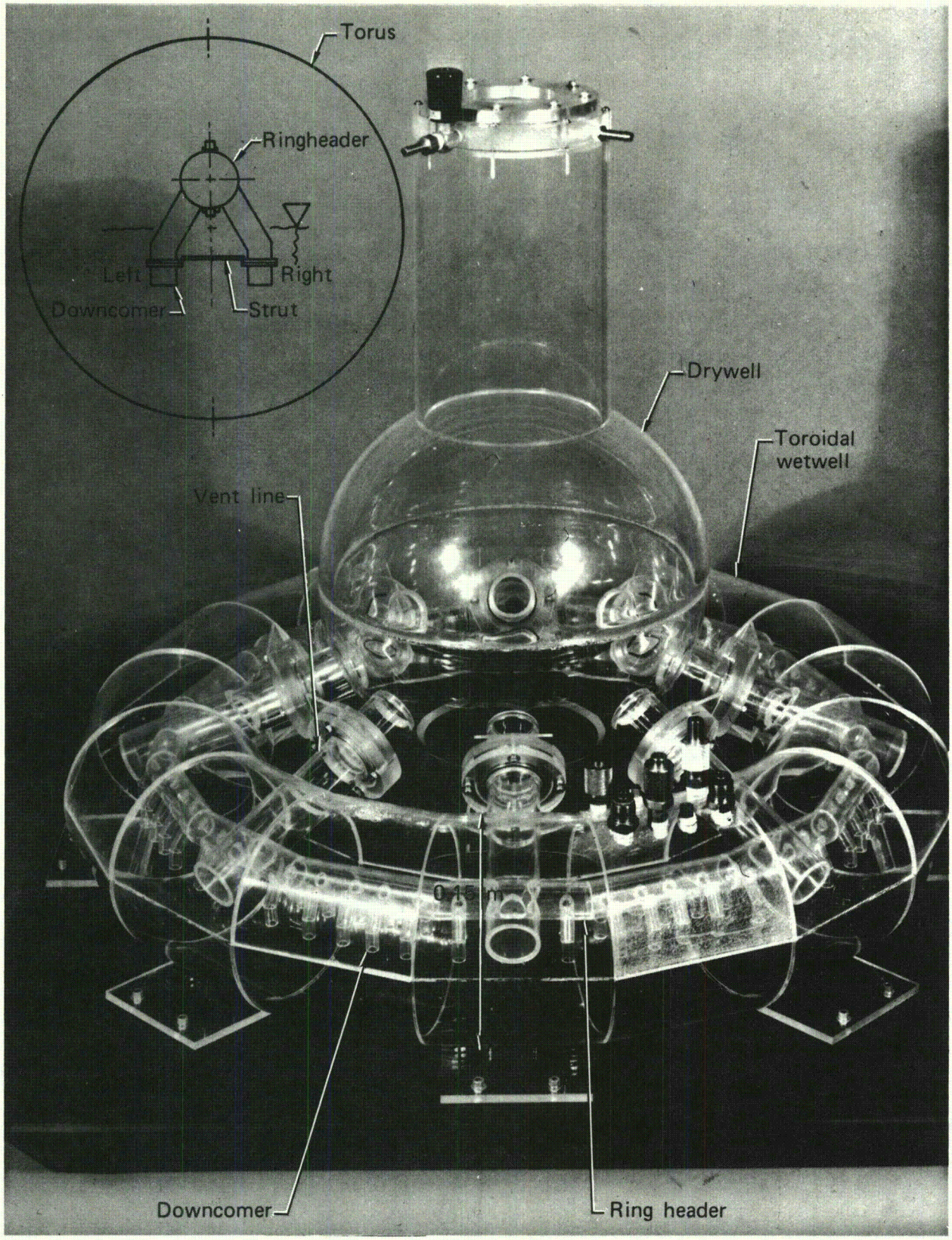


FIG. 1. Scale model (1/64 size) of the Peachbottom Mark I pressure suppression system. The insert is a diagram of the cross section.

condensation chugging are presented and used to calculate structural response. The development of these new algorithms for fluid-structure coupling will allow us to further the analysis of blowdown suppression by the calculation of the coupled loads on containment structures.

The results of the earlier CHAMP work are given in Edwards et al.[2,3]

PELE-IC COMPUTER CODE

INTRODUCTION

The PELE-IC code is a two-dimensional semi-implicit Eulerian hydrodynamic code for the analysis of incompressible flow coupled to flexible structures. The code is written in both plane and cylindrical coordinates and the coupling algorithm is general enough to handle a wide variety of structural shapes.

The basic semi-implicit solution algorithm contained in the SOLA code[1] was used as a foundation for the development of the PELE-IC code. The SOLA code is a simplified version of the earlier MAC code,[4] which uses marker particles to define a free surface. In the PELE-IC code, we use void fractions and interface orientation to track the movement of free surfaces. This gives us great versatility in following fluid-gas interfaces for both bubble definition and water surface motion.

The structural motion is computed by a finite element shell code[5] from the pressure applied at the fluid-structure interface. The fluid code then uses the position and velocity of the structure as boundary conditions. The normal velocities of the fluid at the interface and the structure are corrected iteratively along with the pressure until agreement is obtained.

A simplified theory of condensation on a free surface due to Kowalchuk and Sonin[6] has been included to model steam condensation and oscillatory chugging. Other models can readily be tested in the PELE-IC code.

Typical problem run times are a few minutes on a Control Data Corporation 7600 computer and are such that most problems can be run in a time-sharing environment.

GENERAL DESCRIPTION OF THE SOLUTION ALGORITHMS

The underlying approach used by PELE-IC for the solution of general flow fields is the use of the semi-implicit SOLA solution algorithm. The basic assumption of this approach is that all flow within the computational grid is incompressible. Superimposed on this basic algorithm we have applied the boundary conditions for free surfaces, confined compressible gases, and rigid and flexible moving structures.

This section describes each algorithm. Details are given in Appendix A.

SOLA Algorithm

The differential equations to be solved are those for the conservation of momentum and mass. The equations are written for both plane and cylindrical (axisymmetric) geometries through the use of a symmetry variable ξ. For plane geometry $\xi = 0$ and for cylindrical geometries $\xi = 1$, and x becomes the radial coordinate and y the axial coordinate.

Conservation of Momentum. The Navier-Stokes equations are used to compute the velocity field.

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\xi u^2}{x} = - \frac{\partial p}{\partial x} + \nu\left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \xi\left(\frac{1}{x}\frac{\partial u}{\partial x} - \frac{u}{x^2}\right)\right]$$

$$\frac{\partial v}{\partial t} + \frac{\partial v^2}{\partial y} + \frac{\partial vu}{\partial x} + \xi\,\frac{uv}{x} = - \frac{\partial p}{\partial y} + g + \nu\left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\xi}{x}\frac{\partial v}{\partial x}\right]. \qquad (1)$$

In the PELE-IC code we use the Hirt et al.[1] finite difference method, which preserves the conservative property of these equations. This difference method allows a variable amount of upstream differencing.

Conservation of Mass. The incompressible assumption requires that the divergence of the velocity field be zero.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \xi \frac{u}{x} = 0 \quad . \tag{2}$$

The velocity components (u,v) are in the coordinate directions (x,y) and $p = P/\rho$ is the ratio of the pressure to the constant density of the incompressible fluid. The body acceleration is given by g and the kinematic viscosity is specified by the constant ν. These equations require pressure and velocity boundary conditions at free surfaces and structural interfaces. Due to the velocity coupling technique used, we are not required to set explicit boundary conditions on pressure at structural interfaces. The capability of the code to solve a wide variety of structural problems depends on the proper specification of these boundary conditions.

The solution strategy is to first solve the Navier-Stokes equations explicitly using values from the previous time step with all boundary conditions applied. This yields trial values for the flow field $(\tilde{u},\tilde{v})$, where the tilde denotes that these velocities are tentative. Since these velocities do not necessarily satisfy the continuity equation, we iterate on the pressure field until the incompressibility condition for each computational cell is satisfied within prescribed limits. This is done by adjusting the pressure as described in Appendixes A and B. During the iteration, the boundary conditions imposed by free surfaces and structures are applied, and the iteration is continued until both the incompressibility and boundary conditions are satisfied. This then yields new values for the field variables (p,u,v) at the new time level $t + \delta t$. These new values are then used to locate all free surfaces and move any marker particles. This solution method is given in the simplified flow diagram of Fig. 2.

Free-Surface Algorithm

The geometry considered by the free-surface algorithm is shown in Fig. 3 where we have shown an arbitrarily oriented interface.

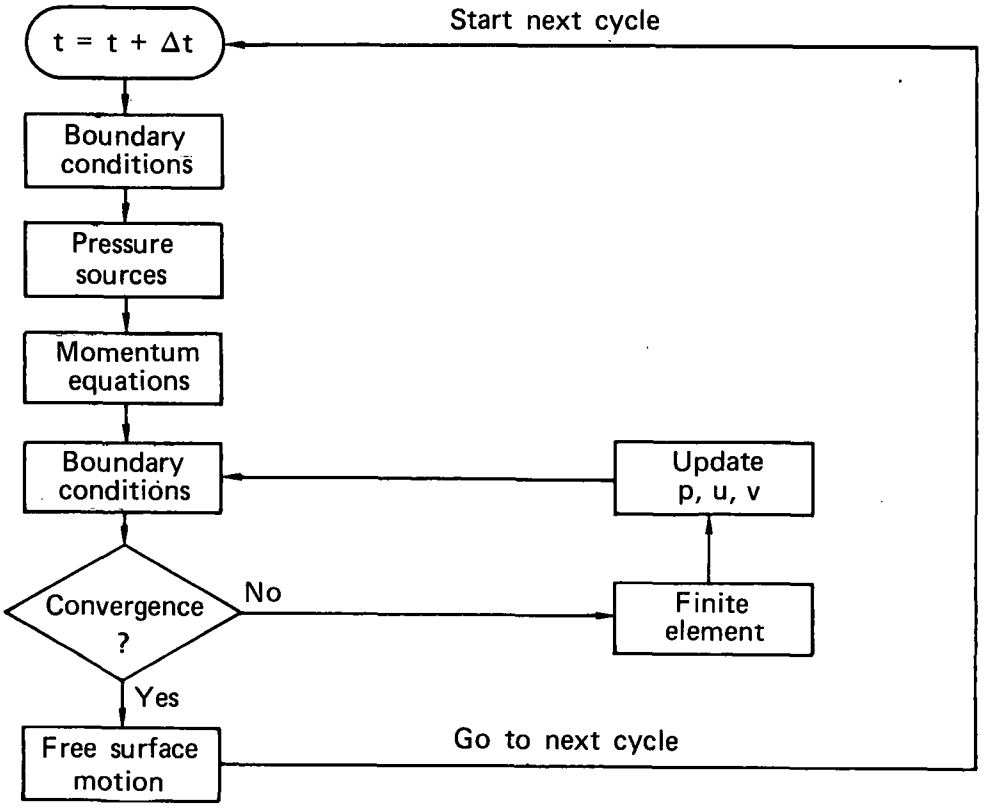


FIG. 2. PELE-IC flow diagram for the solution of incompressible fluid flow coupled to a finite element shell code.

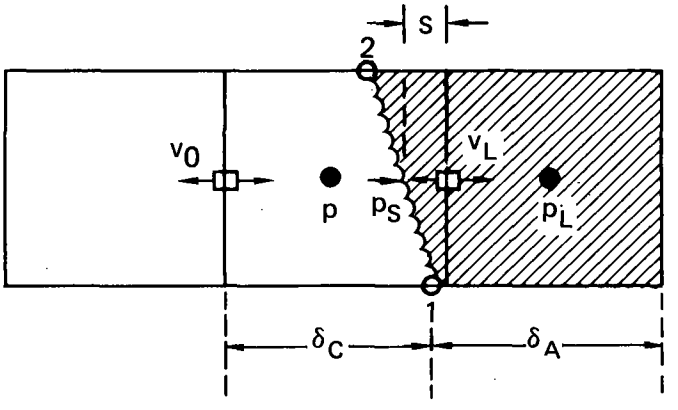


FIG. 3. Geometry for the free surface algorithm. The shaded area is the water.

The free-surface algorithm performs the following four functions:

1. Pressure: The cell-centered pressure, p, used in the equations of motion, is determined from a linear interpolation or extrapolation based on the prescribed boundary surface pressure, p_s, and the current fluid pressure, p_ℓ, in the adjacent liquid cell. The surface boundary pressure can either be a constant or a prescribed function of time. Thus,

$$p = p_\ell + (p_s - p_\ell)\ (\delta a + \delta c)/(\delta a + 2S\delta c)\ , \tag{3}$$

where the cell widths (see Fig. 3) are given by δa and δc, and the surface location is given by $S = 1 - \theta$, with θ being the void fraction of the cell.

2. Surface orientation: The surface orientation is specified by its intercepts on two sides of the cell as illustrated by points 1 and 2 in Fig. 3. These points are determined by the void fractions of the cell and its four nearest neighbors. Within the cell, the interface is considered to be a straight line segment.

3. Fluid advection: The fluid is advected across cell boundaries using the donor cell method where the amount of liquid advected is determined from the contents of the upstream cell. The amount of donor cell advection is determined from the fluid velocity and the orientation of the fluid-void interface. This method helps to provide a more accurate tracking of the free surface without the requirement of using marker particles to define the surface.

4. Cell incompressibility: The incompressibility of the cell is maintained by setting the void side velocity v_0 such that $\nabla \cdot \bar{v} = 0$. This velocity is not used in the calculation but is necessary to maintain continuity in the flow field when the surface motion causes the interface to cross grid lines.

Structure Coupling Algorithm

The geometry considered by the structure coupling algorithm is shown in Fig. 4. This algorithm couples the fluid motion to the structure motion by requiring that the normal velocities be equal at the interface and that the computational cell remain divergentless. This is accomplished in the iteration process by the adjustment of the pressure field.

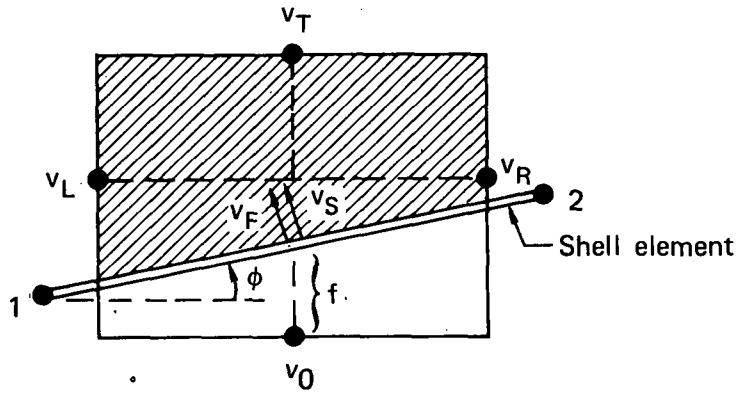


FIG. 4. Geometry for the fluid-structure coupling algorithm.

The cell velocity outside the fluid, v_0, is set such that the interpolated fluid normal velocity, v_f, equals that of the structure, v_s. The interpolated normal fluid velocity is determined from the four velocity components (Fig. 4):

$$v_f = \cos \phi \, (- \bar{v} \tan \phi + v_i) \quad ,$$

where

$$\bar{v} = 1/2 \, (v_L + v_R) \quad , \quad v_i = f v_T + (1 - f) \, v_0 \, , \tag{4}$$

where f gives the location of the structure and ϕ is its orientation angle. The usual SOLA iteration method is then used to adjust the pressure field such that the computational cell remains divergentless within prescribed limits. The iteration process then assures that both conditions are satisfied.

This algorithm applies equally well to both rigid and moving surfaces. For rigid surfaces, the normal fluid velocity is set to zero. For moving surfaces, the finite element module uses the pressure field supplied by the fluid and provides the fluid code with the resultant position and velocity of the interface. This results in an altered pressure field to maintain incompressibility, which will consequently affect the moving structure. The iteration is continued until the normal velocities of the fluid and structure match at the interface. Within a single iteration, all Eulerian zones are adjusted one by one, and then all the shell nodes are simultaneously adjusted by the implicit step.

STRUCTURAL ALGORITHM

Introduction

This section describes the structural module for a shell-of-revolution or cylinder coupled to the fluid code discussed in the preceding section. The module implicitly advances the dynamic shell response one time step using the old interface geometry and nodal velocities and the current pressure distribution. It returns the new interface positions and velocities. We apply the module in the SOLA algorithm iteration loop. In addition to the requirement that the incompressibility condition be satisfied in the iteration, we also need normal velocity compatibility between the fluid and the structure.

We used the thin shell equations originally derived by Love but modified to include deformation due to transverse shear.[7] These equations are discretized spatially by use of the finite element method and temporally by the Newmark implicit time integration scheme.

The element formulation is similar to that described by Hughes and Taylor[8] for plates and extended to a shell by Goudreau[5] and Zienkiewicz et al.[9] The element is a two-node, conical frustrum with three degrees-of-freedom per node. Element shape functions are piecewise linear for displacements and rotations. The shear "locking" associated with low-order functions is removed by the use of one-point quadrature. Large deformation is accounted for by updating the initial coordinates and reformulating the stiffness matrix every time step.

Dynamic Algorithm

This section describes the generalized small displacements of the structure by the vector q measured from an equilibrium state. By the use of Lagrange's equations, it may be shown that the dynamic behavior of the structure is described by

$$\underset{\sim}{M}\underset{\sim}{q} + \underset{\sim}{C}\underset{\sim}{q} + \underset{\sim}{K}\underset{\sim}{q} = \underset{\sim}{P}(t) \quad , \tag{5}$$

where M is the mass matrix, C is the damping matrix, K is the stiffness matrix, $\underset{\sim}{q}$ is a vector of generalized displacements, and $\underset{\sim}{P}$ is the fluid and other loads on the structure.

The Newmark implicit-integration scheme (Goudreau and Taylor[10]) gives

$$\underset{\sim}{q}^{n+1} = \underset{\sim}{q}^{n} + \dot{\underset{\sim}{q}}^{n}\delta t + 1/4 \left(\ddot{\underset{\sim}{q}}^{n} + \ddot{\underset{\sim}{q}}^{n+1}\right)\delta t^2$$

and

$$\dot{\underset{\sim}{q}}^{n+1} = \underset{\sim}{q}^{n} + 1/2 \left(\ddot{\underset{\sim}{q}}^{n} + \ddot{\underset{\sim}{q}}^{n+1}\right)\delta t \quad . \tag{6}$$

If

$$\underset{\sim}{A}^{n} = \underset{\sim}{q}^{n} + \dot{\underset{\sim}{q}}^{n}\delta t + 1/4 \ \ddot{\underset{\sim}{q}}^{n}\delta t^2 \tag{7}$$

and

$$\ddot{\underset{\sim}{q}}^{n+1} = \frac{4}{\delta t^2}\left(\underset{\sim}{q}^{n+1} - \underset{\sim}{A}^{n}\right) \quad , \tag{8}$$

then

$$\left(K + \frac{4}{\delta t^2} M\right)\underset{\sim}{q}^{n+1} = \underset{\sim}{P}^{n+1} - \frac{4}{\delta t^2} M\underset{\sim}{A}^{n} \tag{9}$$

provides the computational algorithm

$$\bar{K}\underset{\sim}{q}^{n+1} = \bar{\underset{\sim}{P}}^{n+1} \quad . \tag{10}$$

$\bar{K}$ and $\bar{\bar{P}}$ are called the effective stiffness matrix and the effective load vector, respectively. For linear problems, $\bar{K}$ has to be formed and factored only when δt changes. This method is unconditionally stable for linear systems and permits time steps large in comparison to those set by stability requirements of explicit schemes.

In the above equations, $\underset{\sim}{q}$ is the total displacement from the initial state. The theory is valid only for displacements less than half of the shell thickness. In applications that yielded larger displacements, it was necessary to modify the theory. At each time step, prior to the SOLA iteration loop, we updated the initial coordinates and recalculated the effective stiffness. This gives an incremental small displacement formulation that, although not exact, provides excellent accuracy for displacements up to three shell thicknesses. We did not update $\bar{K}$ in the iteration loop, but $\bar{P}$ changes with the pressure during the interations.

Finite-Element Discretization

We derive the effective stiffness [Eq. (10)] of a thin shell of revolution. The discretization uses piecewise conical frustra with three degrees-of-freedom per node. Here* u and w are tangential and normal displacements in the middle surface and β is the rotation (Fig. 5). The total displacements are $U = u + \zeta\beta$, $V = 0$, and $W = w + \zeta\beta$, and ζ is a distance measured normal to the middle surface of the shell. The superscript 0 refers to middle-surface quantities. No superscript is shown for middle-surface displacements. The curvatures are κ_s and κ_θ.

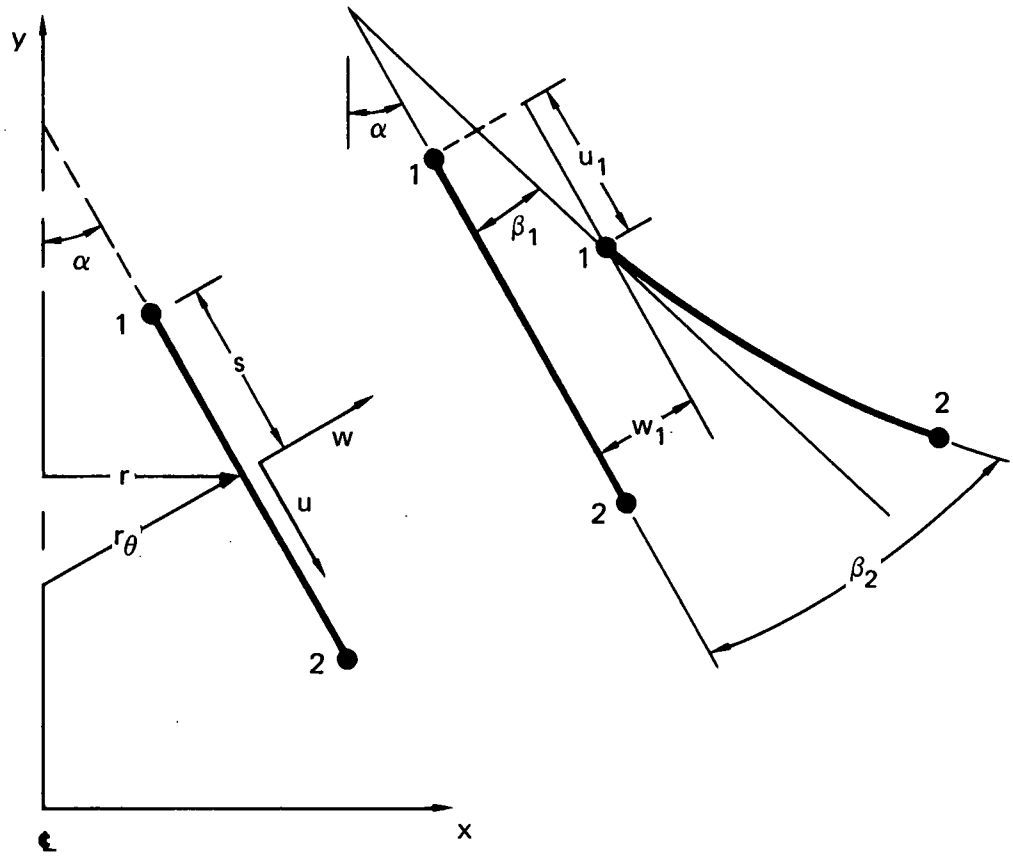


FIG. 5. Coordinate definition for the conical frustrum shell element. Element nodal points are marked 1 and 2.

*Note here that u and w are displacements and not velocity components.

