

Multigrid Methods for Incompressible Heat Flow Problems with an Unknown Interface

C. W. Lan^{*.1} and M. C. Liang[†]

^{*}Chemical Engineering Department, National Taiwan University, Taipei, Taiwan 106, Republic of China; and

[†]Chemical Engineering Department, National Central University, Chung-Li, Taiwan 320, Republic of China

E-mail: lan@ruby.che.ntu.edu.tw

Received April 17, 1998; revised January 29, 1999

Finite-volume/multigrid methods are presented for solving incompressible heat flow problems with an unknown melt/solid interface, mainly in solidification applications, using primitive variables on collocated grids. The methods are implemented based on a multiblock and multilevel approach, allowing the treatment of a complicated geometry. The inner iterations are based on the SIMPLE scheme, in which the momentum interpolation is used to prevent velocity/pressure decoupling. The outer iterations are set up for interface update through the isotherm migration method. V-cycle and full multigrid (FMG) methods are tested for both two- and three-dimensional problems and are compared with a global Newton's method and a single-grid method. The effects of Prandtl and Rayleigh numbers on the performance of the schemes are also illustrated. Among these approaches, FMG has proven to be superior on performance and efficient for large problems. Sample calculations are also conducted for horizontal Bridgman crystal growth, and the performance is compared with that of traditional single-grid methods. © 1999 Academic Press

Key Words: multigrid; finite-volume method; solidification; interface.

1. INTRODUCTION

Solidification processes, such as casting and crystal growth, involve incompressible heat flow and an unknown shape of the solidification (growth) front. 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. Especially, for high Prandtl number materials, the interface is affected significantly by convection. Several numerical approaches [e.g., Refs. 1–8] for solving such problems have been proposed. 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, 8], and from the solution point

¹ To whom correspondence should be addressed.

of view, by decoupled [2–6] and global (coupled) [1, 7, 8] iteration approaches. The global approaches [1, 7, 8] have shown great robustness, and with the iterative matrix solver they are much more efficient than direct solvers for larger problems. However, as compared with traditional decoupled point or line iteration schemes, for large problems, especially for three-dimensional (3D) problems, the iterative matrix solvers still require a tremendous amount of computer memory and often need the use of supercomputers. This has discouraged the performance of their 3D calculations in an engineering workstation. The ψ/ω approach, which requires more equations for 3D problems, is less efficient than the UVP approach. Accordingly, the decoupled UVP method seems to be a better choice for large 3D applications.

The use of a decoupled method [e.g., 2–6], especially the point iteration approach, could greatly reduce the memory required. However, most iterative schemes, e.g., the Gauss–Siedel method, are good smoothers only for short wavelength errors [9]. For long wavelength errors, the convergence rate is slow. For large problems, the convergence rate thus degrades very quickly as the iteration proceeds, especially as the grid is refined. The convergence rate becomes slower as the problem size increases. Accordingly, multigrid (MG) methods that use different meshes to smooth errors with a wider spectrum of wavelengths have been used widely to overcome the difficulty of slow convergence [9, 10]. Very often, if the implementation is right, the convergence is not affected by the problem size, and its operation counts are linearly proportional to the problem size as well. Accordingly, MG methods have been regarded as one of the most effective methods in the computational fluid dynamics (CFD) community [11, 12]. Many studies [e.g., 13, 14] have reported the successful implementation of MG on incompressible flow applications, and its performance is very impressive. However, applications to free-boundary or free-interface problems are very limited. The work by Farmer *et al.* [15] seems to be the only application of MG to the free-surface problem. No applications of MG methods to solidification problems, i.e., with a melt/solid interface, have been reported.

In the present report, approaches implementing the isotherm migration method (IMM) upon a MG incompressible flow solver are presented for solidification problems. The incompressible flow solver design is based on a multiblock and multilevel concept for the treatment of complicated geometry. In each block/level, a collocated-grid finite-volume method (FVM) is used, in which the SIMPLE scheme [16] is used as a smoother. The strategies for interface update based on both V-cycle and full MG (FMG) methods are compared for 2D and 3D problems. A global Newton's method used in our previous report [8] is also tested for comparison purposes. In the next section, the physical model used is described. The FVM/MG implementation based on the multiblock and multilevel concept is discussed in Sections 3 and 4. Section 5 is devoted to results and discussion, followed by conclusions in Section 6.

2. FORMULATION FOR PHYSICAL PROBLEMS

The 2D and 3D steady-state two-phase heat flow problems studied in this report are shown in Figs. 1a and 1b, respectively. The material is partially solidified in a cavity; the interface is coupled with heat transfer and fluid flow. The upper surface can be either a free surface or a solid wall. A simple extension of the two-phase problem is shown in Fig. 1c for horizontal Bridgman (HB) crystal growth, which is one of the major processes

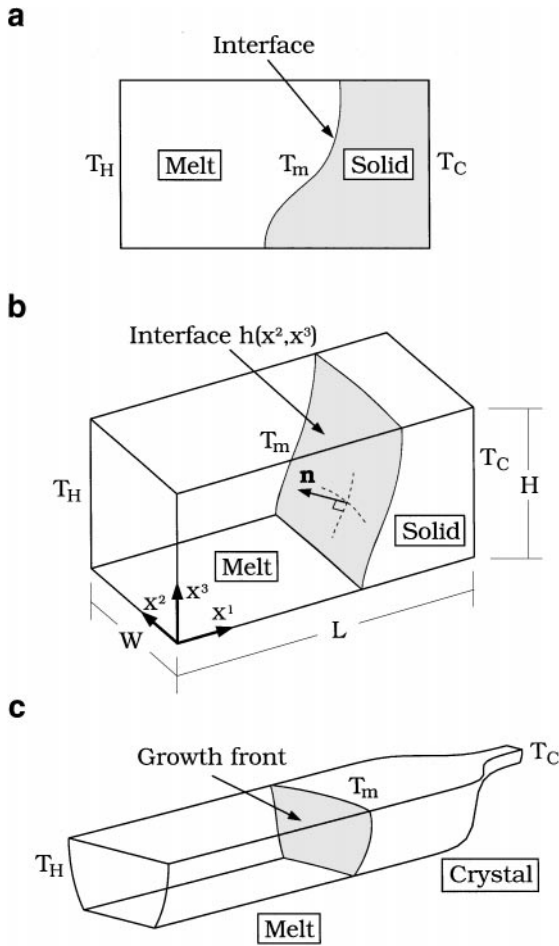


FIG. 1. Schematics of the two- and three-dimensional two-phase heat flow problems: (a) two dimensional (2D), (b) three dimensional (3D), and (c) horizontal Bridgman crystal growth.

for growing GaAs single crystals [17]. In the HB process, the upper surface is usually open to the gas environment and is thus a free surface. To focus on the numerical study, we have purposely chosen Fig. 1b for the formulation of governing equations. Other applications can be extended from here easily [e.g., Refs. 18, 19].

As illustrated in Fig. 1b, the temperature on the two ends is fixed; the left temperature T_H is higher than the melting point (T_m) of the material inside, while the right temperature T_C is lower. The other sides of the system are assumed adiabatic. Again, the case with heat loss or fixed temperature can also be treated easily if needed. The melt/solid interface position $h(x^2, x^3)$ is unknown *a priori*, and needs to be solved simultaneously with other field variables. In the applications of crystal growth, which are our major interests, the movement of the system (in the order of mm/h) is much slower than the melt velocity. Therefore, the contribution due to the movement of the system is considered only at the interface to take the heat of fusion into account. The melt is assumed incompressible and the flow laminar. The dimensionless variables are defined by scaling the length with the height H , velocity with α_m/H , and pressure with $\rho_m \alpha_m^2/H^2$, where α_m is the thermal diffusivity and ρ_m the melt density. The dimensionless temperature (θ) is defined as $\theta \equiv (T - T_m)/(T_H - T_m)$.

