UCRL- 85226
PREPRINT

CONF - 810664 - - 1

# SIMULATION OF THREE-DIMENSIONAL, TIME-DEPENDENT, INCOMPRESSIBLE FLOWS BY A FINITE ELEMENT METHOD

Stevens T. Chan
Philip M. Gresho
Robert L. Lee
Craig D. Upson

This paper was prepared for presentation at the AIAA 5th Computational Fluid Dynamics Conference, Palo Alto, California, June 22-23, 1981

May 1981

Lawrence Livermore Laboratory

# SIMULATION OF THREE-DIMENSIONAL, TIME-DEPENDENT, INCOMPRESSIBLE FLOWS BY A FINITE ELEMENT METHOD

Stevens T. Chan, Philip M. Gresho, Robert L. Lee, and Craig D. Upson

Lawrence Livermore National Laboratory, University of California
Livermore, California 94550

## Abstract

A finite element model has been developed for simulating the dynamics of problems encountered in atmospheric pollution and safety assessment studies. The model is based on solving the set of three-dimensional, time-dependent, conservation equations governing incompressible flows. Spatial discretization is performed via a modified Galerkin finite element method, and time integration is carried out via the forward Euler method (pressure is computed implicitly, however). Several cost-effective techniques (including subcycling, mass lumping, and reduced Gauss-Legendre quadrature) which have been implemented are discussed. Numerical results are presented to demonstrate the applicability of the model.

## Introduction

Our modeling goals include the development and application of cost-effective techniques for simulating the evolution of the velocity, temperature, and pollutant concentration fields associated with air flow over complex terrain in the planetary boundary layer. Typical near-term applications of these models are related to energy production options and include: (1) simulation of the dynamics associated with gravitational spreading and atmospheric dispersion of heavy gas releases into the atmosphere, and (2) nocturnal drainage winds and pollutant transport in regions characterized by rugged, mountainous terrain. Longer term applications will also focus on real time safety issues related to accidental releases of hazardous materials such as radioactive gases (e.g. Three Mile Island).

We plan to achieve these goals by solving the appropriate governing equations (Boussinesq equations or a variant thereof) approximately via a modified finite element method. Since both the physics (e.g., stratified flows) and the terrain are complex, and fairly accurate solutions are desired, it appears that many node points (on the order of $10^4$) will often be required. Further, our longer term goal of providing a code to be used for real time response to emergencies obviously requires that the running speed be significantly faster than real time. It is with these points in mind that we developed the 3-D FEM code which is described in this paper.

In order to obtain a working code in minimum time, we have thus far emphasized simplicity. This led to: (1) the choice of the simplest element - the 8-node isoparametric "brick" employing piecewise trilinear approximating functions for velocity, temperature, and concentration, and piecewise-constant approximation for the pressure, and (2) the use of the simplest time integration method - explicit (forward) Euler. The pressure, being an inherently implicit variable in an incompressible fluid, is, of course, treated implicitly.

In what may be called "Phase 1" of our first 3-D code, we followed most of the rules of the Galerkin method and generated a code with only two ad hoc modifications ("cheats" on the Galerkin method) which we summarize here (see Ref. 1 for further details): (1) the mass matrices, which couple the time derivatives in the honest FEM, were replaced by diagonal (lumped mass) matrices much like the typical finite difference approach, (2) the nonlinear advection terms (e.g. $\underline{u}\cdot\nabla T$) were a priori modified to permit simpler and faster computation (in essence, the conventional triply-subscripted coefficients were replaced by simpler doubly-subscripted ones by employing the element centroid values for the advecting velocities). Even with these simplifications, however, the resulting code was rather expensive and real time simulations appeared to be out of reach; for example, a simulation of heavy gas dispersion required about three hours of computer time on the CRAY-1 (1/3 CPU, 2/3 I/O) to simulate several minutes of real time.[1]

In order to generate a faster, more vectorized code, we modified the "Phase 1" version to "Pnase 2" by two further cost-effective simplifications, which we will summarize and demonstrate in this paper and discuss in more detail in subsequent publications.[2,3] The principal additional simplification is the use of one-point Gauss-Legendre quadrature (rather than 2x2x2 or higher) to evaluate the element level integrals associated with the FEM. This approximation, which tends to result in a discretized model which is perhaps better described as a *blend of finite elements and finite differences*, leads to significant cost reduction in two areas: (1) the element "matrices" are computed as needed ("on the fly") rather than being stored on disk (this storage accounted for most of the I/O cost in the Phase 1 code) and (2) the entire algorithm is more amenable to efficient vectorization. The second major simplification is associated with the time integration aspect of the simulation and is referred to as "subcycling." Briefly, this trick permits us to reduce the frequency of the expensive (again, mostly I/O) pressure update calculation by using a combination of four items: (1) the major time steps are based on temporal accuracy and are dynamically computed via local truncation error estimates, (2) the minor (smaller) time steps, based on stability estimates, are used to compute advection and diffusion with a simple extrapolation approximation employed for the pressure gradient, (3) a mass adjustment scheme is employed at each major time step to re-enforce the satisfaction of the continuity equation, and (4) the corresponding (compatible) pressure field is computed.

In the remainder of the paper, we present one form of the governing conservation equations, briefly describe the finite element spatial discretization process and the time integration scheme, elaborate on the two new cost-effective techniques referred to above, and present two numerical examples which demonstrate the effectiveness of the Phase 2 code by comparing results with those from the Phase 1 version (which has already been briefly "verified" by comparison with some finite difference results,see[1]).

## Governing Equations and Spatial Discretization

The principal set of equations of interest here are the equations of motion, continuity, and energy conservation *for a constant property, incompressible Newtonian fluid* in the Boussinesq approximation (turbulence parameterization via K-theory, or better, will come later):

$$\rho\,(\partial\underline{u}/\partial t + \underline{u}\cdot\nabla\underline{u}) = -\nabla P + \mu\nabla^2\underline{u} - \rho\gamma\underline{g}T \quad , \tag{1a}$$

$$\nabla\cdot\underline{u} = 0 \quad , \tag{1b}$$

$$\partial T/\partial t + \underline{u}\cdot\nabla T = \kappa\nabla^2 T \quad , \tag{1c}$$

where $\underline{u} = (u, v, w)$ is the velocity, P is the pressure, T is the temperature deviation from a reference level, $\rho$ is the density (constant, evaluated at the reference temperature), $\mu$ is the viscosity, $\gamma$ is the volumetric coefficient of thermal expansion, $\underline{g}$ is the gravitational acceleration (directed opposite to the vertical coordinate), and $\kappa$ is the thermal diffusivity. Given appropriate initial data for velocity and temperature (appropriate means, basically, that the initial velocity field must be solenoidal -- $\nabla\cdot\underline{u}_0 = 0$) and appropriate boundary conditions,[4] Eq. (1) can be used to solve for the velocity, pressure, and temperature as functions of space and time.

