JOURNAL OF COMPUTATIONAL PHYSICS **127,** 330–345 (1996)
ARTICLE NO. 0179

A Finite-Volume/Newton Method for a Two-Phase Heat Flow Problem Using Primitive Variables and Collocated Grids

M. C. LIANG AND C. W. LAN*

Chemical Engineering Department, National Central University, Chung-Li, Taiwan 32054, Republic of China

Received July 24, 1995; revised March 18, 1996

A finite-volume/Newton's method is presented for solving the incompressible heat flow problem in an inclined enclosure with an unknown melt/solid interface using primitive variables and collocated grids. The unknown melt/solid interface is solved simultaneously with all of the field variables by imposing the weighted melting-point isotherm. In the finite-volume formulation of the continuity equation, a modified momentum interpolation scheme is adopted to enhance velocity/pressure coupling. During Newton's iterations, the ILU (0) preconditioned GMRES matrix solver is applied to solve the linear system, where the sparse Jacobian matrix is estimated by finite differences. Nearly quadratic convergence of the method is observed. The robustness of the method is further enhanced with the implementation of the pseudo-arclength continuation. The effects of the Rayleigh number and gravity orientation on flow patterns and the interface are demonstrated. Bifurcation diagrams are also constructed to illustrate flow transition and multiple steady states.

1. INTRODUCTION

Two-phase flow problems are important in many solidification processes, such as casting and crystal growth. In such systems, in addition to complicated geometry, simulation could become quite challenging due to the strong coupling of incompressible heat flow and the unknown melt/solid interface shape. Several numerical approaches (e.g., Refs. [1–7]) have been proposed to solve these problems. Among them, they can be categorized, from the formulation point of view, by streamfunction/vorticity (ψ/ω) [2–3, 6–7] and primitive (UVP) variables [1, 4–5], and from solution point of view, by decoupled [2–6] and global (coupled) [1, 7] iteration approaches. Although the ψ/ω formulation could be easily implemented and leads to a smaller set of equations to solve, it is more difficult to be extended to three-dimensional problems.

For the UVP approach, because the pressure does not appear in the equation of continuity, the velocity/pressure decoupling could be quite troublesome and special treatments are required. The well-known approach to overcome this poor linkage is the SIMPLE scheme and its variant on staggered grids [8]. The major drawback in these schemes is from the staggered grid allocation. Using different grids for different variables has made the staggered grid approach less attractive to the problems with complex two- and three-dimensional geometries and free boundaries [9]. The same is true to use the staggered grids in multigrid implementation. An additional drawback is the slow convergent speed of the SIMPLE iterations, and its convergent speed is often mesh dependent. To amend this, Newton's method could be used (e.g., Refs. [10, 13]). Although the convergence rate of Newton's method is quadratic and independent of the grid number used [10], complicated Jacobian matrices need to be estimated, and the computer memory required for the solution of Newton's linear equations is much larger. Nevertheless, Newton-like approaches have proven to be robust and efficient, especially when iterative matrix solvers are used [13].

In contrast to the staggered grid allocation is the nonstaggered or collocated grid arrangement, in which only one grid system is required. The SIMPLE-like approaches can also be used on collocated grids. However, to avoid checkerboard pressure oscillation due to the velocity/pressure decoupling, special formulations, such as the momentum interpolation [8, 14] and the pseudo-compressibility [15] techniques, are necessary. Clearly, if the pressure/velocity decoupling can be avoided, the collocated allocation seems to be a good candidate for two- and three-dimensional free boundary problems, and even for multigrid methods. Interestingly, to the UVP formulation Newton's method has not been used based on the collocated grids. On the other hand, a more complicated structure of the Jacobian matrix due to momentum interpolation is expected, which could be difficult if an analytical Jacobian is required.

When an unknown melt/solid interface is included, an iteration scheme is also required to update the interface location. Two common ways to update the interface shape are through the isotherms [16–17] and energy flux balance methods [6–7]. Again, no matter which approach is used, the convergence characteristics of the two-phase problems strongly rely on the iteration schemes, as well as the grid

* Corresponding author. E-mail: lan@che730.che.ncu.edu.tw.

0021-9991/96 $18.00

arrangements. The easiest iteration scheme is through a decoupled approach by putting the heat flow calculation in inner iteration loops, while updating the interface through outer loops. Unfortunately, this naive implementation usually leads to slow convergence, or even divergence. An alternative to the decoupled successive approximation to update the interface is global iteration, in which all the unknown variables are iterated simultaneously, including the field variables and the unknown interface. Although the global approach for Finite element methods has been adopted for years [1, 16, 18], it is still not widely used for the finite difference and the finite volume methods (FVM). Recently, Dandy and Leal [19] successfully applied Newton's method to the finite difference solution of a rising bubble, in which the shape and the flow field around it were solved globally. Unfortunately, the construction of the Jacobian matrix for the Newton's scheme is still tedious and error-prone. To make Newton's method more attractive to free boundary problems for the finite difference or finite volume methods, the best way is to use a numerical Jacobian. This idea has been implemented successfully by Lan [7] for solving a complicated floating molten zone problem using the ψ/ω formulation. In his calculations, the CPU time for estimating the numerical Jacobian matrix was much less (<20%) than that for the solution of Newton's linear equations. Therefore, Newton's method using a numerical Jacobian provides an efficient way for complicated problems. Again, the global approach based on the UVP formulation for FVM has not been reported. In fact, because the pressure variable does not appear explicitly in the equation of continuity, while the interface variable not in the isotherm condition, zero diagonals in the Jacobian matrix may raise additional difficulties in the iterative matrix solution.

In the present report, a FVM/Newton's scheme using primitive variables on collocated grids and a numerical Jacobian is proposed for solving an unknown interface problem coupled with incompressible heat flow. The field variables and the unknown interface are solved globally. The two-phase incompressible heat flow problem in an inclined enclosure is chosen as an example. Furthermore, whe the problem encounters a limit or bifurcation point, Newton's method may still break down. To amend this, the pseudo-arclength continuation [20] is also implemented. Through the continuation, the flow transition and the bifurcation structure of the complicated two-phase problem can be determined.

2. FORMULATION AND IMPLEMENTATION

The steady-state two-phase heat flow problem studied in this report in an inclined enclosure is shown in Fig. 1. The height H is twice of the width W (aspect ratio $A = 2$). The temperature on two ends is fixed; the lower

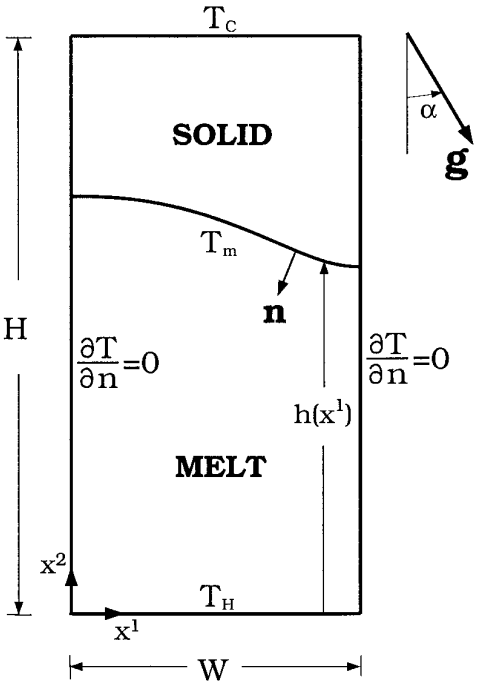

FIG. 1. The schematic of the two-phase heat flow problem.

temperature T_H is higher than the melting point (T_m) of the material inside, while the upper temperature T_C is lower. The other two sides are assumed adiabatic. The melt/solid interface height h is unknown *a priori* and needs to be solved simultaneously with the other field variables. The physical properties of the material are assumed constant, while the melt is incompressible. The dimensionless variables are defined by scaling the length with the width W, velocity with α_m/W, and pressure with $\rho_m\alpha_m^2/W^2$, where α_m is the thermal diffusivity and ρ_m is the density of the melt. The dimensionless temperature (θ) is defined as $\theta \equiv (T - T_m)/(T_H - T_m)$. For the convenience of illustration, T_C and T_H are adjusted so that θ is equal to -1 and 1 at the upper and lower boundaries, respectively. With the Boussinesq approximation [21], the conservation equations in dimensionless form for two-dimensional steady incompressible laminar flow of a Newtonian fluid in the melt and heat conduction in the solid can be described as follows:

Melt,

$$\frac{\partial u_1}{\partial x^1} + \frac{\partial u_2}{\partial x^2} = 0, \tag{1}$$

$$\frac{\partial}{\partial x^1}\left(u_1u_1 - \mathrm{Pr}\,\frac{\partial u_1}{\partial x^1}\right) + \frac{\partial}{\partial x^2}\left(u_2u_1 - \mathrm{Pr}\,\frac{\partial u_1}{\partial x^2}\right) + \frac{\partial P}{\partial x^1} + \mathrm{RaPr}\,\theta\sin\alpha = 0, \tag{2}$$

$$\frac{\partial}{\partial x^1}\left(u_1u_2 - \mathrm{Pr}\frac{\partial u_2}{\partial x^1}\right) + \frac{\partial}{\partial x^2}\left(u_2u_2 - \mathrm{Pr}\frac{\partial u_2}{\partial x^2}\right) + \frac{\partial P}{\partial x^2} - \mathrm{RaPr}\,\theta\cos\alpha = 0, \quad (3)$$

$$\frac{\partial}{\partial x^1}\left(u_1\theta - \frac{\partial \theta}{\partial x^1}\right) + \frac{\partial}{\partial x^2}\left(u_2\theta - \frac{\partial \theta}{\partial x^2}\right) = 0, \quad (4)$$

Solid,

$$\frac{\partial}{\partial x^1}\left(\frac{\partial \theta}{\partial x^1}\right) + \frac{\partial}{\partial x^2}\left(\frac{\partial \theta}{\partial x^2}\right) = 0, \quad (5)$$

subject to the following boundary conditions:

$$\theta(x^1, 0) = 1; \quad \theta(x^1, 2) = -1; \quad (6)$$

$$\partial\theta(0, x^2)/\partial x^1 = \partial\theta(1, x^2)/\partial x^1 = 0; \quad (7)$$

and on all boundaries

$$u_1 = u_2 = 0. \quad (8)$$

At the melt/solid interface, assuming the thermal conductivities of both phases are the same, the thermal flux continuity is imposed as

$$\partial\theta/\partial n|_m = \partial\theta/\partial n|_s, \quad (9)$$

as well as the melting-point isotherm

$$\theta = 0. \quad (10)$$

In the above equations, u_1 and u_2 are the Cartesian velocities in the x^1- and x^2- directions, respectively, and P is the pressure. The Prandtl and Rayleigh numbers are defined as $\mathrm{Pr} \equiv \nu_m/\alpha_m$ and $\mathrm{Ra} \equiv \beta g(T_H - T_m)W^3/(\nu_m\alpha_m)$, where ν_m is the kinematic viscosity, β is the thermal expansion coefficient, and g is the gravity acceleration. The tilt angle α is defined in Fig. 1.