With the Boussinesq approximation [20], the conservation equations in dimensionless form for 3D steady incompressible laminar flow of a Newtonian fluid in the melt, and heat conduction in the solid, can be described as follows:

Melt

$$\frac{\partial u_1}{\partial x^1} + \frac{\partial u_2}{\partial x^2} + \frac{\partial u_3}{\partial x^3} = 0, \quad (1)$$

$$\frac{\partial}{\partial x^1} \left(u_1 u_1 - \text{Pr} \frac{\partial u_1}{\partial x^1} \right) + \frac{\partial}{\partial x^2} \left(u_2 u_1 - \text{Pr} \frac{\partial u_1}{\partial x^2} \right) + \frac{\partial}{\partial x^3} \left(u_3 u_1 - \text{Pr} \frac{\partial u_1}{\partial x^3} \right) + \frac{\partial P}{\partial x^1} = 0, \quad (2)$$

$$\frac{\partial}{\partial x^1} \left(u_1 u_2 - \text{Pr} \frac{\partial u_2}{\partial x^1} \right) + \frac{\partial}{\partial x^2} \left(u_2 u_2 - \text{Pr} \frac{\partial u_2}{\partial x^2} \right) + \frac{\partial}{\partial x^3} \left(u_3 u_2 - \text{Pr} \frac{\partial u_2}{\partial x^3} \right) + \frac{\partial P}{\partial x^2} = 0, \quad (3)$$

$$\begin{aligned} & \frac{\partial}{\partial x^1} \left(u_1 u_3 - \text{Pr} \frac{\partial u_3}{\partial x^1} \right) + \frac{\partial}{\partial x^2} \left(u_2 u_3 - \text{Pr} \frac{\partial u_3}{\partial x^2} \right) \\ & + \frac{\partial}{\partial x^3} \left(u_3 u_3 - \text{Pr} \frac{\partial u_3}{\partial x^3} \right) + \frac{\partial P}{\partial x^3} - \text{Ra Pr} \theta = 0, \end{aligned} \quad (4)$$

$$\frac{\partial}{\partial x^1} \left(u_1 \theta - \frac{\partial \theta}{\partial x^1} \right) + \frac{\partial}{\partial x^2} \left(u_2 \theta - \frac{\partial \theta}{\partial x^2} \right) + \frac{\partial}{\partial x^3} \left(u_3 \theta - \frac{\partial \theta}{\partial x^3} \right) = 0. \quad (5)$$

Solid

$$\kappa_s \left[\frac{\partial}{\partial x^1} \left(\frac{\partial \theta}{\partial x^1} \right) + \frac{\partial}{\partial x^2} \left(\frac{\partial \theta}{\partial x^2} \right) + \frac{\partial}{\partial x^3} \left(\frac{\partial \theta}{\partial x^3} \right) \right] = 0. \quad (6)$$

In the above equations, u_1 , u_2 , and u_3 are the Cartesian velocities in the x^1 , x^2 , and x^3 directions, respectively, and P is the pressure. The Prandtl (Pr) and Rayleigh (Ra) numbers are defined as $\text{Pr} \equiv \nu_m / \alpha_m$ and $\text{Ra} \equiv \beta g (T_H - T_m) H^3 / (\nu_m \alpha_m)$, where ν_m is the kinematic viscosity, β the thermal expansion coefficient, and g the gravitational acceleration. In Eq. (6), κ_s is the thermal conductivity ratio of the solid and the melt.

To solve the previous governing equations, the following boundary conditions are used,

$$\theta(0, x^2, x^3) = \theta_H; \quad \theta(L/H, x^2, x^3) = \theta_C, \quad (7)$$

$$\partial \theta(x^1, 0, x^3) / \partial x^2 = \partial \theta(x^1, W/H, x^3) / \partial x^2 = 0, \quad (8)$$

$$\partial \theta(x^1, x^2, 0) / \partial x^3 = \partial \theta(x^1, x^2, 1) / \partial x^3 = 0,$$

and on all melt boundaries, the no-slip boundary condition is used, i.e.,

$$u_1 = u_2 = u_3 = 0. \quad (9)$$

For the case with a free surface, the shear stress balance on the free surface is imposed,

$$\frac{\partial u_1}{\partial x^3} = \text{Ma} \frac{\partial \theta}{\partial x^1}; \quad \frac{\partial u_2}{\partial x^3} = \text{Ma} \frac{\partial \theta}{\partial x^2}, \quad (10)$$

where the Marangoni number $\text{Ma} \equiv (\partial \gamma / \partial T) (T_H - T_m) H / (\mu_m \alpha_m)$, and μ_m is the melt viscosity. The free surface deformation is neglected in Eq. (10).

At the melt/solid interface, the thermal flux balance is imposed as

$$\partial\theta/\partial n|_m - \kappa_s \partial\theta/\partial n|_s + \text{St Pe}(\mathbf{n} \cdot \mathbf{e}_{x^1}) = 0, \quad (11)$$

as well as the melting-point isotherm

$$\theta = 0, \quad (12)$$

where the Stefan number $\text{St} \equiv \Delta H/(C p_s T_m)$ and the Peclet number $\text{Pe} \equiv U_g \rho_s C p_s H/k_m$; ΔH is the heat of fusion; and $C p_s$ and ρ_s are the specific heat and solid density, respectively. U_g is the solidification rate of the interface and k_m the melt thermal conductivity.

The measure of convective heat transfer is through the \overline{Nu} number as

$$\overline{Nu} = \frac{1}{S} \int_S \frac{\partial\theta}{\partial n} dS, \quad (13)$$

where \mathbf{n} is the unit normal vector at the interface pointing into the melt and S is the surface area of the interface. For the case without convection (purely conductive heat transfer), $\overline{Nu} = (\theta_H - \theta_C)H/L$ for $\kappa_s = 1$.

3. FINITE-VOLUME METHOD

The physical problem can be easily decomposed into a number of blocks for analysis. For example, two blocks are adopted for Fig. 1b: one is for the melt and one for the solid. As the complexity of the problem increases, more blocks are needed. Multiple blocks can also be used in the same media, like in the melt. In each block, the finite-volume method based on the structured mesh used in the previous report [8] is adopted. The grid at the block interface can be coherent or incoherent. Although the incoherent interface is more versatile for a complicated domain, its implementation is tedious. Therefore, in this study only the coherent interface is considered.

3.1. Coordinate Transformation

Due to the unknown and deformed interface shape $h(x^2, x^3)$, structured body-fitted coordinates (ξ^1, ξ^2, ξ^3) are adopted for each block. The algebraic coordinate transformation for both melt and solid phases are performed as follows:

Melt

$$x^1 = \bar{\xi}^1(\xi^1)h(\xi^2, \xi^3), \quad (14)$$

$$x^2 = \bar{\xi}^2(\xi^2)W/H, \quad (15)$$

$$x^3 = \bar{\xi}^3(\xi^3). \quad (16)$$

Solid

$$x^1 = h(\xi^2, \xi^3) + \bar{\xi}^1(\xi^1)(L/H - h(\xi^2, \xi^3)), \quad (17)$$

$$x^2 = \bar{\xi}^2(\xi^2)W/H, \quad (18)$$

$$x^3 = \bar{\xi}^3(\xi^3). \quad (19)$$

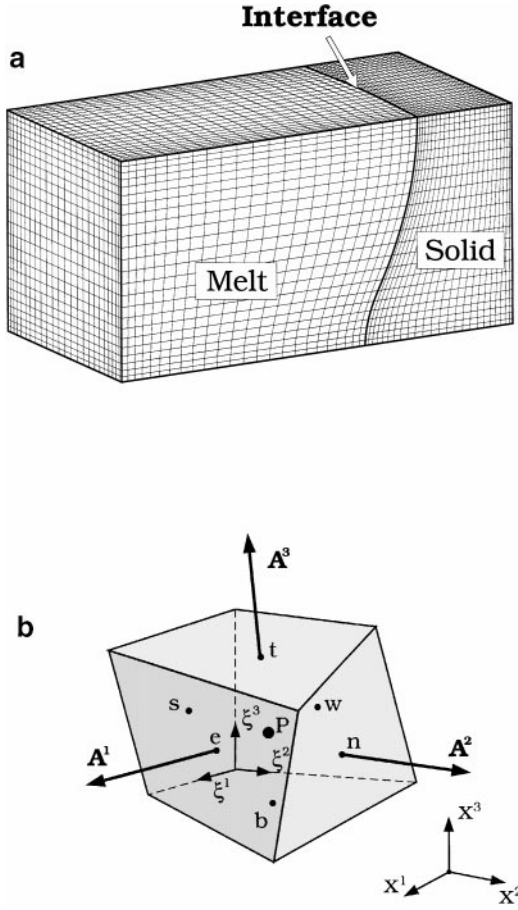


FIG. 2. (a) A sample mesh and (b) a 3D finite-volume and labeling scheme.

Here $\bar{\xi}^1(\xi^1)$, $\bar{\xi}^2(\xi^2)$, and $\bar{\xi}^3(\xi^3)$ are stretch functions ranging from 0 to 1 for adjusting grid distribution. A hyperbolic tangent function with the following form, for example, for $\bar{\xi}^1$, is used,

$$\bar{\xi}^1(\xi^1) = 0.5 \left[1 + \tanh \left(\frac{B[(\xi^1 - 1)/(N_{\xi^1} - 1) - 0.5]}{0.5B} \right) \right], \quad (20)$$

where B is a stretch constant and N_{ξ^1} the number of control volumes (CVs) in the 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. 2a, are calculated according to the transformation. The cell faces can be represented by surface vectors, i.e., \mathbf{A}^1 , \mathbf{A}^2 , and \mathbf{A}^3 . For convenience, $\Delta\xi^1 = \Delta\xi^2 = \Delta\xi^3 = 1$, so the CVs in the computational domain (ξ^1, ξ^2, ξ^3) have unit volume. In the implementation, ξ^1 , ξ^2 , and ξ^3 are simply the loop indices I , J , and K for the three coordinates. The values of variables in each CV are defined at its geometric center. Since some variables on boundaries are unknown, to make coding easier the CVs on the boundaries are also assigned, but their volume is zero. Grids highly stretched toward the boundaries can be obtained easily by simply increasing the stretch constant B .