## Basic Spatial Discretization

The finite element spatial discretization of Eq. (1) is performed using the Galerkin method via the following expansion in the piecewise polynominal basis functions typical of the FEM,

$$\underline{u}^h(\underline{x},t) = \sum_{i=1}^{n} \underline{u}_i(t)\varphi_i(\underline{x}) \quad , \tag{2a}$$

$$T^h(\underline{x},t) = \sum_{i=1}^{n} T_i(t)\varphi_i(\underline{x}) \quad , \tag{2b}$$

and

$$P^h(\underline{x},t) = \sum_{i=1}^{m} P_i(t)\psi_i(\underline{x}) \quad , \tag{2c}$$

where, in the discretized domain, there are n nodes for velocity and temperature and m "nodes" for pressure (one per element); the superscript h indicates a finite dimensional approximation.

Inserting Eq. (2) into the weak (Galerkin) form of Eq. (1) (which reduces differentiability requirements: $\varphi_i$ can then be continuous with piecewise-discontinuous first derivatives and $\psi_i$ can be piecewise discontinuous), leads to the following set of ordinary differential equations (ODE's) -- the Galerkin FEM equations, written in a compact matrix form,

$$M\dot{U} + [K + N(U)]\,U + CP = f \tag{3a}$$

$$C^T U = 0 \quad , \tag{3b}$$

and

$$M_S\dot{T} + [K_S + N_S(U)]\,T = f_S \quad . \tag{3c}$$

Now U is a global vector of length 3n containing all nodal velocity components, P is a global m-vector of elemental pressures, and T is a global n-vector of nodal temperatures; f is a global vector which incorporates buoyancy forces and any traction boundary conditions on velocity, and $f_S$ incorporates any of the natural boundary conditions associated with Eq. (1c). Specified nodal values of velocity and temperature are imposed directly on the assembled system. M is the mass matrix, $M_S$ is an appropriate submatrix of M, K is the viscous matrix, $K_S$ is the thermal diffusion matrix, C is the pressure gradient matrix and its transpose, $C^T$, is the divergence matrix, N(U) and $N_S(U)$ are the advection matrices. (For further details see Ref. 4.)

## Extension to Anelastic Equations

One of the numerical examples to be discussed does not employ the Boussinesq equations since the density variation is quite large. We have, accordingly, generalized the anelastic equations[5] to permit large density changes on the premise that the important fluid dynamics are still basically incompressible and that acoustic waves are therefore unimportant and can be filtered a priori. The key ingredients of these equations are: (1) the replacement of the continuity equation by $\nabla\cdot(\rho\underline{u}) = 0$ where $\rho$ is obtained from an equation of state (ideal gas law), (2) the replacement of $\underline{u}$ by $\rho\underline{u}$ (momentum) as a principal unknown (i.e. $\underline{u}_i$ is replaced by $(\rho\underline{u})_i$ in Eq. (2a). For further discussion of these equations, see Ref. 1.

## One-Point Quadrature

As mentioned earlier, we have resorted to the approximation referred to as 1-point quadrature, in which all element integrals are approximated by their value at the element centroid multiplied by the element volume. The reasons for resorting to such a simplification are: (1) the I/O cost of storing (more accurate) element level information on disc and retrieving it at every time step is very high (especially on the CRAY, where CPU performance is quite high relative to I/O) and (2) the cost of recomputing all integrals at every time step using a more accurate Gauss rule is also too high - by about an order of magnitude.

We note first that the idea is not new; it has already been successfully employed in explicit FEM solid mechanics codes,[6,7] and it is this fact which encouraged us to try it. This trick, however, is not totally free of problems, as we discuss below.

Element Volume. Since it has been claimed[8] that convergence of the FEM requires, among other things, an exact calculation of element volume, we have not (yet) cheated in this area. We use a 2x2x2 Gauss rule to compute element volumes and store the resulting vector in core.

Gradient and Divergence. Although we are well aware that 1-point quadrature cannot integrate the C matrix (see Eqn. 3) correctly on the general distorted element,[9] and thus we probably do not retain the element level mass balances associated with the correct integration, preliminary results seem to indicate that the resulting errors are quite acceptable. We will continue to explore this point and report further findings in a subsequent publication.[2]

Advection. We first describe the resulting approximation to $\underline{u}\cdot\nabla T$ on a regular mesh: the average (centroid) velocity in an element is multiplied by the average temperature gradient in the element and the result is averaged over all elements (typically 8) sharing the node in question. For distorted elements, the interpretation appears to be similar except that a volume-weighted final average is employed (using, as mentioned earlier, exact element volumes). This approximation appears to be reasonable and well-behaved.

Diffusion. The biggest problem thus far encountered with 1-point quadrature is the approximation to the Laplacian operator (K, $K_S$ in Eq. 3). This problem can be described very simply — the diffusion matrix is singular -- but is subject to various interpretations. In the Lagrangian codes of solid mechanics, the problem is described in terms of so-called hourglass patterns[10] and is interpreted in terms of zero energy deformations or mode shapes. In fluid mechanics (in an Eulerian reference frame), it seems more appropriate to associate the problem with the common "2Δx" waves; these waves (null vectors of the singular matrix) are not diffused by the 1-point quadrature diffusion matrix. While there is only one such wave in 2-D (associated with alternating nodal values of +1 and -1), there are four in 3-D and the associated nodal values (on a single element) are shown in Fig. 1. The first of these is the three-dimensional wave and the other three are two-dimensional, one in each plane.