The measure of convective heat transfer is through the Nu number as

$$\mathrm{Nu} = \int_0^1 \frac{\partial\theta}{\partial n}\,dx^1, \quad (11)$$

where **n** is the normal vector pointing toward the melt. For the case without convection (purely conductive heat transfer), Nu = 1.

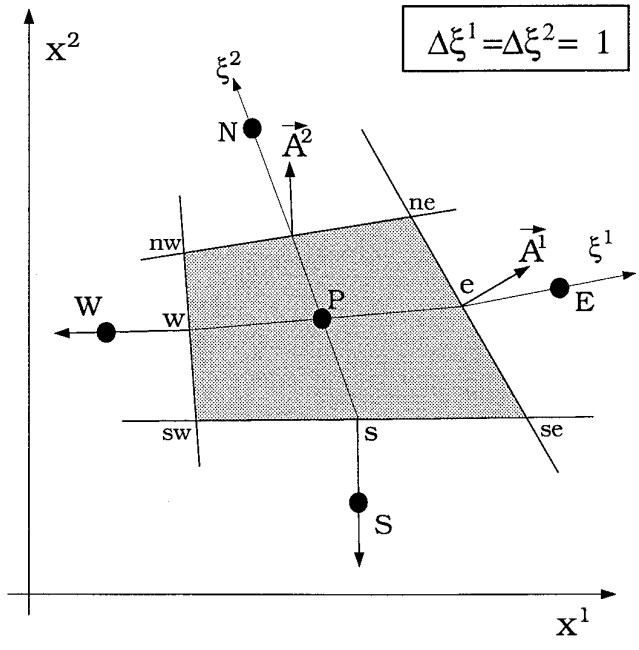

FIG. 2. Two-dimensional finite volume and labeling scheme.

3. FINITE VOLUME FORMULATION

3.1. *Coordinate Transformation*

Due to the unknown and deformed interface shape $h(x^1)$, body-fitted coordinates (ξ^1, ξ^2) are adopted. The algebraic coordinate transformation for both melt and solid phases are performed as follows:

Melt,

$$x^1 = \overline{\xi^1}(\xi^1), \quad (12)$$

$$x^2 = \overline{\xi^2}(\xi^2)h(\xi^1). \quad (13)$$

Solid,

$$x^1 = \overline{\xi^1}(\xi^1), \quad (14)$$

$$x^2 = h(\xi^1) + \overline{\xi^2}(\xi^2)(2 - h(\xi^1)), \quad (15)$$

where $\overline{\xi^1}(\xi^1)$ and $\overline{\xi^2}(\xi^2)$ are stretch function ranging from 0 to 1 for adjusting grid distribution. A hyperbolic tangent function with the following form, for example, for $\overline{\xi^1}$, is

$$\overline{\xi^1}(\xi^1) = 0.5\left[1 + \tanh\left(\frac{B[(\xi^1 - 1)/(N_{\xi^1} - 1) - 0.5]}{0.5B}\right)\right], \quad (16)$$

where B is a stretch constant and N_{ξ^1} the number of control volumes (CVs) in ξ^1 direction. The coordinate transformation defines the boundaries of the CVs. For example, the corner coordinates of each CV in physical space, as shown in Fig. 2, are calculated according to the transformation,

TABLE I

Form of Governing Equations

$$\frac{\partial}{\partial \xi^1}(c_1\phi + D_1) + \frac{\partial}{\partial \xi^2}(c_2\phi + D_2) + JS_\phi = 0$$

ϕ	c_1	c_2	D_1	D_2	S_ϕ
1	U^1	U^2	0	0	0
u_1	U^1	U^2	$-\frac{\Pr}{J}\left(B^{1j}\frac{\partial u_1}{\partial \xi^j} + A_j^1\omega_1^j\right)$	$-\frac{\Pr}{J}\left(B^{2j}\frac{\partial u_1}{\partial \xi^j} + A_j^2\omega_1^j\right)$	$-\frac{A_1^j}{J}\frac{\partial \wp}{\partial \xi^j}$
u_2	U^1	U^2	$-\frac{\Pr}{J}\left(B^{1j}\frac{\partial u_2}{\partial \xi^j} + A_j^1\omega_2^j\right)$	$-\frac{\Pr}{J}\left(B^{2j}\frac{\partial u_2}{\partial \xi^j} + A_j^2\omega_2^j\right)$	$-\frac{A_2^j}{J}\frac{\partial \wp}{\partial \xi^j}$
θ	U^1	U^2	$-\frac{1}{J}\left(B^{1j}\frac{\partial \theta}{\partial \xi^j}\right)$	$-\frac{1}{J}\left(B^{2j}\frac{\partial \theta}{\partial \xi^j}\right)$	0

Note. $U^i = A_j^i u_j$, $B^{ij} = A_k^i A_k^j$, $\omega_j^i = (\partial u_i/\partial \xi^k)A_j^k$.

while the cell faces are straight lines between the corners. The cell faces can be represented by surface vectors, i.e., $\mathbf{A}^1$ and $\mathbf{A}^2$. For convenience, $\Delta\xi^1 = \Delta\xi^2 = 1$, so that the CVs in the computational domain (ξ^1, ξ^2) have a volume of 1. The values of variables in each CV are defined at the geometric center. Since some variables on the boundaries are unknown, to make coding easier the CVs on the boundaries are also assigned, but their volume is zero. Highly stretched grids toward the boundaries can be obtained easily by simply increasing the value of B.

3.2. *Finite Volume Integration*

After the coordinate transformation, the governing equations can be rewritten in a general conservation-law form in (ξ^1, ξ^2) as

$$\frac{\partial}{\partial \xi^i}(C_i\phi + D_i) + JS_\phi = 0, \tag{17}$$

where $J = (\partial x^1/\partial \xi^1)(\partial x^2/\partial \xi^2) - (\partial x^1/\partial \xi^2)(\partial x^2/\partial \xi^1)$. The variable ϕ and its associated coefficients are listed in Table I. For the convenience of representation, tensor notation (summation on free indices) is adopted here. Also, in Table I, A_j^i represents the x^j component of the surface vector $\mathbf{A}^i$ (see Fig. 2). More importantly, the driving forces due to pressure gradients and the buoyancy force $\mathbf{F}$ are combined and redefined through a new variable $\wp$ as

$$\frac{\partial \wp}{\partial \xi^j} \equiv F_k\frac{\partial x^k}{\partial \xi^j} - \frac{\partial P}{\partial \xi^j}, \tag{18}$$

where F_k is the kth component of the body force $\mathbf{F}$; $\mathbf{F} \equiv$ PrRa $\theta[-\sin(\alpha)\mathbf{e}_1 + \cos(\alpha)\mathbf{e}_2]$. The advantages of using this variable will be discussed later. The finite-volume method simply integrates Eq. (17) over each CV in the computational domain (ξ^1, ξ^2). In fact, the integration performed over the physical domain (x^1, x^2) is the same as that over the computational domain after coordinate transformation, and $\Delta V = J$. After applying the Gauss theorem, the integration over each CV results in a flux balance equation,

$$I_e - I_w + I_n - I_s + \int_{\Delta V} S_\phi dV = 0, \tag{19}$$

where I_i, $i = (e, w, n, s)$, represents the fluxes of ϕ across the faces of CV. Each of the fluxes I_i, is made of two distinct parts, namely a convective contribution I^C $(=C_i\phi)$ and a diffusive contribution I^D $(=D_i)$. Both I^C and I^D are approximated with the central difference scheme. Taking I_e (for $\phi = \theta$) as an example,

$$I_e^C = U_e\phi_e = (A_j^1 u_j)_e\phi_e \tag{20}$$

and

$$\begin{aligned} I_e^D &= -\frac{1}{J_e}\left[B^{1j}\frac{\partial \phi}{\partial \xi^j}\right]_e \\ &\approx \frac{1}{\mathbf{A}_e^1 \cdot \mathbf{PE}}[B_e^{11}(\phi_E - \phi_P) + B_e^{12}(\phi_{ne} - \phi_{se})], \end{aligned} \tag{21}$$

where $\mathbf{PE}$ is the vector from points P to E. The definition of B^{ij} is given in Table I. A weighted linear interpolation is used to calculate the face value:

$$\phi_e = \frac{\overline{Pe}}{\overline{PE}}\phi_E + \frac{\overline{eE}}{\overline{PE}}\phi_P, \tag{22}$$

where $\overline{Pe}$ is the arclength between points P and e, while $\overline{eE}$ is that between points e and E. This weighted scheme is particularly useful at boundaries where the CVs have zero volume. Values at the corners of the CV are obtained by taking a weighted average value from their nearby nodal points.