TABLE I
Form of Governing Equations

$\frac{\partial}{\partial \xi^i} (C_i \phi + D_i) + J S_\phi = 0$			
ϕ	C_i	D_i	S_ϕ
1	U^i	0	0
u_1	U^i	$-\frac{\text{Pr}}{J} \left(B^{ij} \frac{\partial u_1}{\partial \xi^j} + A_j^i \omega_1^j \right)$	$-\frac{A_1^i}{J} \frac{\partial \wp}{\partial \xi^j}$
u_2	U^i	$-\frac{\text{Pr}}{J} \left(B^{ij} \frac{\partial u_2}{\partial \xi^j} + A_j^i \omega_2^j \right)$	$-\frac{A_2^i}{J} \frac{\partial \wp}{\partial \xi^j}$
u_3	U^i	$-\frac{\text{Pr}}{J} \left(B^{ij} \frac{\partial u_3}{\partial \xi^j} + A_j^i \omega_3^j \right)$	$-\frac{A_3^i}{J} \frac{\partial \wp}{\partial \xi^j}$
θ	U^i	$-\frac{1}{J} \left(B^{ij} \frac{\partial \theta}{\partial \xi^j} \right)$	0

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

3.2. Finite-Volume Integration

After the coordinate transformation, the governing equations can be rewritten in a general conservation-law form for each block in (ξ^1, ξ^2, ξ^3) as

$$\frac{\partial}{\partial \xi^i} (C_i \phi + D_i) + J S_\phi = 0, \quad (21)$$

where

$$J = \det|\partial x^i / \partial \xi^j|.$$

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. 2b). 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

$$\frac{\partial \wp}{\partial \xi^j} \equiv F_k \frac{\partial x^k}{\partial \xi^j} - \frac{\partial P}{\partial \xi^j}, \quad (22)$$

where F_k is the k th component of the body force \mathbf{F} ; $\mathbf{F} \equiv \text{Pr Ra } \theta \mathbf{e}_3$. The advantages of using this variable have been discussed in our previous report [8].

The finite-volume method simply integrates Eq. (21) over each CV in the computational domain (ξ^1, ξ^2, ξ^3) . In fact, the integration performed over the physical domain (x^1, x^2, x^3) is the same as that over the computational domain after coordinate transformation, and $\Delta V = J$. After the Gauss theorem is applied, the integration over each CV results in a flux balance equation,

$$I_e - I_w + I_n - I_s + I_t - I_b + \int_{\Delta V} S_\phi dV = 0, \quad (23)$$

where I_i , $i = (e, w, n, s, t, b)$, represents the fluxes of ϕ across the faces of the CV. Each of the fluxes I_i is made of two distinct parts, namely a convective contribution $I_i^C (= C_i \phi)$

and a diffusive contribution $I_i^D (=D_i)$. Both I_i^C and I_i^D are approximated with the central difference scheme. A detailed implementation of this boundary condition and related discussions can be found elsewhere [8]. It should be noticed that the FVM formulation for each block is the same. The interface between blocks is set up by equating the fluxes from both sides.

Furthermore, 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 the Rhie–Chow momentum interpolation scheme [21] 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. Furthermore, for incompressible flow, the pressure does not appear explicitly in the continuity equation and only the pressure gradient is meaningful. Therefore, the concept of the pressure correction scheme from the SIMPLE method [16] is adopted.

4. SOLUTION SCHEME

4.1. Strong Implicit Procedure

The approximation from the previous section leads to a general nonlinear form as

$$a_P^\phi \phi_P - \sum_{nb} a_{nb}^\phi \phi_{nb} = S_\phi \Delta V, \quad (24)$$

for variable ϕ , and $\phi = u_1, u_2, u_3, P$, and θ at every CV center for each block. A similar formulation can also be obtained for the block interfaces. The strong implicit procedure (SIP) based on the incomplete LU decomposition [22] is then used to solve Eq. (24). To ensure the diagonal dominance, the deferred correction method [23] is also used. With the deferred correction, the central difference scheme is allowed for the convection terms for all of the cases considered here.

The assembly of Eq. (24) for all variables and all blocks leads to a sparse block matrix. In each block for each variable, there are 9 diagonals for 2D problems, and 19 diagonals for 3D problems. In addition to SIP, other techniques, like the Gauss–Siedel (GS) method or preconditioned conjugate gradient (PCG) [24] methods, can also be used. However, for memory and performance, the SIP is adopted for each block, and the global iteration is performed variablewise and blockwise. The performance is found satisfactory for coarse grids, but becomes poor as the grid is refined. In fact, to most of the iterative methods, the convergence becomes slower as the number of equations increases. The convergence rate is fast only at the initial stage of the iterations, and it degrades rapidly as the iteration proceeds further. As mentioned before, this is because only the short wavelength errors are smoothed out efficiently. The long wavelength errors are hard to remove in the fine grid.

4.2. Multigrid Methods

An efficient way of overcoming the difficulties of slow convergence is to use different meshes for iterations, i.e., the MG methods. The concept of MG is simple, but its implementation is quite tedious. Careful design of the data structure is crucial to a successful implementation. If a wide spectrum of the wavelength for errors can be seen from

different grids, the convergence is expected to be faster. Therefore, to implement MG for each block, we further adopt several grid levels (multilevel) for the MG acceleration. For simplicity, the finest grid, which describes the shape of each block most accurately, is set up first. After setting up the finest grid, as the level is down for one level, eight contiguous control volumes (the kid volumes) are coalesced into a larger finite volume (the parent volume). From there, a simple parent–kid link-list data structure is then set up for computation. Furthermore, there are several approaches to performing MG iterations. Among them, the full approximation scheme (FAS) proposed by Brandt [25], which has been widely used [14, 15], is adopted here. Since the detailed description of FAS can be found elsewhere [25], only a brief description is presented here.

Let us take two grids (levels), the coarse grid (CG) and the fine grid (FG), as an example. During iteration, the FG formulation of Eq. (24) is rewritten in a matrix form,

$$A_f \phi_f = S_f, \quad (25)$$

where A_f is the coefficient matrix and S_f the source term. Since the solution of Eq. (25) is the most time consuming part, we only iterate it several times. After the iterations, say three nonlinear iterations, we can estimate the FG residual R_f by

$$R_f = S_f - A_f \phi_f. \quad (26)$$

This residual is then passed to CG to get the defect D_c . The corresponding CG residual \overline{R}_c and solution $\overline{\phi}_c$ are estimated first by linear interpolation (the so-called restriction step),

$$\overline{R}_c = I_f^c R_f, \quad (27)$$

$$\overline{\phi}_c = I_f^c \phi_f, \quad (28)$$

where I_f^c is the restriction operator for interpolating the values from FG to CG. Since the structure grids are used here, i.e., eight FG CVs are coalesced into a CG CV, the value on the CG is simply the volume-weighted average from the FG values. The defect on CG, D_c , is then estimated by

$$D_c \equiv \overline{A_c \phi_c} - \overline{S_c} + \overline{R}_c. \quad (29)$$

Now, the next step is to get a solution on CG (ϕ_c) with this defect by solving

$$A_c \phi_c = S_c + D_c. \quad (30)$$

Since the solution on the CG is much cheaper than that on the FG, enough nonlinear iterations (but less than 20) are performed on the CG. After the solution is obtained, the correction value $\Delta \phi_c$ is calculated by subtracting the old value interpolated from FG, $\overline{\phi}_c$, i.e.,

$$\Delta \phi_c = \phi_c - \overline{\phi}_c. \quad (31)$$

The FG correction $\Delta \phi_f$ is again obtained by interpolation,

$$\Delta \phi_f = I_c^f \Delta \phi_c, \quad (32)$$

where I_c^f is the prolongation operator for interpolating the values from CG to FG. Equations (31) and (32) are the so-called prolongation step. Linear interpolation is found to be adequate in this study. The solution on the FG is then updated by

$$\phi_f = \phi_f^{\text{old}} + \Delta\phi_f. \quad (33)$$

This new solution is used as a guess for Eq. (25) for further nonlinear iterations on FG. The MG iteration continues until the solution on the FG converges. Again, the MG acceleration is done block by block.