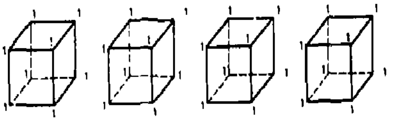

*Fig. 1 "2Δx" wave patterns on a single element.*

The "patch job" for this deficiency, which is also borrowed from the solid mechanics community, is to augment the singular matrix with an "hourglass correction matrix", a procedure which appears to be both simple and effective. Rather than using the more rigorous (and expensive) corrections suggested by Kosloff,[10] we employ the simpler procedure developed by Hallquist and Goudreau;[11] viz. the hourglass matrix is defined, element-wise, by an outer product of the associated null vector with itself. This matrix is then multiplied by an "appropriate" scalar ("tuning"; see below) and added to the singular 1-point diffusion matrix. Figure 2 shows the stencils (for a cube of side length $\ell$) associated with several discrete approximations to $\nabla^2$ as well as the three-dimensional hourglass matrix, $K_{HG1}$. (Note: All the matrices except $K_{HG1}$ have been multiplied by $48\ell^2$.) Noteworthy here is that $K_{1p}$ and even $K_{2p}$ (2-point quadrature - exact for a brick) look rather "suspicious" in that the coefficients closest to the central node are either positive or zero, in marked contrast to the well-known finite difference stencil.

A better perspective may be gained by studying the behavior of these schemes on an appropriate prototypical test case, pure transient diffusion. If the "heat equation,"

$$\partial T/\partial t = \alpha\nabla^2 T \quad , \qquad (4)$$

is solved on the 3-D infinite span with an initial condition given by

$$T(0) = e^{ik(x+y+z)} , \qquad (5)$$

the "sine-wave" diffuses according to the exact solution,

$$T(x,y,z,t) = T(0)e^{-3\alpha k^2 t} \quad , \qquad (6)$$

Through a Fourier analysis, the effective diffusivity ($\bar{\alpha}$) associated with each scheme can be obtained. Figure 3 shows the ratio of the effective diffusivity to the desired diffusivity, $\bar{\alpha}/\alpha$, of the discrete forms mentioned above, as a function of wave number, with $k\ell$ varying from 0 to $\pi$ (the associated wavelength varies from infinity to the limit of the grid, $2\Delta x$). The suspicion associated with the stencils corresponding to $K_{1p}$ and $K_{2p}$ is indeed verified - as is the singularity of $K_{1p}$ to the $2\Delta x$ wave. As expected, $K_{FD}$ performs much better. The much-improved behavior of the augmented 1-point matrix is shown by the dashed line (here, and in the numerical results presented below, a scalar multiplier of 1/2 was employed).

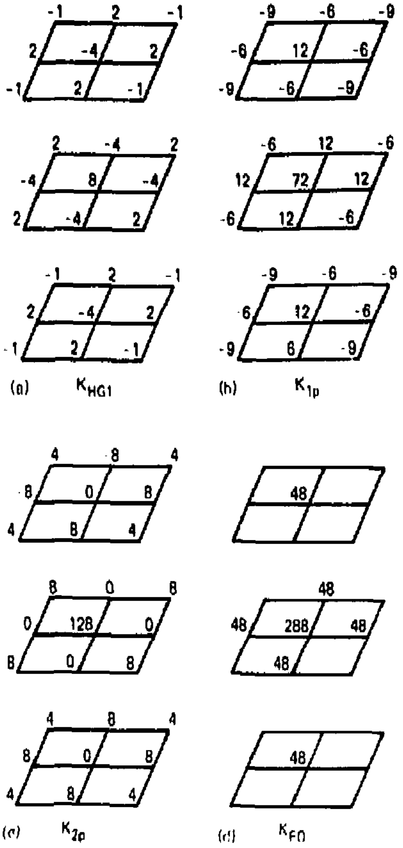

Fig. 2 The 3-D "hourglass" matrix stencil (a) and various stencils (for a cube) associated with discrete approximations to the Laplacian operator: (b) one-point quadrature stencil; (c) 2 x 2 x 2 quadrature stencil; (d) finite difference stencil.

It is noteworthy that the poor performance of $K_{2p}$ is caused solely by the process of mass lumping. If the consistent mass matrix were employed, the (honest) FEM result would actually display overdamping rather than underdamping.

In Ref. 2 we will discuss the "hourglass correction matrices" more carefully, interpret them as "balancing truncation error" terms, and show that they vanish in the limit of mesh refinement.

## Time Integration of the FEM Equations

### Time Stepping Procedure

The forward Euler method of time integration, applied to Eqn. (3a) gives

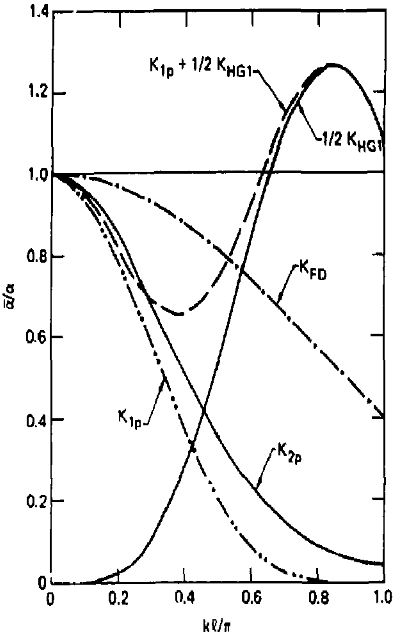

Fig. 3 The ratio of effective diffusivity to the desired diffusivity vs non-dimensional wave numbers associated with various discrete approximations to the transient heat equation.

$$U_{n+1} = U_n + \Delta t M^{-1}[f_n - KU_n - N(U_n)U_n - CP_n] \qquad (7)$$

where $U_n$ is the vector of nodal velocities at time $t_n$ and $\Delta t$ is the step-size. Before this equation can be used to advance the velocity, however, the pressure at time $t_n$ must be computed. This is done by combining Eq. (3a) with a time-differentiated version of Eq. (3b) ($C^T\dot{U}=0$ since $C^TU=0$ for all time) to generate the consistent discretized Poisson equation for the pressure, evaluated at time $t_n$,

$$(C^TM^{-1}C)P_n = C^TM^{-1}[f_n - KU_n - N(U_n)U_n] \qquad (8)$$

The sequence of steps for advancing the velocity and pressure from $t_n$ to $t_{n+1}$ is thus (given that $U_n$ is available and that it satisfies $C^TU_n = 0$):