The contribution of the source term can be approximated by

$$\int_{\Delta V} S_\phi dV = (S_\phi)_P \Delta V. \tag{23}$$

The implementation of the boundary conditions for the velocity and temperature variables are straightforward by introducing values or fluxes. However, care should be taken for the pressure boundary condition on solid wall. With a simple extrapolation of the pressure from interior values to the solid boundaries [22–24], some spurious velocities could appear near the boundaries. These errors, even though they are small, are still not trivial near the static solution (no convection) when the inclined angle α is almost zero, i.e., the traditional Rayleigh–Benard problem. Taking a zero pressure gradient [14], i.e., $\partial P/\partial n = 0$, gives a slightly better result. However, the small spurious velocities still cannot be eliminated. In fact, since the flow is introduced by the source term in the momentum equations, both pressure and nonpressure gradients there could induce flow. Therefore, the pressure on the boundaries is extracted from the extrapolated value of $\wp$. As a result, no spurious velocity is found in all of computation here. Using $\frac{\partial \wp}{\partial n} = 0$ gives a good result as well. In the implementation of $\frac{\partial \wp}{\partial n} = 0$, the skewness of the mesh is also considered, and the second-order backward difference is used. The detailed implementation of the pressure boundary conditions is further described in Appendix.

3.3. *Momentum Interpolation*

Since the pressure variable does not appear explicitly in the continuity equation, the use of linearly interpolated velocity at the cell faces for collocated grids could lead to the velocity/pressure decoupling or the so-called checkerboard oscillation of pressure. In order to amend this, the idea of Rhie–Chow momentum interpolation scheme [14] is adopted. In other words, the velocity values required for the continuity equation are interpolated from the momentum equations, rather than linearly from the adjacent nodal values. To implement this, the formulation of the momentum equation for u_i from the previous discretization can be rewritten as

$$u_{iP} = \frac{1}{a_P}\left(\sum_{nb} a_{nb}u_{inb} + S_{u_i}\Delta V\right), \tag{24}$$

where nb means the neighbor points. The face velocities required for the continuity equation, e.g., for face e, is evaluated by taking a weighted average fro the above equation except the source term:

$$u_{i_e} = \overline{u_{ie}} - \left(\frac{\overline{S_{ui}\Delta V}}{a_P}\right)_e + \left(\frac{S_{ui}\Delta V}{a_P}\right)_e. \tag{25}$$

The overbar indicates the weighted average from adjacent nodal points. The contravariant velocity required for the mass fluxes now becomes

$$U_e^i = (A_k^i u_k)_e \approx \overline{U_e^i} + C_e^{ij}\left[\left(\frac{\overline{\partial \wp}}{\partial \xi^j}\right)_e - \left(\frac{\partial \wp}{\partial \xi^j}\right)_e\right], \tag{26}$$

where

$$C^{ij} = \frac{A_k^i A_k^j}{a_P}.$$

Since the derivative values in Eq. (26) are approximated by the central difference, the cross-derivative terms in the bracket cancel, leaving the diagonal term alone. The contravariant velocities at other faces are calculated in a similar manner.

Clearly, from the above equation the pressure gradient at face e used for the continuity equation is calculated by a $1 - \delta$ difference using points P and E. This is similar to that in the staggered grid allocation that stores the velocity variables at cell faces and the momentum equations are approximated there. In fact, this is also the key idea of the Rhie–Chow scheme. However, in the original Rhie–Chow scheme, only the pressure gradient is introduced, while nonpressure source terms are excluded. As it is, some spurious velocities are introduced caused by the slight unbalance of source terms. It should be noticed that the source terms are evaluated at cell faces for staggered grids. To better mimic the staggered grid allocation, all of the source terms need to be included in the Rhie–Chow interpolation.

Furthermore, due to the incompressibility nature, only pressure gradients are meaningful to the solution. Therefore, to obtain a unique solution, a pressure variable in the melt is set to zero as a reference; any pressure in the melt (not on boundaries) can be used as the reference. Although one pressure at the lower left corner was required to be fixed in the staggered allocation to avoid the breakdown of LU decomposition [25], it is found unnecessary here.

3.4. *Pseudo-Arclength Continuation*

In order to enhance the robustness of the solution scheme and to locate the solution from one to the other, the pseudo-archlength continuation [20] is adopted. Also, in many nonlinear systems, a limiting (turning) or a bifurcation point in parameter space may exist. To such a point, Newton's method may fail to get a convergent solution due to the singular Jacobian matrix. To overcome this, continuation is usually helpful. The idea of continuation is to trace the solution along a branch through an arclength s by including an additional equation for the continuation parameter p,

$$\left.\frac{\partial \mathbf{x}^T}{\partial s}\right|_{s_0} \cdot [\mathbf{x}(s) - \mathbf{x}(s_0)] + \left.\frac{\partial p}{\partial s}\right|_{s_0} [p(s) - p(s_0)] - (s - s_0) = 0, \tag{27}$$

where $\mathbf{x} = (\theta, u_1, u_2, P, h)^{\mathrm{T}}$ and $s - s_0$ is the step size along the solution branch, where the arclength $s \equiv \|(\mathbf{x}, p)^{\mathrm{T}}\|_2$. The direction vectors ($\partial \mathbf{x}/\partial s|_{s_0}$ and $\partial p/\partial s|_{s_0}$) can be computed by the backward finite differences from previous two solution vectors along the same solution branch. To start the continuation, two solutions for p and $p + \Delta p$ are necessary, where Δp is a small increment of p. In the present study, the Raleigh number Ra or the tilt angle α is used as the parameter.

4. SOLUTION SCHEME

4.1. *Newton's Method*

After the discretization for both governing and boundary equations, as well as the continuation equation, a set of nonlinear algebraic equations can be obtained,

$$\mathbf{f}(\mathbf{y}) \equiv 0, \tag{28}$$

where $\mathbf{y} = [\mathbf{x}, \text{Ra or } \alpha]^{\mathrm{T}}$. In Eq. (28), one of the discretized continuity equations is replaced by the constant pressure condition (simply set $P = 0$ here) to ensure a unique solution. This nonlinear equation set is solved by Newton's method simultaneously for all variables. Starting from an initial approximation to the unknown vector $\mathbf{y}^0$, successive updates are constructed as

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \boldsymbol{\delta}^{n+1}, \tag{29}$$

and the correction vector $\boldsymbol{\delta}^{n+1}$ is the solution of the linear equation set

$$\tilde{J}\boldsymbol{\delta}^{n+1} = -\mathbf{f}(\mathbf{y}^n). \tag{30}$$

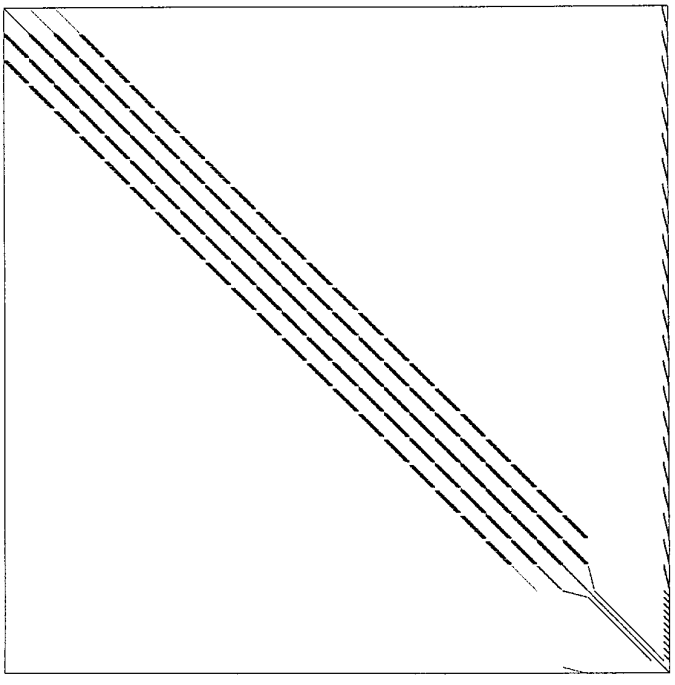

FIG. 3. Sparse structure of the Jacobian matrix for NEQ = 2414.

The components of the Jacobian matrix $\tilde{J}$, formed by explicit differentiation as $\tilde{J}_{ij} \equiv \partial f_i/\partial y_j$, represent the sensitivity of the residual vector to perturbations in the solution vector. They are estimated by the forward difference as described next.

4.2. *Jacobian Matrix*

Because pressure variable does not appear in the equation of continuity, zeros may appear in the diagonal of the Jacobian matrix. Particularly, the incomplete LU (ILU) decomposition used in the preconditioning for the iterative matrix solvers may fail because no pivoting is implemented [25]. Therefore, variable ordering could be critical to the success of the ILU preconditioner. However, in the present problem, if the velocity variables of each CV are ordered first, the ordering of other variables does not affect much on the performance of the preconditioning. For convenience, and also to achieve a tight band structure of the Jacobian matrix, the equations and unknowns are ordered, starting at ξ^2, $\xi^1 = 1$, in the order of appearance for u_1, u_2, P, and θ in the melt, and then θ in the solid. After all field equations are ordered, the isotherm condition (for interface variables $h(x^I)$) is taken into account, followed by the continuation equation for the parameter. The sparse structure of the Jacobian matrix according to this ordering is shown in Fig. 3. The sparsity of this matrix is only about 0.6%. The vertical band entries on the right shows the dependence of field variables on the interface variables,

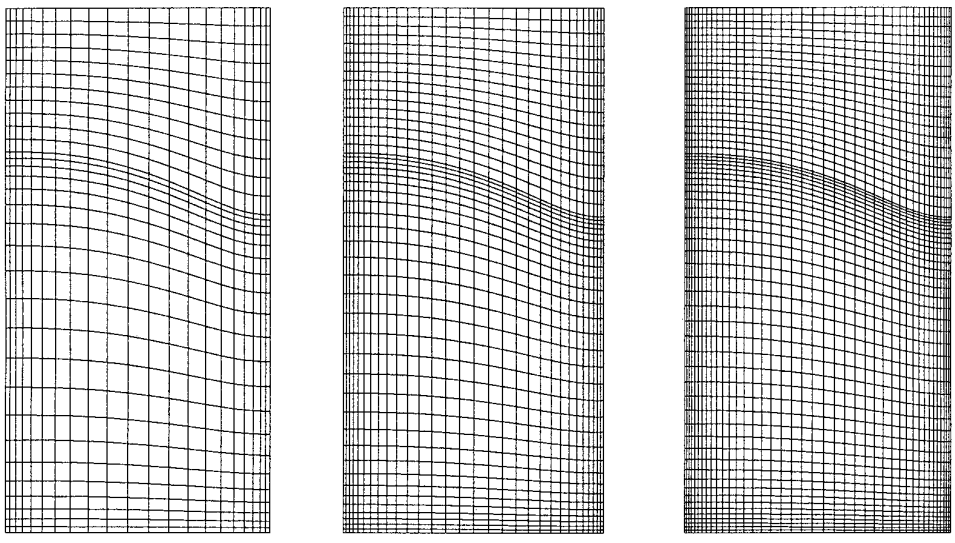

FIG. 4. Three different meshes used for calculations: (a) 23 × 23 (melt), 23 × 13 (solid); (b) 33 × 33 (melt), 33 × 18 (solid); (c) 43 × 43 (melt), 43 × 23 (solid); Ra = 5000 and $\alpha = \pi/2$.

which is typical for free boundary problems [7, 16, 18]. A full row at the bottom of the matrix is due to the arclength continuation, but it cannot be seen from the figure. The topology of the Jacobian matrix provides useful information for the design of preconditioners for iterative matrix solvers [26]. It is also useful for the identification of independent columns for sparse finite difference evaluation, which will be discussed shortly.