The strains are

$$\begin{aligned}
\varepsilon_s &= \varepsilon_s^0 + \zeta\kappa_s \\
\varepsilon_\theta &= \varepsilon_\theta^0 + \zeta\kappa_\theta \\
\gamma_{sn} &= \mu_s^0 \quad ,
\end{aligned} \tag{11}$$

where

$$\begin{aligned}
\varepsilon_s^0 &= \frac{du}{ds} \\
\varepsilon_\theta^0 &= \frac{u}{r}\frac{dr}{ds} + \frac{\omega}{r_\theta} = \frac{1}{r}\,(u \sin\alpha + w\cos\alpha) \\
\kappa_\theta &= \frac{\beta}{r}\sin\alpha \\
\kappa_s^0 &= \frac{d\beta}{ds} \\
\mu_s^0 &= \frac{d\omega}{ds} + \beta \quad .
\end{aligned} \tag{12}$$

Defining

$$\bar{D} \equiv \frac{Eh}{1 - \nu^2} \quad \text{and} \quad D \equiv \frac{\bar{D}h^2}{12} \quad ,$$

where E is Young's modulus, h is the shell thickness and ν is Poisson's ratio. The stress resultants are

$$\begin{aligned}
Q &= \frac{5}{6}\,Gh\mu_s^0 \\
N_s &= \bar{D}\left(\varepsilon_s^0 + \nu\varepsilon_\theta^0\right) \\
N_\theta &= \bar{D}\left(\varepsilon_\theta^0 + \nu\varepsilon_s^0\right) \\
M_s &= \bar{\bar{D}}\left(\kappa_s^0 + \nu\kappa_\theta^0\right) \\
M_\theta &= \bar{\bar{D}}\left(\kappa_\theta^0 + \nu\kappa_s^0\right) .
\end{aligned} \tag{13}$$

The variational statement for static equilibrium is

$$\int_{s_1}^{s_2} \left[N_s \delta\varepsilon_s^0 + N_\theta \delta\varepsilon_\theta^0 + M_s \delta\kappa_s^0 + M_\theta \delta\kappa_\theta^0 + Q\delta\mu_s^0 - \underset{\sim}{q}\delta\omega \right] r ds = 0 \quad . \tag{14}$$

The finite element expansion is

$$u = \phi\hat{u}, \; w = \phi\hat{w}, \; \beta = \phi\hat{\beta} \quad , \tag{15}$$

where $\phi(s)$ is the piecewise-linear interpolation basis, u is the field variable, and $\hat{u}$ is a vector of element nodal point displacements. Substituting the stress resultants of Eq. (13) and isolating the three kinematic variations gives the desired element stiffness*:

$$\begin{bmatrix} A & B & 0 \\ B^T & C & D \\ 0 & D^T & E \end{bmatrix} \begin{Bmatrix} \hat{u} \\ \hat{w} \\ \hat{\beta} \end{Bmatrix} = \begin{Bmatrix} 0 \\ P \\ 0 \end{Bmatrix} \tag{16}$$

or

$$\underset{\sim}{K}\underset{\sim}{q} = \underset{\sim}{P} \quad . \tag{17}$$

The details of the calculation of A, B, C, D, and E are given in Appendix B. The interpolation function is

$$\phi(s) = \frac{1}{\Delta s} \langle s_2 - s, \; s - s_1 \rangle \quad . \tag{18}$$

Equation (17) is in local coordinates and a rotation is required before assemblage into the global stiffness matrix of Eq. (10). The details are given in Appendix B with a formulation of the lumped-mass matrix and specialization to the equations to plane geometry.

---

*Here P is not the cell pressure.

SPECIAL FEATURES

Marker Particles

Marker particles have been incorporated into the PELE-IC code to track the fluid movement for flow visualization. The method is based on the algorithm in the SMAC code.[11] We identify particles by color to determine the flow paths from different regions of the fluid. Outer and inner boundaries are used to compute the movement of particles around an obstacle.

Particle movement is determined at the end of each cycle using the fluid velocity. The velocity used to move the particle is computed from an area-weighting scheme involving the nearest cell velocities (Fig. 6). Marker particles are not used to define free surfaces.

The cell in which the particle lies is defined as the donor cell. A reference cell is further defined to determine the contribution of the surrounding fluid motion. The reference cell is the cell lying between the lower pair of u velocities when computing the u_p velocities for a given particle. Similarly, the left-most pair of v velocities is used to define the reference cell when computing the v_p velocity for a given particle. A1, A2, A3, and A4 are fractional areas of the given donor cell. Let (IR, JR) be the index for the reference cell and (ID, JD) be the index for the donor cell. The particle velocities (u_p, v_p) are computed from the following formulas:

$$u_p = A2*U(ID-1,JR+1) + A1*U(ID,JR+1) + A3*U(ID-1,JR) + A4*U(ID,JR)$$
$$v_p = A2*V(IR,JD) + A1*V(IR+1,JD) + A3*V(IR,JD-1) + A4*V(IR+1,JD-1) \quad .$$

The particle velocities (u_p, v_p) have now been computed from the neighboring fluid velocities (u,v). The next position is finally computed by

$$x^{n+1} = x^n + u_p \delta t$$
$$y^{n+1} = y^n + v_p \delta t \quad .$$

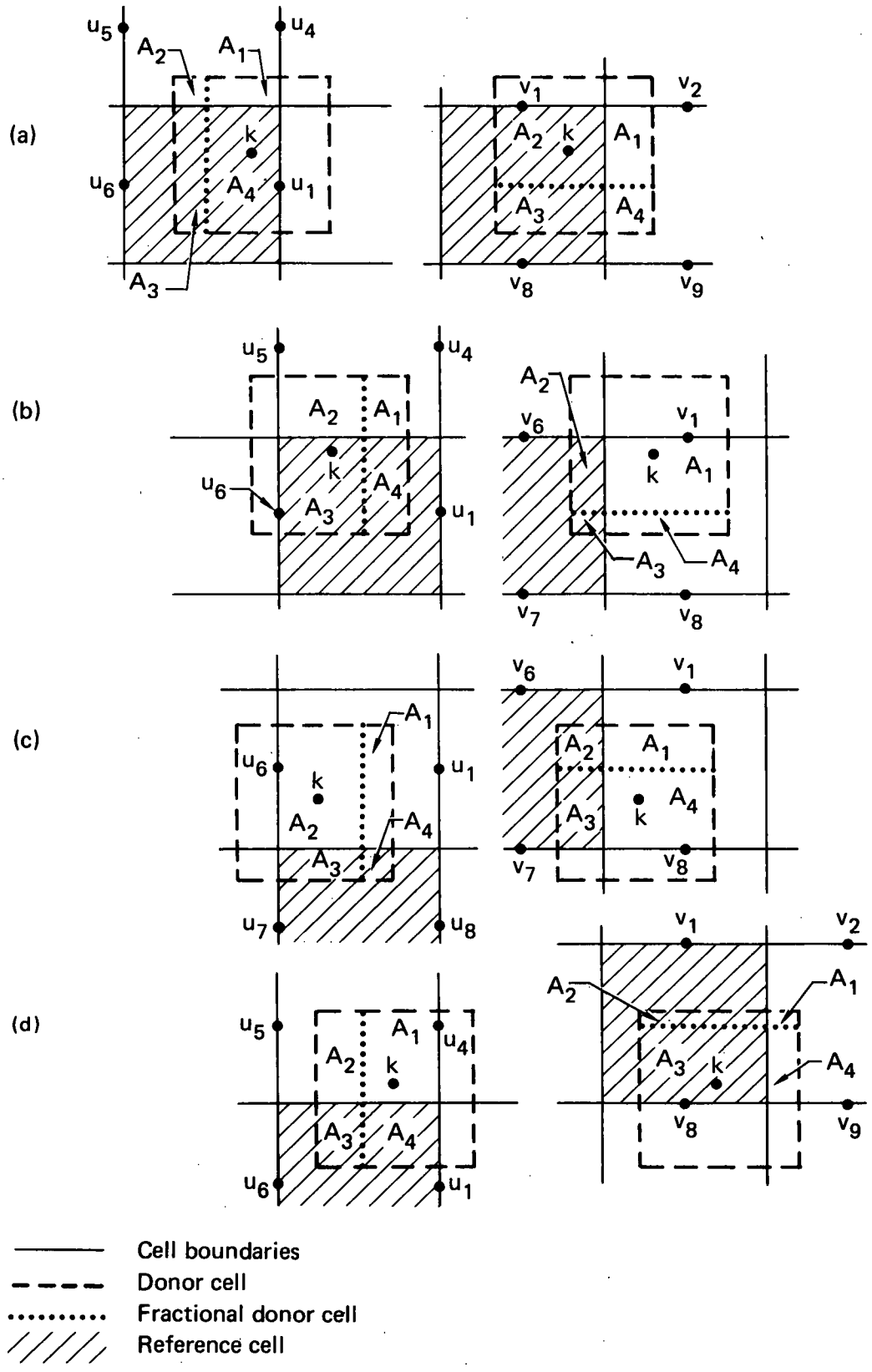


FIG. 6. Marker particle weighting scheme for a given particle (K) in each of the four quadrants of the cell. An area-velocity weighting scheme is used to compute the displacement. Each reference cell is shaded.

Figure 7 shows examples of particle motion resulting from a prescribed periodic drywell pressure.

Chugging Model

In a loss-of-coolant accident, steam entering the wet well condenses as it comes in contact with the water. The condensation reduces the pressure in the bubble forming in the wet well, allows the bubble to collapse and causes water to return up the downcomer. The process is then repeated as the continued flow of high-pressure steam from the dry well pushes the interface back into the wet well. To check the ability of PELE-IC to calculate chugging-like phenomena, the one-dimensional steam-flow relationships of Kowalchuk and Sonin[6] were incorporated into our computer code. The steam is modeled as a simple lumped control volume with vapor entering from the dry well and leaving via condensation. Work is done by pressure forces on the moving boundary; the steam mass changes with time.

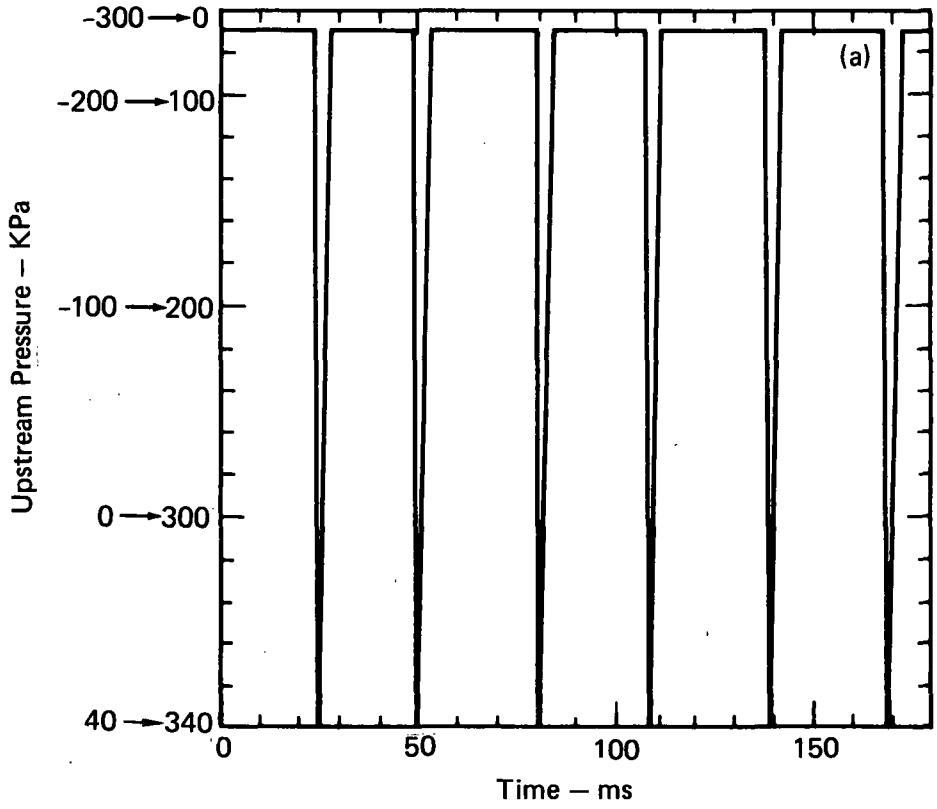


FIG. 7. Calculated marker particle location in UCLA tank and downcomer pipe. (a) Upstream pressure time history.

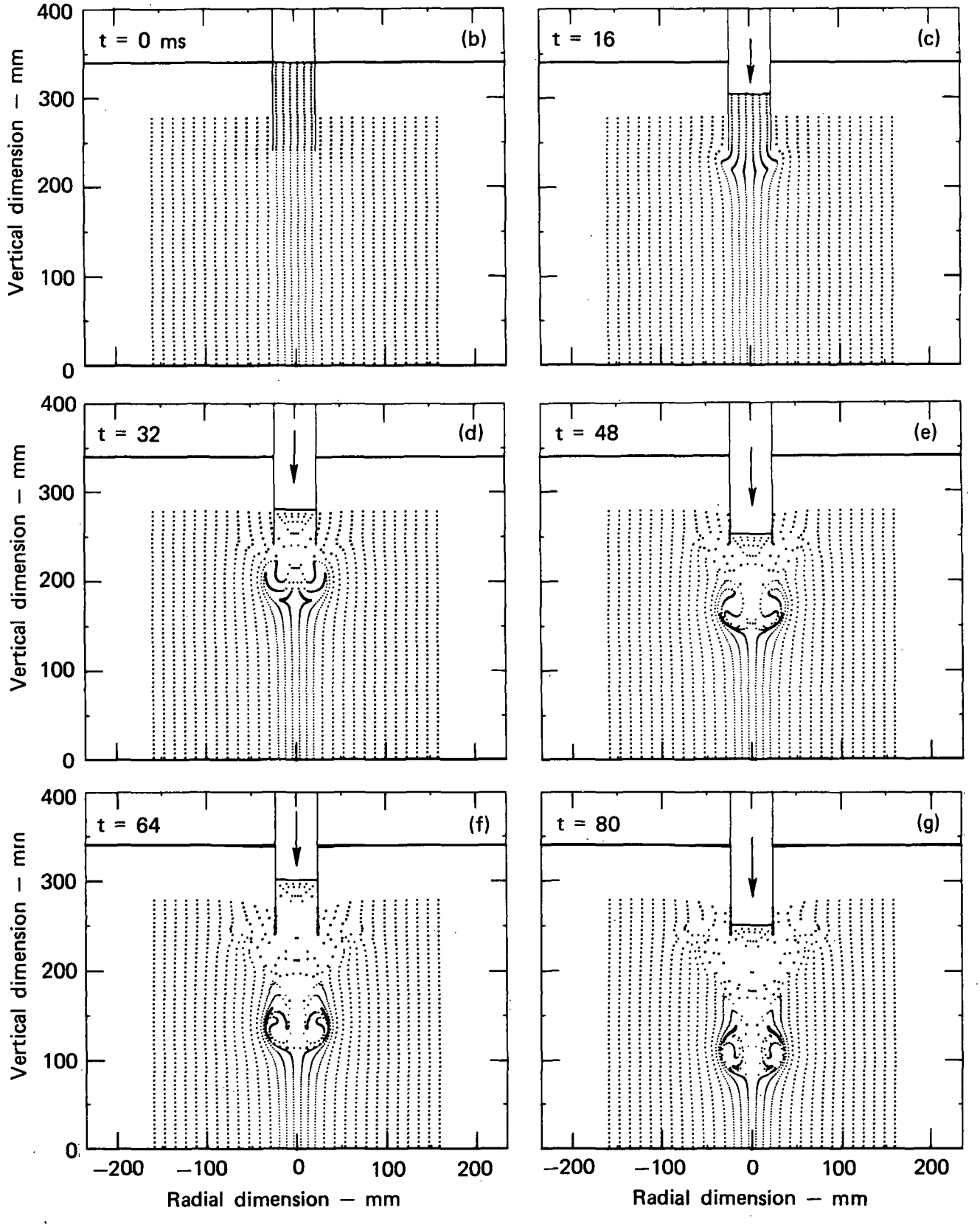


FIG. 7. Calculated marker particle location in UCLA tank and downcomer pipe. (b)-(g) Original location.

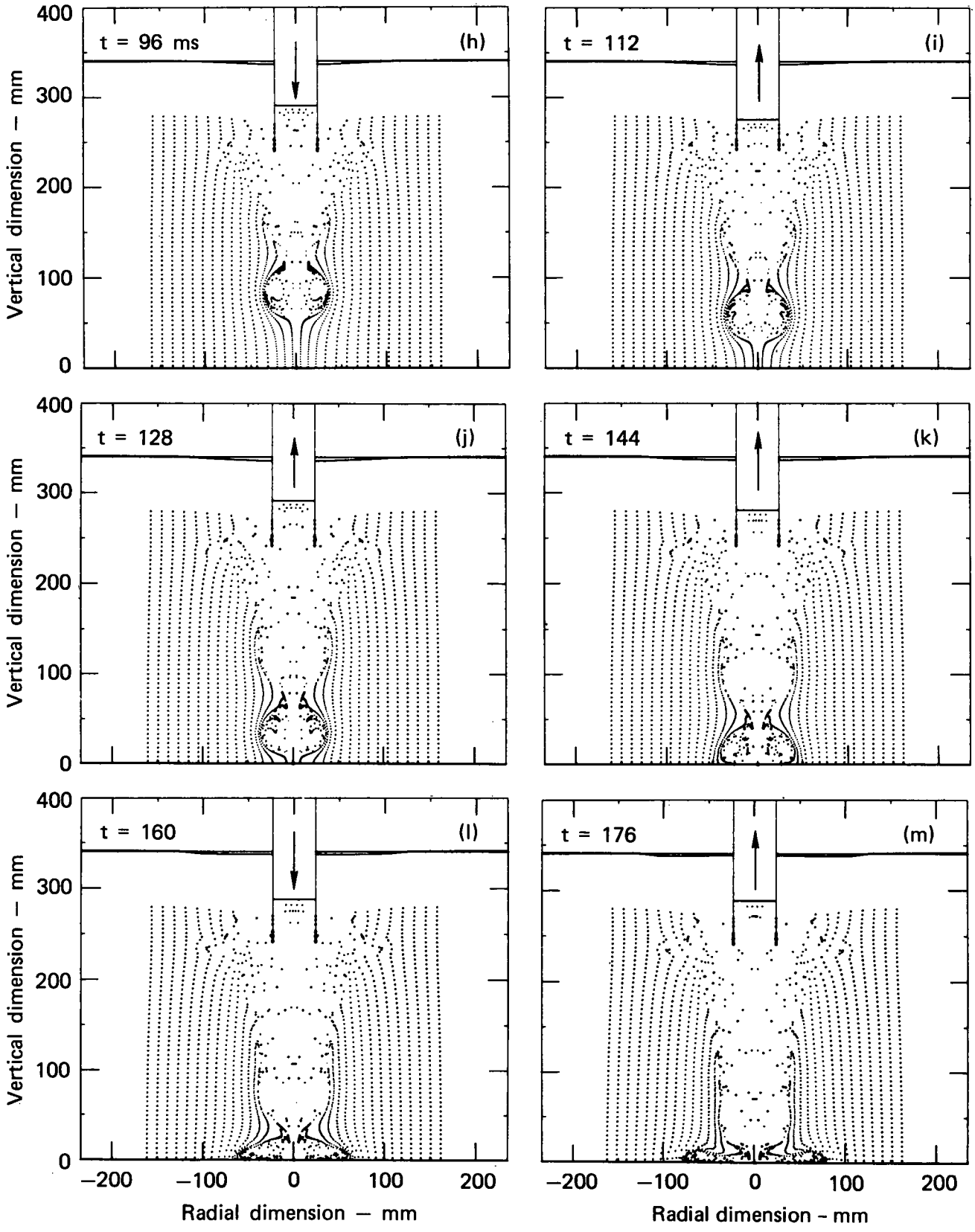


FIG. 7. Calculated marker particle location in UCLA tank and downcomer pipe. (h)-(m) Location at various times.

The condensation rate is approximated by modeling two mechanisms; the transport of steam to the surface and the rate at which latent heat is transported into the water. The equation predicting the smaller amount of condensation drives the process and this value is utilized. In the first mechanism, a thermal flux model is assumed. In the second case, an expressio is needed for the heat-transfer rate at the interface. When the fluid is in the downcomer, a simple one-dimensional heat conduction model is considered. The model uses a surface temperature equivalent to the saturation temperature and a constant fluid temperature equal to that of the wet well far from the interface. The thermal diffusivity is assumed to be proportional to the time averaged flow speed. When the fluid is outside the downcomer, the rate at which latent heat is transported into the water is proportional to a condensation Stanton number based on water properties. The heat transfer rate is also proportional to the time averaged mean flow speed and the temperature difference between the saturated steam and the wet well water. Empirical values are picked for proportionality constants in the thermal flux mechanisms, the thermal diffusivity relationship, and for Stanton numbers.

The resulting equations include the conservation of mass and energy for the steam, an orifice flow relation for steam inflow, the condensation mechanisms, and the Clapeyron equation to relate the saturation temperature to steam pressure. These are coupled via the pressure and steam volume to PELE-IC. They were coded as subroutines with the GEAR numerical integration scheme utilized in the solution of the ordinary differential equations.[12]

PROGRAM VERIFICATION

INTRODUCTION

The objective of the program verification was to check PELE-IC results against analytical solutions. Three types of verification calculations were chosen: a draining tank, an expanding spherical bubble, and a vibrating submerged plate. All test problems use the conservation equations but exercise different boundary conditions needed for applications. The calculations assumed an inviscid fluid.

Our first calculation is a tank draining through an outlet tube at the tank bottom. The pipe flow pattern is similar to that of a downcomer pipe during vent clearing following a loss-of-coolant accident. These calculations require rigid internal structures such as pipes and straight walls. The calculation checks mass conservation, general flow behavior, and the free-surface algorithm.

The expansion of a spherical bubble was calculated to test the logic of the air-water interface used to describe bubble expansion. The expanding (and contracting) bubble is similar to the bubble phenomenon at the end of a downcomer pipe accompanying blowdown.

To test the fluid-structure coupling algorithm, a series of submerged vibrating plate calculations were done. For these calculations, we used a circular, rigid vertical cylinder with a flexible bottom. We also calculated the response of a circular plate submerged in a semi-infinite fluid. Preliminary calculations of more general spherical and infinitely long cylindrical tanks in plane geometry containing water are presented.

TANK DRAINING

We calculated cylindrical tanks with vertical and inclined walls. All calculations used gravitational loads; in addition, some tanks were pressurized. The tanks were modeled with rigid walls with a valve at the end of an outlet pipe that was opened at the start of the calculation. The calculations simulated the initial transient response due to fluid acceleration and the later quasi-steady-state flow. The drain times for the unpressurized tanks were from 2 to 6 s and for the pressurized tanks less than 1 s.

A one-dimensional solution derived from control volume theory is given by the second-order differential equation

$$-\ddot{H}\left(H + L\theta + \frac{H^2 B}{R_e}\right) = \frac{\delta P}{\rho} + \frac{1}{2}\left(1 - \theta^2 + \frac{4BH}{R_e}\right)\dot{H}^2 + g(L + H) \quad , \tag{19}$$

with the initial condition for H(t)

$$H(0) = H_0 \ , \ \dot{H}(0) = 0 \quad , \tag{20}$$

where H is the water depth in the tank, $\dot{H}$ and $\ddot{H}$ are its time derivatives, L is the outlet pipe length, θ is the ratio of the water surface area to the pipe area (A/A_p), B is the inverse slope of the tank wall $(r - R_e)/H$, A is the top surface area (πr^2), A_e is the tank bottom area (πR_e^2), A_p is the pipe area $(\pi R_p)^2$, δP is the constant pressure difference, ρ is the mass density, and g is the gravitational constant. The above equation was solved numerically. Two cases were calculated: a funnel (slanted walls) where $A_p = A_e$, and a vertical-walled tank*,[13] where $A = A_e$ and $B = 0$.

All calculations used a time step of 0.3 ms with a grid of 10 by 10 mm cells. One cell was used across the pipe radius. The zoning for the funnel is shown in Fig. 8. The funnel tank walls cross many cells whereas the

*For the special case of an unpressurized vertical-walled tank without an outlet tube, $\delta P = L = B = 0$, this equation reduces to Hansen's formula for a draining tank.