(1) Form the acceleration vector (sans the pressure gradient)

$$A_n = M^{-1}[f_n - KU_n - N(U_n)U_n] \qquad (9)$$

(2) Solve the linear algebraic system (discrete Poisson equation) for the compatible pressure via

$$(C^TM^{-1}C)\ P_n = C^TA_n \qquad (10)$$

(3) Update the velocity, accounting for the pressure gradient,

$$U_{n+1} = U_n + \Delta t(A_n - M^{-1}CP_n) \qquad (11)$$

(4) Finally, in an "uncoupled step," update the temperature (and concentration, when appropriate), again using the forward Euler method,

$$T_{n+1} = T_n + \Delta t\ M_S^{-1}[f_{S_n} - K_ST_n - N_S(U_n)T_n]. \qquad (12)$$

This is the basic scheme used in our Phase 1 code and is a straightforward and legitimate method for solving the ODE's of Eqn. (3). Several additional comments are appropriate before discussing the less legitimate shortcuts employed in Phase 2 (subcycling):

(1) The presence of $M^{-1}$ in the above equations explains why we employ the lumped mass approximation ($M^{-1}$ is then a vector rather than a full matrix). While this approximation is known to reduce the accuracy (especially in regard to phase speed errors[12]), it is simple and probably cost-effective for very large problems.

(2) The pressure "Laplacian" matrix, $C^TM^{-1}C$, is symmetric and invariant with time. Hence we have chosen to solve (10) via direct methods, using a recently developed profile (or skyline) method.[13] This matrix is formed and factored in the pre-processor code and stored on disk for later retrieval by the main code. During the time integration, each pressure update is obtained by reading the disk file and performing one forward reduction and a back substitution.

(3) The explicit Euler method, while simple and fast, has one serious disadvantage (as do essentially all explicit schemes): it is only conditionably stable. The integration may become unstable if any one of several stability limits is exceeded: (1) the diffusive limit, (2) a linear advective-diffusive stability limit, and (3) certain types of nonlinear advection stability limits.[14] While we are not able to predict these $\Delta t$ limits a priori for the nonlinear FEM system on an arbitrary mesh, we have generally been reasonably successful by satisfying the following step-size restrictions, which come from analyzing the constant coefficient advection-diffusion equation via second-order centered finite differences on a grid with fixed $\Delta x$, $\Delta y$, and $\Delta z$:

(a) Diffusive limit:

$$\Delta t \leq \frac{1/2\kappa}{1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2} \qquad (13)$$

(b) Linear advective-diffusive limit:

$$\Delta t \leq \frac{2\kappa}{u^2 + v^2 + w^2}\ , \qquad (14)$$

where $\kappa$ is the diffusivity ($\mu/\rho$ for the Navier-Stokes equations). While the diffusive limit is well-known (e.g. Roache,[15]), there has been some confusion regarding the advective-diffusive limit.[16]

## Subcycling

The principal shortcoming of our Phase 1 code, which caused it to be uncomfortably expensive, is related to the following two points: (1) the stability-limited step-size is often much smaller than would be needed to accurately integrate the ODE's, and (2) the pressure updates, while not so expensive in CPU cost (typically ∿ 10% of the total cost of a time step), are inordinately expensive in I/O cost - reading the large disc file (1-2 million words) containing the factored Laplacian matrix requires several seconds.

Based on the premise that the pressure and the associated continuity constraint equations have little or no effect on the stability of the explicit scheme, we have

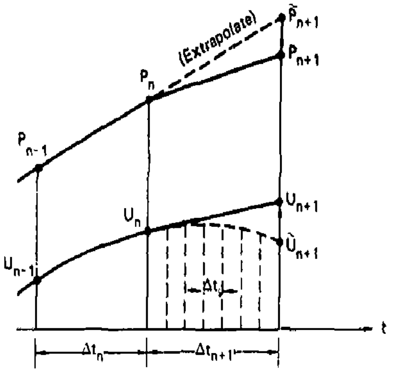

Fig. 4 Schematic of the subcycling process.

devised a subcycling strategy which permits less frequent updates of the pressure and a concomitant savings in computer time. The major ingredients of subcycling have already been described; here we summarize the entire process with reference to the schematic in Fig. 4.

We assume that the minor time step ($\Delta t_s$) has been pre-selected based on the more stringent of the criteria in (13) and (14) and will be used throughout the time integration. This is the subcycle step-size; the rest of the scheme is as follows:

(1) Assume that U, $\dot{U}$, and P are known at $t_n$ and $t_{n-1}$. Estimate the local time truncation error vector, $d_n$, for U via

$$d_n \equiv U_n - U(t_n) \approx -\frac{(\Delta t_n)^2}{2}\ddot{U}_n \approx -\frac{\Delta t_n}{2}(\dot{U}_n - \dot{U}_{n-1}) \quad , \tag{15}$$

where $U(t_n)$ represents the (unknown) exact solution on the assumption that $U_{n-1}$ is exact, and $\dot{U}_n = (U_n - U_{n-1})/\Delta t_n$. A similar error estimate is then made for T and a combined, relative RMS norm, e, is computed. The next (major) time step size is then obtained from

$$\Delta t_{n+1} = \Delta t_n (\epsilon/e)^{1/2} \quad , \tag{16}$$

where we have used the fact (from (15)) that the local error varies like $(\Delta t)^2$ and $\epsilon$ is an (input) error tolerance parameter related to the desired local *accuracy. A typical value for $\epsilon$ is .001 which implies* that a local error of about 0.1% is acceptable.

(2) Assuming that $\Delta t_{n+1} >> \Delta t_s$ (otherwise we do not subcycle), an estimate of the pressure at $t_{n+1}$ ($\tilde{P}_{n+1}$) is obtained via linear extrapolation. This extrapolated pressure is used (along with $P_n$ via a linear interpolation) to estimate the pressure gradient ($\nabla P$) during the subcycle steps; i.e. the continuity equations are completely ignored during subcycling. After the appropriate number ($\approx \Delta t_{n+1}/\Delta t_s$) of minor steps is completed, during each of which the advection and diffusion terms in both momentum and energy equations are updated (in part, to maintain stability), the mass adjustment process is invoked; i.e.,

(3) Given $\tilde{U}_{n+1}$ and $\tilde{P}_{n+1}$ as estimates to $U_{n+1}$ and $P_{n+1}$, the velocity is "adjusted" according to

$$U_{n+1} = \tilde{U}_{n+1} - M^{-1}C\lambda_{n+1} \quad , \tag{17}$$

where $\lambda_{n+1}$ is a vector of Lagrange multipliers, obtained by first solving the linear system

$$(C^T M^{-1} C)\lambda_{n+1} = C^T\tilde{U}_{n+1} \quad . \tag{18}$$

*It can be shown[17] that this procedure is a minimal least* squares velocity adjustment, from $\tilde{U}_{n+1}$ to $U_{n+1}$, subject to the constraint that $C^T U_{n+1} = 0$; i.e. $U_{n+1}$ is mass consistent whereas $\tilde{U}_{n+1}$ is not.