Furthermore, because the interface variable $h(x^1)$ does not appear explicitly in the equation, there is no entry on the diagonal of the Jacobian matrix for the isotherm condition. This causes the breakdown of the ILU(0) preconditioning as well. To resolve this difficulty, $h(x^1)$ is introduced to the isotherm condition as a weighting factor. For example, for the nodal point i on the interface the isotherm condition can be rewritten as

$$f_{h(\xi_i^1)} = (\theta(\xi_i^1, 1) - 0)[h(\xi_{i-1}^1) + h(\xi_i^1) + h(\xi_{i+1}^1)], \quad 1 < i < N_{\xi_1}. \tag{31}$$

Through this modification, zeros could be avoided on the main diagonal. In addition, the number of entries from the Eq. (31) is much less than that from the flux balance formulation [7]. To calculate the Jacobian components for this equation only the perturbations on $\theta(\xi_i^1, 1), h(\xi_{i-1}^1), h(\xi_i^1)$, and $h(\xi_{i+1}^1)$ need to be considered. Other choices of the weighting factor are possible, but the present formulation is found effective.

Jacobian matrix with a known sparse pattern could be estimated either analytically or numerically. However, getting an analytical Jacobian for a large and complicated problem is usually difficult and error-prone, even with the help of symbolic differentiation, especially as complexity increases. The finite difference approach is the easiest way to compute the derivatives and has the advantage that one needs only the residual function $\mathbf{f}(\mathbf{y})$ as a "black box." For a residual fector $\mathbf{f}(\mathbf{y})$ with dimension NEQ (the total number of equations), the simplest way to approximate $\tilde{J}$ is to use the forward difference:

$$\frac{\partial f_i}{\partial y_j} \approx \frac{f_i(\mathbf{y} + \varepsilon_j \mathbf{e}_j) - f_i(\mathbf{y})}{\varepsilon_j}, \tag{32}$$

where $\mathbf{e}_j$ is the column j of the identity matrix and ε_j is a suitable step-length. For a dense matrix, the calculation of Eq. (32) requires NEQ computations of the residual vector. In other words, we need to calculate $\mathbf{f}(\mathbf{y} + \varepsilon_j \mathbf{e}_j)$ for each j. However, for a sparse matrix, each f_i depends only on a few x_j. Therefore, we can take this fact to reduce the number of residual evaluations [27–28]. For example, for a case with 33 × (33 + 18) grid points (or NEQ = 4949), if the last row (from the pseudo-arclength continuation) is excluded, the columns of $\tilde{J}$ can be grouped into 54 groups that no two have entries in the same row. Accordingly, only 54 residual evaluations are needed to estimate $\tilde{J}$. The components of Jacobian from the pseudo-arclength continuation are evaluated analytically. The small step size ε_j for y_j is estimated by

$$\varepsilon_j = \sqrt{u}\max(|y_j|, \mathrm{RTOL}_j|y_j| + \mathrm{ATOL}_j), \qquad (33)$$

where RTOL_j and ATOL_j are the relative and absolute error tolerances specified, and u is the unit roundoff error of the computer. For all variables, RTOL and ATOL are chosen in the order of 1×10^{-6}. The values ranging from 10^{-2} to 10^{-8} do not affect much the convergence characteristics. In this study, all calculation are performed in the HP9000/735 workstation with 80 DRAM.

4.3. *Solution of Linear Equations*

Solving the linear Eq. (30) requires the use of an efficient sparse matrix package. We have tested both direct matrix [29] and iterative matrix solvers [26, 30–31]. Among them, ILU (0) preconditioned GMRES method seems to be a promising one from the consideration of CPU time, memory, and robustness. GMRES is a technique introduced by Saad and Schultz [30] for solving general large sparse nonsymmetric linear system of equations by minimizing the 2-norm (Euclidean norm) of the residual vector. It has also been used widely in fluid flow computation [7, 13, 25]. With a suitable preconditioner, its efficiency and convergence could be enhanced significantly [13, 25, 32].

The incomplete LU decomposition without fill-in, ILU (0) [33], is one of the simplest and most popular preconditioners. The principle is to find a pair of unit lower triangular L and upper triangular U matrices such that $L + U$ has the same structure as the original matrix. The common way to obtain ILU (0) is to perform Gaussian elimination and replace any fill-in by a zero during the process. Since ILU (0) does not perform pivoting during factorization, it may fail due to zero diagonal elements in the Jacobian matrix as described previously. Furthermore, increasing the dimension of Krylov subspace in GMRES could reduce the number of iterations to convergence, but the memory space required is increased as well. The size of this subspace is chosen to be 100, and the convergence criterion with the L_2 norm of 10^{-9} is used.

5. RESULTS AND DISCUSSION

5.1. *The Performance of the Scheme*

The performance of the scheme is examined first through mesh refinements. Figure 4 shows three different meshes for Ra = 5000 and $\alpha = \pi/2$. Through the stretch function, more finite volumes are generated near the boundaries to enhance the accuracy of calculations. The numbers of unknowns are 2414, 4949, and 8384, respectively, for meshes (a) to (c). The calculated flow patterns and the interface shape are illustrated in Fig. 5(a). As shown, they

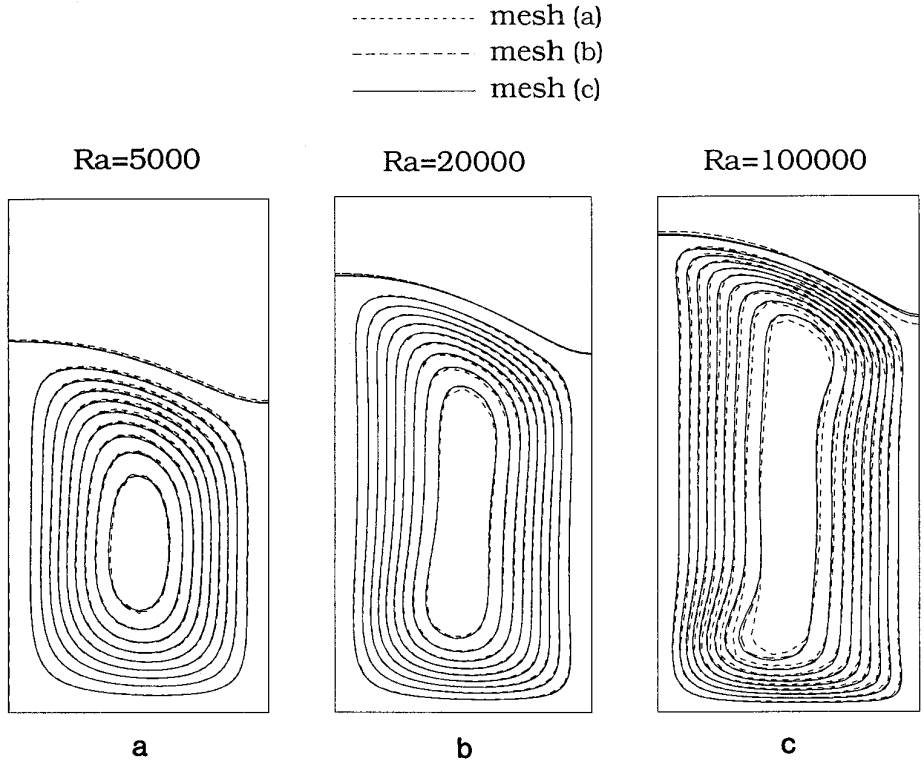

FIG. 5. Calculated results based on the meshes in Fig. 4 for (a) Ra = 5000, (b) Ra = 20000, and (c) Ra = 10^5; $\alpha = \pi/2$. Some calculated values are listed in Table II.

TABLE II

Comparison of Some Calculated Values Based on Different Meshes (from Fig. 5)

	$Ra = 5 \times 10^3$				$Ra = 2 \times 10^4$				$Ra = 1 \times 10^5$			
Mesh	Nu	h_{min}	h_{max}	$\lvert\psi\rvert_{max}$	Nu	h_{min}	h_{max}	$\lvert\psi\rvert_{max}$	Nu	h_{min}	h_{max}	$\lvert\psi\rvert_{max}$
(a)	1.583	1.192	1.448	3.551	2.596	1.381	1.701	6.141	4.661	1.502	1.862	10.33
(b)	1.580	1.194	1.445	3.580	2.568	1.389	1.694	6.226	4.494	1.531	1.852	10.45
(c)	1.579	1.196	1.444	3.589	2.557	1.391	1.691	6.264	4.426	1.539	1.847	10.55

are all very close to one another. Even at higher Ra numbers, as shown in Figs. 5 (b)–(c), the calculated results are not affected by the mesh much. The calculated Nu, the minimum (h_{min}) and maximum (h_{max}) interface heights, and the maximum stream function $|\psi|_{max}$ are listed in Table II for further comparison. Furthermore, from the results of different mesh sizes, mesh-independent solutions can be approximated from the Richardson extrapolation for second-order schemes. Solution errors on various meshes can then be estimated by subtracting the mesh-independent solutions. If we plot the estimated relative errors of the calculated Nu (E_{Nu}%) as a function of grid size using a logarithmic scale, the order of the present method can be estimated. Taking $Ra = 10^4$ and 10^5 as an example, as shown in Fig. 6, both curves indicate a slope of around 2. In other words, second-order accuracy of the scheme is retained.