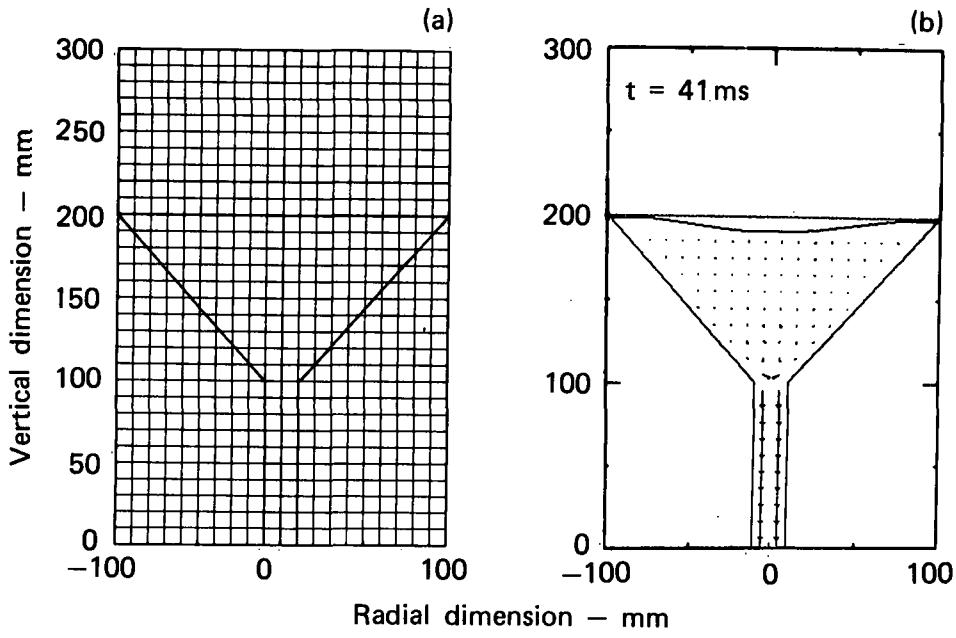
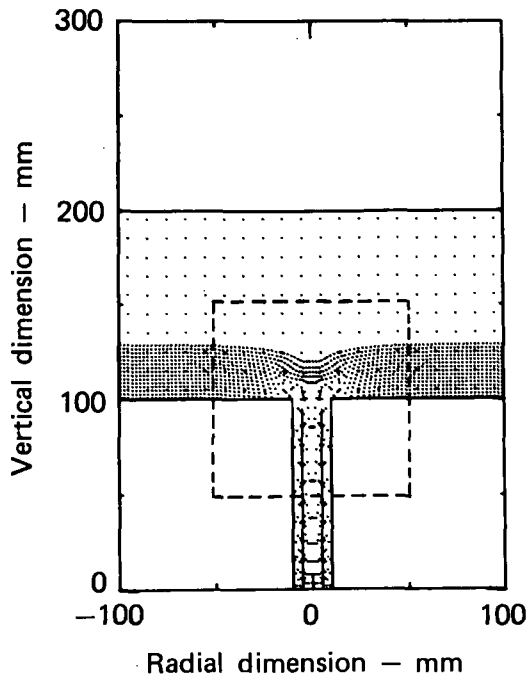


FIG. 8. Draining funnel. (a) Grid. (b) Flow field at 41 ms.

vertical-walled tank had rigid walls at grid lines. Another calculation of a vertical-walled tank with walls between grid lines gave results within one percent of the calculation with walls on gridlines.

Streaklines in the tank near the outlet pipe entrance display physically reasonable streak patterns. Streaklines are inferred from the marker particle calculations shown in Fig. 9.

A comparison of the calculated and theoretical time histories of the tank water level (H/H_0) and the outlet pipe velocity ($\theta \cdot H$) is shown in Fig. 10 for a tank and in Fig. 11 for a draining funnel. The comparisons are very good. Further comparisons of drain time, time to reach maximum outlet velocity, and percentage errors are given in Table 1. The effect of the length of the outlet pipe on the duration of the transient response is shown in Fig. 12.



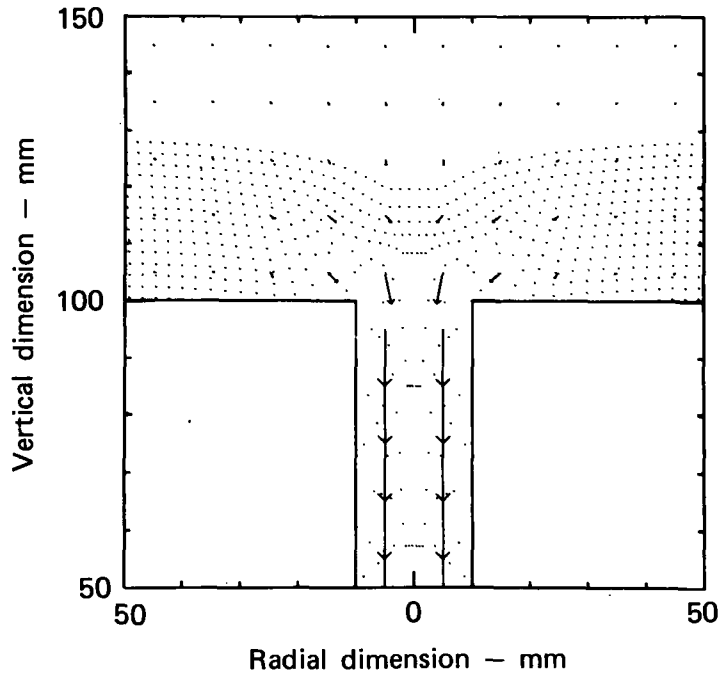


FIG. 9. Streaklines for an unpressurized tank at about 91 ms. The velocity vectors are represented by the arrows and the marker particles by dots.

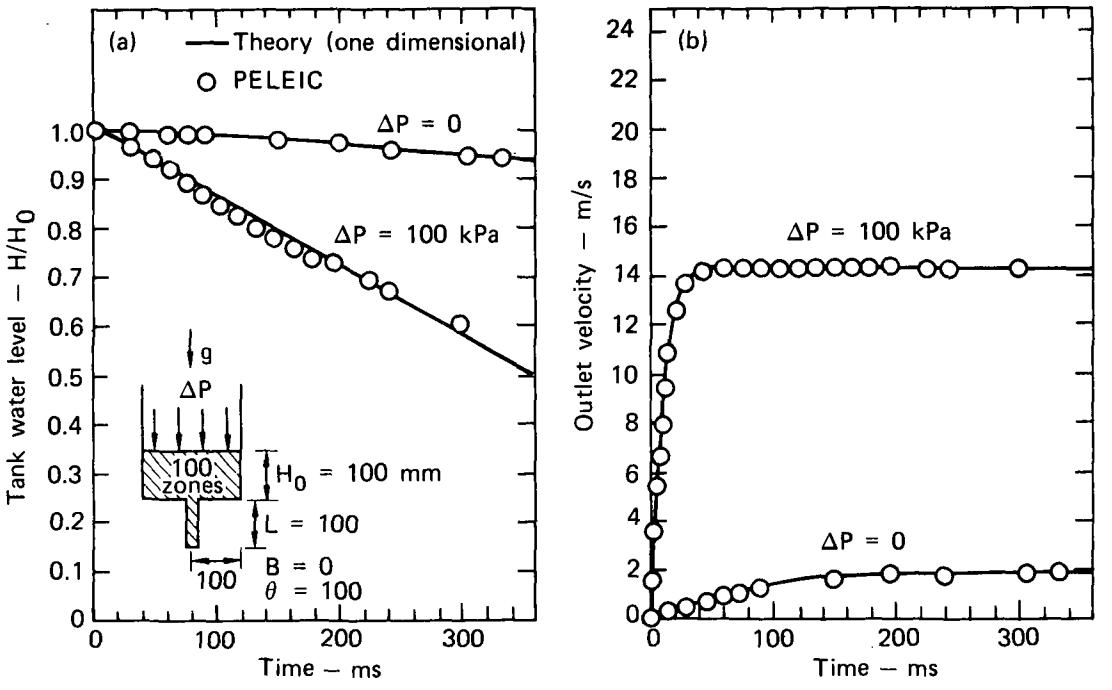


FIG. 10. Draining tank verification problem for pressurized and unpressurized tanks. (a) Comparison of calculated and theoretical water levels, H/H_0, vs time (H is averaged over the top surface). (b) Comparison of outlet velocity vs time.

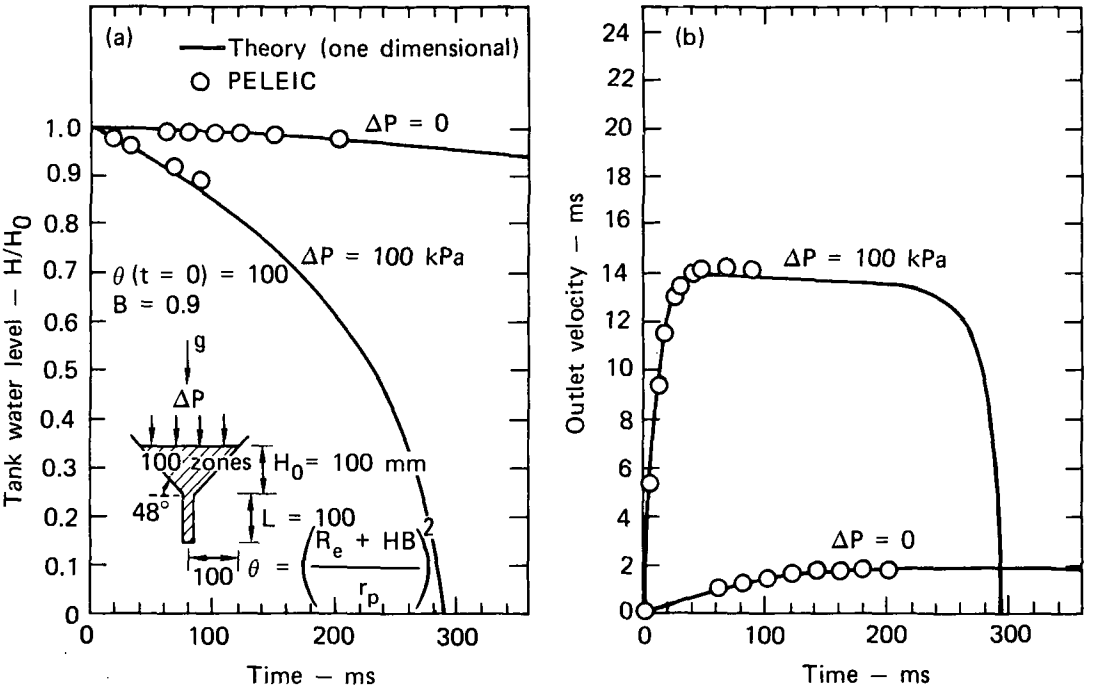


FIG. 11. Draining funnel verification problem for pressurized and unpressurized tanks. (a) Comparison of calculated and theoretical water, H/H_0, vs time (H is averaged over the top surface). (b) Comparison of outlet velocity vs time.

TABLE 1. Comparison of tank draining data (H_0 = L = 100 mm, $\theta(t = 0) = 100$).

Geometry	Tank	Tank	Funnel	Funnel
Pressure, kPa	0	100	0	100
Drain time, s	5.96	0.708	2.25	0.144
Maximum time calculated, s	0.900	0.300	0.200	0.090
Time of maximum outlet velocity, s				
Theory	0.342	0.075	0.348	0.060
PELE-IC	0.345	0.082	---	0.078
Maximum error in surface height, %	1	4	0.4	1.8
Maximum error in outlet velocity, %	8[a]	3	0.2	1.9

[a]This value occurs during the transient response.

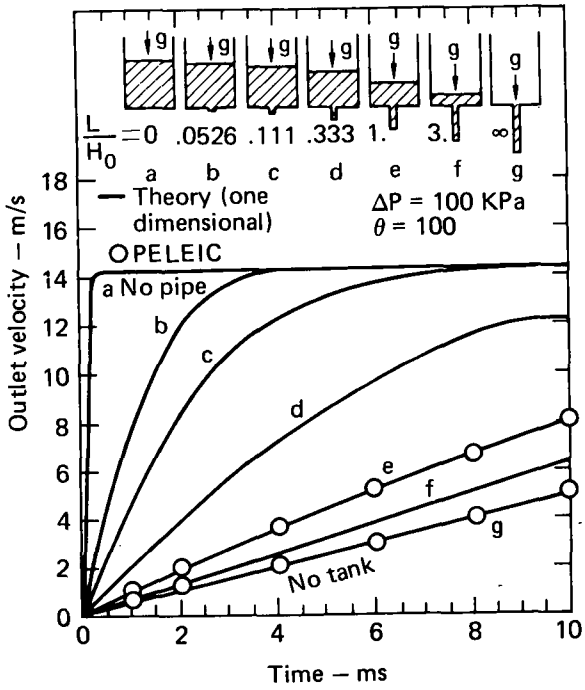


FIG. 12. Transient outlet velocity for various pipe lengths. Initial water height is 200 mm for all cases. L/H_0 is the ratio of outlet-pipe length to initial water depth in the tank.

BUBBLE GROWTH

We verified the logic for recognition and movement of an interface between two phases by simulating the growth of a spherical bubble in an infinite medium. We compared the results with a known analytic solution and obtained good agreement. Comparisons with bubble growth in air blowdown experiments are given in the Applications section.

We simplified the analysis of a bubble in a infinite fluid by assuming that the surface tension and viscous effects are negligible and that the fluid is incompressible and homogeneous. An equation-of-motion for the radius of the bubble under these assumptions was given by Besant in 1859. Rayleigh[14] described the oscillatory expansion and collapse of an adiabatic bubble with an initial pressure. In this simulation we model the growth of a bubble under a constant driving overpressure. The equation-of-motion for the case is given by Parker et al.[15]

$$R\ddot{R} + \frac{3}{2}\dot{R}^2 = \frac{\delta P}{\rho} \quad , \tag{21}$$

where δP is the driving overpressure and ρ is the density of the liquid.

We solved this equation numerically for the radius of the bubble as a function of time. We used a constant driving overpressure of 20 kPa and modeled the initial bubble diameter with eight zones. The initial radius of the bubble, R_0, was 20 mm; the initial velocity was zero everywhere. The outer boundary of the grid was at 200 mm with a continuative boundary condition.

We plotted the bubble radius (Fig. 13) as a function of time and compared it with the analytical solution. The calculation accurately tracks the bubble growth. The bubble maintained sphericity to about 4 percent after doubling its radius. The spherical geometry was meshed in cylindrical (r,z) coordinates.

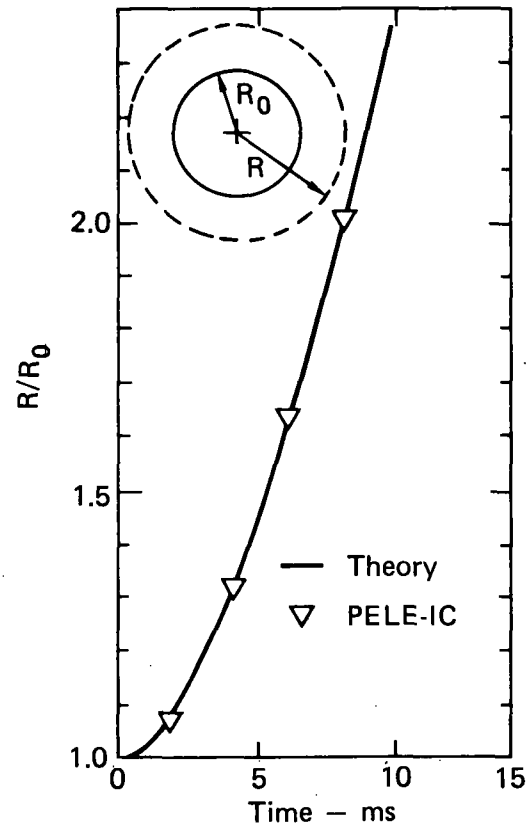


FIG. 13. Comparisons of calculated and analytical solution for spherical bubble growth as a function of time.

VIBRATION OF A SUBMERGED PLATE

The vibrational frequency of a plate decreases when it is submerged in a fluid. This is due to an increase in the effective inertia of the submerged plate and has been described in terms of an added hydrodynamic mass by Stokes[16] and Lamb.[17] In our calculations, the hydrodynamic mass enters naturally through the coupling algorithm.

We verified the accuracy of the fluid-structure coupling algorithm by calculating a circular flexible plate in contact on one side with a fluid. We considered two cases: (1) the fluid is unconfined, and (2) the fluid is confined in a cylindrical tank with rigid side walls with a diameter equal to that of the plate. We show that our solution for the unconfined fluid converges at infinite fluid depth to the Lamb solution, and we describe the transition from free vibrations in vacuum to vibrations in the unconfined fluid. For vibrations in a confined fluid, we show that our calculated frequency agrees with the frequency computed using the added mass technique.

We first calculated the fundamental period of a circular plate in a flat wall, in contact on one side with an infinite amount of water. The solution to this problem was obtained by Lamb using the Rayleigh-Ritz method.[17] His solution is

$$f = \frac{0.4745}{\sqrt{1+\beta}} \frac{hc}{R^2} \quad , \tag{22}$$

where f is the frequency, h is the plate thickness, c is the speed of propagation of extensional waves in a thin plate of the material, R is the radius of the plate, $\beta = 0.6689\ \rho R/h\rho_p$, ρ is the density of water, and ρ_p is the density of the plate.

To obtain a solution to this problem with the PELE-IC code, we modeled the geometry with a finite depth of water, gradually increasing the depth to approach that of the Lamb solution. A typical simulated geometry is shown in Fig. 14. To load the plate we applied a step pressure pulse of 5 kPa at the

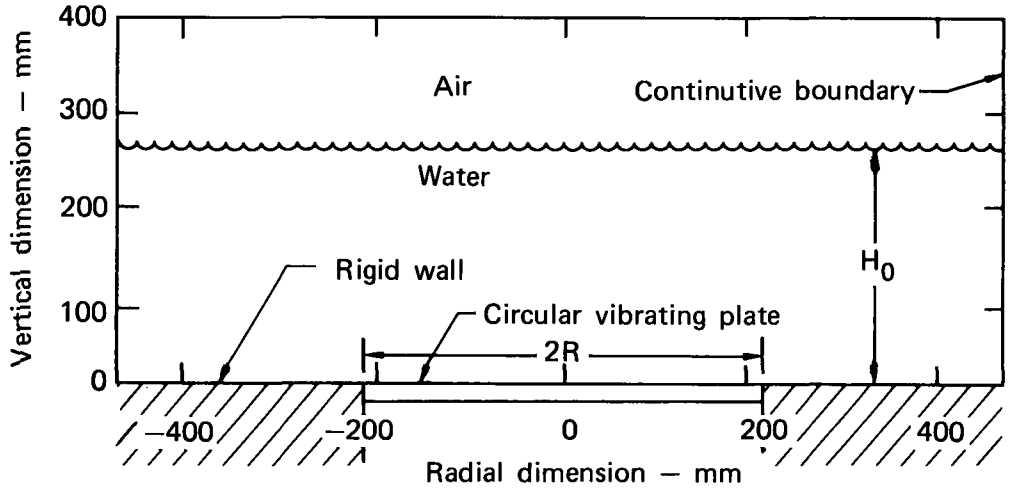


FIG. 14. Typical simulated geometry for a circular plate in contact with an unconfined amount of water.

top surface of the water for 11 ms, and excited the first vibrational mode of the plate. The amplitude was chosen so that the plate deflection was less than the plate thickness. To obtain the natural frequency, the simulations were carried out to a minimum of three complete oscillations. The computer time on a CDC 7600 computer ranged from 1 min for the shallow water calculations to about 5 min for the deeper water.

These calculations used a 2-mm-thick circular steel plate having a 211.5-mm radius. We took Young's modulus as 206 GPa, Poisson's ratio as 0.29, and the density as 7.87 kg/m^3. A clamped circumferential boundary condition was used.

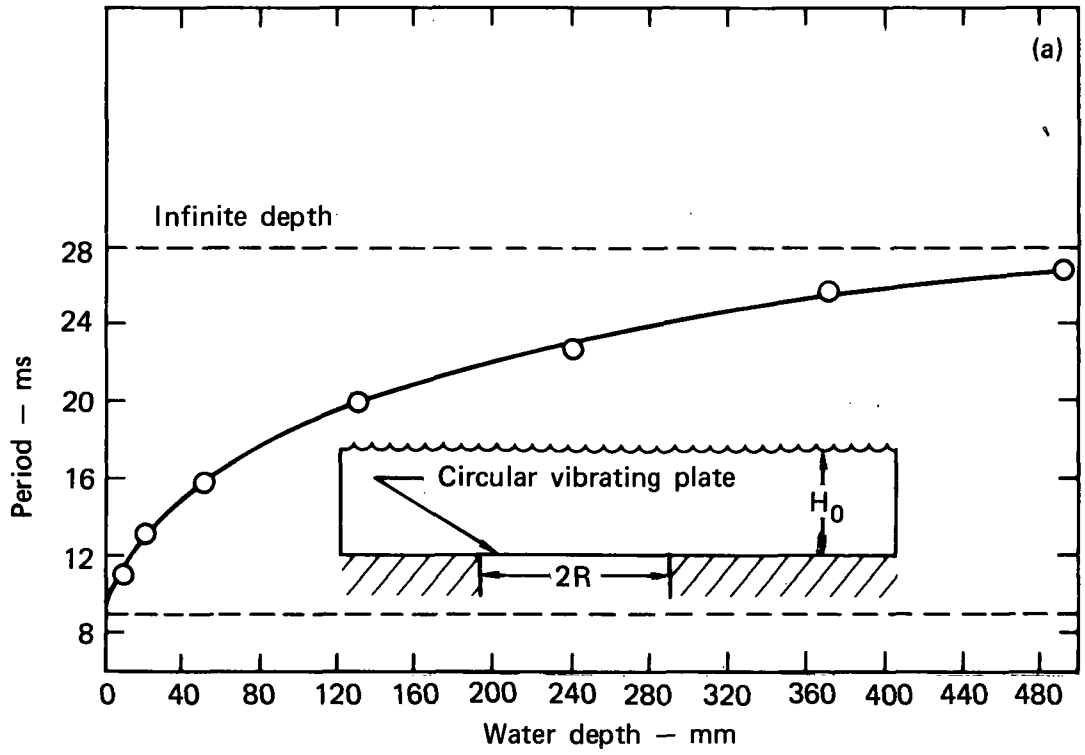
The results are shown in Fig. 15(a) for the various water depths. The Lamb solution predicts a period of 27.9 ms; we calculated a period of 27 ms at a water depth of 1.16 plate diameters. Our solution approaches the Lamb solution asymptotically.

Figure 15(b) is a plot of the normalized added mass vs water depth in units of plate radius. This curve describes in a normalized form, independent of plate characteristics, the transition from free vibrations in vacuum to vibrations in contact with unconfined incompressible fluid at infinite depth. Table 2 summarizes the results of the seven calculations.

The results in Fig. 15(a) can be generalized using the concept of added hydrodynamic mass. The added mass is zero for free vibrations and converges to a constant value in the limit of vibrations under infinite depth. Since the square of the period of oscillation is proportional to the total effective mass (equal to the mass of plate plus the added hydrodynamic mass), one can define a normalized added mass, m, for any depth H:

$$m = \frac{\text{added mass at given depth}}{\text{added mass at infinite depth}} = \frac{\tau_H^2 - \tau_0^2}{\tau_\infty^2 - \tau_0^2} , \tag{23}$$

where τ_H is the period for depth H, τ_0 is the period in vacuum, and τ_∞ is the period at infinite depth. The normalized added mass for any



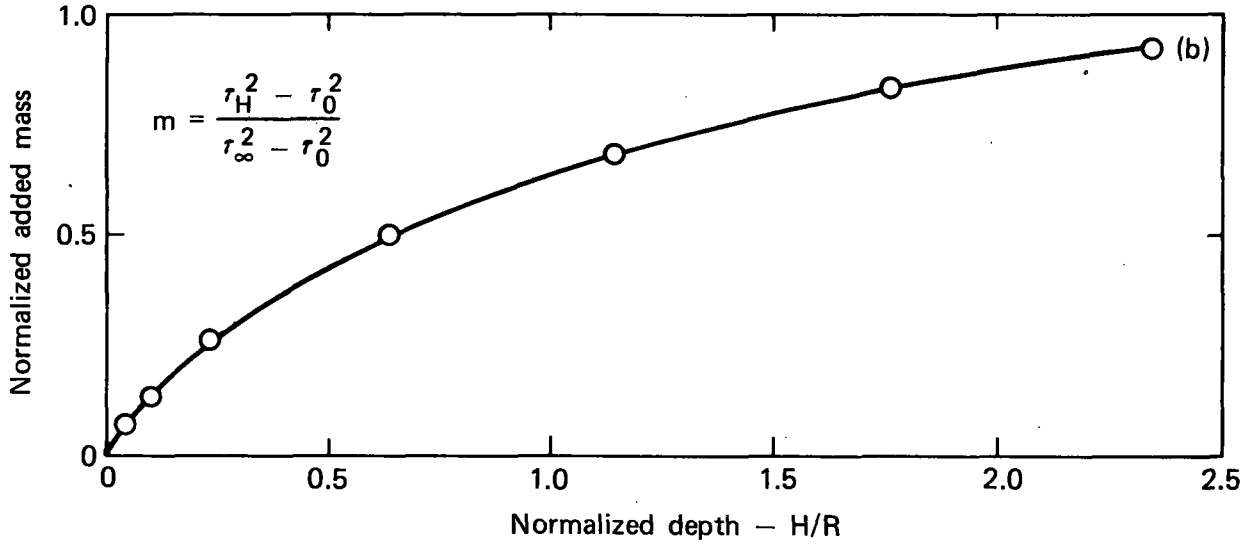


FIG. 15. Circular clamped plate in contact on one side with an unconfined amount of water. (a) Period of oscillation vs normalized water depth. (b) Normalized added mass vs water depth.