(4) Finally, the compatible pressure field is obtained by solving an equivalent linear system for $P_{n+1}$, a la Eqn. (10) with n replaced by n+1; i.e., the mass consistent velocity field is employed to get $P_{n+1}$.

It is clear that two re-solutions using the same factored matrix are required at each major timestep. Hence, subcycling can only be cost-effective when the subcycle ratio, $\Delta t_{n+1}/\Delta t_s$, is significantly greater than 2 *(based on our current limited experience, this ratio* can vary from ∿3 to 10 or more, depending on the "dynamics"); it can be especially effective if the solution is approaching a steady state.

If and when this procedure (or a variant thereof) has proven to be sufficiently robust, only major time step results would be reported. Thus far, we have been more carefully monitoring the behavior relative to the more rigorous approach.

## Numerical Results

Two numerical experiments have been chosen to illustrate the performance of the Phase 2 code and also, *hopefully, to justify the liberties taken by* under-integration and subcycling. *The two simulations,* namely, (1) confined Bénard-like thermo-convection in a cavity and (2) the atmospheric dispersion of a heavy gas over complex terrain, exhibit very different dynamics and thus provide a fairly wide range of test conditions.

### *Natural Convection in a 3-D Cavity*

For our first example we present the time-dependent, oscillatory motion of a fluid contained in a three dimensional rectangular enclosure which is heated from below (T = 0.5), cooled from above (T = − 0.5), and insulated laterally. These boundary conditions will cause fluid motion if the temperature difference (i.e., Rayleigh *number) between the upper and lower isothermal plates is* sufficiently large.

A sequence of numerical experiments (see Ref. 18 for details) led to a three cell oscillatory solution which is schematically illustrated in Fig. 5; the computational domain comprises 8192 elements (16 x 16 x 32), 9537 nodes, and approximately 46,000 equations. Note that the cell axes are parallel to the smaller horizontal dimension (z) and alternate in rotational direction. The fluid motion is a function of the Rayleigh number, Ra, the Prandtl number, Pr, and the aspect ratios of the cavity. For the present simulation, $Ra = \rho g \gamma \ell^3 \Delta T/\mu\kappa = 50{,}000$, $Pr = \mu/\rho\kappa = 0.5$, and the aspect ratios are 3.3 and 1.9 (based on the vertical dimension, $\ell$). For this pair of *aspect ratios, the prevailing flow pattern consists of three* cells (see Ref. 19 for impressive experimental visualizations).

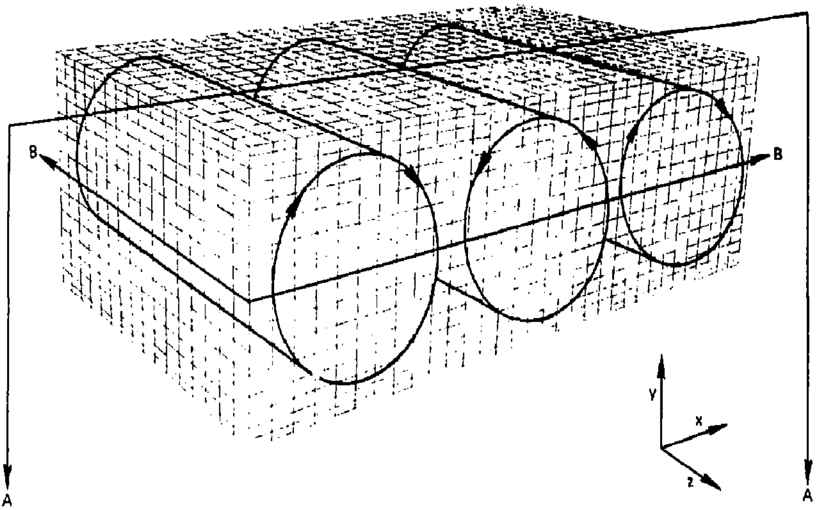

Fig. 5 Perspective view of the computational domain and three roll cells.

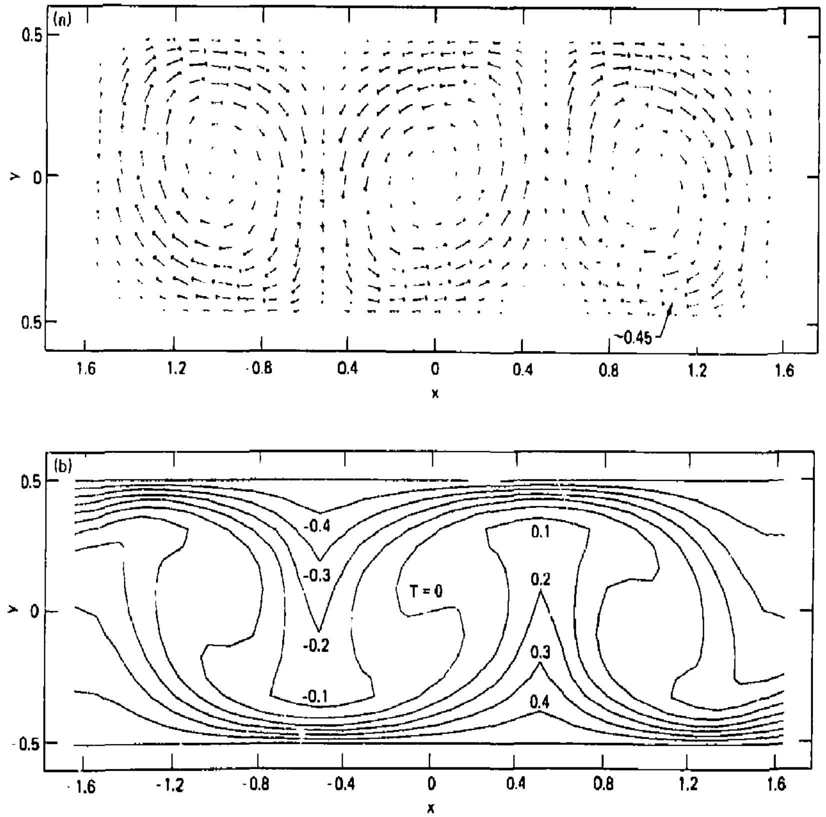

Fig. 6 Predicted results on the mid-cavity vertical plane (cross-section A-A): (a) velocity field; (b) temperature distribution.