The detailed CPU time used in various phases of the solution and the iteration numbers required for convergence in the calculations at various Ra numbers are summarized in Table III. In the column for iteration number, the numbers in the parenthesis are the iteration numbers of GMRES at each Newton's iteration. The memory required is listed in the last column; $\theta = 1$, $h(x^1) = 1$, and $u_1 = u_2 = P = 0$ are used as the initial guess for all cases. As shown, the computational effort for a numerical Jacobian is about half of that for Newton's linear equation at smaller Ra numbers. However, as the number of unknowns is increased, the iteration numbr required for GMRES increases; the increase is about twice from meshes

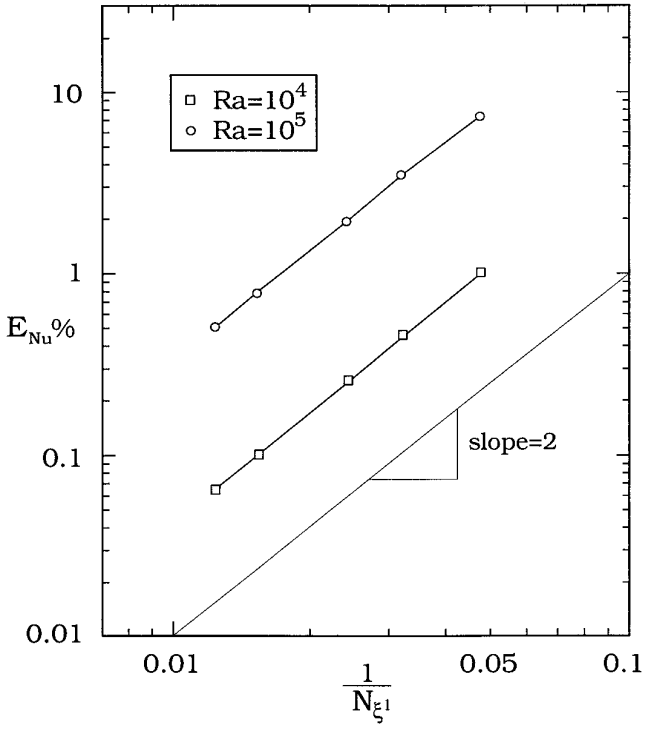

FIG. 6. Estimated errors of Nu as a function of grid sizes for $Ra = 10^4$ and 10^5; $\alpha = \pi/2$.

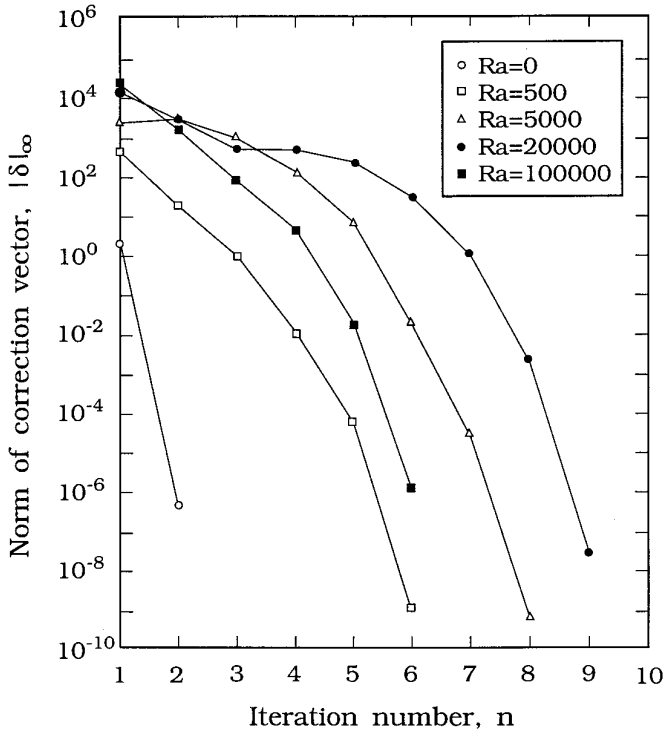

FIG. 7. Convergence of Newton's iterations for various Ra numbers; $\alpha = \pi/2$.

TABLE III

Comparison of Performance of Newton's Method on Different Meshes; $\alpha = \pi/2$

Meshes (NEQ)	Ra	Nu	CPU(s) for residual	CPU(s) for Jacobian	CPU(s) for GMRES	Number of Newton/GMRES iterations	Memory (Mbytes)
23 × (23 + 13)	0	1.0000	0.0300	4.982	5.270	2/63 (27,36)	
(2414)	500	1.0199	0.0300	9.537	16.72	5/214 (40,44,44,42,44)	
	1000	1.0731	0.0300	10.67	20.23	6/261 (40,43,44,44,45,45)	5.02
	3000	1.3533	0.0300	12.64	25.19	7/328 (46,42,45,48,47,47,53)	
	5000	1.5831	0.0400	12.66	26.76	7/352 (55,44,51,51,50,49,52)	
33 × (33 + 18)	0	1.0000	0.0500	11.06	16.26	2/96 (43,53)	
(4949)	500	1.0200	0.0700	24.28	57.43	6/350 (57,61,57,56,62,57)	
	1000	1.0734	0.0700	24.37	60.60	6/367 (56,61,63,60,64,63)	10.39
	3000	1.3524	0.0700	27.71	72.25	7/438 (58,58,65,65,65,66,61)	
	5000	1.5799	0.0800	31.06	86.83	8/527 (62,58,71,68,66,69,66,67)	
43 × (43 + 23)	0	1.0000	0.1200	20.74	42.27	2/133 (60,73)	
(8384)	500	1.0200	0.1200	45.06	161.5	6/505 (75,87,85,82,88,88)	
	1000	1.0735	0.1300	45.78	163.0	6/510 (76,81,89,84,88,92)	17.74
	3000	1.3521	0.1200	51.91	201.6	7/603 (70,73,85,93,91,91,100)	
	5000	1.5786	0.1400	58.18	250.4	8/795 (76,83,90,97,91,92,97,169)	

(a) to (c). Accordingly, to the finest mesh the computational effort for GMRES is about four times that for a Jacobian. It should be pointed out that, due to the coupling of the pressure and velocities in the equation of continuity through momentum interpolation, the number of groups for forming the Jacobian in all cases is 54, which is about 60% more than that for the ψ/ω formulation. This increases the computation time substantially. Also, if a better initial guess is used, the CPU time required for cases with higher Ra numbers could be reduced significantly. The memory required is linearly proportional to NEQ. For higher Ra numbers, the upwind scheme may be more robust. However, for the cases considered, the use of upwind degrades the accuracy, but it does not improve much the performance of GMRES (the difference is less than five iterations).

The convergence history on the L_∞ norm of the correction vector $\boldsymbol{\delta}$ is further illustrated in Fig. 7. As shown, nearly quadratic convergence of Newton's iterations is obtained when the solution is achieved for all cases. When the flow is absent (Ra = 0), only one Newton's iteration is required. For Ra greater than 5000, a better initial guess is necessary.

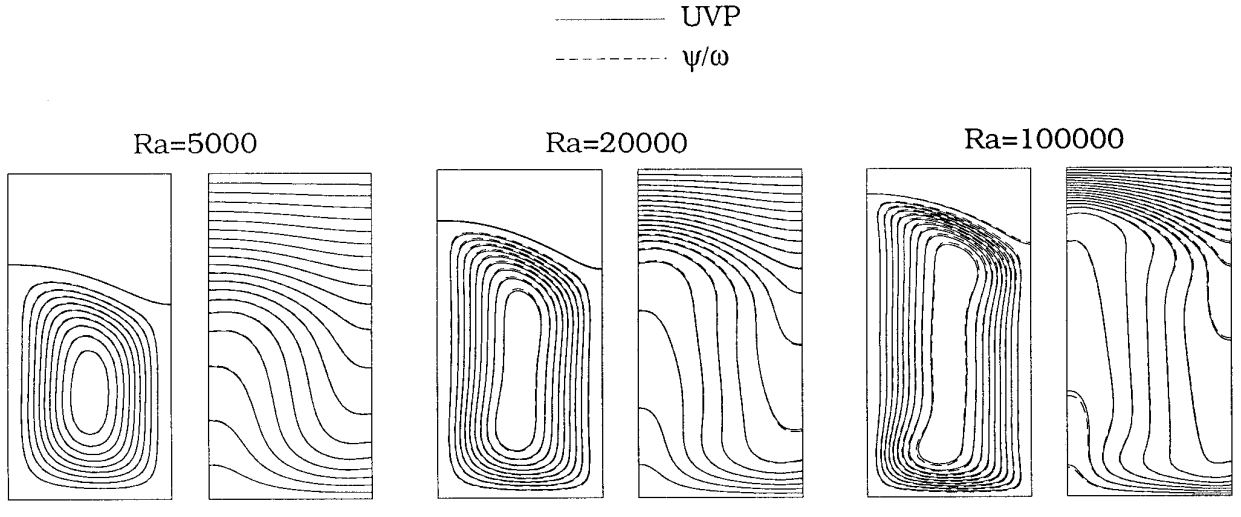

FIG. 8. Comparison of flow patterns, isotherms, and interface shape from primitive variable (UVP) and streamfunction/vorticity (ψ/ω) schemes: (a) Ra = 5000, (b) Ra = 20000, (c) Ra = 10^5; $\alpha = \pi/2$. The solid line is for the UVP scheme and the dashed-line is for the ψ/ω scheme. Some calculated values are listed in Table IV.

TABLE IV

Comparison of Some Calculated Values from UVP and ψ/ω Approaches (from Fig. 8)

	Ra = 5000				Ra = 20000				Ra = 100000			
Methods	Nu	h_{min}	h_{max}	$\lvert\psi\rvert_{max}$	Nu	h_{min}	h_{max}	$\lvert\psi\rvert_{max}$	Nu	h_{min}	h_{max}	$\lvert\psi\rvert_{max}$
UVP	1.579	1.196	1.444	3.589	2.257	1.391	1.691	6.264	4.426	1.539	1.847	10.55
ψ/ω	1.582	1.202	1.448	3.649	2.256	1.398	1.698	6.401	4.408	1.550	1.848	10.90

In Fig. 7, the solutions of Ra = 5000 and 20000 are used as initial guesses for Ra = 20000 and 10^5, respectively.