TABLE 2. Period of oscillation and normalized added mass at various depths for a circular plate in contact with unconfined water.

Depth, mm	Period, ms	Depth/Radius	Normalized added mass
0	8.9	0.0	0.
10	11.0	0.047	0.059
20	13.25	0.095	0.14
50	16.0	0.235	0.25
130	20.5	0.615	0.49
240	23.5	1.13	0.67
370	26.0	1.75	0.85
490	27.0	2.32	0.93

depth is a dimensionless number between 0 and 1 and is a measure of how close the frequency of oscillation is to the Lamb frequency at infinite depth.

The pressure and velocity distribution during vibration are shown in Figs. 16 to 19 for a ratio of water depth to plate radius of 1.18. The five frames in each of these figures describe the plate and fluid response for the first 20 ms of vibration. The 5 kPa overpressure is removed at 11 ms. The first deflection peak of 2 mm occurs at 9.5 ms. At 20 ms, the plate has moved back toward the fluid and is at its upper maximum swing of 1.43 mm. Gravity was included in these calculations.

For small oscillations of a circular plate closing the bottom of a cylindrical rigid walled tank, the added hydrodynamic mass is the mass of water in the tank. At infinite depth the frequency of oscillation for a confined fluid approaches zero. At any depth, H, we define

$$\lambda = \frac{\text{added hydrodynamic mass}}{\text{plate mass}} = \frac{\rho H}{\rho_p h} \quad , \tag{24}$$

where ρ is the density of water, H is the water depth, ρ_p is the density of the plate, and h is the plate thickness.

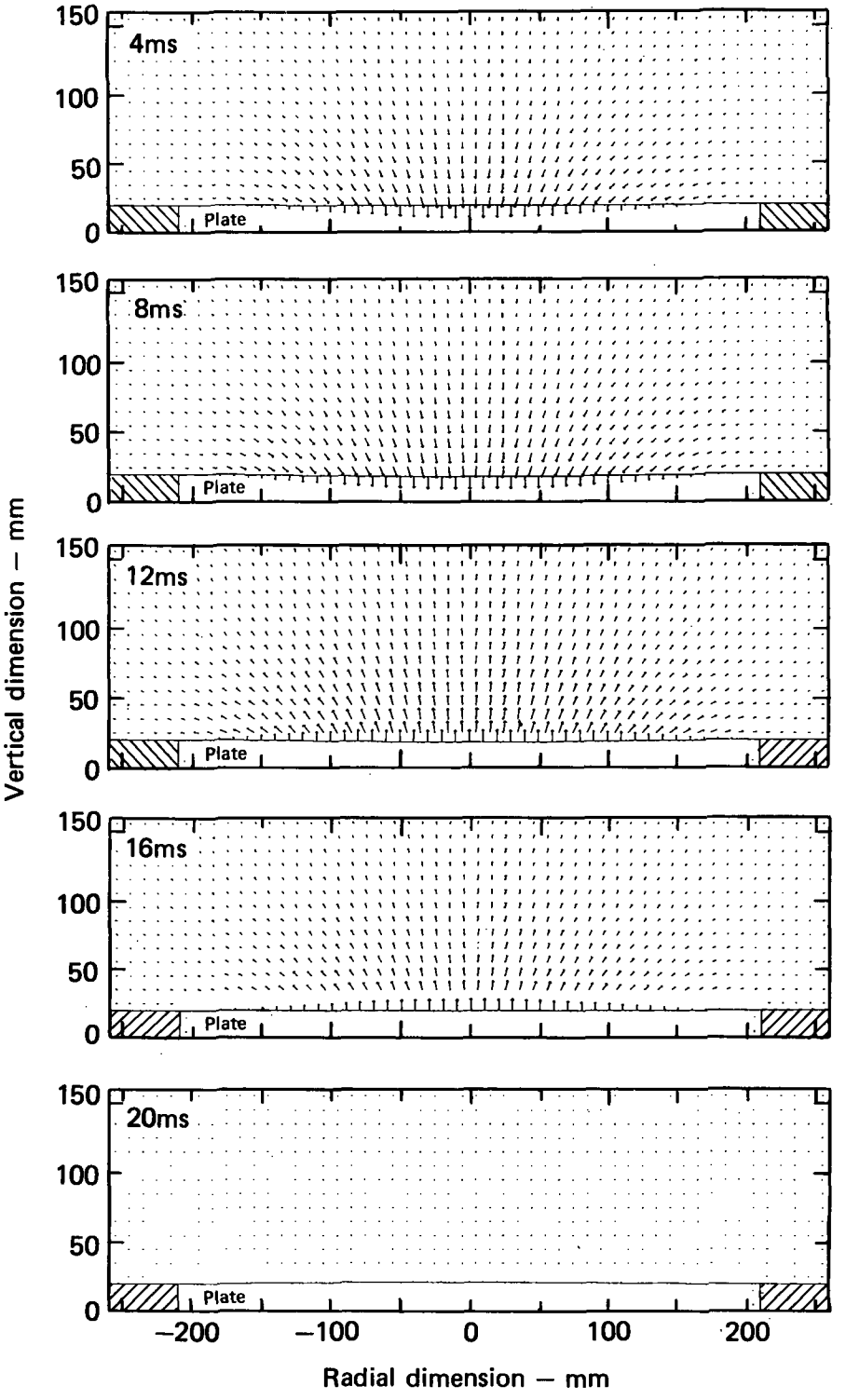


FIG. 16. Fluid velocity distribution above a vibrating plate. Velocity vectors are shown at 5 times. H_0 is 250 mm. A constant 5 kPa pressure was applied to the water surface for 11 ms.

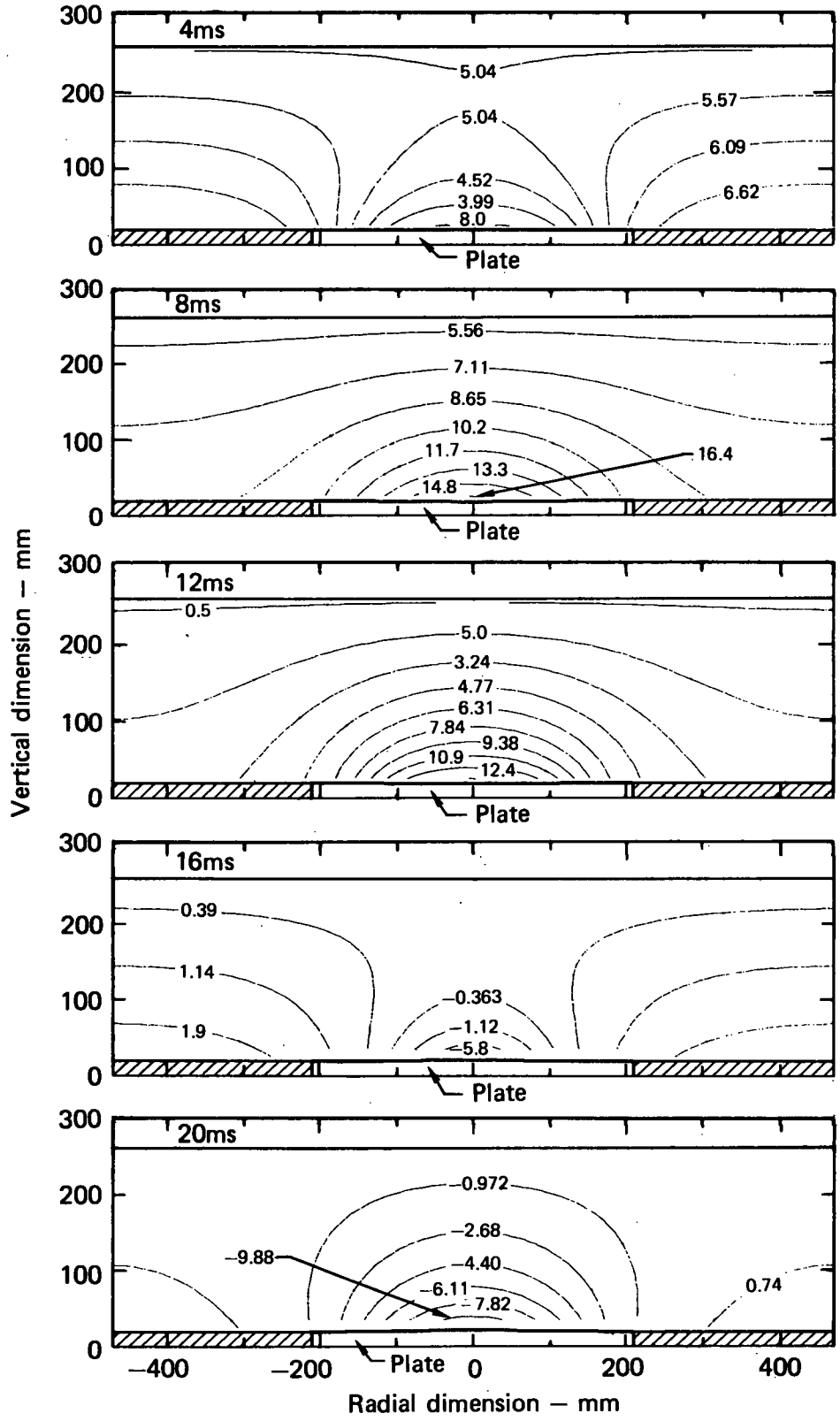


FIG. 17. Pressure contours in a fluid above a vibrating plate. H_0 is 250 mm.

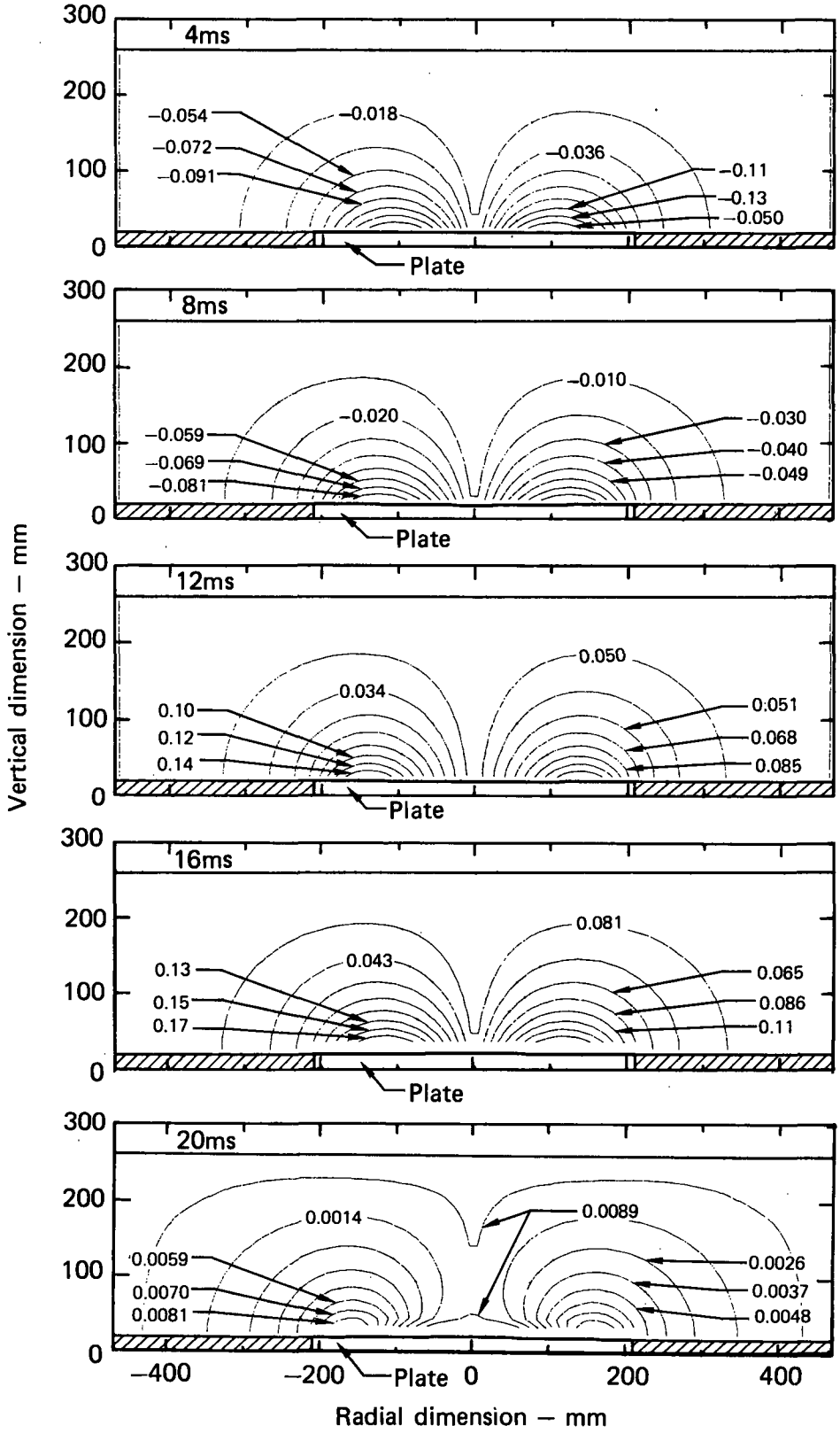


FIG. 18. Radial velocity contours in a fluid above a vibrating plate. Units are m/s. H_0 is 250 mm.

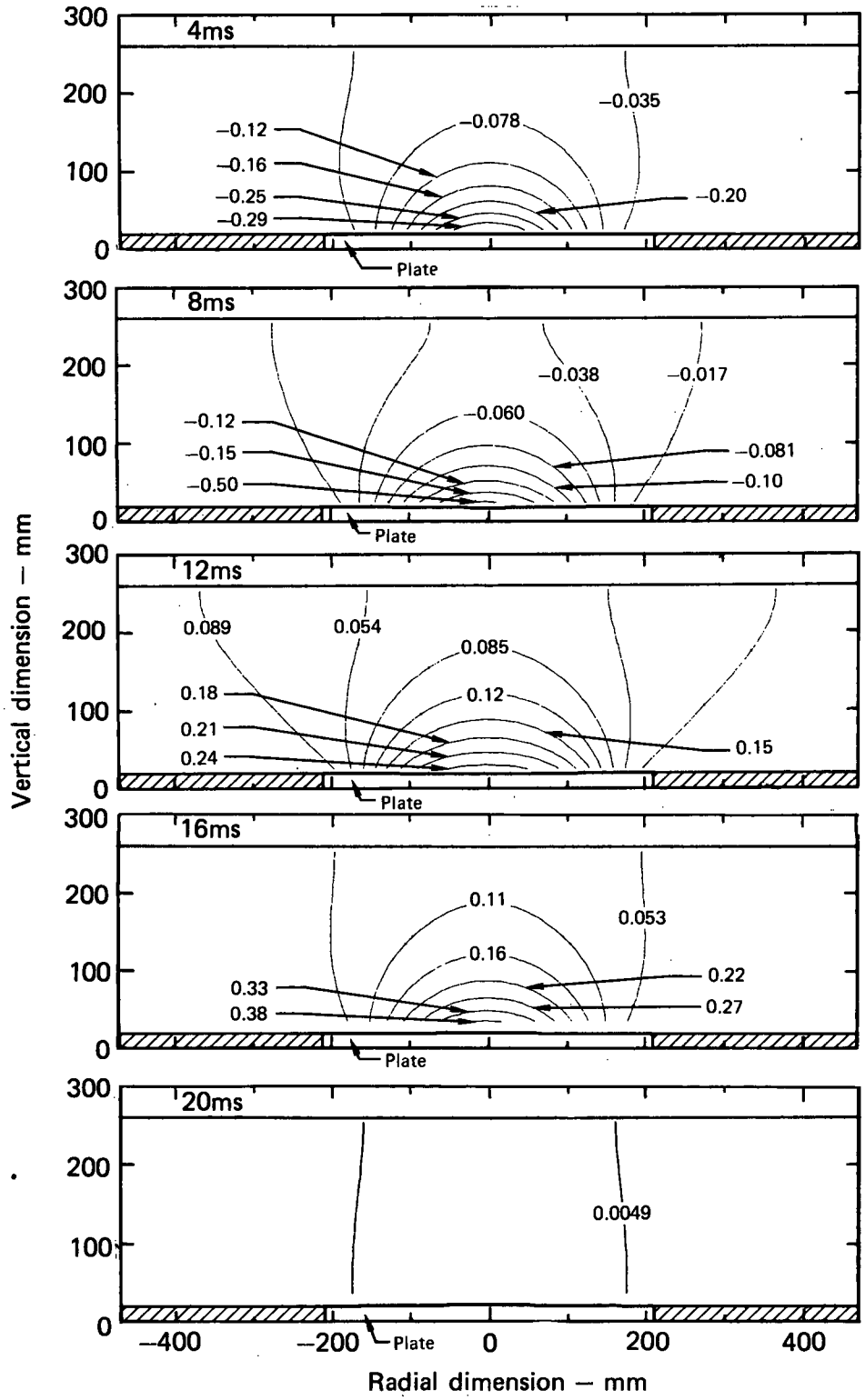


FIG. 19. Axial velocity contours in a fluid above a vibrating plate. H_0 is 250 mm.

The period of oscillation τ_H at any depth is

$$\tau_H = \tau_0 \sqrt{1+\lambda} \ , \tag{25}$$

where τ_0 is the period of oscillation in vacuum.

We simulated the vibrations of 235-mm radius circular steel plates at the bottom of a tank with rigid walls. Table 3 compares the period of oscillation to the predicted value for four calculations. These calculations were for deflections less than 0.1-mm plate thicknesses. The absolute difference between the PELE-IC and predicted periods is less than four times the time step used in the calculation. Thus, the accuracy can be increased by using smaller time steps. The time steps for these calculations varied between 0.1 and 0.4 ms.

TABLE 3. Calculated and predicted periods of oscillation for steel plates in contact with a confined amount of water (plate radius = 235 mm).

Plate thickness, mm	Period in vacuum, ms	Water depth/ plate thickness	PELE-IC calculated period, ms	Predicted period, ms	Difference, %
2	11.0	60	29.5	32.3	9
5	4.4	48	11.8	11.7	1
10	2.2	24	4.1	4.4	7
10	2.2	12	3.6	3.5	3

NONLINEAR PLATE VIBRATION COMPARISON

We calculated the nonlinear transient response of a flexible plate submerged in a cylindrical water tank in two geometries. Since no analytical solution was available for verification, we compared the plate response from PELE-IC with that of NIKE2D,[18] an implicit, large-deformation finite element code.

The period of a submerged vibrating plate for small deflection is predicted accurately by PELE-IC as shown in the previous section. For plate deflections greater than half the plate thickness, geometric nonlinearities occur due to membrane stresses[19] that stiffen the plate and reduce the natural period of vibration. The two nonlinear calculations reported here use the tank geometries used in studies at the University of California at Los Angeles and at the Massachusetts Institute of Technology.* (See Applications section.) For the purpose of the calculations, we modeled the rigid bottoms in these tanks as thin flexible plates, choosing the thickness and loading function such that we obtained deflections of two plate thicknesses for the UCLA tank and three plate thicknesses for the MIT tank. No gravitational loads were used in these calculations.

Although the calculational methods reported here are radically different, implicit finite element vs semi-implicit finite difference coupled to a thin shell code, the bottom plate deflections and bottom pressures are markedly similar as shown in Fig. 20 and Fig. 21.

All calculations used rigid tank walls and an applied pressure on the top surface of the water. A 4-node incompressible fluid element and an 8-node isoparametric element were used for the vibrating plate in the finite element calculation. The finite difference calculations (run with a stabilizing viscosity) used rectangular cells for the fluid and finite element shell elements for the plate. The stiffness matrix was re-formed at each time step for both calculations. The PELE-IC and NIKE2D calculations used the same zoning in the water and plate; however, the time step for the PELE-IC

*The MIT tests used tanks with diameters of 140, 280, and 550 mm; only the 140-mm tank was used in this calculation.

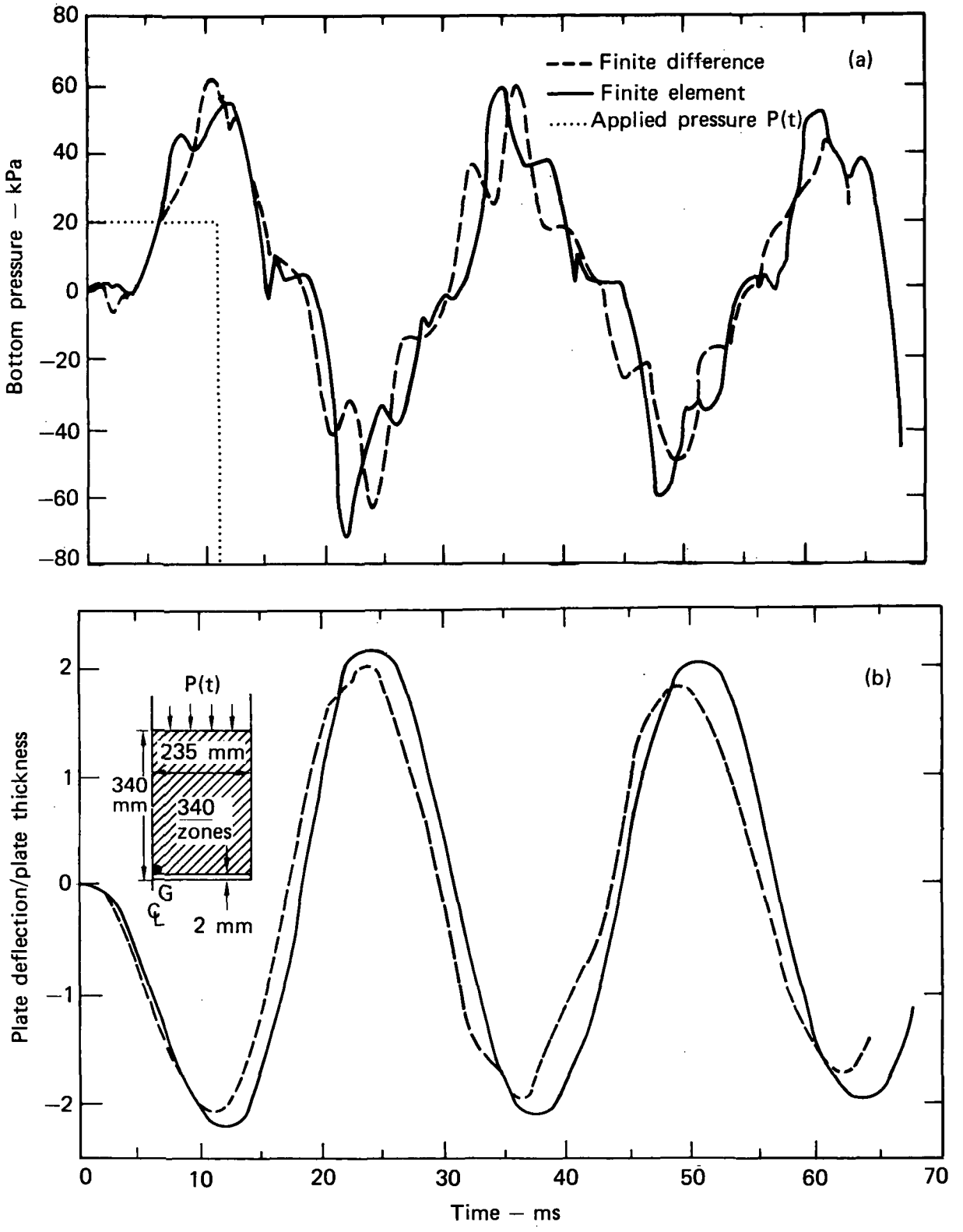


FIG. 20. Comparison of PELE-IC and NIKE2D calculations of a submerged steel plate vibrating in a rigid-wall tank of water for the UCLA configuration. (a) Bottom center pressure. (b) Plate center deflection.