Since the main purpose here is to illustrate the differences caused by reduced integration and subcycling, we have chosen to present only a portion of the numerical results for this problem. Here we limit ourselves to a fully developed, transient, three cell solution integrated in time for five non-dimensional time units (which corresponds to about half of the oscillation period). The first run employs exact (2 x 2 x 2) quadrature on the element integrals and the second run, starting with the same initial condition, uses 1-point quadrature and invokes subcycling.

Since the solutions at the end of the two runs are visually indistinguishable; most of the results to be presented are from run 2. Fig. 6(a) shows the three convection cell velocity field on the mid-cavity vertical plane, *as indicated by cross-section A-A in Fig. 5.* The temperature distribution on the same plane is shown in Fig. 6(b), in which the relatively isothermal vortex cores and the strong gradients near the walls due to the fluid motion are apparent. The vectors on the horizontal mid-plane (cross-section B-B of Fig. 5) are shown in Fig. 7. This clearly reveals the three-dimensionality of the problem along with the complex fluid motion within and *between the convection cells. Figure 8 shows the time* histories for u and v at a typical node. The small, abrupt jumps (solid curves) indicate the mass adjustment process invoked at the end of each major time step. For these runs $\Delta t_S$ was 0.04 (based on stability), and $\varepsilon$ was $10^{-3}$, leading to an average subcycle ratio of 7 and a maximum of 19. By repeating run 2 with no subcycling, we learned that the major discrepancy is caused by reduced quadrature; subcycling errors are much smaller. Based on the maximum flow velocity, the results of run 2 in Fig. 8 were less than 1.3% in error for u and less than 2.2% in error for the vertical velocity (v).

Even though this simple and repetitive mesh permitted an efficient run using the Phase 1 code (Run 1) - only the element matrices associated with 64 of the 8192 elements needed to be stored, and this was done in core — the cost differential was still substantial. Run 1 required 1.34 seconds of CPU and 12.3 seconds of I/O (mainly to retrieve the factored Laplacian matrix at each time step) per time step. On the other hand, Run 2 required only 0.44 second of CPU and 2.63 seconds of I/O per subcycle step.

Simulation of a Heavy Gas Release

This example concerns the simulation of the dynamics associated with the gravitational spread and vapor dispersion of LNG (liquefied natural gas) spills in the atmosphere. Since the NG (natural gas) density at its boiling temperature is significantly greater than that of the ambient air (by approximately 60%), the use of the Boussinesq equations for modeling such flows is probably inappropriate.[20] We have therefore employed the generalized anelastic formulation described earlier (see Ref. 1 for more details). In addition to solving equations in the form of (1), we also solved the concentration equation for NG mass fraction (designated as $\omega$). Again, the same simulation was performed with both codes. Since most of the results from the two simulations differ by only a few per cent and are again hardly distinguishable graphically, we present herein, unless noted otherwise, only the results obtained with our Phase 2 code.

Figure 9 shows the topography at the LNG experimental facility near China Lake, California. The horizontal extent of the computational domain, also shown in the figure, is 500 m x 400 m and the vertical extent is only 20 m. The same graded mesh consisting of 6400 elements (40 x 20 x 8) was used in both simulations. The total system comprises 7749 nodal points and approximately 45,000 equations. Figure 10 is an upwind view of a portion of the mesh in the spill pond vicinity.

The initial condition for the simulation was a steady isothermal wind field (∿3 to 4 m/sec at the top of the grid with the nominal wind direction from left to right) without NG vapor. Constant diffusivities, typical of the planetary boundary layer (0.4 $m^2$/sec vertically and 2.0 $m^2$/sec horizontally), were used throughout the simulation. The boil-off of LNG was simulated as a source area over 12 of the 30 elements comprising the spill pond. Over this area, a vertical injection velocity of ∿0.1 m/sec, along with a temperature of -160°C (NG *boiling temperature), and a constant rate of NG mass flux* were specified. Away from the source area, we used $\underline{u} = 0$ and $\partial T/\partial n = \partial\omega/\partial n = 0$ at the ground. The remaining boundary conditions employed were: specified $\underline{u}$, T, $\omega$ at the inlet plane, natural boundary conditions at the outlet, and symmetry conditions at the top and two lateral surfaces.

*A sample of the numerical results* (for the region enclosed by dashed lines in Fig. 9) is shown in figures 11 through 14. Fig. 11(a) shows the initial horizontal velocity (with topography contours superimposed) on the surface defined by the nodes located 1 m above the ground, which already indicates some terrain effects (e.g., "downslope" flow to the lower right of the spill pond) even though the flow field is isothermal. Fig. 11(b) shows the horizontal velocity on the same "plane" after 104 seconds, and demonstrates clearly the gravity-induced flow in <u>all</u> directions near the vapor source area and also the accentuated "downslope" flow. In Fig. 12, concentration contours (in volume fraction) on the same plane as predicted by the two codes are compared. As seen in the figure, the gravitational spread in all directions and the shift of the cloud centerline away from the mean wind direction (both of which have been observed at China Lake) have been correctly predicted by both codes. Furthermore, the agreement between the two predictions regarding the predicted flammable zone (5 to 15% NG vapor) is also very good. Figures 13(a) and (b) show, respectively, the initial velocity and the resulting velocity and concentration along a longitudinal plane through the pond center line at t = 104 seconds; again the influence of the dense cloud is apparent, this time in the form of a rollup vortex near the upwind edge of the cloud. In Fig. 14, we compare the results of velocity components (u,v,w) and density ($\rho$) versus time for a representative point as predicted by the two codes. The location of the point is 133 m downwind from the spill point, 1 m above ground, and in the region where the terrain has a slope of ∿7° (see Fig. 9). This is the region where the terrain is fairly complex and numerous distorted elements (rather than regular prisms) are therefore employed. Despite the fact that element level mass balances are not maintained by reduced quadrature, the results obtained with 1-point quadrature appear to compare quite well with those obtained using more exact quadrature. The greatest discrepancy appears to exist in the vertical velocity component (v), where a difference of about 20 percent occurs. This velocity component, however, is at least an order of magnitude smaller than the other component (u) and therefore any inaccuracy introduced by the combined effects of 1-point quadrature and subcycling is expected to have a stronger effect on such a component rather than on the larger and presumably more important ones. With a tighter error tolerance parameter ($\varepsilon$), the accuracy can, of course, be improved, but at additional computational cost.