4.3. V-Cycle, FMG, and Interface Update

The MG procedure described above can be easily extended to more than two grids. Now, we must find the following: (1) a better MG procedure and (2) a better strategy for updating the interface. Two procedures for MG iterations are used here. With a known geometry, for each cycle, as illustrated in Fig. 3a, the procedure based on the previous section starts from the finest grid (top level) to the coarsest grid (lowest level) for restriction, and the prolongation is from the lowest level upward to the top level. Because of its shape, it is also called the V-cycle scheme. After the fixed-geometry MG V-cycle iteration is converged, the interface update based on the isotherm migration method (IMM) [8] is performed on the top level, as shown by the dashed line with an arrow in Fig. 3a. The IMM is simply to locate the melting point isotherm by interpolation. After the interface shape is obtained, the grid is generated

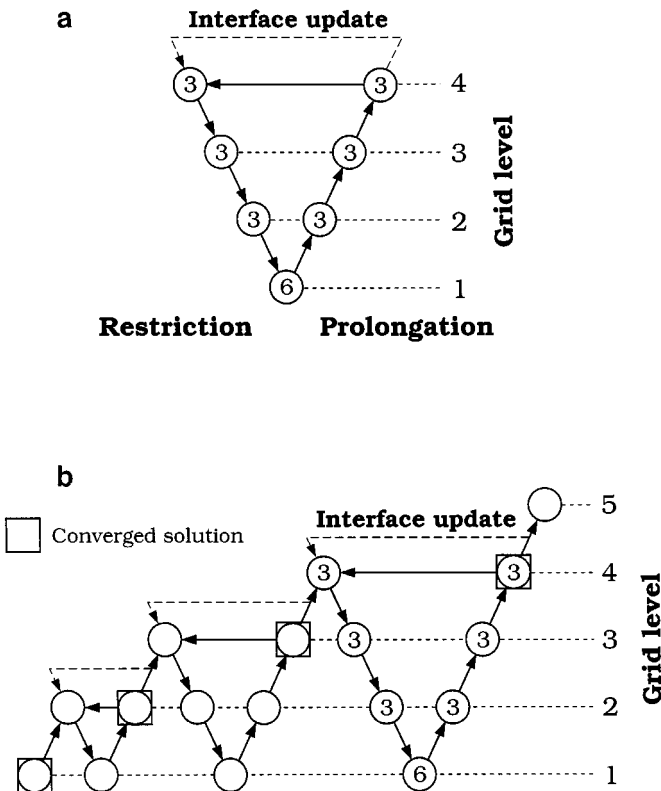


FIG. 3. Schemes for multigrid iteration and interface update: (a) V-cycle and (b) full multigrid (FMG).

using Eqs. (17)–(19) before entering the next fixed-grid V-cycle. The same procedure continues until the residual for the interface update is reduced to an acceptable tolerance.

Since the initial guess is often far away from the final solution, it may make more sense to obtain a rough solution from the coarsest grid first. This is the original concept of FMG, as illustrated in Fig. 3b. As shown, the solution starts from the coarsest grid. After the solution is obtained, one (or more than one) grid level is added. The addition of the grid level continues until the top level is reached. The interface update can be added onto the top level like that used in the V-cycle. However, a more consistent way is to add it between levels, as shown by the dashed line with an arrow in Fig. 3b. In fact, FMG with the interface update can be seen as a combination of V-cycles with different levels.

There is an additional advantage of FMG. Since the solution is converged for each level of grids, the grid-independent solution can be obtained easily from different levels through Richardson extrapolation. As a result, the estimation of discretization errors and the order of convergence is possible after two levels of grids have been used. In other words, local mesh refinement may be implemented if a refinement tree is set up. Furthermore, since the solutions for coarser grids are only the initial guess for the finest grid, the convergence criteria can be looser. Therefore, another modification can be made by tightening up the convergence tolerance as the grid level goes up. Such a strategy can be used for both field and interface variables.

It should be pointed out that the IMM scheme can also be used for single-grid approaches somewhat differently. The interface is usually updated more frequently during iterations. For example, one could update the interface every 50 nonlinear iterations of the field variables. Synchronized iterations can also be used by updating the interface right after each temperature iteration. However, such an interface update scheme is less stable because the isotherms are usually not smooth enough. Hence, it often requires underrelaxation. For example, one may use

$$h^{\text{new}} = h^{\text{old}} + \alpha \Delta h \quad (34)$$

or

$$h^{\text{new}} = h^{\text{old}} + \alpha(\theta - 0), \quad (35)$$

where α is the relaxation parameter, but this does not force the interface temperature to be the melting point. Using the Stefan condition (Eq. (11)) is also a good way for single-grid methods to update the interface through its residual R_h by

$$h^{\text{new}} = h^{\text{old}} + \alpha R_h. \quad (36)$$

However, the interface temperature has to be set at the melting point ($\theta = 0$). Also, the synchronized iteration scheme is necessary to keep the interface shape stable.

5. RESULTS AND DISCUSSION

Since the performance of IMM relies on the convergence of fixed-grid calculations, the MG performance on the fixed-grid cases, as well as the benchmark comparison, is presented first in the next section. The applications of MG/IMM schemes will be presented in Section 5.2, followed by an example on the crystal growth application, where both buoyancy and thermocapillary flows are important, in Section 5.3.

5.1. Performance on Fixed-Boundary Problems

The performance on a 2D heated square problem [13, 26] for various Ra numbers ($Pr = 0.71$) is examined first; $L/H = 1$. Although $Pr = 0.71$ is usually used for air in the benchmark solution, for most oxides Pr is about the same order. Therefore, for the convenience of illustration, $Pr = 0.71$ is used throughout this report even for the cases with phase change unless otherwise stated. The no-slip boundary condition is used for all boundaries. The top and bottom walls are insulated, while the hot wall ($\theta = 1$) is on the left and the cold wall ($\theta = 0$) is on the right. For both single-grid (SG) and MG, $u_1 = u_2 = P = \theta = 0$ is used as the initial guess. Five levels of meshes, from 10×10 to 160×160 , are used. For $Ra_T = 10^5$, less than 50 fine-grid iterations are used for both V-cycle MG and FMG to achieve a normalized residual norm of 10^{-7} . Using the finest grid alone, it will take more than 1000 iterations to reach a residual norm of only 10^{-2} . The calculated \overline{Nu} are in excellent agreement, up to four to five digits, with the benchmark results [13, 15, 27] for $Ra = 10^4$ to 10^6 . The cases with a 45° skewed parallelogram are also conducted, which are based on nonorthogonal grids and $Pr = 0.1$ or 10 . Excellent agreement is obtained as well. The CPU time required for MG methods is linearly proportional to the number of equations (NEQ); i.e., CPU time \propto NEQ n , $n \sim 1$. For the single-grid approach, $n \sim 1.8$, while $n \sim 1.6$ for Newton's method [8] using the ILU(0) preconditioned GMRES iterative matrix solver [28, 29], and the dimension of the Krylov subspace is 50. Due to the small convergence radius, for Newton's method the solution at $Ra = 10^4$ needs to be used as an initial guess for higher Ra . Furthermore, for all of the methods, the memory required is about linearly proportional to NEQ, but the memory required for MG is still much less than that for Newton's method. More importantly, the CPU time required for NEQ = 140,000 is less than 50 s for one calculation on an IBM/590 workstation, which is one to two orders of magnitude faster than that by the single-grid solution as well as Newton's method. The memory required is only 30 Mbytes.

The convergence for a 3D heated box problem ($L/H = 1$ and $W/H = 1$) using a similar initial guess is illustrated in Fig. 4. As shown, the convergence of MG still outperforms that

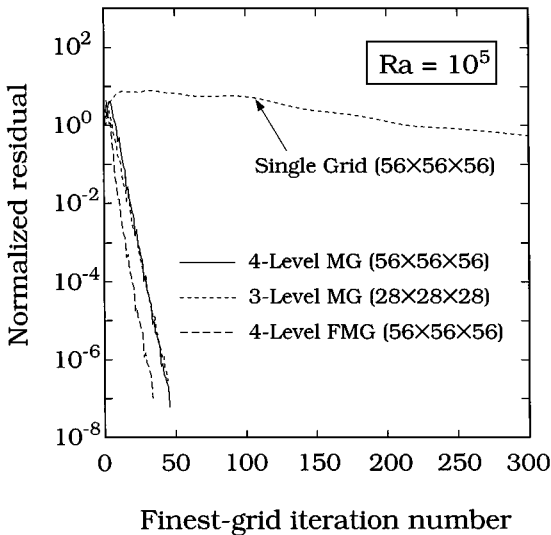


FIG. 4. Convergence of single-grid and multigrid methods for the 3D heated box problem at $Ra = 10^5$.