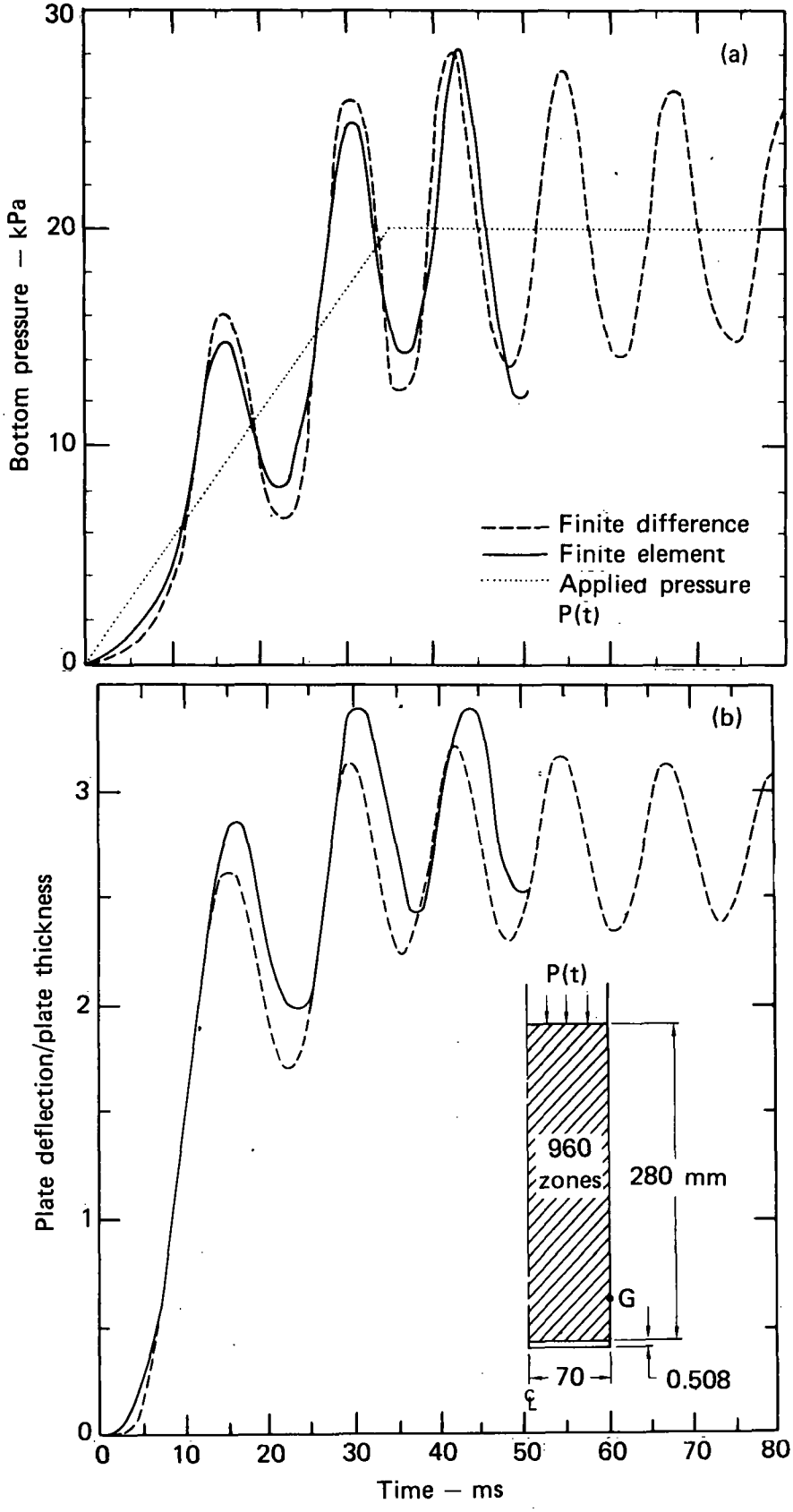


FIG. 21. Comparison of PELE-IC and NIKE2D calculations of a submerged aluminum plate vibrating in a rigid-wall tank for the small MIT configuration. (a) Bottom center pressure. (b) Plate center deflection.

calculations (0.05 ms) was ten times less than that of the NIKE2D calculations. PELE-IC ran about three times faster than NIKE2D on the CDC 7600 computer due in part to the higher order finite elements in the plate for the NIKE2D calculations.

Additional comparisons of the calculated normalized frequencies and maximum plate deflections, for the two geometries are given in Table 4. The frequencies are normalized using Eq. (25).

TABLE 4. Comparisons between finite element and PELE-IC calculations.

Configuration	Tank diam., mm	Water depth, mm	Plate thickness, mm	Calculated frequency / Linear theory frequency		Calculated maximum def. / Plate thickness	
				PELE-IC	NIKE2D	PELE-IC	NIKE2D
UCLA[a]	450	340	2.00	$\frac{39.5}{20.0} = 1.98$	$\frac{40.0}{20.0} = 2.00$	2.05	2.20
MIT[b]	140	280	0.51	$\frac{79.4}{17.9} = 4.43$	$\frac{76.9}{17.9} = 4.30$	3.05	3.25

[a]Steel plate: E = 206 GPa, ν = 0.29, ρ = 7.86 kg/m^3.

[b]Aluminum plate: E = 69. GPa, ν = 0.29, ρ = 2.7 kg/m^3.

TORUS CALCULATIONS

The submerged plate calculations described earlier showed the effectiveness of the fluid-structure coupling algorithm for a simple rectilinear structure. This section presents some preliminary calculations using a shell geometry that tests the algorithm for the more general Lagrangian structure overlay. This fluid-structure coupling algorithm was described on pages 9 to 10.

We tested the curved structure algorithm in a simulation of air blowdown from a downcomer submerged in a spherical shell half-filled with water. Figure 22 shows the simulated geometry and the calculated fluid velocity distribution at 32 ms. The top of the shell was pinned at the shell-downcomer intersection. We applied a linear pressure ramp increasing from 0 to 10 kPa in 15 ms and then a constant pressure of 10 kPa after 15 ms.

We changed the stiffness of the shell and verified that in the limit of very large Young's modulus the calculation gives the same results as a fixed boundry of the same curvature. No instability developed with decreasing structural stiffness and the general shell response appeared normal. Figure 23 shows the pressure contours inside the spherical shell at 32 ms. The pressure in the bubble at this time is 10 kPa and the contour levels vary from 9 kPa to 3 kPa at regular intervals. This calculation was for a sphere diameter to thickness ratio of 100. Ratios of 40 and 20 were also calculated. Coarse zoning (δx = 10 mm) was used in the fluid.

We also calculated the air blowdown in plane geometry (an infinitely long cylinder), for an aspect ratio of 100. The sphere is a stiffer structure than the cylinder. At time equal to 32 ms the maximum deflection at the bottom of the long cylinder was 1.5 times the deflection at the bottom of the sphere.

We are now comparing similar calculations with results from the literature to further verify the accuracy of the coupling algorithm for a more general shell structure.

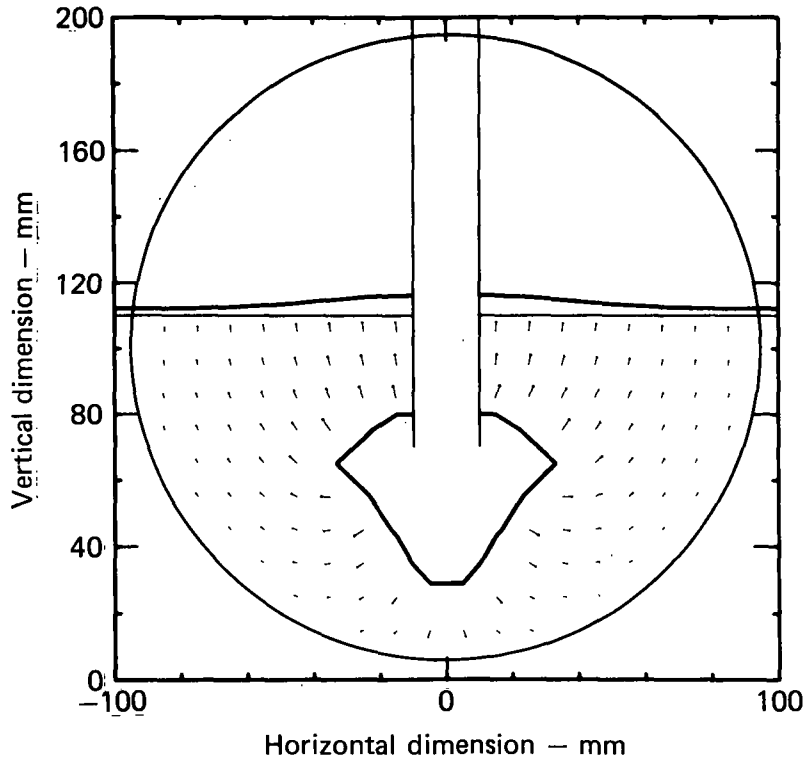


FIG. 22. Bubble growth calculation in a flexible spherical shell.

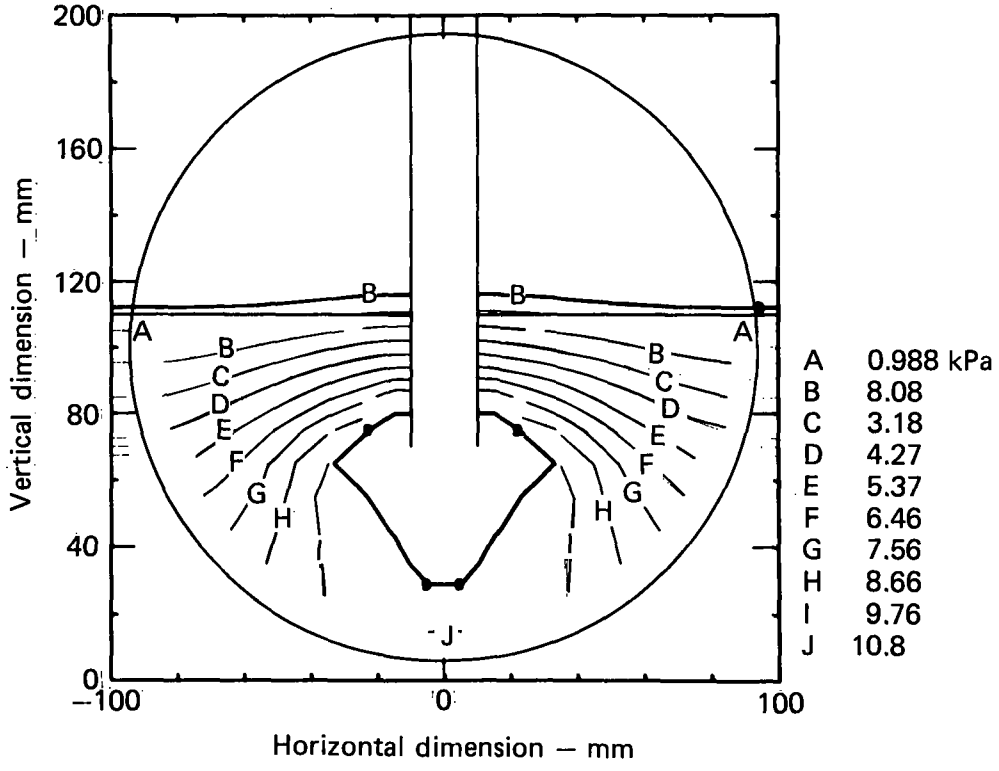


FIG. 23. Pressure contours at t = 32 ms. Vent clearing occurred at 25 ms.

APPLICATIONS

INTRODUCTION

This section presents the PELE-IC calculations of air blowdown experiments at the Massachusetts Institute of Technology[20] and University of California at Los Angeles.[21] These experiments used small vertical cylindrical tanks with a single downcomer. The experiments are not prototypical but were designed to calibrate computer codes and to describe the general blowdown phenomenology. Also presented are results obtained using some simple computer models to describe the oscillatory steam bubble condensation.

EXPERIMENTAL COMPARISONS OF AIR BLOWDOWN

To further assess the fluid mechanics in PELE-IC and to gain experience in calculating wetwell conditions, we calculated the results for two small-scale experiments by Anderson et al.[20] at MIT and Chan et al.[21] at UCLA. These experiments simulate wetwell dynamics by blowing air into a vertical pipe (downcomer) submerged in a rigid vertical cylindrical tank. The pressure time history in the downcomer and tank were measured and high-speed photography measured bubble growth.

In the experiment, air is suddenly released from the drywell and forms a bubble at the exit of the downcomer. The bubble grows and rises until it breaks through the fluid surface. The fluid pressure at the bottom of the tank reaches a maximum immediately after vent clearing and eventually drops to the static pressure head in the tank.

The tank geometries are compared in the inserts of Fig. 24 and in Table 5. For both geometries the water depths and downcomer pipe submergence depths are about the same; however, the UCLA tank has an air volume seven times greater than that of the MIT tank. Both tanks were constructed of rigid Plexiglas and

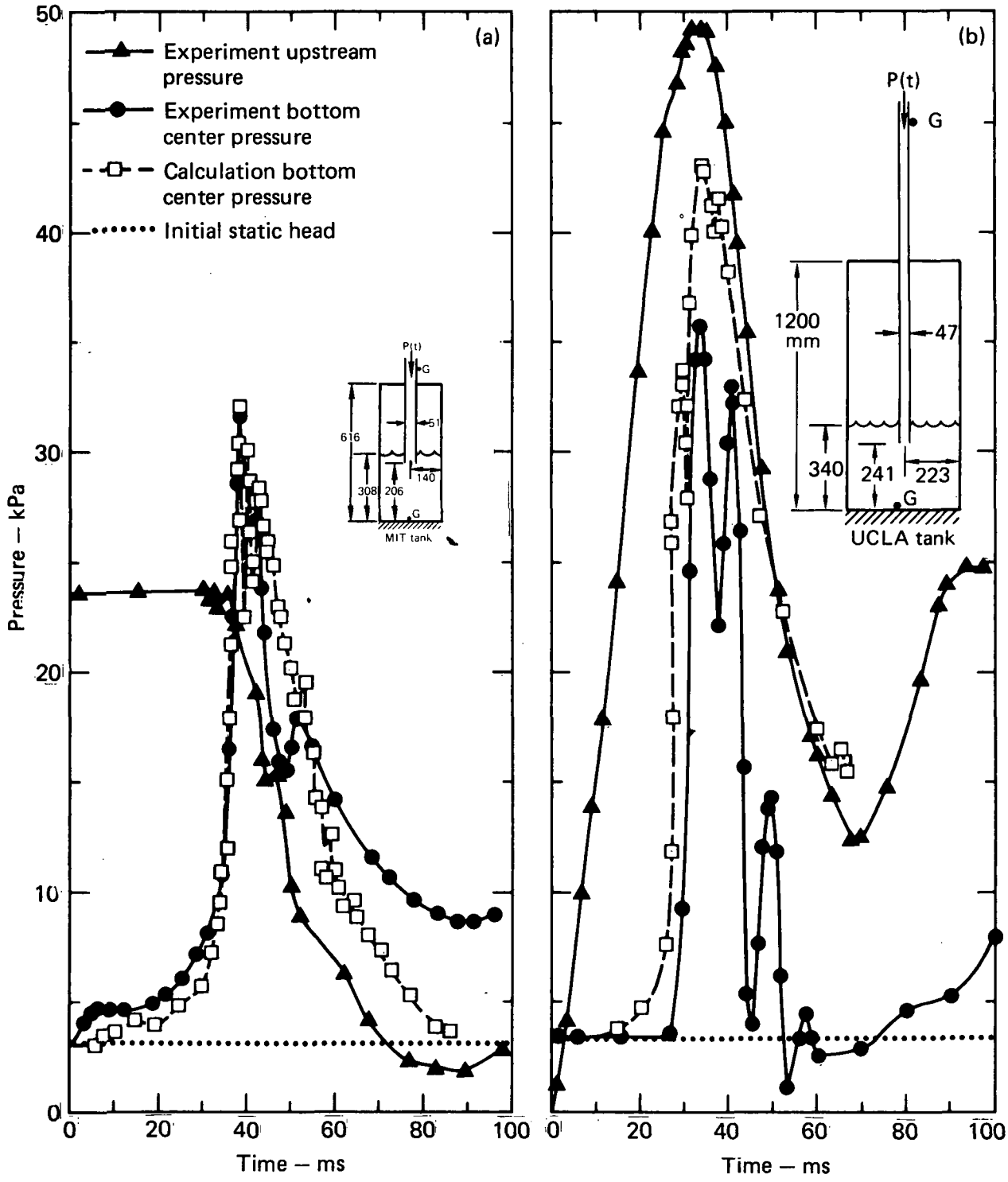


FIG. 24. Comparison of calculated and experimental pressure time histories for air blowdown. The upstream and tank bottom pressure gages are indicated by the letter G. (a) MIT configuration. (b) UCLA configuration.

TABLE 5. Comparisons of MIT and UCLA tank geometries.[a]

Configuration	Inner tank diameter, D	Inner pipe diameter, d	Tank height	Water depth	Distance to downcomer[b]	Volume of tank filled, %
UCLA	D = 450 mm	0.104D	2.67D	0.755D	0.762D	28
	9.57d	d = 51 mm	25.5d	7.25d	7.23d	
MIT	D = 280 mm	0.182D	2.2D	1.1D	0.736D	50
	5.49d	d = 47 mm	12d	6.0d	4.0d	

[a]Each dimension is given in terms of the tank diameter D and pipe diameter d.

[b]Distance from tank bottom to downcomer pipe exit.

had pressure transducers located in the downcomer pipe and at the tank bottom as shown in the inserts of Fig. 24.

A calculational cell size approximately 10 by 10 mm was chosen with two cells across the outlet of the pipe radius. We used 682 zones in the MIT calculation and 1140 for UCLA. Time steps for the UCLA and MIT calculations were 0.1 and 0.2 ms, respectively.

A comparison of the experimental and calculated results is shown in Fig. 24. Also shown is the downcomer pressure that was recorded experimentally and used as the driving pressure in the code. The static pressure is indicated by the dashed lines in Fig. 24. The tank (bottom center) pressure history curve from the calculation was normalized so that experimental and analytical vent clearing coincide. Data are shown for the initial pressure response and bubble growth. Comparison of vent clearing and peak pressure times are given in Table 6.

TABLE 6. Air blowdown--peak pressure and vent clearing times.

	MIT	UCLA
Vent clearing time, ms		
Experiment[a]	30 ± 7	25 ± 2
Calculation	31	32
Peak bottom pressure time, ms		
Experiment[b]	39 ± 3[c]	33[d]
Calculation	33	39

[a]Based on bubble growth photographs.
[b]Based on pressure history curves.
[c]Tic marks on pressure history curve unevenly spaced.
[d]First peak only, time zero taken at t = 20 ms.

Both the experimental and analytical results display similar qualitative features. Both experiments and calculations show multiple pressure peaks. Our calculations are in much better agreement with the MIT data than those of

UCLA. The quantitative differences are in the maximum pressures and the width of the pressure pulses. For the MIT calculation, the peak pressure is 2 percent higher than the experiment while the UCLA calculation peak pressure is about 20 percent higher. The calculated pressure pulses are wider than the experiments. At late times the differences between the analytical and experimental results are due to bubble communication with the structure and free surface in the experiments.

A comparison of downcomer water level for the UCLA geometry is shown in Fig. 25. Results have been normalized so that vent clearing coincides in both cases. A comparison of calculated and measured bubble growth is shown in Fig. 26. The coarseness of the zone size causes the dimpling of the bottom bubble surface in the analytical results (Fig. 26(a)). After vent clearing, the bubbles from both UCLA and MIT experiments (Fig. 26(b)) are flat and then grow to a strawberry shape. Calculated bubble growth histories also show the strawberry bubble shape.

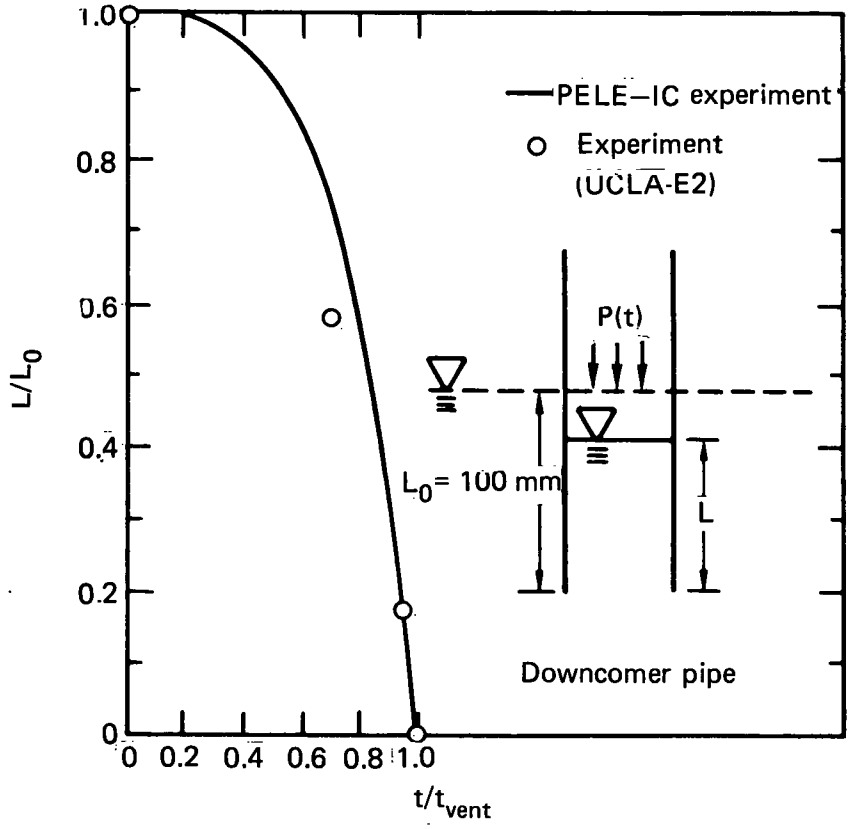


FIG. 25. Vent clearing comparison between calculation and the UCLA experiment.