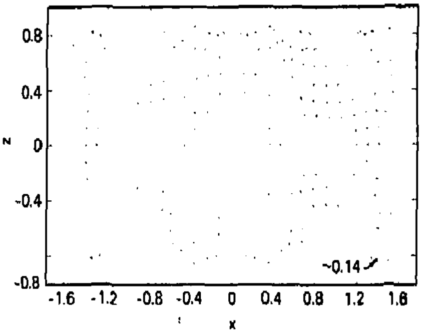

Fig. 7 Velocity vectors on the mid-horizontal plane (cross-section B-B) of the cavity.

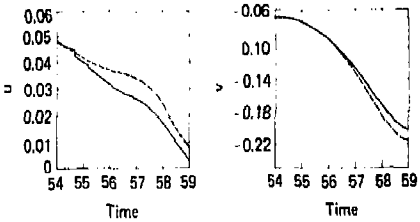

Fig. 8 Comparison of time histories for u and v at a representative point: —— one-point quadrature with subcycling; - - - 2 x 2 x 2 quadrature.

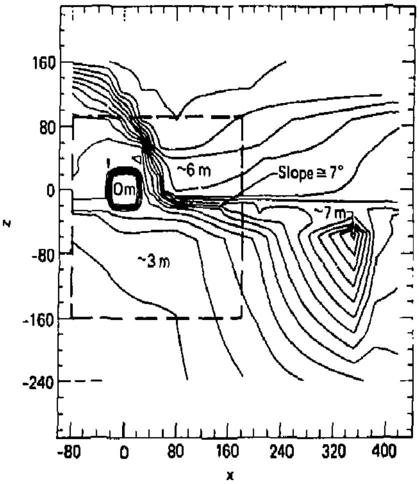

Fig. 9 Topography at China Lake LNG spill facility.

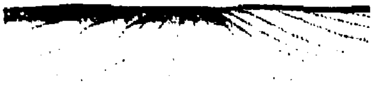

Fig. 10 Upwind view of a portion of the mesh layout on the ground in the spill pond vicinity.

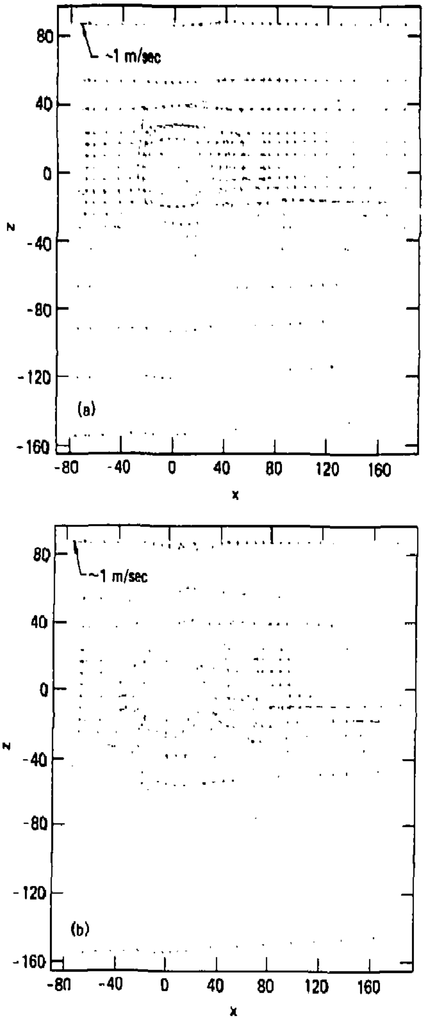

Fig. 11 Horizontal velocities 1m above the ground: (a) t = 0; (b) t = 104 seconds after the LNG spill.

Regarding computation time for this particular simulation, the new code has gained at least an order of magnitude improvement in both *CPU* and I/O over the old code. With an error tolerance parameter (ε) of $10^{-3}$, the new code required only 36 major time steps, with an average of 7 subcycles (the maximum number of subcycles was 14) to simulate 104 seconds of real time. The average cost per minor time step (0.4 second of real time) is approximately 0.3 second of CPU and 1.3 seconds of I/O. On the other hand, the old code required approximately 7 seconds of CPU and 13 seconds of I/O per time step.

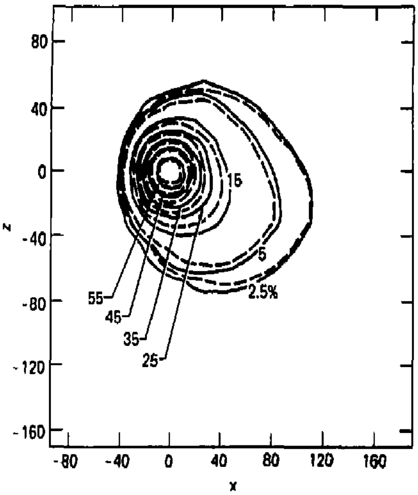

Fig. 12 Comparison of predicted NG concentrations (volume fraction) 1$^m$ above the ground at t = 104 seconds after the LNG spill: — one-point quadrature with subcycling; - - - 2 x 2 x 2 quadrature.

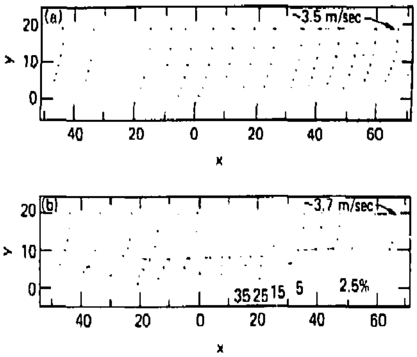

Fig. 13 Velocity and concentration on a longitudinal plane through the pond center line: (a) velocity at t = 0; (b) velocity and concentration at t = 104 seconds.