TABLE II

Grid	Rayleigh number			
	10^3	10^4	10^5	10^6
$7 \times 7 \times 7$	1.068	2.126	5.210	8.650
$14 \times 14 \times 14$	1.070	2.073	4.569	10.045
$28 \times 28 \times 28$	1.071	2.060	4.396	9.014
$56 \times 56 \times 56$	1.071	2.057	4.352	8.735
Grid-independent	1.071	2.056	4.337	8.642
Costa <i>et al.</i> [30] ($60 \times 60 \times 60$)	1.071	2.054	4.346	8.69
Fusegi <i>et al.</i> [31] ($64 \times 64 \times 64$)	1.085 ^a	2.100	4.361	8.77

^a This result of Fusegi *et al.* for $Ra = 10^3$ was obtained with a nonuniform grid of $32 \times 32 \times 32$.

of the single-grid method. Again, the convergence of MG is not affected by the problem size; the convergence rate for three and four levels is almost the same. Furthermore, the FMG seems to work better than the V-cycle MG. Clearly, for FMG, by the time the finest grid is added, its initial guess is good already even though the convergence rate remains the same. Furthermore, the benchmark comparison with previous calculations [30, 31] for \overline{Nu} is illustrated in Table II. Again, the agreement is very good, but our results are closer to those by Costa *et al.* [30].

The flow patterns and isotherms calculated for the two-block case for $Ra = 10^5$ are shown in Fig. 5. Due to the side wall insulation, the flow is quite similar to 2D flow; the flow in the third dimension is weak. In addition, since the interface is no longer at a fixed temperature, the flow is slightly different from that in a heated box. The comparison of CPU time required for different methods is summarized in Fig. 6, and again FMG appears to be the best method. Due to memory limitation, Newton's method is excluded from the comparison for 3D calculations. For $NEQ \approx 10^6$, the FMG takes about 10 min CPU time on an IBM workstation, and it is about two orders of magnitude faster than using the finest grid alone.

5.2. Performance on Free-Interface Problems

After the MG methods for the fixed-interface problems have been demonstrated, we are now ready to move on to the calculations including an unknown interface. When the melt/solid interface is included, the strategies on locating the interface are important. Figure 7a illustrates the IMM convergence of the interface shape ($Ra = 10^4$) by the V-cycle MG; the MG/IMM iteration scheme is shown in Fig. 3a. As shown, it requires more than 5 fixed-grid iterations to converge; the initial guess is the same as before and $h = 1$. However, if we monitor the maximum interface position at $x^2 = 1$ during iterations, as shown by the dashed line in Fig. 7b, 10 iterations are needed to converge. Nevertheless, the final interface position is quite far away from the initial guess, which also illustrates the robustness of the IMM scheme here. The convergence of $|\Delta h|_{\max}$ is further illustrated in Fig. 7c. Fourteen interface updates are needed to reduce the norm to 10^{-4} for the V-cycle MG (again, the dashed line). Nevertheless, the number of fine-grid iterations for each interface shape is also significantly reduced as convergence is approached. As a result, the overall CPU time required is much less than that for 14 fixed-grid calculations. If $Ra = 10^5$ is used, the interface position will move toward the cold wall even more. The CPU time required will be a bit more as well.

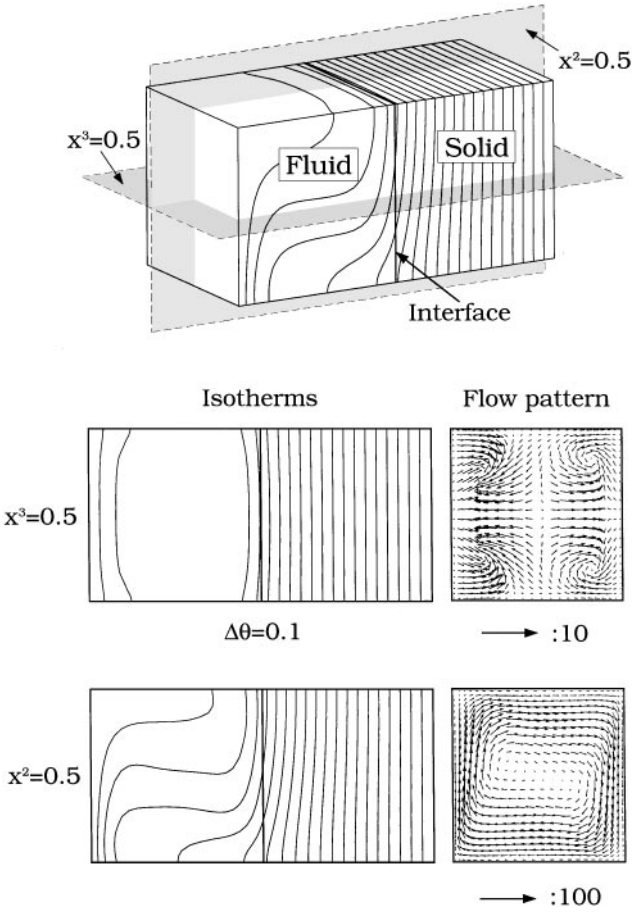


FIG. 5. Calculated temperature and velocity fields for the 3D two-block problem at $Ra = 10^5$.

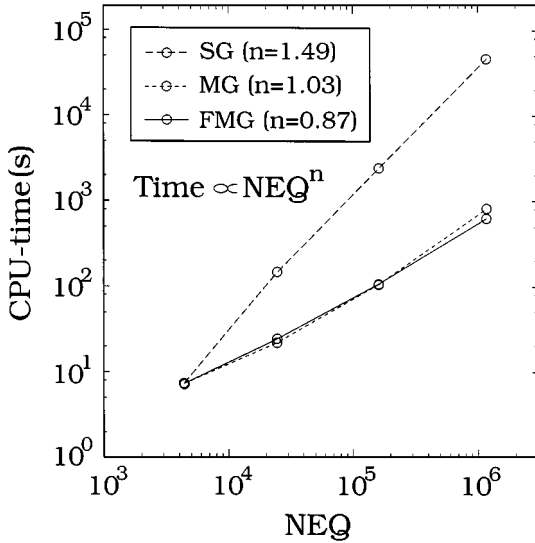


FIG. 6. CPU time required for different schemes for the problem in Fig. 5.

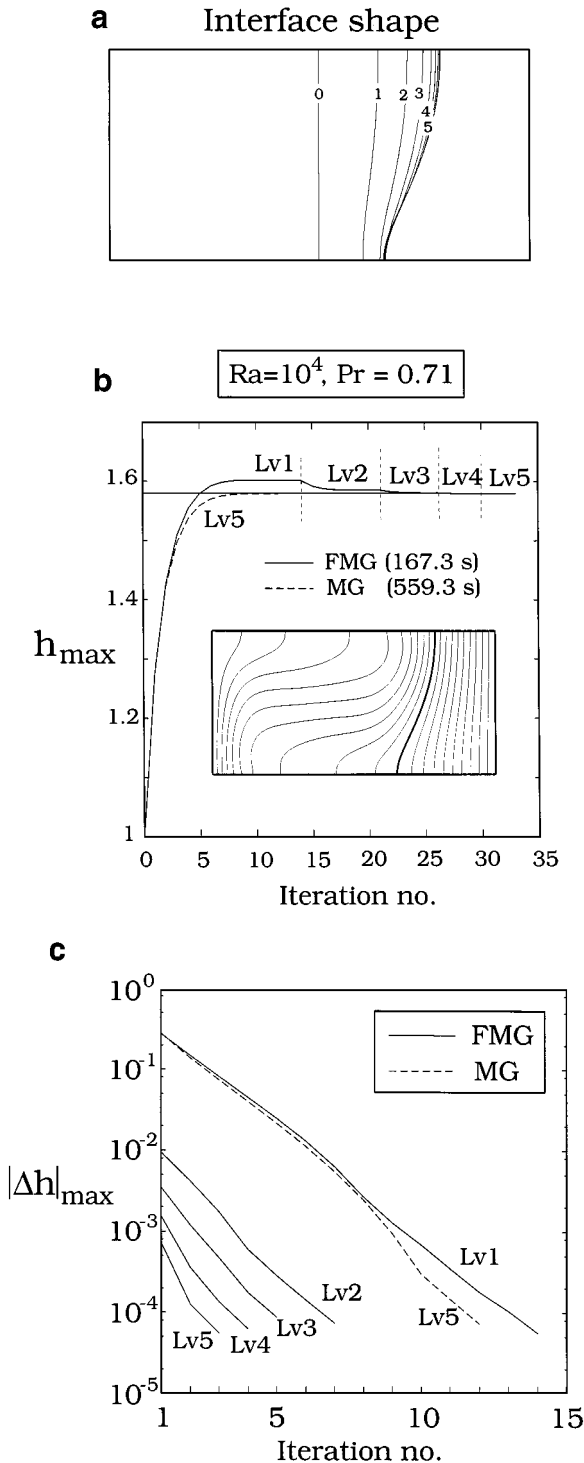


FIG. 7. Multigrid methods for the 2D free-interface problem ($Ra = 10^4$): (a) interface update using the V-cycle method (Fig. 3a); (b) convergence of the maximum interface position for V-cycle (dashed line) and FMG (the scheme is shown in Fig. 3b); and (c) convergence of interface corrections for V-cycle and each level of FMG. The converged isotherms are also shown in (b).