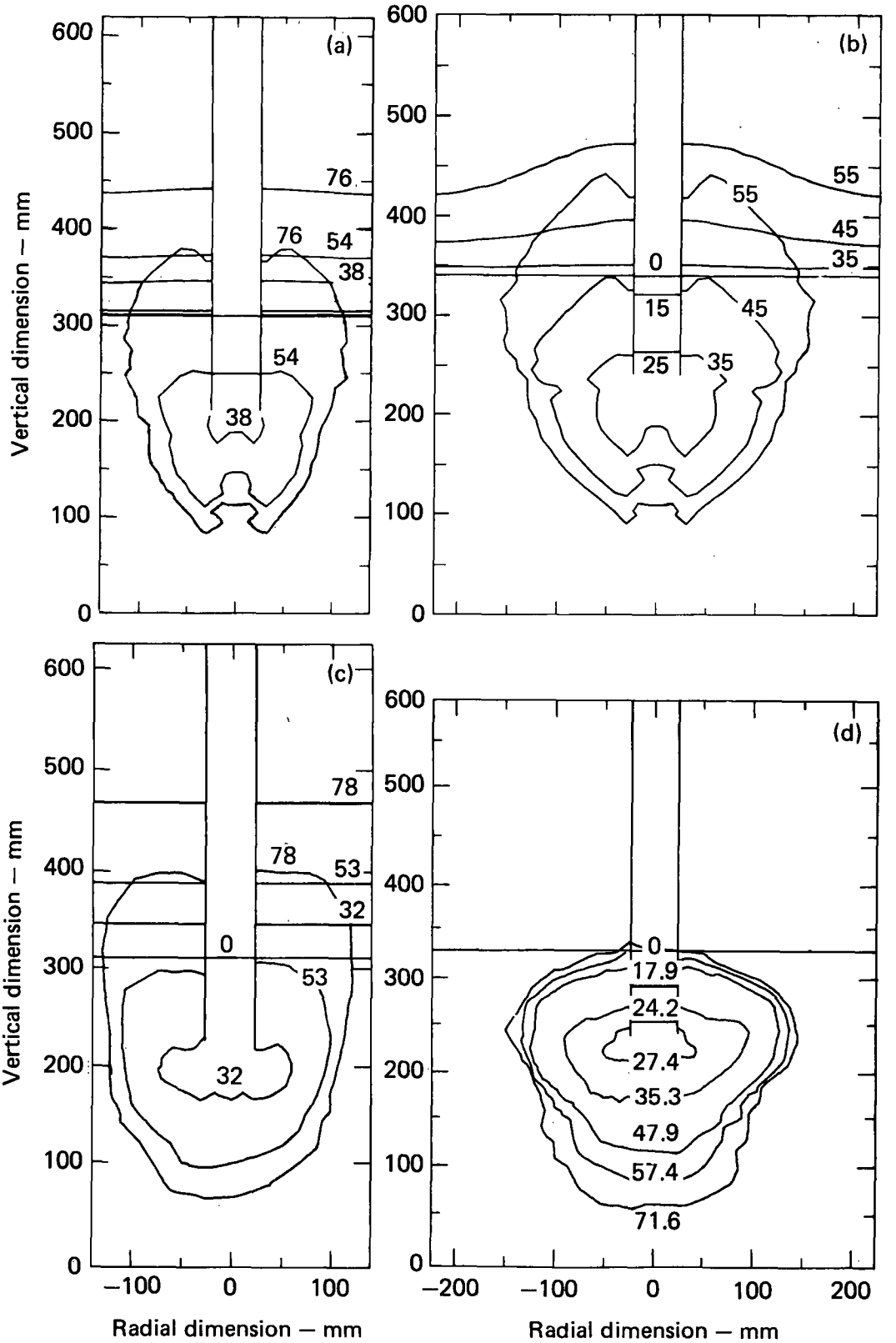


FIG. 26. Bubble growth sequence for air blowdown in cylindrical tanks. (a) Calculated for MIT tank. (b) UCLA tank. (c) MIT experimental. (d) UCLA experiment.

CHUGGING

The steam condensation model of Kowalchuk and Sonin[5] was coupled with PELE-IC as indicated earlier. The calculational geometry with fluid velocity vectors at vent clearing is shown in Fig. 27. The chugging parameters correspond to calculation 1 of Kowalchuk and Sonin.[5] For this case, the condensation is governed by the thermal conduction into the water.

The periodic solutions for interface movement, steam bubble pressure and steam temperature are shown in Figs. 28 through 30, respectively. A reasonable periodicity (about 1 s) was obtained after two cycles. Figures 28 and 29 demonstrate the essentials of the chugging phenomenon. After the interface clears the vent, the pressure drops due to the combined effects of expanding volume and resulting increased condensation. The buoyancy of the fluid then overcomes the downward pressure force and pushes the fluid back into the pipe. The momentum of the fluid is eventually reversed by the larger constant pressure from the dry well and the process repeated.

We compared our results with those of Kowalchuk and Sonin.[6] Although both responses looked similar to those of Figs. 28 through 30, quantitatively they are quite different. Their pressure drop with vent clearing is a factor of four greater, and their period is a factor of two longer than that calculated with PELE-IC.

The purpose of the liquid model in the coupling scheme is to determine the steam volume given the steam pressure. As a check, the Kowalchuk and Sonin equations used with PELE-IC were separated-out, and the volume profile reported by their complete analysis was used as input and the steam pressure determined. These results were completely consistent with those reported by the complete Kowalchuk and Sonin relations, thus verifying the coding and demonstrating that discrepancies appear because of the two liquid models.

Since previous tests of bubble growth using PELE-IC, i.e., the spherical bubble verification, have demonstrated the validity of pressure-volume relationships, the Kowalchuk and Sonin liquid model probably causes the differences. Not only is their model one-dimensional, but it also uses two

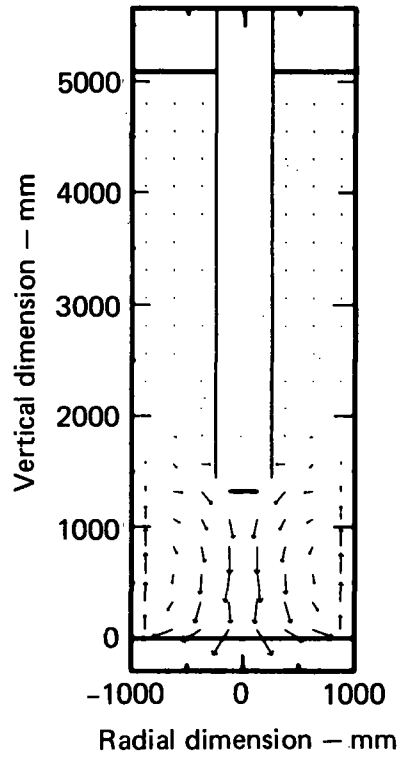


FIG. 27. Velocity distribution at vent clearing in a chugging simulation.

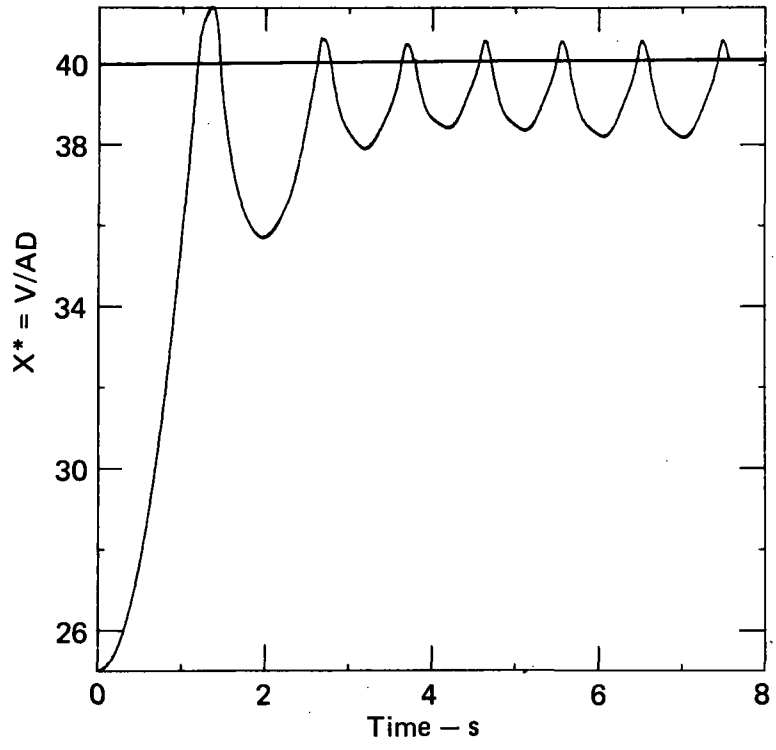


FIG. 28. Periodic movement of the steam-water interface during chugging. The steam volume in the downcomer and bubble is V. The area of the pipe is A and the pipe diameter is D.

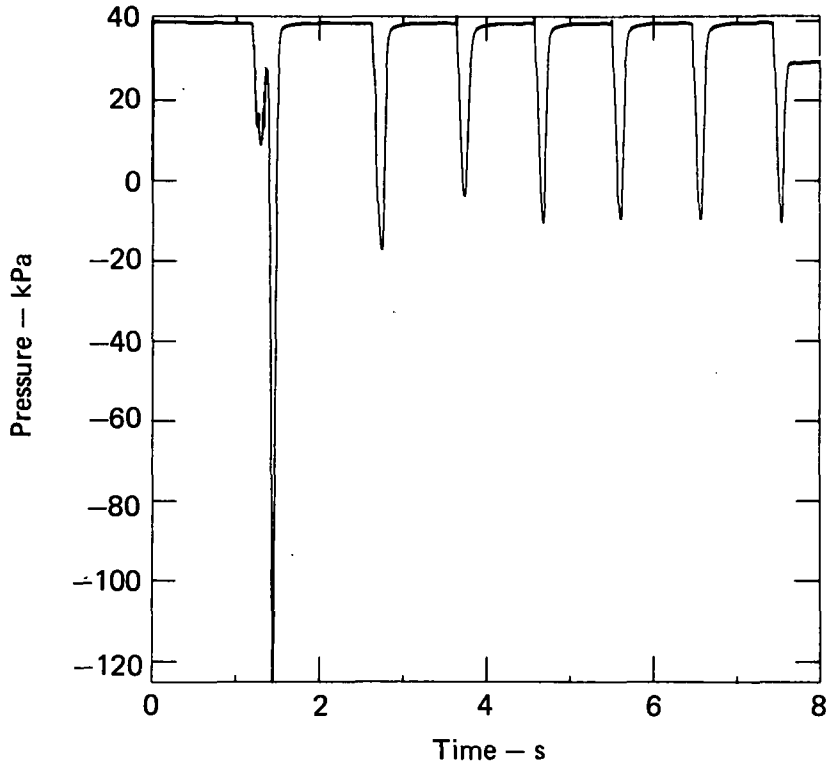


FIG. 29. Chugging pressure oscillations at the steam-water interface.

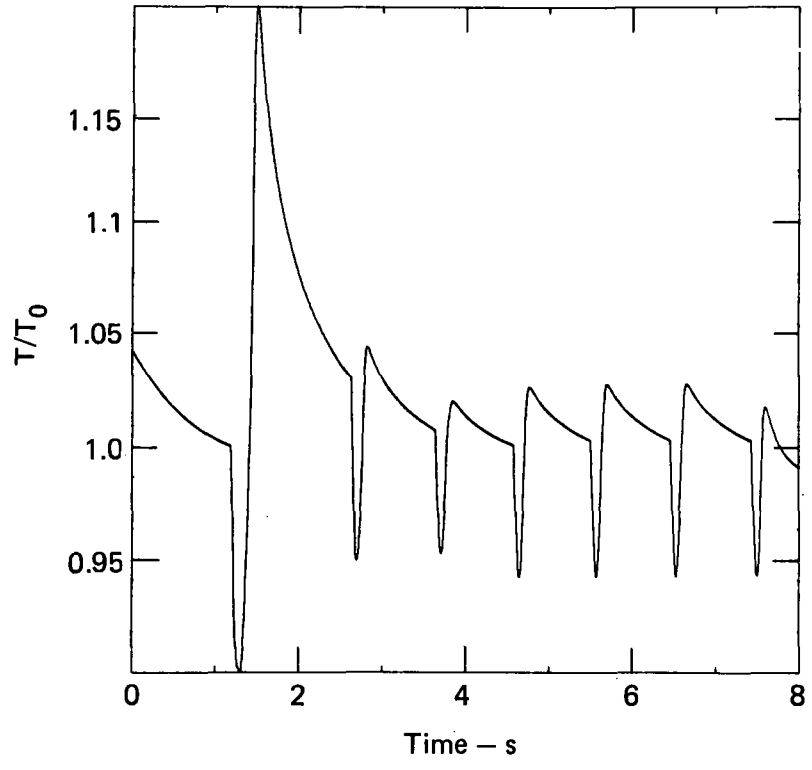


FIG. 30. Chugging steam temperature vs time, T_0 = 423K.

arbitrary parameters. The first is inserted to account for kinetic energy losses at the pipe exit. A standard loss coefficient times velocity head relationship is utilized. The second parameter accounts for the inertia of the fluid being accelerated outside the pipe.

To check the one-dimensional assumption, the depth of the tank was doubled and the chugging code run with PELE-IC. The results were virtually the same as those with the originaly geometry. It was therefore concluded that the discrepancies were due to the kinetic energy loss term and the fluid inertia approximation, the latter being responsible for the major differences.

SUMMARY AND CONCLUSIONS

We have coupled an incompressible Eulerian hydrodynamic algorithm to a Lagrangian finite-element shell algorithm for analysis of pressure suppression. A computer program calculates loads and structural response from air and steam blowdown and the oscillatory condensation of steam bubbles in a water pool. The fluid, structure, and coupling algorithms are verified by calculating solved problems from the literature and from air and steam blowdown experiments. The application is boiling water reactor pressure suppression designs.

The foundation of the program is the semi-implicit, two-dimensional SOLA algorithm. This plane and axisymmetric algorithm uses an upwind finite difference method to obtain a trial velocity field from the Navier-Stokes equations. This field is then modified iteratively by zonal pressure adjustment to satisfy the incompressibility condition. The shell structure uses conventional thin-shell theory with transverse shear. A finite-element spatial discretization employs piecewise-linear interpolation functions and one-point quadrature applied to conical frustra (or cylindrical plates). We used the Newmark implicit time-integration method implemented as a one-step module. The algorithms are strongly coupled in the iteration loop using the iterated pressure in the fluid to drive the structure.

We developed two new algorithms to treat free surface and fluid-structure boundaries. One is an air-water surface algorithm that models bubble growth and pool swell, and the other is a fluid-structure coupling algorithm that correctly couples the Lagrangian structure overlaying the Eulerian grid.

The air-water-surface algorithm calculates void fraction and surface orientation to define the free surface. The algorithm performs four functions: (1) maintains the correct surface pressure, (2) maintains incompressibility in the fluid component of the zone, (3) determines the surface orientation, and (4) advects the fluid consistent with the surface

orientation and the flow field. The first two are done in the iteration loop and the others are done explicitly using the updated velocity field.

The coupling algorithm requires normal velocity compatibility at the fluid-structure interface and incompressibility of the computational Eulerian zone overlaid by the structure. The fluid zones containing the structure have sides that are outside the physical model. Zone-side velocity outside the structure is set at each iteration cycle to maintain fluid-structure velocity compatibility inside the zone. We then adjust the pressure in these zones the same way as in the interior zones to satisfy the incompressibility condition. The iterated pressure is applied to the structure using the current time level configuration and the iteration is continued until the two conditions are satisfied.

Verification calculations are presented for tank draining, spherical bubble growth, submerged plate vibration, and computer simulations of air blowdown experiments. These calculations demonstrated remarkable ability to reproduce the solutions from the literature. The draining tank and the spherical bubble growth calculations show we satisfied the conservation laws and accurately tracked the motion of the free surfaces both in the form of pool swell and bubble growth. The coupling algorithm works well for the case of a submerged flexible plate as demonstrated by calculation of the Lamb problem and when compared against solutions, in simple geometries, from a finite element code containing a suitable fluid element. The hydrodynamic mass appears naturally as a result of the fluid-structure coupling algorithm.

Comparisons with blowdown experiments done both at the MIT and UCLA show that, while we are able to calculate the general phenomenology, we are unable to match these experiments in the detail we expected from the earlier calculations. For the MIT blowdown experiment, agreement is moderately good both in amplitude and shape of the wet-well bottom pressure. For the analogous UCLA experiments, agreement is poor (Fig. 24) both in amplitude and temporal pulse width. Comparison of calculated air bubbles with the MIT and UCLA experiments shows that the calculated geometry has the characteristic strawberry shape of the experimental bubble. The calculated volume also shows

reasonable agreement with the MIT data after the data are normalized so that vent clearing times agree. Again, poor agreement is obtained with the UCLA results.

The superior agreement we obtained in simulating the classical solutions suggest that there are additional physical models required in the code to simulate the details of the experiments. Assumptions neglecting viscosity, surface tension, increase in fluid compressibility with dissolved air, and uniform pressure in the bubble and downcomer probably contribute to the lack of some detail in these simulations. However, the general behavior appears to be correctly modeled and is probably adequate to obtain good load estimates and response for pressure-suppression structures.

The future work is directed to applications of the program. We are calculating problems from the fluid-structure literature for axisymmetric shells and cylinders. Submerged plate vibration data are now available from the MIT group. We plan to do simulations of the Livermore fifth scale[22] air blowdown experiments and other larger air and steam experiments. The objective is to provide the Nuclear Regulatory Commission with a verified computer program suitable for calculation of boiling water reactor pressure suppression phenomena.

APPENDIX A. STRUCTURE OF THE CODE AND SOLUTION ALGORITHMS

This section presents more details on the algorithms used and the structure of the code. Directions for the use of the code are given in the PELE-IC Users Manual.[23]

FINITE DIFFERENCE MESH

The finite difference mesh used for the numerical solution of the governing equations consists of rectangular cells of variable width δx and height δy. The generator allows one to increase or decrease the spacing geometrically to provide a smooth transition between all sizes. This mesh is laid out as shown in Fig. A.1 with the indicies and labels used by the code. The problem region is surrounded by a single layer of boundary cells used to impose the desired external boundary conditions.

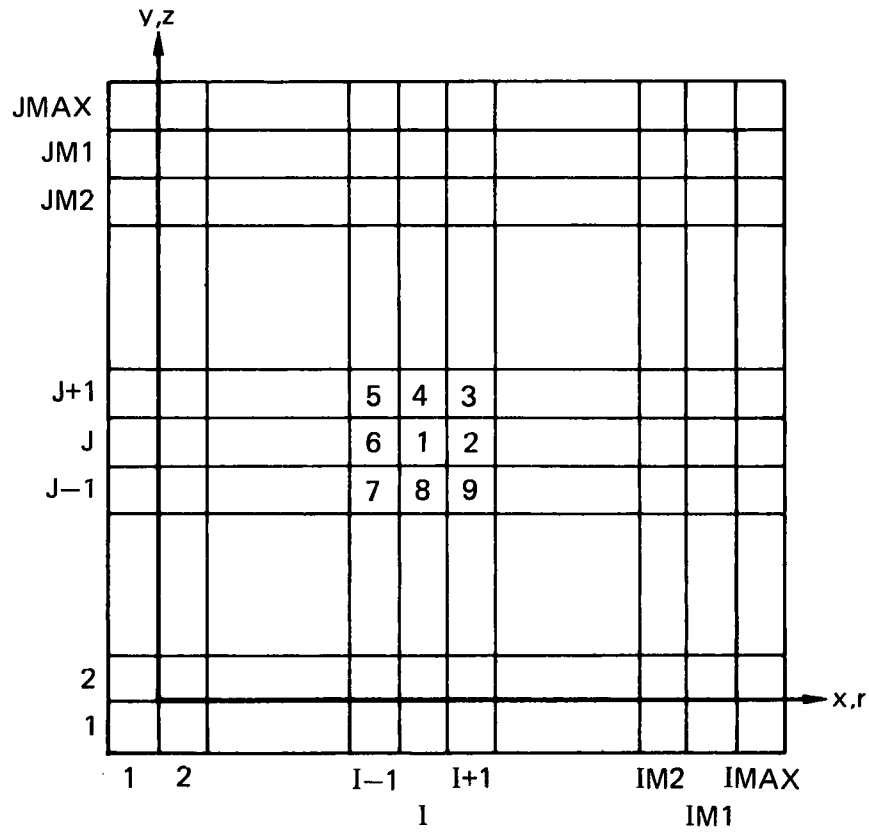


FIG. A.1. General mesh arrangement showing indexing conventions and fictitious boundary cells.

The indexing for cell (I,J) and its eight nearest neighbors is shown in Fig. A.1. This cell is identified by the intersection of I and J grid lines at the upper right-hand corner of the cell. The cell center coordinates are designated by XX(I) and YY(J), and the grid line coordinates are given by XT(I) and YT(J). For problems with cylindrical symmetry (axisymmetric), the x axis becomes the radial coordinate and the y axis is the axial coordinate.

The components of the fluid velocity, u and v are specified on cell sides and represent average values of the velocity components normal to their respective side. This placement of the velocity components makes it easy to preserve the conservative property of the differential equations in the finite difference equations. The space average pressure p is associated with the cell center. These variables are shown in Fig. A.2 with the notation used by the code.

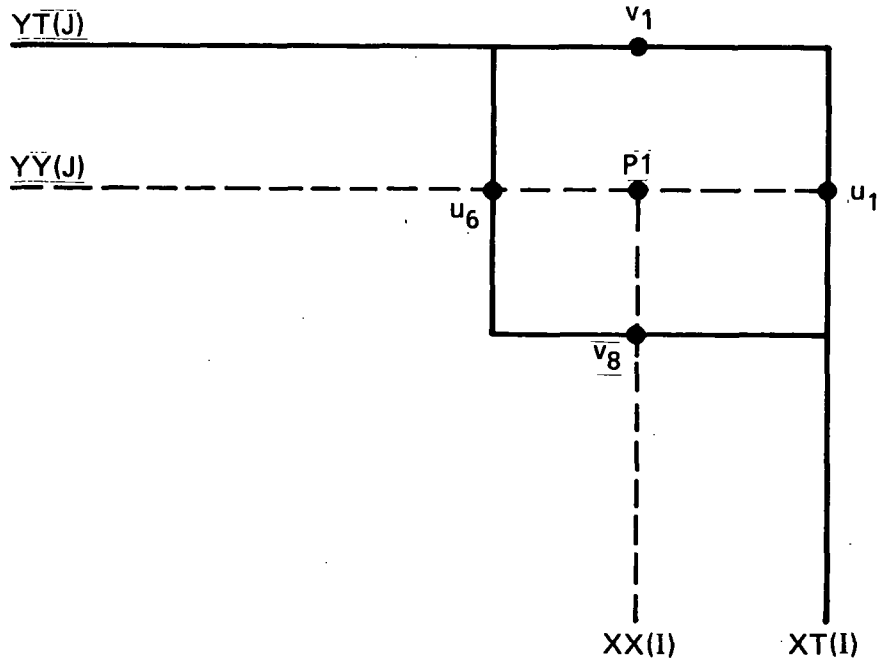


FIG. A.2. Location of finite difference variables in typical cell.

These variables, when related to conventional differencing systems, are indexed, according to the system used, as

$$
\begin{aligned}
p_1 &= p_{ij} \quad \text{or} \quad p_{ij} \\
u_1 &= u_{ij} \quad \text{or} \quad u_{i+1/2,j} \\
v_1 &= v_{ij} \quad \text{or} \quad v_{i,j+1/2} \\
u_6 &= u_{i-1,j} \quad \text{or} \quad u_{i-1/2,j} \\
v_8 &= v_{i,j-1} \quad \text{or} \quad v_{i,j-1/2}
\end{aligned}
\tag{A.1}
$$

The grid size is indicated by the maximum values, IMAX and JMAX, for the I and J indices, respectively. The numbering system used by the code is sequential by column and thus the index for a specific cell is given by

$$K = J + (I - 1)*JMAX.$$

The variables for each cell are stored using the K index, i.e.,

p(K)	pressure,
u(K)	x-velocity on the right side, and
v(K)	v-velocity on top side.

In addition to these primitive variables, each cell has a structured word, CI(K), containing 10 tags used to identify the characteristics of the cell sides as shown in Fig. A.3.

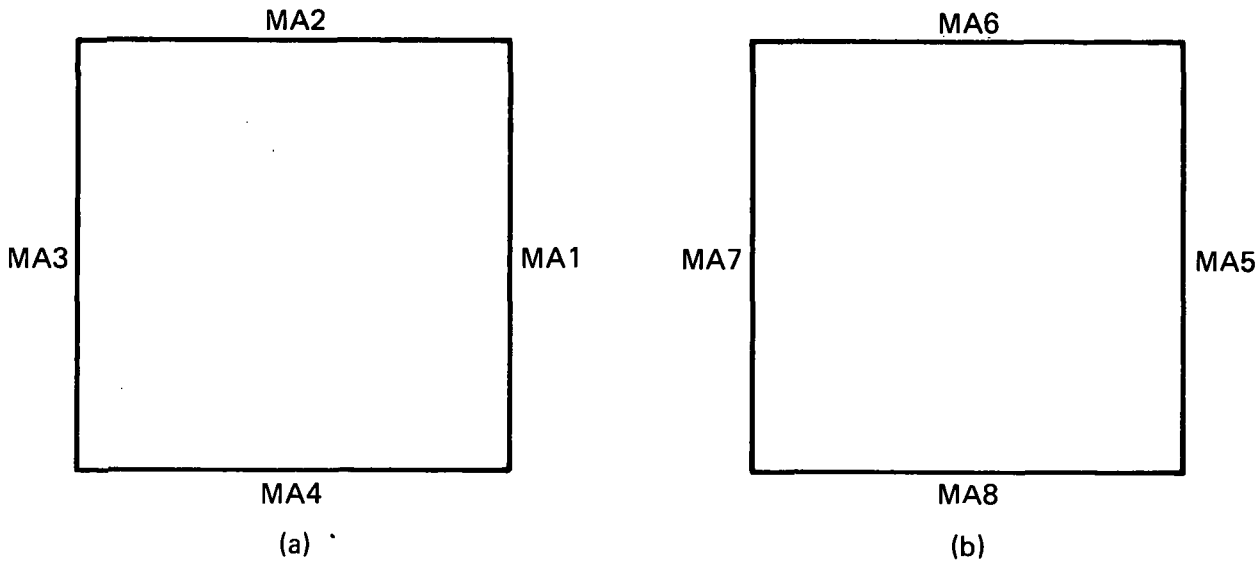


FIG. A.3. Location of tags for cell sides.
(a) Border tags. (b) Intercept tags.