## Summary and Conclusions

In this paper we have presented a viable algorithm for numerically simulating three-dimensional, time-dependent, incompressible flows via a modified Galerkin finite element method. The model is simple and fast in that we use the simple 8-node isoparametric "brick" element and employ the simplest time integration method - explicit (foward) Euler, which is made feasible by lumping the mass matrix. To solve for the pressure, we form the consistent Poisson equation from the discretized momentum and continuity equations, which is then effectively solved via a disk-based skyline solver. To further improve computational efficiency, numerical techniques including reduced Gauss-Legendre quadrature and subcycling in conjunction with the use of variable time steps, have been introduced for the first time and demonstrated to be very cost-effective. With these newly devised techniques, an improvement of an order of magnitude in computational costs, yet with comparable accuracy as compared with the more rigorous algorithm, has been shown to be attainable.

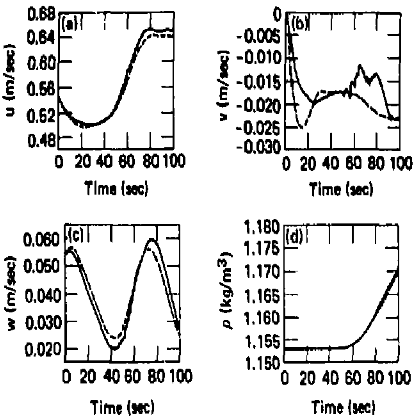

Fig. 14 Comparison of time histories for velocity and density of a representative point: — one-point quadrature with subcycling; - - - - 2 x 2 x 2 quadrature.

As with essentially all explicit schemes, the present algorithm must respect certain stability time step limits, which can sometimes be very stringent-especially the linear advective-diffusive limit for some cases of interest to us. Also, even with subcycling, the I/O cost is still the major part of the total computational cost. We are currently working toward further reducing these costs such as exploring the feasibility of adding balancing diffusion$^3$ and the use of a more efficient technique (less I/O cost) for solving the pressure equation.

## Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. The authors are grateful to Dr. John O. Hallquist of LLNL for discussions on hourglass modes, and to Dr. Michael J. P. Cullen of the British Meteorological Office for discussions on 2Δx waves and related problems.

## References

1. Gresho, P. M., Chan, S. T., Lee, R. L., and Upson, C. D., "Solution of the Time-Dependent, Three-Dimensional Incompressible Navier-Stokes Equations via FEM," to be presented at the International Conference on Numerical Methods for Laminar and Turbulent Flow, Venice, Italy, July 1981.

2. Chan, S. T., and Gresho, P. M., "Solution of the Multi-Dimensional, Incompressible, Navier-Stokes Equations Using Low-Order Finite Elements and One-Point Quadrature" (in preparation).

3. Gresho, P. M., and Chan, S. T., "Subcycling - A Cost-Effective Explicit Technique for Solving the Time-Dependent, Multi-Dimensional, Incompressible Navier-Stokes Equations" (in preparation).

4. Gresho, P. M., Lee, R. L. and Sani, R. L., "On the Time-Dependent Solution of the Incompressible Navier-Stokes Equations in Two- and Three-Dimensions," in C. Taylor and K. Morgan, Recent Advances in Numerical Methods in Fluids, Pineridge Press Ltd., Swansea, U. K., 1980.

5. Ogura, Y., and Phillips, N., "Scale Analysis of Deep and Shallow Convection in the Atmosphere," J. Atm. Sci., 19, 1962, pp. 173-179.

6. Hallquist, J. O., "User's Manual for DYNA2D - An Explicit Two-Dimensional Hydrodynamic Finite Element Code with Interactive Rezoning," UCID-18756, July 1980.

7. Key, S. W., Beisinger, Z.E., and Krieg, R. D., "HONDO-II: A Finite Element Computer Program for the Large Deformation, Dynamic Response of Axisymmetric Solids," Sandia Report SAND 78-0422, Oct. 1978.

8. Zienkiewicz, O. C., The Finite Element Method, 3rd Edition, McGraw-Hill Book Co. (U.K.) Ltd., 1977.

9. Leone, J. M., Gresho, P. M., Chan, S. T., and Lee, R. L., "A Note on the Accuracy of Gauss-Legendre Quadrature in the Finite Element Method," Int. J. Num. Meth. Eng., Vol. 14, 1979, pp. 769-773.

10. Kosloff, D. and Frazier, G. A., "Treatment of Hourglass Patterns in Low Order Finite Element Codes," Int. J. Num. & Ana. Meth. Geomech., Vol. 2, 1978, pp. 57-72.

11. Hallquist, J. O., and Goudreau, G. L. (private communication).

12. Gresho, P. M., Lee, R. L. and Upson, C. D., "FEM Solution of the Navier-Stokes Equations for Vortex Shedding Behind a Cylinder: Experiments with the Four-Node Element," in Finite Elements in Water Resources, Proceedings of Third Inter. Conf., Univ. of Miss., USA, 1980.

13. Taylor, R. L., Wilson, E. L. and Sackett, S. J., "Direct Solution of Equations by Frontal and Variable Band, Active Column Methods," Proc. U.S. European Workshop on Nonlinear Finite Element Analysis in Structural Mechanics, Bochum, W. Germany, July, 1980, pp. 521-552.

14. Hirt, C. W., "Heuristic Stability Theory for Finite-Difference Equtions," J. Comp. Phys., 2, 1968, pp. 339-355.

15. Roache, P., Computational Fluid Dynamics, 2nd Ed., Hermosa Press, Albuquerque, New Mexico, 1976.

16. Hindmarsh, A., and Gresho, P., "The Stability of Explicit Euler Time-Integration for Certain Finite Difference Approximations of the Multi-Dimensional Advection-Diffusion Equation," submitted to Int. J. Num. Meth. Fluids.

17. Sani, R. L., Gresho, P. M., Tuerpe, D. R., and Lee R. L., "The Imposition of Incompressibility constraints via Variational Adjustment of Velocity Fields," presented at the the International Conference on Numerical Methods in Laminar and Turbulent Flow, Swansea, United Kingdom, July 1978.

18. Upson, C. D., Gresho, P. M., Chan, S. T., and Lee, R. L., "A Thermal Convection Simulation in Three Dimensions by a Modified Finite Element Method," to be presented at the Second National Symposium on Numerical Methods in Heat Transfer, Univ. of Maryland, College Park, Maryland, September 1981.

19. Stork, K. and Müller, U., "Convection in Boxes: Experiments," J. Fluid Mechanics, 54 part 4, 1972, pp. 599-611.

20. Daley, B. and Pracht, W., "Numerical Study of Density - Current Surges," Phys. of Fluids, 11, 1, 1968, p. 15.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government thereof, and shall not be used for advertising or product endorsement purposes.