For FMG, the convergence of the maximum interface position is also shown in Fig. 7b (solid lines). Apparently, the convergence of the interface improves progressively as the grid level increases. Clearly, it takes much more interface updates, but mainly on the coarser meshes (lower levels). As the finest level is added, only two to four iterations are required to achieve the same accuracy as that by the V-cycle MG/IMM. As a result, the CPU time for FMG/IMM is only about one-third of that for the V-cycle MG/IMM. The difference can be further illustrated in Fig. 7c for the convergence history of the interface norm $|\Delta h|_{\max}$ for both methods. Interestingly, the convergence rate for the interface using IMM seems to be the same for all levels of meshes; it converges almost linearly. In other words, the major difference between the V-cycle MG/IMM and FMG/IMM is the initial guess of the interface shape for the finest grid that requires most of the CPU time. As shown in Fig. 7c, for FMG, by the time the finest level is added, the initial norm $|\Delta h|_{\max}$ is small already. As a result, for the problem with an unknown interface shape, FMG/IMM requires about three times more CPU time than that needed for a fixed-grid problem. Furthermore, the convergence of the FG iterations is progressively speeded up due to the better initial guess from previous solutions. Therefore, the overall performance leads to the CPU time being less than three times of that for a fixed-grid problem.

Although the FMG/IMM scheme seems to be satisfactory, further improvement is still possible. By examining Fig. 7c, it is interesting to note that $|\Delta h|_{\max}$ always has a jump from one grid level to the other due to interpolation. We also observe that the magnitude of the jump is not affected much by the convergence of the previous level if the convergence of the previous level is down to a certain degree (usually about a two-order decrease in the residual norm). Therefore, the convergence criteria for coarser grids for both field and interface variables can be loosened. By doing so, we can further reduce the CPU time by 15 to 30% for most cases.

Similar convergence is also obtained for the 3D heated box problem with an interface ($Ra = 10^4$). The results are summarized in Fig. 8. As shown in Fig. 8a, the V-cycle MG/IMM is still about three times slower than the FMG/IMM. Although the FMG/IMM still requires about four fixed-grid iterations, the total CPU time for about one million unknowns is less than 50 min on an IBM RS6000/590 workstation with a good quality of convergence. The convergence rates of the interface at different levels by FMG, again, are very similar to those of the 2D case, as shown in Fig. 8b. The converged result for temperature and velocity fields is shown in Fig. 8c.

For a further illustration on the robustness and the convergence speed, the result for $Ra = 10^5$ is placed in Fig. 9 for comparison; $h = 1$ and zero velocity and pressure are also used for the initial guess. As shown in Fig. 9a, it requires more interface updates for both MG and FMG. The convergence rate is also slightly slower than that in the previous case. However, due to the much stronger flow, the interface position is farther away from the initial position and the isotherms become much more distorted. Even for single-grid calculations, the convergence for higher Ra is slower as well. In other words, the speedup of the MG schemes remains satisfactory. More importantly, when using the single-grid approach with the finest grid, it is difficult to achieve the same convergence criteria. Furthermore, since the side walls for the previous cases are insulated, the interface deformation in the third dimension is not much. A more deformed interface will be illustrated in Section 5.3 for crystal growth applications.

Since different levels of grids are used, the order of convergence can be estimated easily. The grid-independent solution is again obtained by Richardson extrapolation from the

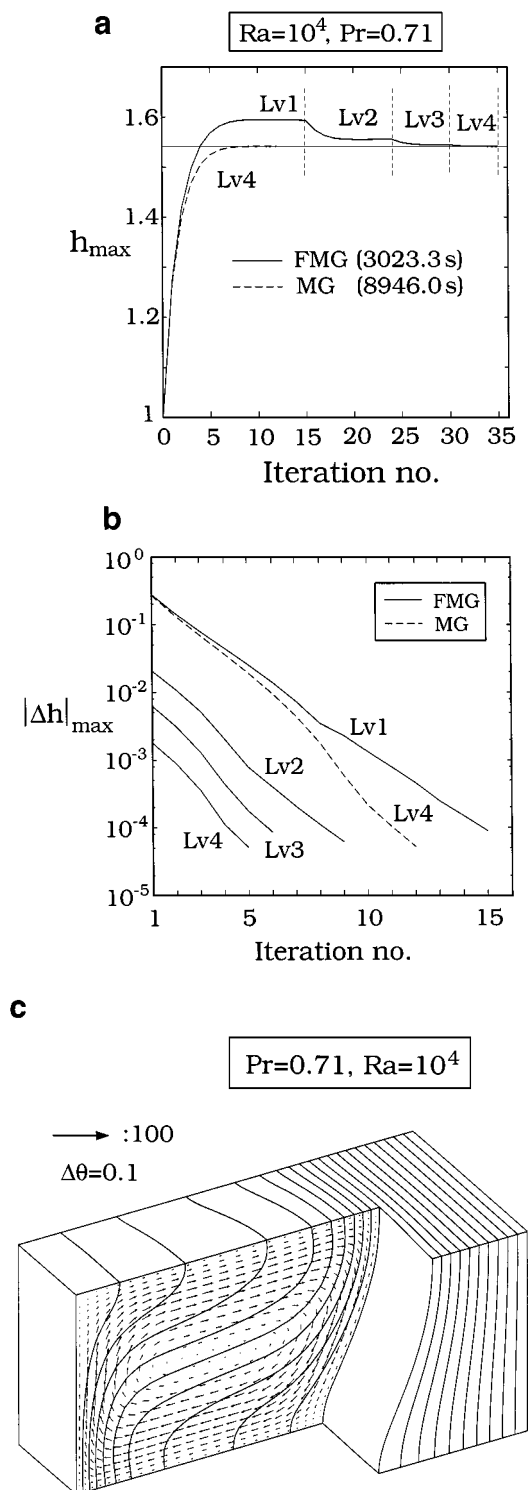


FIG. 8. Multigrid methods for the 3D free-interface problem ($Ra = 10^4$): (a) convergence of the maximum interface position for V-cycle (dashed line) and FMG (the scheme is shown in Fig. 3b); (b) convergence of interface corrections for V-cycle and each level of FMG; and (c) converged isotherms and velocity fields.

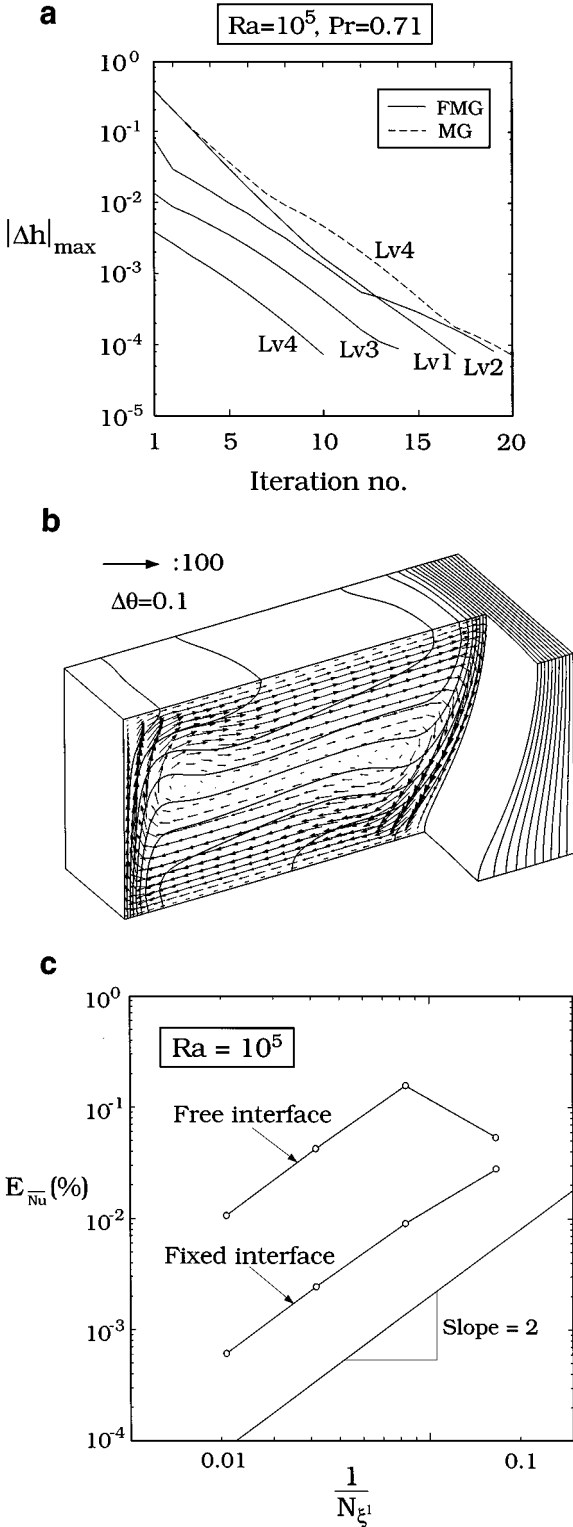


FIG. 9. Multigrid methods for the 3D free-interface problem ($Ra = 10^5$): (a) convergence of interface corrections for V-cycle and each level of FMG; (b) converged isotherms and velocity fields; and (c) estimated errors of \overline{Nu} as a function of grid size for the 3D fixed-grid and free-interface problems.