Tags MA1 through MA4 are called border tags and indicate the status of each side and have the following values:

1. Rigid boundary, used for all cells;
2. Liquid side, used for mixed cells only;
3. Mixed side, same;
4. Void side, same.

Tags MA5 through MA8 are called intercept tags and give the positive fluid-void fractional intercepts for each side if there is a free surface in the cell.

Tag MA9 equals 0 if the cell is in the problem and 3 if it is outside as could be in the case when there is a structure in the problem. The calculation ignores the cells tagged as outside the problem. Tag MA10 gives the number of structual intercepts in a cell and can have values from 0 to 2.

SOLA ITERATION TECHNIQUE

Introduction

The velocities computed from the Navier-Stokes equations will not, in general, satisfy the continuity equation. In the MAC method, this incompressibility constraint is imposed by adjusting the cell pressures. For example, if the divergence of a cell is negative corresponding to a net flow of mass into the cell, the pressure is increased to eliminate the inflow. Likewise, when there is a net flow from the cell, the pressure is decreased to draw it back. Because there is one pressure variable for each cell, the divergence for each cell can be driven to zero in this way. The pressure adjustment must be done iteratively, however, because when one cell is adjusted the velocities of its neighbors are also affected. The iteration proceeds by sweeping the mesh rows from left to right starting with the bottom row and working upward. For each cell encountered, the divergence, D, is computed using the most current velocity values available. The pressure change, δp, required to make $D = 0$ is then determined. The iteration proceeds until all computation cells satisfy the criteria that $|D| \leq \varepsilon$, where ε is a small number selected by the user.

General Case

The divergence of the current velocity field is

$$\frac{(u_1 - u_6)}{\delta x} + \frac{(v_1 - v_8)}{\delta y} + \xi \frac{(u_1 + u_6)}{2x} = D \quad . \tag{A.2}$$

The velocity field required for zero divergence is

$$\frac{(u_1^n - u_6^n)}{\delta x} + \frac{(v_1^n - v_8^n)}{\delta y} + \xi \frac{(u_1^n + u_6^n)}{2x} = 0 \quad . \tag{A.3}$$

Subtracting Eq. (A.2) from Eq. (A.3) one obtains

$$\frac{(\Delta u_1 - \Delta u_6)}{\delta x} + \frac{(\Delta v_1 - \Delta v_8)}{\delta y} + \xi \frac{\Delta \bar{u}}{x} = -D \quad ,$$

but

$$\Delta u_1 = \frac{\delta p \delta t}{\delta x} \quad , \qquad \Delta u_6 = \frac{-\delta p \delta t}{\delta x} \quad ,$$

$$\Delta v_1 = \frac{\delta p \delta t}{\delta y} \quad , \qquad \Delta v_8 = \frac{-\delta p \delta t}{\delta y} \quad , \tag{A.4}$$

$$\Delta \bar{u} = 0 \quad ,$$

where

$$\bar{u} = \frac{1}{2}(u_1 + u_6) \quad .$$

Upon substitution of the velocity changes into the correcting equation we have

$$\frac{1}{\delta x}\left(\frac{2\delta p \delta t}{\delta x}\right) + \frac{1}{\delta y}\left(\frac{2\delta p \delta t}{\delta y}\right) = -D \quad . \tag{A.5}$$

Yielding the required pressure change for the iteration,

$$\delta p = \frac{-D}{2\delta t\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2}\right)} \quad . \tag{A.6}$$

When applied to the momentum equation, these corrections have the form

$$u^{n+1} = \tilde{u} - \frac{\delta p \delta t}{\delta y}$$

where the tilde velocities contain all the terms from the Navier-Stokes equations, which were solved using the values known at time t^n. In practice, we multiply δp by an over-relaxation parameter, ω, to help accelerate the convergence.[24] For stability ω has the restriction $0 < \omega < 2$.

Special Case of a Rigid Boundary

The geometric factor in Eq. (A.6) is altered if the cell is bounded by a rigid boundary since, in that case, there is a reduction in the number of degrees of freedom, i.e., there are fewer velocity components we are free to adjust.

As an example, let us consider the bottom boundary of the cell to be rigid and its normal velocity is zero for all time, i.e., $v_8 = 0$. Proceeding in the same manner as before we find

$$\frac{1}{\delta x}\left(\Delta u_1 - \Delta u_6\right) + \frac{1}{\delta_y}\Delta v_1 + \frac{\xi \Delta \bar{u}}{x} = -D \quad . \tag{A.7}$$

Upon substitution of the pressure change relations, we have

$$\frac{1}{\delta x}\left(\frac{2\delta p \delta t}{\delta x}\right) + \frac{1}{\delta y}\left(\frac{\delta p \delta t}{\delta y}\right) = -D \quad , \tag{A.8}$$

which yields the required pressure change for iteration

$$\delta p = \frac{-D}{2\delta t\left(\frac{1}{\delta x^2} + \frac{1}{2\delta y^2}\right)} \quad . \tag{A.9}$$

Similarly for a rigid boundary on a side of the cell

$$\delta p = \frac{-D}{2\delta t\left(\frac{1}{2\delta x^2} + \frac{1}{\delta y^2}\right)} \quad . \tag{A.10}$$

For the corner cell on axis where $u_6 = v_8 = 0$, we have

$$\delta p = \frac{-D}{\delta t\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{\xi}{2x\delta x}\right)} \quad . \tag{A.11}$$

Since we are computing the new pressure field at a cell center based on the cell velocity divergence, we are not required to apply explicitly boundary conditions on pressures at structural interfaces.

FREE-SURFACE TRACKING

Accurate free-surface tracking is necessary to allow the application of appropriate velocity and pressure boundary conditions. Our approach uses the concepts of void fractions and surface orientation to define the free surface. This method automatically guarantees mass conservation in the surface cells. The method is analogous to the marker-particle method since the intercepts used to designate the surface orientation can be considered the equivalent of two marker particles per cell.

Surface Orientation

The cell intercept of a free surface depends on the relative void fractions in each of the two cells within a common side. We can distinguish the four cases shown in Fig. A.4.

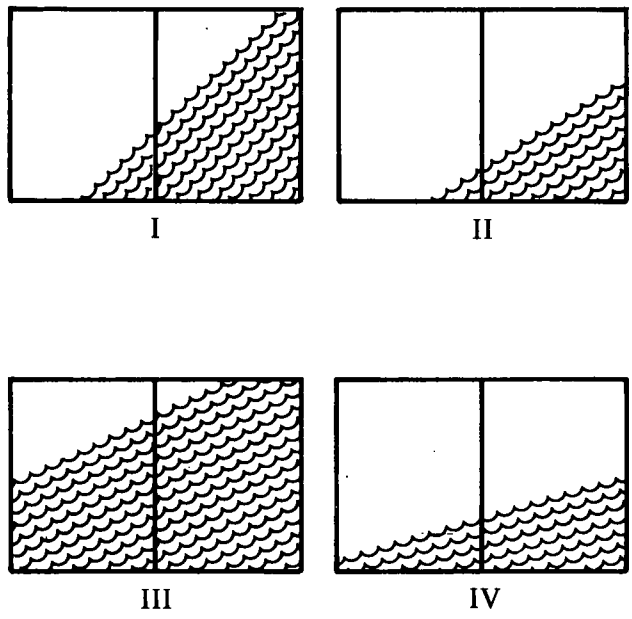


FIG. A.4. Four surface orientations distinguished by the code in determining the interface intercepts.

Figure A.4 shows the condition for side 1 of a cell. Other sides are just rotatations of this basic orientation, and their intercepts are found in the same manner. In the solution algorithm, we indicate the larger of the two void fractions as θ_L and the smaller as θ_S. The cell widths associated with these void fractions are x_L and x_S, respectively, and we define their ratio as

$$R \equiv x_L/x_S \quad .$$

Once the intercept is found, we store the fraction of the cell side as measured in the positive direction.

For finding the intercept f, we make the assumption that the surface is a straight line in the two cells. This will give an approximate location that will be corrected for mass conservation when the advection is calculated.

The formula used to find the cell side intercepts are found by simple geometric considerations with the following results:

Case I

$$f = \frac{\sqrt{R\ (1 - \theta_L)/\theta_S}}{1 + \sqrt{R(1 - \theta_L)/\theta_S}} \quad .$$

Case II

$$f = 2\left\{\sqrt{R(1 - \theta_L)\left[R(1 - \theta_L) + (1 - \theta_S)\right]} - R(1 - \theta_L)\right\} \quad .$$

Case III

$$f = 1 - 2\left\{\sqrt{\frac{\theta_S}{R}\left[\frac{\theta_S}{R} + \theta_L\right]} - \frac{\theta_S}{R}\right\} \quad .$$

Case IV

$$f = \frac{R\theta_S + \theta_L}{R + 1} \quad .$$

For cells on axis, we make the orientation assumptions shown in Fig. A.5.

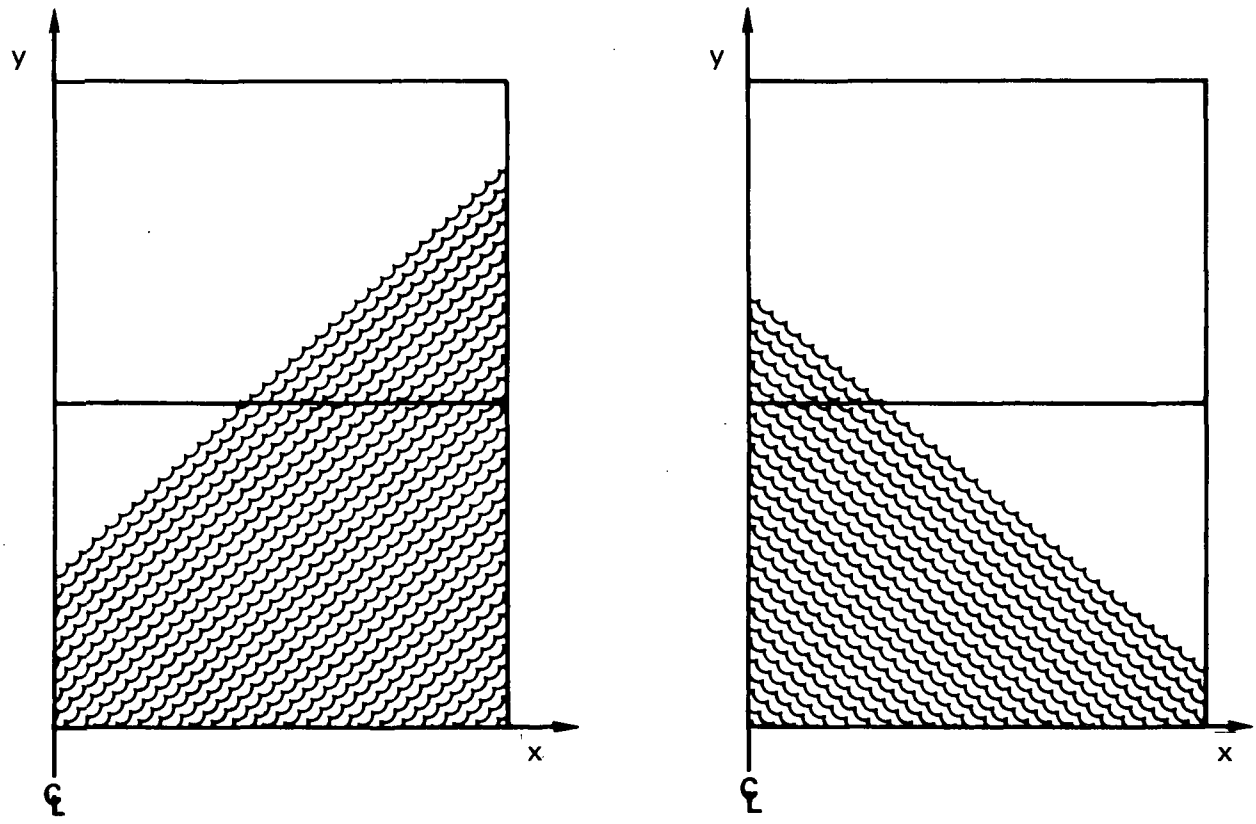


FIG. A.5. Surface of orientation assumptions for cells on axis.

The liquid profile in a mixed cell is assumed to be either a trapezoid or a triangle depending on the location of the intercept as determined above.

Trapezoidal Region

The geometry of the trapezoidal region is as shown in Fig. A.6.

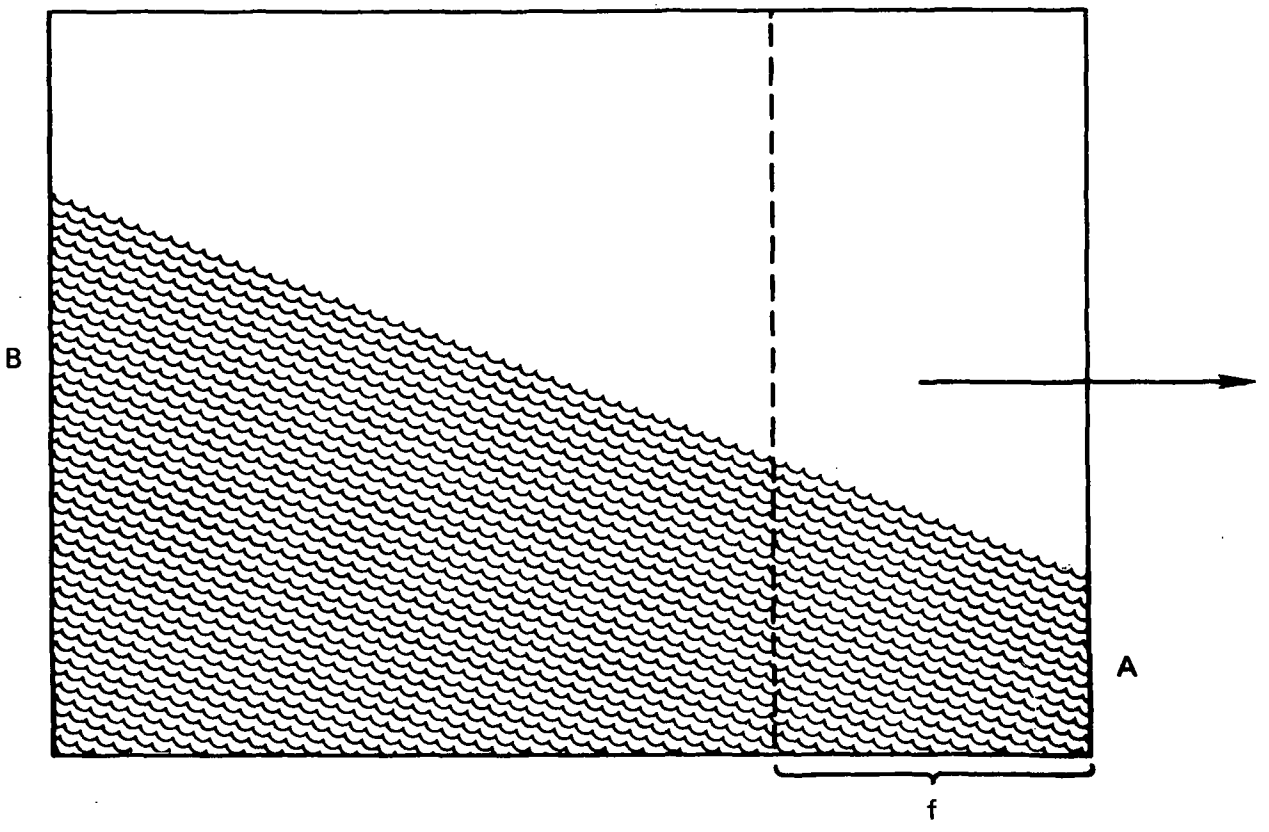


FIG. A.6. Trapezoidal region used in the fluid advection algorithm.

From geometric considerations the fraction of the total liquid advected is given by

$$F = \frac{[A(2-f) + Bf]f}{A + B} \quad .$$

Triangular Region

The geometry of the triangular region is as shown in Fig. A.7.

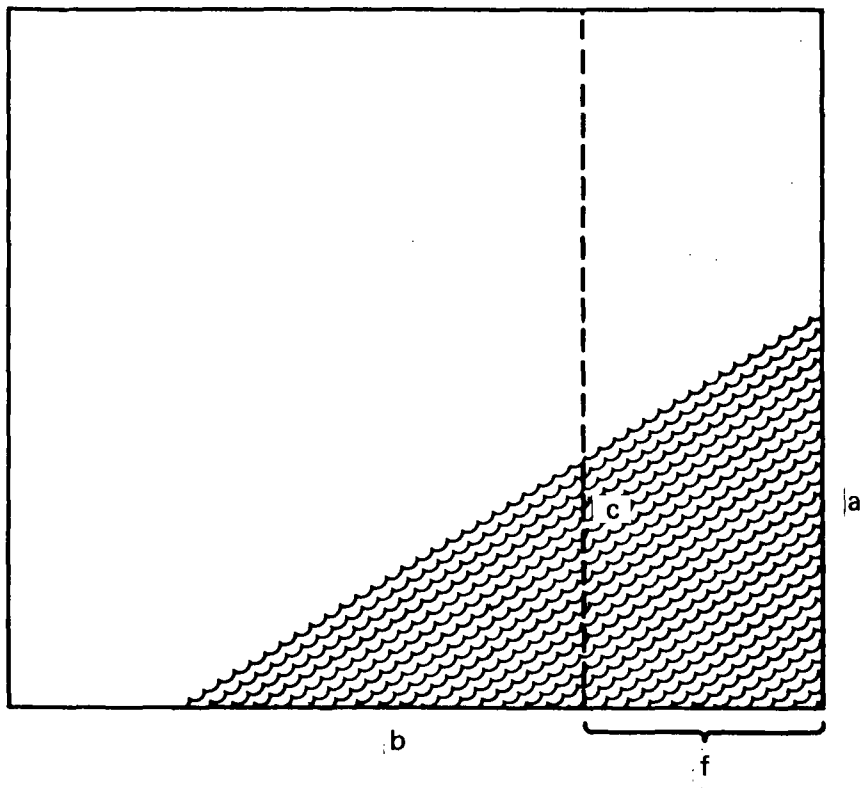


FIG. A.7. Triangular region used in the fluid advective algorithm.

From geometric considerations the fraction of the total liquid advected is given by

$$F = \frac{(a + c)f}{ab} \quad .$$

FLUID-STRUCTURE COUPLING

In coupling the fluid to the structure we require that the normal velocities at the I or J line intercept be equal. The fluid velocity is found by interpolation between the four cell side velocities and the structure velocity is found by interpolating between the nodal values. We use the structure's normal velocity to set the cell side velocity which is outside the structure. This then causes the cell not to satisfy the divergence criteria and the pressure must then be adjusted using the SOLA algorithm.

This different pressure field causes a different structure response; therefore, the iteration proceeds until both the fluid divergence condition and the structural response are compatible.

The velocity set depends on whether the I or J intercept is used. We choose the I intercept if tan $\phi \leq 1$ and the J intercept otherwise, where ϕ is the slope angle of the shell element. When two cell sides lie outside the structure then one velocity is set equal to that of the adjacent fluid cell (slave velocity) before the I or J intercept velocity (master velocity) is set.

If there are no intercepts in the cell, then the two required velocities are slaved to their nearest neighbor.

Setting the I Intercept Velocity

The geometry being considered is shown in Fig. A.8.

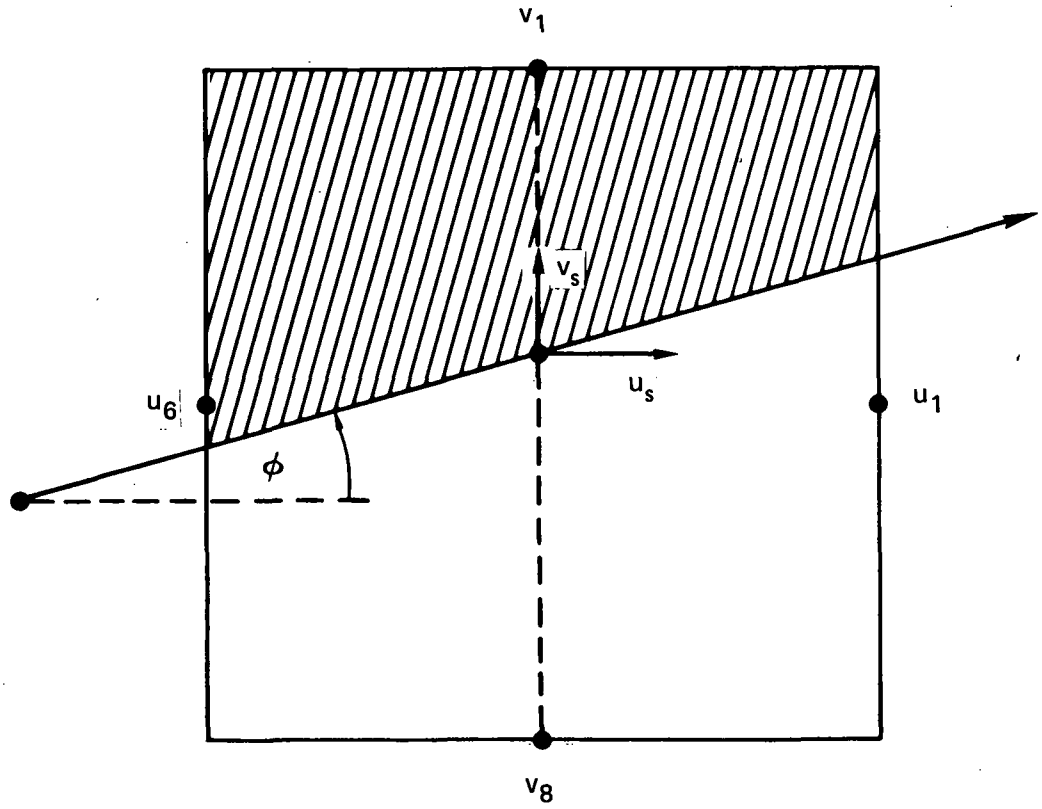


FIG. A.8. Coupling of an unblended fluid-structure interface.

The normal velocity of the structural element at the I intercept is given by

$$w_s = \cos \phi \, (-u_s \tan \phi + v_s) \quad ,$$

where u_s and v_s are the interpolated structure velocity components, and that of the fluid is

$$w_f = \cos \phi \, (-\bar{u} \tan \phi + v_f),$$

where ϕ is the angle of the structural element, $\bar{u} = 0.5 * (u_1 + u_6)$ is the interpolated x-velocity along the I line, and v_f is the v-velocity interpolated between v_1 and v_8 and is dependent on the structure's position in the cell, F. These two velocities must be equal. Thus,

$$v_f = v_s + \tan \phi \, (\bar{u} - u_s).$$

From the interpolation we have

$$v_8 = v_1 + (1 - F)*(v_f - v_1),$$

which yields the value of the velocity outside the structure:

$$v_8 = v_1 + (1 - F)*(v_s - v_1) + \tan \phi \, (\bar{u} - u_s).$$

Note that as F approaches 1, v_8 becomes almost equal to v_1 and the coupling to the structure becomes very weak. To avoid this condition we blend cells whenever F is greater than 0.5. In this case, the geometry being considered is that shown in Fig. A-9.