5.2. *Validation of the Method*

The validation of a numerical method is usually difficult, particularly when its benchmark solution is not available. Three approaches are used to assure the correctness of the numerical approach. First of all, the calculated results for Ra = 5000, 20000, and 10^5 ($\alpha = \pi/2$) using mesh (c) are compared with those obtained from the ψ/ω formulation [7]. As shown in Fig. 8, they are in excellent agreement for streamlines, isotherms, and the melt/solid interface. Some calculated results are listed in Table IV for further comparison. Since pressure is removed from the ψ/ω formulation, its number of unknowns is less. Furthermore, the number of groups for Jacobian estimation is also reduced to 33. Therefore, the CPU time required for the ψ/ω approach takes only one-half that for the primitive variables. Therefore, for two-dimensional cases the ψ/ω formulation still outperforms the UVP one. Nevertheless, as pointed out previously, its extension to three-dimensional problems is much more difficult.

Second, further validation is performed by comparing our calculated results with the benchmark solutions for the heated square problem [34–35]; the interface is fixed. The comparison is listed in Table V. As shown, the calculated results are still in excellent agreement with the benchmark solutions up to Ra = 1×10^6. Second-order accuracy of the present scheme is retained for this problem as well. Finally, a special debugging procedure [36] is used to check the code. In brief, a proposed analytical solution is assumed first; e.g., let $\theta = (x^1)^3(x^2)^4$, and then substitute this solution into the governing equations to generate a source term. Artificial boundary conditions are also obtained from the solution form. In other words, the proposed solution now becomes the analytical solution of the new governing equations with an artificial source term and boundary conditions. The FVM solution of this new set of partial differential equations is then compared with the proposed analytical solution. Again, excellent agreement is found.

5.3. *Test Examples*

The above calculations can be extended to the systematic study of flow bifurcation and transition. By using the pseudo-arclength continuation, one could calculate the solutions from one to the other. Particularly, the method is still robust even when a limiting point is encountered. Figure 9 shows the bifurcation diagram for Ra = 2000. As shown the solution is symmetric with respect to $\alpha = 0$. A no-flow solution (Nu = 1) is found for $\alpha = 0, \pm\pi$. At $\alpha = 0$, the configuration becomes a typical Rayleigh–Benard problem. Since the lower boundary is hotter and, thus, the melt is lighter there, the top-heavy arrangement is potentially unstable. However, because Ra is not large enough, the flow is inhibited by its viscosity leading to a static solution (Nu = 1). At $\alpha = \pm\pi$, the thermal configuration is physically stable; the melt is on the top with respect to the gravity. Since the lateral temperature gradient is zero, no flow is induced. Heat transfer in both cases are through conduction alone. However, as the system is tilted, natural convection is induced due to lateral temperature gradients. As a result, heat transfer is enhanced leading to

TABLE V

Comparison of Calculated Nu with Benchmark Solutions [34, 35] for the Heated Square Problem

Ra \ Mesh	23 × 23	33 × 33	43 × 43	63 × 63	83 × 83	Davis and Jones [34]	Hortmann et al. [35]
10^3	1.1163	1.1172	1.1175	1.1177	1.1179	1.118	—
10^4	2.2551	2.2493	2.2472	2.2457	2.2453	2.243	2.2448
10^5	4.6521	4.5812	4.5545	4.5351	4.5293	4.519	4.5216
10^6	9.3914	9.2031	9.0477	8.9194	8.8759	8.800	8.8251

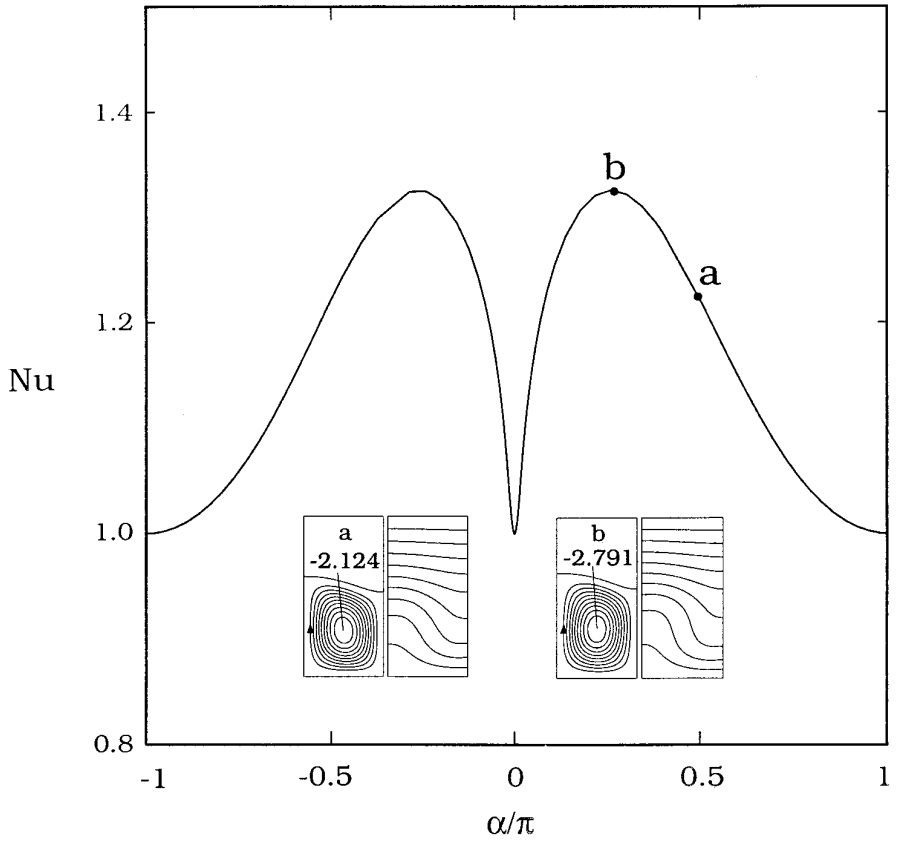

FIG. 9. Bifurcation diagram for Ra = 2000. Typical flow patterns and isotherms are also included.

the increase of the Nu number. The maximum Nu number occurs at the tilted angle about $\pm\pi/4$, where the convection is expected to be the strongest.

When the Ra number is increased to 3000, as shown in Fig. 10, multiple steady states appear near $\alpha = 0$. For example, at $\alpha = 0$, there are three solutions: b, b′, and the static solution (Nu = 1). Typical flow patterns and isotherms along the solution branch are also illustrated in the same figure. More interestingly, the calculated solutions for b and b′ are mirror images of each other; they have the same Nu number as well. In fact, at $\alpha = 0$, because the top-heavy static solution is not a unique state any more, it becomes unstable. A small disturbance could turn this state to a more stable state b or b′, depending on the perturbation direction. In other words, solutions b and b′ are physically stable. Because stability exchanges at simple limiting points [37], in Fig. 10 the solution branch (including the static solution) below two limiting points is not stable. Far away from the zero tilt angle, no multiple steady states are observed and the solution is stable.

Further, increasing the Ra number could lead to more complicated solution structures near $\alpha = 0$. As shown in Fig. 11, in which Ra = 7000, in addition to the main solution branch from $\alpha = -\pi$ to π, an isola appears near the static state at $\alpha = 0$. We also highlight the isola, and the associated flow patterns and isotherms (a, b, and c) are also illustrated. Apparently, its flow structure changes and becomes two-cell. As shown, for the points c and h, the two flow cells are symmetrical with respect to the centerline. More interestingly, the flow directions of c and h are just the opposite. Accordingly, the interface for c becomes concave, while that for h convex. Again, the static solution (on the isola) is not stable, and the solutions c, f, and f′ are the more stable states. From the exchange of stability at simple turning points, h seems to be an unstable solution. However, without a careful examination of its eigenvalues, the conclusion cannot be drawn. The two-cell flow structure of the tilted cases, e.g., a and b, are asymmetrical. In fact, below the turning point (near point g) along the main solution branch, the flow structure is two-cell as well. For example, the flow pattern of g consists of a larger cell, and a smaller cell near the lower right corner. When Ra = 10000 in Fig. 12, the bifurcation diagram becomes even more complicated near $\alpha = 0$. The size of isola also increases, and the solution multiplicity increases as well.

If we use the previous calculations at $\alpha = 0$ as starting points, the bifurcation diagram using the Ra number as the parameter can be constructed easily. To distinguish

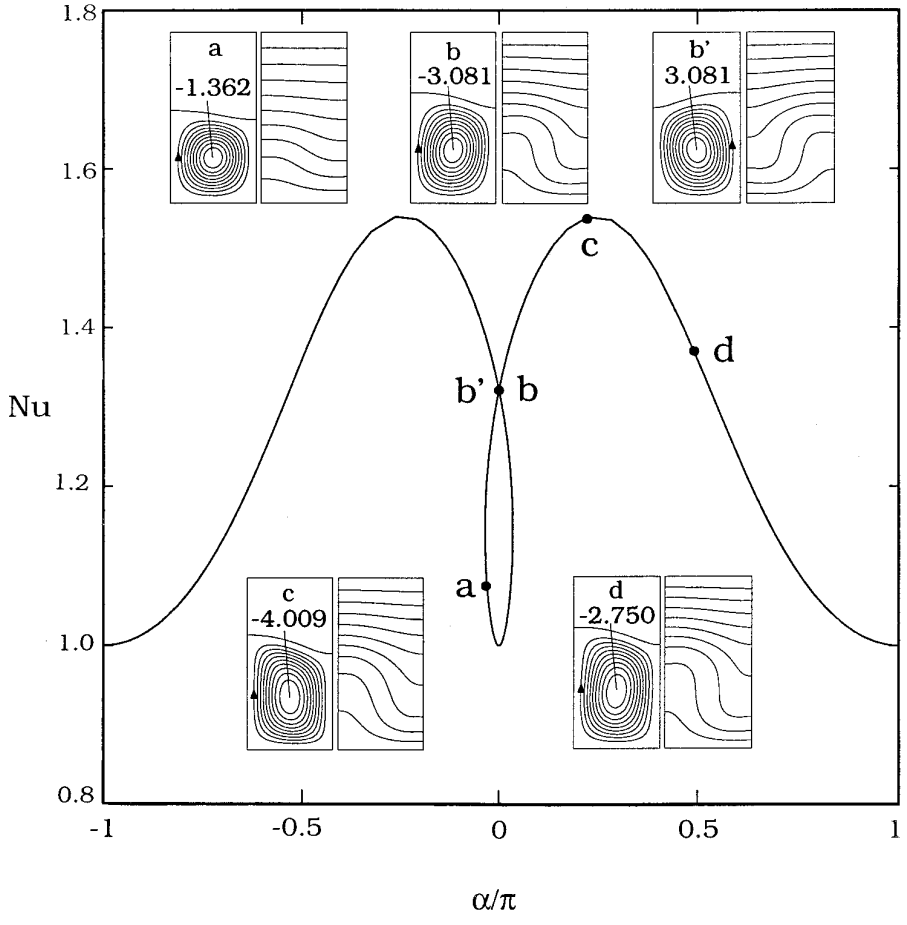

FIG. 10. Bifurcation diagram for Ra = 3000.