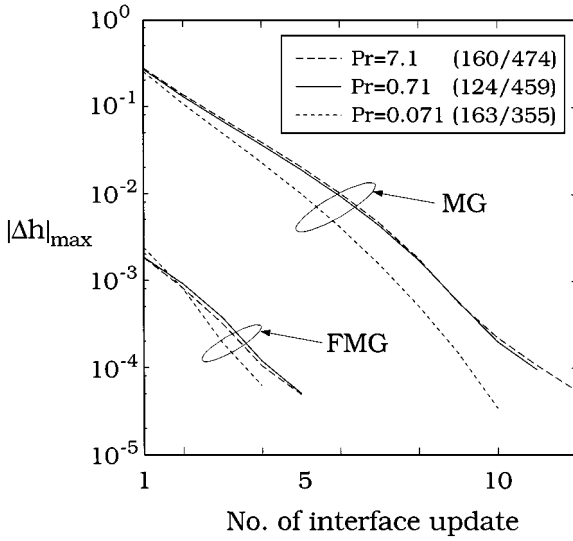


FIG. 10. Effects of Prandtl number (Pr) on the interface convergence of the V-cycle MG and FMG; only the finest grid results are shown. The numbers of finest grid iterations of FMG and V-cycle MG are indicated in parentheses in the legend.

solutions of the finest two grids. For the results in Figs. 5 and 9b, the error on \overline{Nu} , i.e., $E_{\overline{Nu}}$, as a function of grid spacing on the computational domain is plotted in Fig. 9c. As shown, as the grid is fine enough, second-order accuracy for both cases is obtained. Apparently, our FVM approximation is second-order accurate and it is the same as we did before for the 2D problems solved by Newton's method [8].

Since the heat transfer and fluid flow are strongly coupled in our study, the Prandtl number can also affect the convergence. The convective heat transfer becomes more important with the increasing Prandtl number; the interface deformation is increased as well. To illustrate this, we use $Ra = 10^4$, and the effects of different Prandtl numbers on the interface convergence and the total number of finest grid iterations for MG and FMG are shown in Fig. 10. As shown, for the interface update, the lower the Prandtl number, the faster the convergence speed. However, we are also aware that the convergence of the field variables, especially the velocity, slightly decreases as the Prandtl number increases. As a result, the total fine-grid iteration, and thus the CPU time, is not affected much by the Prandtl number.

5.3. Application to Crystal Growth

In previous calculations, we have demonstrated the convergence and robustness of the MG/IMM schemes, and the FMG/IMM has proven to be the most effective. The final example is to apply the FMG/IMM scheme to the HB crystal growth (Fig. 1c). As mentioned previously, the HB method is one of the major processes for growing GaAs single crystals for optoelectronic devices. The control of the growth interface has been of major interest to crystal growers because it directly controls the crystal quality. However, because the problem is 3D and the interface shape is unknown and strongly affected by the heat flow, both buoyancy and thermocapillary flows, its modeling is regarded as a great challenge in the crystal growth community. The first 3D coupled solution for the open boat configuration was conducted in our previous paper [19] using a SG/SIMPLE approach. The slow convergence

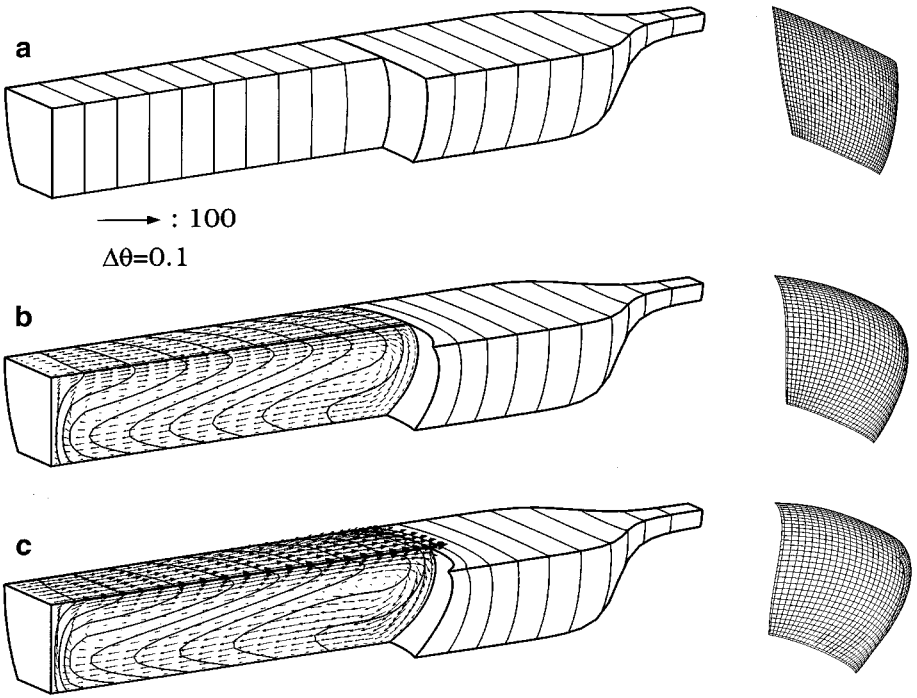


FIG. 11. Sample calculated results for the horizontal Bridgman crystal growth: (a) conduction mode ($Ra = Ma = 0$); (b) buoyancy mode ($Ra = 10^4$, $Ma = 0$); and (c) buoyancy and thermocapillary modes ($Ra = 10^4$, $Ma = 10^3$). The interface shapes are also shown on the right-hand side for comparison; for clarity of illustration, the results are based on the grid level 3 ($196 \times 34 \times 34$).

of the method for fine grids thus motivates the development of the present methods. To simplify the calculation, the crucible is neglected here; in practice, the crucible is very thin. The aspect ratios $L/H = 8$ and $W/H = 2$ are now much larger. Also, the thermocapillary convection is considered ($Ma = 10^3$). Crystal growth rate ($Pe = 10^{-2}$), the heat of fusion ($St = 20$), and the heat exchange with the environment ($Bi = 5$) are also included. A linear heating profile (in the x^1 direction) is imposed for the environment. Again, $Pr = 0.71$ is also used.

The calculated results are shown in Fig. 11. For the case without convection in Fig. 11a, the isotherms and the interface shape are not deformed much; the interface shape is also plotted on the right of the figure for comparison. As the buoyancy force ($Ra = 10^4$) is considered in Fig. 11b, due to the flow the isotherms become quite distorted. As shown, the interface deformation in the x^2 direction increases. As the Marangoni convection is taken into account ($Ma = 10^3$) in Fig. 11c, the velocity at the surface becomes much larger due to the thermocapillary force. As a result, the interface deformation increases. Since the driving force for the flow at the free surface is proportional to the thermal gradients there, the flow near the crucible is driven outward; the velocity vectors are more or less perpendicular to the isotherms at the interface. As a result, near the crucible, the interface shape is sharpened by the flow. In fact, such a small contact angle could induce parasitic nucleation and results in a polycrystalline growth. The convergence of this example is still very good, which will be illustrated shortly. Therefore, the schemes presented here seem to be promising for crystal growth applications as well as for other solidification problems.

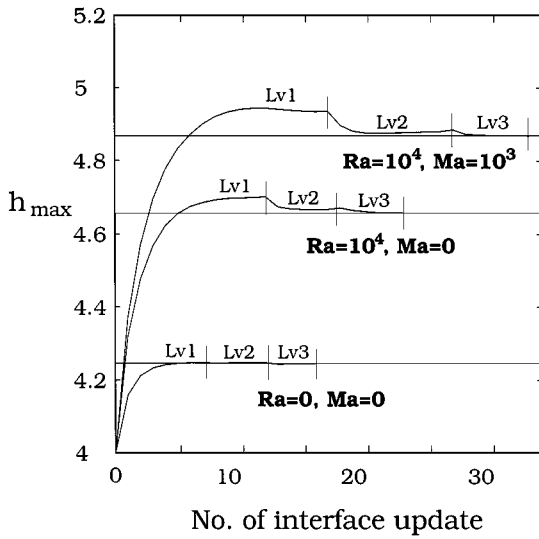


FIG. 12. Convergence of the maximum interface position for different convection modes using FMG for Fig. 11.