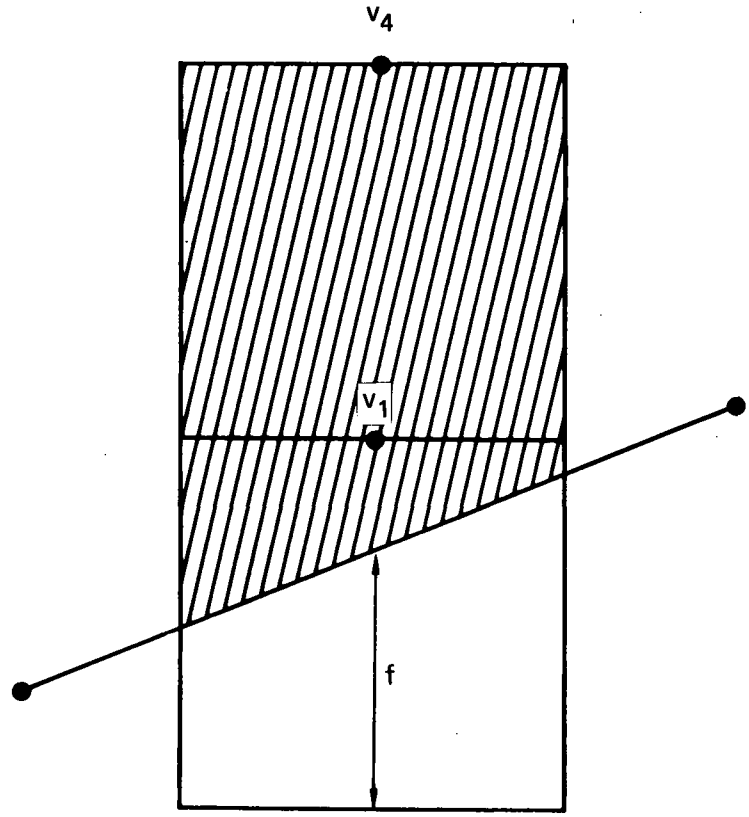


FIG. A.9. Coupling of a blended fluid-structure interface.

The velocity coupled to the finite element shell is v_1 instead of v_8. For equal velocities, we find the cell side velocity v_1 from an interpolation between v_4 and v_s with the resulting formula

$$v_1 = v_4 + \frac{\Delta y_4}{\Delta y_4 + (1 - F)\,\Delta y_1} \left[(v_s - v_4) + \tan \phi\, (\bar{u} - u_s)\right]$$

and we note that as F approaches 1 then v_1 approaches v_s, which gives us the strong coupling desired.

By performing this blending we avoid discontinuities as the structure crosses a grid line.

When we perform this blending we also maintain compatibility in the blended cell by setting

$$u_1 = u_4 \quad \text{and} \quad u_6 = u_7$$

and setting v_8 such that $\nabla \cdot \bar{v} = 0$. The pressure in cell 1 is then set equal to the pressure in cell 4.

FLOW DIAGRAM

The computer code PELE-IC is compartmentalized into a series of special purpose subroutines (Fig. A.10). In this section we show how these subroutines are put together to form the code. In the next section the function of each of the major subroutines is briefly described.

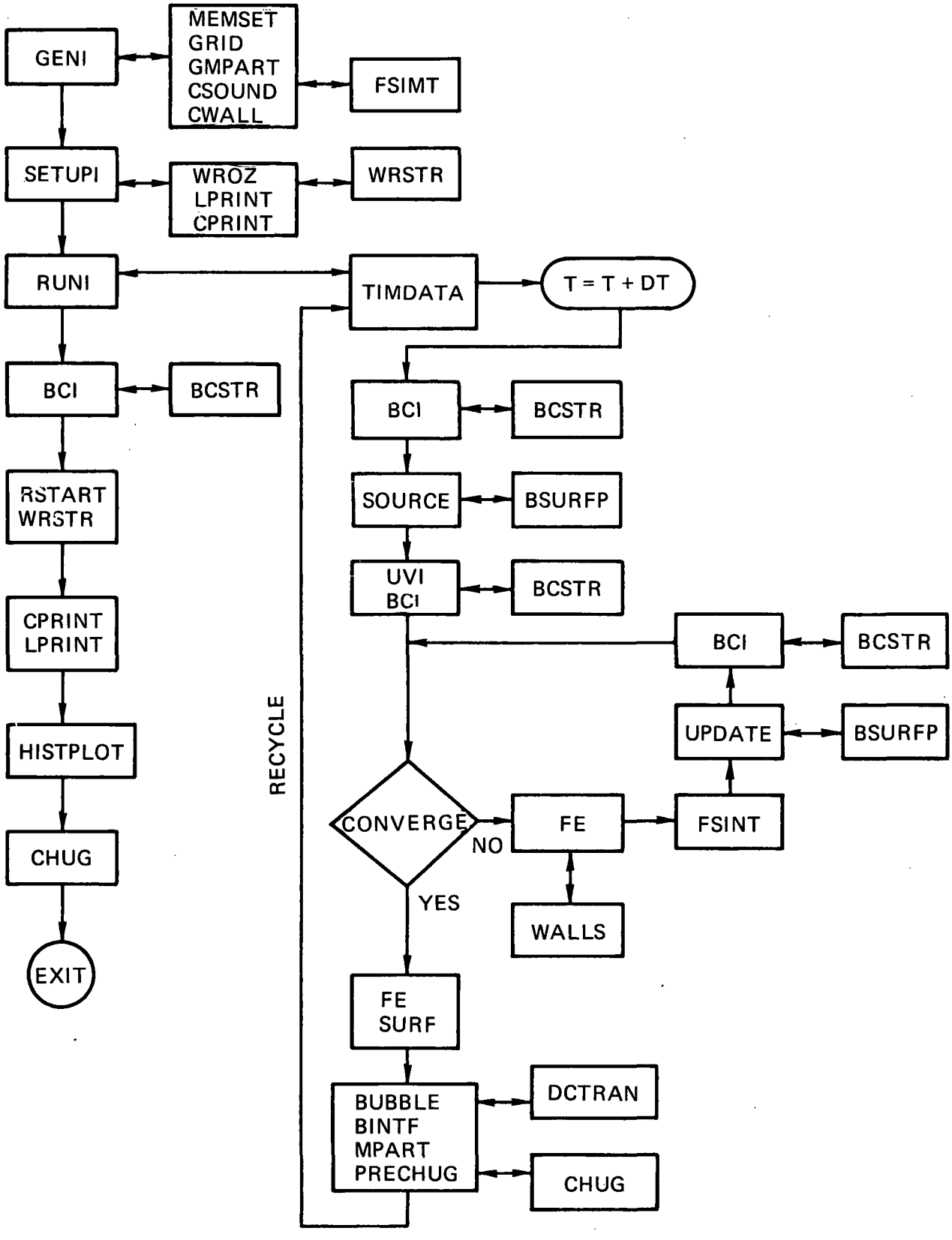


FIG. A.10. PELE-IC flow diagram.

FUNCTIONS OF THE MAJOR SUBROUTINES

The code consists of a large number of subroutines, each designed to perform a separate function. Thus, code improvements and new features can readily be made by changing a specific subroutine or by adding a new subroutine. The following list of major subroutines includes a brief description of the function of each. All subroutines have names that indicate their general function.

BCI-- Sets up Boundary Conditions.
This subroutine supplies the boundary conditions for the edges of the computational grid, all free surfaces, and rigid structures that follow grid lines. The boundary conditions for arbitrary curved structures in the grid are obtained by a call to subroutine BCSTR.

BCBUB-- Boundary Conditions for a bubble interface.
The velocity boundary conditions are set for all cells that contain a phase interface.

BCSTR-- Curved Structure Boundary Conditions.
This subroutine uses the intercept array BSTR to set the cell side velocity such that a linear interpolation to the fluid structure matches the normal velocity of the structure. The I or J-line velocity is used depending on the slope of the structure element.

BUBTAG-- Tag Mixed Cells.
This subroutine sets the tags MA1-MA4, which identify the neighbors of all mixed cells.

BINTF-- Locates Bubble Interface.
This subroutine locates the interface orientation for free surfaces by interrogating neighboring cells. The cell intercept tags MA5 to MA8 in variable CI(K) are set to indicate the intersections of the free surface with the cell sides. These intercepts are used by DCTRAN to calculate the advection of fluid.

BSURFP-- Bubble Surface Pressure.
This subroutine completes the interpolation to find the cell-centered pressure that gives the prescribed pressure time history on the bubble-fluid interface. The subroutine uses the interface orientation to interpolate from a full fluid cell.

BUBBLE-- Advections for Mixed Cells.
This subroutine is the controller for the advection of liquid fractions in mixed cells. It calls subroutine DCTRAN when there is advection between mixed cells. Liquid fractions are advected to maintain conservation of mass. After the advection is calculated for all cells, the new void fractions are computed.

CHUG-- Chugging Equation Solver.
This main routine sets up the call to the Gear package (a differential equation solver) to solve the mass conservation and energy for the Massachusetts Institute of Technology chugging model of Kowalchuk and Sonin.

CONE-- Finite Element Stiffness Matrix.
This subroutine forms the local stiffness matrix and uses ROTATE and ADD to change to global coordinates and assemble the global matrix, respectively.

CWALL-- Generates the Curved Wall.
This subroutine in the generator package is used to generate circular or arbitrarily shaped structural boundaries in the grid.

DCTRAN-- Advection between Mixed Cells.
This subroutine computes the special fluid advection based on the interface orientation of the donor cell. This algorithm allows us to track the interface without the use of marker particles and maintains conservation of mass.

DIFFUN-- Chugging Differential Equations Definition.
This subroutine provides the mass conservation and energy equation to be solved by the gear package.

FE-- Controller for Coupling to Finite Element Code.
This subroutine sets up the current pressure loading on the structure elements for use by the finite element code each iteration. It resets the curved structure array, CSTR, after each iteration for use by the fluid code. After convergence it updates the structures velocity and displacement arrays.

FSINT-- Fluid/Structure Interface.
This subroutine loads the intercept array BSTR, from the curved structure array CSTR, and sets the tag ICSTR that indicates the number of cell intercepts on the structure. Whenever two intercepts

occur in a single cell, the routine selects the master side using the slope of the element to decide between the I and J intercepts. This master side is then used by the coupling algorithm.

GENI-- Generator.

This subroutine reads and processes all input cards for the generation of the problem. The input data is handled by a set of 13 control cards for specifying different types of input information. The grid data arrays are dynamically dimensioned; therefore, we make a call to MEMSET to determine the computer storage requirements for the input grid size.

GMPART-- Sets up Marker Particles.

This subroutine sets up different color marker particles in specified regions of the grid to monitor the fluid flow in the problem. This is done as part of the problem generation.

HISTPLOT--Makes Time History Plots.

This subroutine makes time history plots of source pressure, top surface level, bottom surface level, and pressure for the specified cell.

MEMSET-- Set Memory Size.

This subroutine dynamically sets the computer memory for the size of the problem after the grid size cards are read in.

MPART-- Updates Marker Particle Positions.

This subroutine calculates and relocates the marker particles based on the fluid velocities. The particle velocities used to relocate the particle are computed from an area-weighting scheme involving the nearest cell velocities. The algorithm is based on one used in the MAC code.

PRECHUG--PELE-IC Chugging Interface.

This subroutine acts as an interface between PELE-IC and the Kowalchuk-Sonin chugging model. It computes and provides the chugging model with the variables it needs such as total steam volume, water surface area in contact with the steam, and average surface velocity.

RUNI-- Controller which Runs the Calculation.

This subroutine is the controller for processing the calculation. It calls all major computational subroutines and controls printouts and writing of the restart dumps.

SHELL-- Controller for the Finite Element Code.
This subroutine is the controller for the thin shell finite element code. It calls CONE to form the stiffness matrix, does the Newmark dynamics, and calls SYMBC to invert the stiffness matrix. A switch allows the stiffness reformulation to be bypassed during the fluid-structure compatibility iterations.

SOURCE-- Applies Boundary Pressure Sources.
This subroutine determines the current boundary pressure from the inputed pressure time history. It calls subroutine BSURFP whenever a mixed cell is found.

SURF-- Computes New Surface Positions.
This subroutine tracks free top and bottom surfaces by explicit liquid advection using the updated velocity field.

SYMBC-- Finite Element Solver.
This subroutine solves the symmetric banded equations of the shell code in a single-subscript arithmetic using Gauss elimination.

TAGSTR-- Sets Tags for Zones Outside Structure.
This subroutine sets the tag MA9 = 3 for all cells outside the structure. Other subroutines then skip over all cells with this tag set, thus reducing the number of cells swept during the calculation.

TIMDATA--Accumulate Time History Data.
This subroutine accumulates time history data for source pressure, top surface level, bottom surface level, and pressure at specified cells. These data are then used by subroutine HISTPLOT for graphics output.

UPDATE-- Updates Cell Pressures and Velocities.
This subroutine updates the pressure field and finds the new velocities during the iteration. It applies the special boundary conditions for free surfaces. It also calls subroutine BCI to apply all other boundary conditions and then makes a divergence check on all cells. If the divergence criteria is met, a flag is set to end the iteration.

UVI -- Solves the Navier-Stokes Equations
This subroutine solves the Navier-Stokes equations explicitly to obtain the trial velocities $\tilde{u}$ and $\tilde{v}$ to start the iteration.

APPENDIX B. STRUCTURE ALGORITHM DETAIL

This appendix presents some details required for the calculation of the stiffness matrix of Eq. (16). For constant material properties within an element define

$$X = \int_{s_1}^{s_2} (\phi')^T \; (\phi') \; r\,ds$$

$$Y = \int_{s_1}^{s_2} (\phi')^T \; (\phi) \; r\,ds \qquad \text{(B.1)}$$

$$Z = \int_{s_1}^{s_2} (\phi)^T \; (\phi) \; r\,ds$$

$$\bar{Y} = \int_{s_1}^{s_2} (\phi')^T \; (\phi) \; ds$$

$$\bar{Z} = \int_{s_1}^{s_2} (\phi)^T \; (\phi) \; \frac{ds}{r} \quad ,$$

where ϕ (s) is the piecewise-linear interpolation function

$$\phi(s) = \frac{1}{\Delta s} \left\langle s_2 - s \, , \, s - s_1 \right\rangle \, . \tag{B.2}$$

Equations (B.1) are integrated using one-point quadrature putting $r = \bar{r}$ $= 1/2(r_1 + r_2)$ to give

$$X = \frac{\bar{r}}{\Delta s} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$Y = \frac{\bar{r}}{2} \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

$$Z = \frac{\bar{r}\Delta s}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{B.3}$$

$$\bar{Z} = Z/\bar{r}^2$$

$$\bar{Y} = \frac{1}{\bar{r}} Y \, .$$

The constants of the stiffness matrix K of Eq. (16) are

$$A = \bar{D} \left[X + \nu \sin (V + V^T) + \sin^2 \bar{Z} \right]$$

$$B = \bar{D} \left[\nu Y + \sin \alpha \, \bar{Z} \right] \cos \alpha$$

$$C = \bar{D} \, \bar{Z} \cos^2 \alpha + \frac{5}{6} Gh \, X \tag{B.4}$$

$$D = \frac{5}{6} Gh \, Y$$

and

$$E = \bar{D} \left[X + \nu(\bar{Y} + \bar{Y}^T) \sin \alpha + \sin^2 \alpha \bar{Z} \right] + \frac{5}{6} Gh \, Z \, .$$

The above equations are in a local coordinate system. The rotation to global coordinates is obtained with the rotation matrix

$$T = \begin{bmatrix} sn & cs & 0 \\ -cs & sn & 0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{B.5}$$

where we redefine sn = sin (90-α) and cs = cos (90-α). Then

$$K' = T^T KT , \tag{B.6}$$

where K is given by Eq. (16). The result is

$$K' = \left[\begin{array}{ccc|c} Asn^2 - (B+B^T)sncs + Ccs^2 & Bsn^2 + (A-C)cssn - B^Tcs^2 & -Dcs \\ B^Tsn^2 + (A-C)sncs - Bcs^2 & Csn^2 + (B+B^T)\,cssn + Acs^2 & +Dsn \\ -D^Tcs & D^Tsn & E \end{array}\right] . \tag{B-7}$$

The mass matrix is lumped by the assumption

$$M = \int_L \rho h \, \phi \, r ds , \tag{B.8}$$

which with one point quadrature becomes

$$M_1 = M_2 = \frac{1}{2} \rho h \, \bar{r} \, \Delta s . \tag{B.9}$$

For plane geometry, the shell of revolution becomes a curved plate deformed cylindrically. The constitutive form stays the same for plane strain, but for plane stress, $\bar{D} = Eh$. In the finite element formulation (using $\bar{r} = 1$ and $\bar{Y} = \bar{Z} = 0$), we have

$$\begin{bmatrix} DX & 0 & 0 \\ 0 & \frac{5}{6}\,Gh\,X & \frac{5}{6}\,Gh\,Y \\ 0 & \frac{5}{6}\,Gh\,Y^{T} & \bar{D}X + \frac{5}{6}\,Gh\,Z \end{bmatrix} \begin{Bmatrix} \hat{u} \\ \hat{w} \\ \hat{\beta} \end{Bmatrix} = \begin{Bmatrix} 0 \\ P \\ 0 \end{Bmatrix} . \qquad (B.10)$$

Note that membrane action and bending are uncoupled in the element and curvature induced coupling is introduced only by the local to global transformation at the nodes.

REFERENCES

1. C. W. Hirt, B. D. Nichols, and N. C. Romero, SOLA--A Numerical Solution Algorithm for Transient Fluid Flow, Los Alamos Scientific Laboratory, Los Alamos, N. Mex., LA-5652 (1975).
2. L. L. Edwards, R. B. Hickman, J. K. Hobson, and T. C. Michels, CHAMP: A Coupled HEMP and Multifluid Eulerian Program for Fluid Flow Simulations, Lawrence Livermore Laboratory, Livermore, Calif., UCRL-52444 (1978).
3. L. L. Edwards, Computer Simulations of a 1/5 -Scale Experiment of a Mark I Boiling Water Reactor Pressure-Suppression System under Hypothetical LOCA Conditions, Lawrence Livermore Laboratory, Livermore, Calif., UCID-17782 (May 1978).
4. F. H. Harlow and J. E. Welch, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow," Phys. Fluids 8, 2182 (1965).
5. G. L. Goudreau, A Computer Module for One Step Dynamic Response of an Axisymmetric or Plane Linear Elastic Thin Shell, Lawrence Livermore Laboratory, Livermore, Calif., UCID-17730 (1978).
6. W. Kowalchuk and A. Sonin, A Model for Condensation Oscillations in a Vertical Pipe Discharging Steam into a Subcooled Water Pool, MIT Report for NRC (1978).
7. Harry Kraus, Thin Elastic Shells (John Wiley & Sons, Inc., New York, 1967).
8. T. J. R. Hughes and R. L. Taylor, "A New Finite Element for Plate Bending Analysis," SMIRT Proc. M 2/1 (1977).
9. O. C. Zienkiewicz, J. Bayer, K. Morgan, and E. Omate, "A Simple and Efficient Element for Axisymmetric Shells," Int. J. Numerical Methods in Eng. 11 (10), 1545 (1977).
10. G. L. Goudreau and R. L. Taylor, "Evaluation of Numerical Integration Methods in Elasto Dynamics," in Computational Methods in Applied Mechanics and Engineering, vol. 2, N1 (1973) pp. 69-97.
11. A. A. Amsden and F. H. Harlow, The SMAC Method: A Numerical Technique for Circulating Incompressible Fluid Flows, Los Alamos Scientific Laboratory, Los Alamos, N. Mex., LA-4370 (1970).

12. A. C. Hindmarsh, Gear Ordinary Differential Equation System Solver, Lawrence Livermore Laboratory, Livermore, Calif., UCID-30001, Rev. 3 (1974).
13. A. G. Hansen, Fluid Mechanics (John Wiley & Sons, Inc., New York, 1967), pp. 134-138.
14. J. W. S. Rayleigh, Phil Mag. 34, 94 (1917).
15. J. D. Parker, J. H. Boggs and E. F. Blick, Introduction to Fluid Mechanics and Heat Transfer (Addison-Wesley Publishing Company, Inc.; Reading, Mass., 1969), p. 345.
16. G. G. Stokes, "On Some Cases of Fluid Motion," Proc. Cambridge Philosophical Soc. 8, 1-5, 137 (1843).
17. H. Lamb, "On The Vibrations of an Elastic Plate in Contact with Water," Proc. Roy. Soc. (London) A, XCVIII, 206-216 (1921).
18. J. O. Hallquist, "NIKE2D--An Implicit, Finite Deformation, Finite Element Code for Analyzing the Static and Dynamic Response of 2-D Solids, to be published as a Lawrence Livermore Laboratory UCRL Report.
19. S. Timoshenko and A. Woinowsky-Krieger, Theory of Plates and Shells (McGraw-Hill, Inc., New York, 1959), 2nd ed. p. 407.
20. W. G. Anderson, P. W. Huber, and A. A. Sonin, Small Scale Modeling of Hydrodynamic Forces in Pressure Suppression Systems, Massachusetts Institute of Technology, Final Report, NUREG/CR-0003, prepared for the U.S. Nuclear Regulatory Commission, Dec. 1977.
21. C. K. Chan, V. K. Dhir, H. H. Chiou, C. Y. Lin, B. K. C. Lee, and I. Catton, Suppression Pool Dynamics and Annual Report, University of California, Los Angeles, Report NUREG-0264-3, prepared for the U.S. Nuclear Regulatory Commission, July 1, 1976 to June 30, 1977.
22. J. H. Pitts and E. W. McCauley, "Vertical Forces Induced in a 1/5 scale Mark I BWR Toroidal Wetwell Under LOCA Conditions," Paper B4/6 in Trans. 4th International Conference on Structural Mechanics in Reactor Technology, San Francisco (1977).
23. W. H. McMaster and E. Y. Gong, PELE-IC Users Manual, Lawrence Livermore Laboratory, Livermore, Calif., UCRL-52609 (1979).
24. W. F. Ames, Numerical Methods for Partial Differential Equations (Barnes & Noble, New York, 1969).

NM/dls

NRC FORM 335 (7-77)

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

1. REPORT NUMBER *(Assigned by DDC)*: NUREG/CR-0607

4. TITLE AND SUBTITLE *(Add Volume No., if appropriate)*: Coupled Fluid-Structure Method for Pressure Supression Analysis

2. *(Leave blank)*

3. RECIPIENT'S ACCESSION NO.

7. AUTHOR(S): W.H. McMasters and others.

5. DATE REPORT COMPLETED: MONTH February YEAR 1979

9. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS *(Include Zip Code)*:
Lawrence Livermore Laboratory
Livermore, CA. 94550

DATE REPORT ISSUED: MONTH May YEAR 1979

6. *(Leave blank)*

8. *(Leave blank)*

12. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS *(Include Zip Code)*:
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D. C. 20555

10. PROJECT/TASK/WORK UNIT NO.

11. CONTRACT NO.: NRC FIN No.A0116

13. TYPE OF REPORT

PERIOD COVERED *(Inclusive dates)*

15. SUPPLEMENTARY NOTES

14. *(Leave blank)*

16. ABSTRACT *(200 words or less)*

This report discusses an effective method of fluid-structure coupling for application in blowdown suppression analysis in boiling-water reactors. A computer program couples an incompressible, Eulerian fluid algorithm with a finite element shell algorithm. The fluid algorithm provides a trial velocity field using the Navier-Stokes equations that, at each time step, is corrected iteratively to satisfy the incompressibility condition and fluid-structure interface compatibility. The code is verified by comparing calculations of classical problems and experiments. Computer simulations of air blowdown and steam chugging in pool-suppression geometries are also presented.

17. KEY WORDS AND DOCUMENT ANALYSIS

17a. DESCRIPTORS

17b. IDENTIFIERS/OPEN-ENDED TERMS

18. AVAILABILITY STATEMENT: Unlimited

19. SECURITY CLASS *(This report)*: Unclassified

20. SECURITY CLASS *(This page)*

21. NO. OF PAGES

22. PRICE: S

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D. C. 20555

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
U.S. NUCLEAR REGULATORY
COMMISSION