the symmetric solutions, we redefine the Nu by subtracting it by 1 (the static solution) and adding a minus sign to the subtracted value when the flow cell (choosing the left cell for two-cell structures) is counterclockwise in direction. With this newly defined Nu number, the bifurcation diagram is constructed (Fig. 13) and some typical flow patterns are illustrated. As shown, there are four critical Ra numbers (Ra_{c1} = 2301, Ra_{c2} = 6577, Ra_{c3} = 8845, and Ra_{c4} = 8863) in this diagram. For the traditional Raleigh–Benard problem, the width goes to infinity. In such a case, the Ra_{c1} is 1708. However, as the aspect ratio increases, the Ra_{c1} increases [38–39]. For the fixed boundary case with the aspect ratio of 1 for the melt region, the Ra_{c1} is slightly higher, being 2585 [38–39]. At the first bifurcation point, in addition to the static state, the flow bifurcates into two symmetrical solution branches, i.e., the supercritical pitchfork bifurcation, and this is similar to the fixed boundary case [38]. Across Ra_{c1}, the static solution changes its stability and becomes unstable, while the other two solution branches are stable. Also, comparing the values of Ra_{c1} for single-phase (2585) and two-phase (2301) systems suggests that the presence of the melt/solid interface could destabilize the static. A similar conclusion was drawn by Chang and Brown [1] for the axisymmetrical case. Furthermore, because of the effect of the melt/solid interface, the flow structure could become even more complicated as the Ra number increases.

Beyond Ra_{c2}, the flow becomes two-cell. However, the solution branches for c and d are not perfectly symmetrical to each other any more, due to the interface; the interface for c is concave, while that for d is convex. More interestingly, beyond Ra_{c3} or Ra_{c4} new solution branches come out from the symmetrical ones. The flow structure for the new solution branches happens to be asymmetrical two-cell, such as e and f, or g and h. Furthermore, the solutions e and f, or g and h, have the same Nu number, because they are perfectly symmetrical to each other. In other words, beyond Ra_{c4}, there are nine solutions (including the static solution). Similar bifurcation structure was reported for Lapwood convection [40].

The bifurcation diagram on the Nu $-$ α plane for $\alpha \neq 0$ can be constructed easily as well. In brief, if the tilt angle

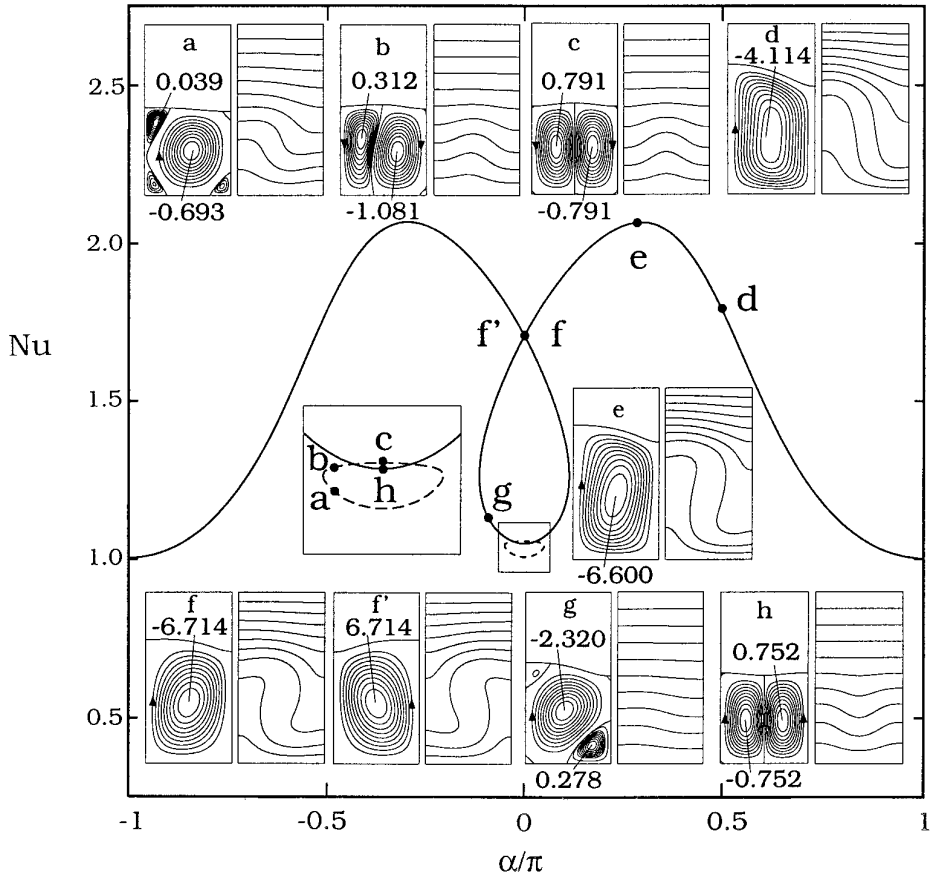

FIG. 11. Bifurcation diagram for Ra = 7000. A small region near $\alpha = 0$ is highlighted for the illustration of solutions a, b, c, and h. The dashed-line indicates a different solution branch (an isola).

α is slightly greater than 0, except the first lower branch, the other branches become isolated, while the critical Ra numbers shift backward. The imperfect bifurcation can be understood by comparing the buckling of an initially bent rod. With an increasing load (like the Ra number), the preferred bending direction (like the flow direction) is determined by the initial bend. For example, with a small α, the counterclockwise flow (the lower solution branch) is stable and dominant with the increasing Ra number from Ra = 0. The stable isolated solutions (clockwise flow) is possible at larger Ra numbers, but they need enough perturbation from the initial counterclockwise flow. Furthermore, from the bifurcation results of Figs. 9–13, a cusp catastrophe [41] in the Ra $-$ α bifurcation plane can be constructed easily. Furthermore, because of the robustness of the present scheme, no trouble is encountered in constructing the whole bifurcation diagrams using Ra or α as the parameter.

Although the illustrated examples so far are only two-dimensional, extension of our scheme to three-dimensional problems is quite straightforward. We are currently applying the scheme to the three-dimensional modeling of

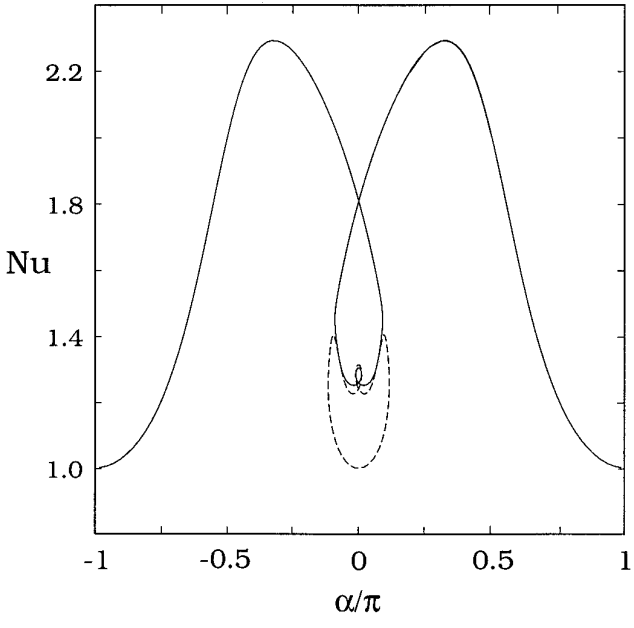

FIG. 12. Bifurcation diagram for Ra = 10000. The dashed-line indicates a different solution branch.