Furthermore, the performance of the interface update also depends on the convection modes. As shown in Fig. 12, the case with both flow modes takes the most interface updates for convergence because its solution is the farthest away from the initial guess. Even so, the number of interface updates for the top level ranges only from 5 to 10, which is still quite satisfactory. On the contrary, using the finest grid alone for Fig. 11c, the convergence is very slow. A comparison of their convergence is shown in Fig. 13, where we represent the convergence of the maximum interface position as a function of the number of fine-grid

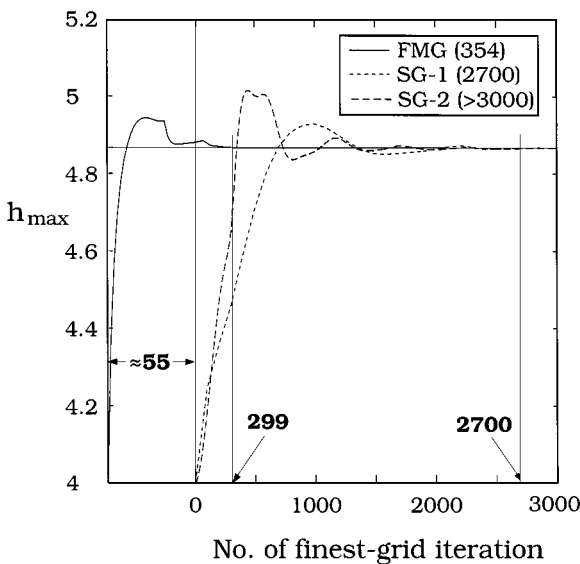


FIG. 13. Comparison of the interface convergence for FMG and single-grid methods for Fig. 11c. SG-2 updates the interface synchronically with the field variables using Eq. (35) with $\alpha = 0.1$.

iterations. As shown, FMG takes about 350 fine-grid iterations for convergence, while the single-grid methods take more than 3000 iterations. In fact, to make the comparison possible, the convergence criteria for the single-grid iterations are also loosened to about 0.01 for both field and interface variables. Two interface migration schemes are used for single-grid iterations. One updates the interface every 100 iterations of the field variables. The other updates the interface synchronically with the field variables, which requires a relaxation parameter of 0.1 to ensure the stability; see also Eq. (35). Through this comparison, it is clear that FMG outperforms the traditional single-grid approaches. The method is not only efficient, but also robust in crystal growth calculations.

6. CONCLUSIONS

Efficient multigrid methods are presented for solving 2D and 3D incompressible flow problems with an unknown melt/solid interface, mainly for solidification applications. The implementation concept is based on a multiblock and multilevel approach that allows complicated geometry of the problem to be treated. The present schemes are one to two orders of magnitudes faster than the global Newton's method using the GMRES iterative matrix solver and the single-grid SIMPLE method. The CPU time required for the present schemes is linearly proportional to the problem size. Both V-cycle and full multigrid methods are tested for calculating the interface shape using the isotherm migration method. It is found that the full multigrid method is the most effective, and its best performance requires about two to three times more CPU time than is needed for a similar fixed-grid problem. The 2D and 3D cases with strong buoyancy convection are successfully tested on a workstation with good quality of convergence. The performance of the scheme is not affected much by the Prandtl number. Although the convergence for stronger flows is slower, its speedup is not degraded much in our study as long as the coarsest grid remains stable during iterations. The application to a 3D crystal growth problem with buoyancy and thermocapillary flows is further illustrated, and the present schemes outperform the traditional schemes. In principle, the present schemes may be easily extended to time-dependent problems as well as to other free-boundary problems. Local refinement may also be another extension.

ACKNOWLEDGMENTS

The authors are grateful for the support from the National Science Council and the National Center for High Performance Computing of the Republic of China under Grant NSC87-2214-E002-035.

REFERENCES

1. J. Chang and R. A. Brown, Natural convection in steady solidification: Finite element analysis of a two-phase Rayleigh–Benard problem, *J. Comput. Phys.* **53**, 1 (1984).
2. C. J. Kim and M. Kaviany, Numerical method for phase change problems with convection and diffusion, *Int. J. Heat Mass Transfer* **35**, 457 (1992).
3. M. Lacroix, Predictions of natural-convection-dominated phase-change problems by the vorticity–velocity formulation of the Navier–Stokes equations, *Numer. Heat Transfer* **22**, 79 (1992).
4. R. Viswanath and Y. Jaluria, Computation of different solution methodologies for melting and solidification problems in enclosures, *Numer. Heat Transfer, Part B* **24**, 77 (1993).
5. S. K. Sinha, T. Sundararajan, and V. K. Garg, Study of the effects of macrosegregation and buoyancy-driven flow in binary mixture solidification, *Int. J. Heat Mass Transfer* **36**, 2349 (1993).

6. J. C. Chen, C. F. Chu, and W. F. Ueng, Thermocapillary convection and melt–solid interface in the floating zone, *Int. J. Heat Mass Transfer* **37**, 1733 (1994).
7. C. W. Lan, Newton’s method for solving heat transfer, fluid flow and interface shapes in a floating molten zone, *Int. J. Numer. Methods Fluids* **19**, 41 (1994).
8. M. C. Liang and C. W. Lan, A finite-volume/Newton method for a two-phase heat flow problem using primitive variables and collocated grids, *J. Comput. Phys.* **127**, 330 (1996).
9. A. Iserles, *A First Course in the Numerical Analysis of Differential Equations* (Cambridge Univ. Press, Cambridge, 1996), p. 227.
10. W. Hackbusch, *Multi-grid Methods and Applications* (Springer-Verlag, Berlin, 1985).
11. S. P. Vanka, Block-implicit multigrid solution of Navier–Stokes equations in primitive variables, *J. Comput. Phys.* **65**, 138 (1986).
12. W. Hackbusch and U. Trottenberg (Eds.), *Proceedings, Third European Multigrid Conference* (Birkhausen, Basel, 1991).
13. I. Demirdzic, Z. Lilek, and M. Peric, Fluid flow and heat transfer problem for non-orthogonal grids: Benchmark solution, *Int. J. Numer. Methods Fluids* **15**, 329 (1992).
14. Z. Lilek, S. Muzaferija, and M. Peric, Efficiency and accuracy aspects of a full-multigrid simple algorithm for three-dimensional flows, *Numer. Heat Transfer, Part B* **31**, 23 (1997).
15. J. Farmer, L. Martinelli, and A. Jameson, Fast multigrid method for solving incompressible hydrodynamic problems with free interfaces, *AIAA J.* **32**, 1175 (1994).
16. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, Washington, DC, 1980).
17. E. Monberg, in *Handbook of Crystal Growth 2a: Basic Techniques*, edited by D. T. J. Hurle (North-Holland, Amsterdam, 1994), p. 52.
18. M. C. Liang and C. W. Lan, Three-dimensional convection and solute segregation in vertical Bridgman crystal growth, *J. Cryst. Growth* **167**, 320 (1996).
19. M. C. Liang and C. W. Lan, Three-dimensional thermocapillary and buoyancy convections and interface shapes in horizontal Bridgman crystal growth, *J. Cryst. Growth* **180**, 587 (1997).
20. L. D. Landau and E. M. Lifshitz, *Fluid Mechanics* (Pergamon Press, Elmsford, NY, 1987), 2nd ed., Vol. 6, p. 217.
21. C. M. Rhie and W. L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, *AIAA J.* **21**, 1527 (1983).
22. H. L. Stone, Iterative solution of implicit approximations of multidimensional partial differential equations, *SIAM J. Numer. Anal.* **5**, 530 (1968).
23. P. K. Khosla and S. G. Rubin, A diagonally dominant second-order accurate implicit scheme, *Comput. Fluids* **2**, 207 (1974).
24. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics* (Springer-Verlag, Berlin, 1996).
25. A. Brandt, Multilevel adaptive computations in fluid dynamics, AIAA Paper 79, 1455 (1979).
26. G. De Vahl Davis, Natural convection of air in a square cavity: A bench mark numerical solution, *Int. J. Numer. Methods Fluids* **3**, 249 (1983).
27. M. Hortmann, M. Peric, and G. Scheuerer, Finite volume multigrid prediction of laminar natural convection: Bench-mark solutions, *Int. J. Numer. Methods Fluids* **11**, 189 (1990).
28. J. A. Meijerink and H. A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M -matrix, *Math. Comput.* **31**, 148 (1977).
29. Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
30. M. Costa, A. Oliva, and C. D. Perez-Segarra, Three-dimensional numerical study of melting inside an isothermal horizontal cylinder, *Numer. Heat Transfer, Part A* **32**, 531 (1997).
31. T. Fusegi, J. M. Hyun, K. Kuwaharas, and B. Farouk, A numerical study of three-dimensional natural convection in a differential heated cubical enclosure, *Int. J. Heat Mass Transfer* **34**, 1543 (1991).