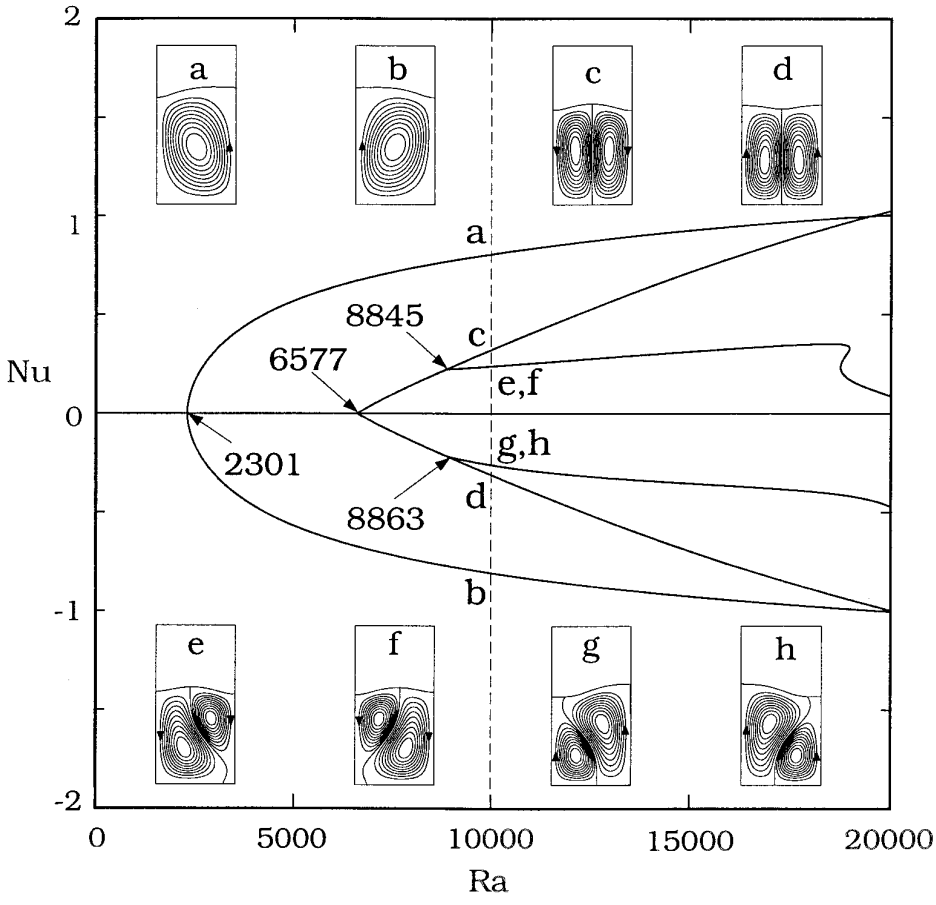

FIG. 13. Bifurcation diagram for $\alpha = 0$. The arrows indicate the critical Ra numbers.

Bridgman crystal growth. Fluid flow, heat and mass transfer, and the growth interface are computed simultaneously by Newton's method. Some preliminary results can be found elsewhere [42].

6. CONCLUSIONS

The numerical solution of the two-phase incompressible heat flow in an inclined enclosure with an unknown melt/solid interface by a FVM/Newton's method using primitive variables and collocated grids is proposed. The velocity/pressure coupling is enhanced through a modified Rhie–Chow scheme. During Newton's iterations, the Jacobian is estimated numerically, while the linear equations are solved by the ILU (0) preconditioned GMRES solver. With the implementation of the pseudo-arclength continuation, this approach proved to be robust and efficient; nearly quadratic convergence of the method is observed. The solution is also validated by the streamfunction/vorticity solution, and they are in excellent agreement. Flow transition and bifurcation of the problem are illustrated as well.

APPENDIX

The implementation of the pressure boundary conditions is further described here. Take a wall cell, labeled by W as an example. As shown in Fig. 2, the cell P is next to it, while the cell E is next to the cell P. Because the cell volume is zero at the wall, the points W and w are at the same point, i.e., $\overline{Ww} = 0$. The linear extrapolation of the variable $\wp$ from the cells E and P to the wall cell at W can be represented as

$$\wp_W - \wp_P = \frac{\overline{WP}}{\overline{PE}}(\wp_P - \wp_E). \tag{A1}$$

With the definition of $\wp$ in Eq. (18), the above equation can be written in terms of pressure and body force:

$$\begin{aligned} &-F_k|_w(x_P^k - x_W^k) + (P_P - P_W) \\ &= \frac{\overline{WP}}{\overline{PE}}[-F_k|_e(x_E^k - x_P^k) + (P_E - P_P)]. \end{aligned} \tag{A2}$$

Therefore, the pressure P at wall (P_W) can then be represented explicitly as

$$P_W = P_P - F_k|_w(x_P^k - x_W^k) + \frac{\overline{WP}}{\overline{PE}}[F_k|_e(x_E^k - x_P^k) - (P_E - P_P)]. \tag{A3}$$

The extrapolated value is used for the pressue boundary condition at the solid wall.

The implementation of $\partial\wp/\partial n = 0$ is straightforward as well. Its value at the solid wall, labeled by w, can be written in terms of coordinates ξ^i:

$$\left.\frac{\partial\wp}{\partial n}\right|_w = \frac{1}{J|A^1|}\left(A_j^1 A_j^i \frac{\partial\wp}{\partial\xi^i}\right)\Bigg|_w = 0. \tag{A4}$$

Using Eq. (18) gives

$$\left[A_j^1 A_j^i\left(F_k\frac{\partial x^k}{\partial\xi^i} - \frac{\partial P}{\partial\xi^i}\right)\right]_w = 0. \tag{A5}$$

After finite difference approximation, we obtain the equation for the pressue boundary condition,

$$(A_j^1 A_j^1)_w[F_k|_w(x_P^k - x_W^k) - (P_P - P_W)] + (A_j^1 A_j^2)_w[F_k|_w(x_{nw}^k - x_{sw}^k) - (P_{nw} - P_{sw})] = 0, \tag{A6}$$

where A_j^2, $j = 1, 2$, at w are evaluated using points W and P. The second term of Eq. (A6) is due to the nonorthogonal mesh on the boundary.

Both implementations, Eqs. (A3) and (A6), are tested in this study, but the extrapolation scheme gives a slightly better result as compared with the mesh-independent solution. Nevertheless, second-order accuracy is retained for both approaches.

ACKNOWLEDGMENTS

This work is supported by the National Science Council of the Republic of China under Grant NSC84-2215-E008-011. C.W.L. also acknowledges Professor Youcef Saad of the University of Minnesota for making SPARSKIT2 available to us.

REFERENCES

1. J. Chang and R. A. Brown, *J. Comput. Phys.* **53,** 1 (1984).
2. C. J. Kim and M. Kaviany, *Int. J. Heat Mass Transfer* **35,** 457 (1992).
3. M. Lacroix, *Numer. Heat Transfer* **22,** 79 (1992).
4. R. Viswanath and Y. Jaluria, *Numer. Heat Transfer B* **24,** 77 (1993).
5. S. K. Sinha, T. Sundararajan, and V. K. Garg, *Int. J. Heat Mass Transfer* **36,** 2349 (1993).
6. J. C. Chen, C. F. Chu, and W. F. Ueng, *Int. J. Heat Mass Transfer* **37,** 1733 (1994).
7. C. W. Lan, *Int. J. Numer. Methods Fluids* **19,** 41 (1994).
8. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, Washington, DC, 1980).
9. M. Peric, R. Kessler, and G. Scheuerer, *Comput. & Fluids* **16,** 389 (1988).
10. J. W. MacArthur and S. V. Patanker, *Int. J. Numer. Methods Fluids* **9,** 325 (1989).
11. H. E. Bailey and R. M. Beam, *J. Comput. Phys.* **93,** 108 (1991).
12. D. A. Knoll and P. R. McHugh, *Int. J. Numer. Methods Fluids* **17,** 449 (1993).
13. P. R. Mchugh and D. A. Knoll, *Int. J. Numer. Methods Fluids* **19,** 439 (1994).
14. C. M. Rhie and W. L. Chow, *AIAA J.* **21,** 1527 (1983).
15. D. Kwak, S. E. Roger, U. K. Kaul, and J. C. L. Chang, NASA Tech. Memo 88319, 1986 (unpublished).
16. H. M. Ettouney and R. A. Brown, *J. Comput. Phys.* **49,** 118 (1983).
17. C. W. Lan and S. Kou, *Int. J. Numer. Methods Fluids* **12,** 59 (1991).
18. S. F. Kistler and L. E. Scriven, *Int. J. Numer. Methods Fluids* **4,** 207 (1984).
19. D. S. Dandy and L. G. Leal, *Int. J. Numer. Methods Fluids* **9,** 1469 (1989).
20. H. B. Keller, "Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems," in *Applications of Bifurcation Theory*, edited by P. H. Rabinowiz (Academic Press, New York, 1977).
21. L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, 2nd ed., Vol. 6 (Pergamon Press, Elmsford, NY, 1987), p. 217.
22. M. C. Melaaen, *Numer. Heat Transfer B* **21,** 1 (1992).
23. I. Demirdzic, Z. Lilek, and M. Peric, *Int. J. Numer. Methods Fluids* **16,** 1029 (1993).
24. W. Rodi, S. Majumdar, and B. Schonung, *Comput. Methods Appl. Mech. Eng.* **75,** 369 (1989).
25. P. Chin, E. F. D'Azevedo, P. A. Forsyth, and W. P. Tang, *Int. J. Numer. Methods Fluids* **15,** 273 (1992).
26. Y. Saad, RIACS Tech. Rep. 90.20, May 1990 (unpublished).
27. A. R. Curtis, M. J. D. Powell, and J. K. Reid, *J. Inst. Math. Appl.* **13,** 117 (1974).
28. T. F. Coleman, B. S. Garbow, and J. J. More, *ACM Trans. Math. Software* **10,** 329 (1984).
29. I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices* (Clarendon, Oxford, 1986), p. 268.
30. Y. Saad and M. H. Schultz, *SIAM J. Sci. Stat. Comput.* **7,** 856 (1986).
31. M. Seager, Lawrence Livermore National Laboratory Technical Rep., UCRL-100267, December 1988 (unpublished).
32. K. Ajmani, W. F. Ng, and M. S. Liou, *J. Comput. Phys.* **110,** 68 (1994).
33. J. A. Meijerink and H. A. van der Vorst, *Math. Comput.* **31,** 148 (1977).
34. G. De Vahl Davis, *Int. J. Numer. Methods Fluids* **3,** 249 (1983).
35. M. Hortmann, M. Peric, and G. Scheuerer, *Int. J. Numer. Methods Fluids* **11,** 189 (1990).
36. T. M. Shih, *Int. J. Numer. Methods Eng.* **21,** 1027 (1985).
37. G. Iooss and D. D. Joseph, *Elementary Stability and Bifurcation Theory*, 2nd ed. (Springer-Verlag, New York, 1990), p. 19.
38. J. M. Luijkx and J. K. Platten, *J. Non-equilib. Thermodyn.* **6,** 141 (1981).
39. C. P. Jackson and K. H. Winters, *Int. J. Numer. Methods Fluids* **4,** 127 (1984).
40. D. S. Riley and K. H. Winters, *J. Fluid Mech.* **204,** 325 (1989).
41. K. A. Cliffe and K. H. Winters, *J. Comput. Phys.* **54,** 531 (1984).
42. M. C. Liang and C. W. Lan, *J. Crystal Growth*, in press.