Application of a Multigrid Method to a Buoyancy-Induced Flow Problem

C. P. Thompson† and G. K. Leaf
Mathematics and Computer Science Division
S. P. Vanka
Materials and Components Technology Division

Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois

INTRODUCTION

The numerical prediction of buoyancy-induced flows provides special difficulties for standard numerical techniques associated with velocity-buoyancy coupling. We present a multigrid algorithm based upon a novel relaxation scheme that handles this coupling correctly. Numerical experiments have been performed that show that this approach is reasonably efficient and robust for a range of Rayleigh numbers and a variety of cycling strategies.

1. OVERVIEW

The multigrid concept has emerged as one of the most promising for the solution of certain types of partial differential equations. There are extremely fast and robust codes available for single elliptic equations (see, for example [6]), and the techniques have been successfully applied to some systems of elliptic pdes. The philosophy of multigrid algorithms is (in a certain sense) to find an efficient smoother, i.e. a relaxation scheme which reduces high frequency errors, and organize a hierarchy of grids so that this rate of convergence applies to all error modes.

The aim of this paper is to present a multigrid algorithm and, in particular, a novel relaxation scheme that is effective for buoyant flows. This is an extension of the block-implicit scheme developed by Vanka [5].

Natural convection flows cause special numerical problems for iterative schemes (see, e.g., [7] and [9]). The new feature, not present in forced flows, is the coupling between momentum equations and the temperature equation through the buoyancy source term. Conventional (i.e., segregated) schemes that update the velocity fields independently of the temperature field suffer from a severe restriction: the effective time-step taken in this type of iterative procedure is limited by the buoyant

This work was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under contract W-31-109-Eng-38.

†Permanent adddress: Harwell Laboratory, Oxfordshire, England. On leave at Numerical Algorithms Group, Inc., Downers Grove, IL 60515.

time-scale (see [8] and [9]). This situation is true for both steady and transient problems and can be extremely restrictive because conduction time-scales are often between three and six orders of magnitude longer than the buoyant time-scale. Thus, in processes where buoyancy, convection, and conduction are present, special techniques are required for efficient solution.

In Section 2 we describe the "laminar double-glazing problem." This test case has been widely studied [10], and many solution algorithms have been applied to it. It has several interesting features. There is a high degree of nonlinearity in the problem, causing a significant degree of structure in the resulting flows. Narrow boundary layers are found for some parameter values. Unlike some other "benchmark" problems, this one is free from singularities and also has a simple geometric configuration. Also, accurate answers are available for this test case [12], and comparisons with the current results are presented.

In Sections 3 through 7 we discuss various features of the present multigrid method, starting with an overview and continuing with various details of our algorithm, concentrating on the relaxation (Section 5) and the treatment of the coarse grid (Section 7).

Finally, in Sections 8 and 9 we present the efficiency of the algorithm in terms of work units, compare the accuracy to other solutions, and discuss possible extensions of the technique.

Table 3 shows the average rate of convergence and the times per cycle for the various Rayleigh-number/grid-size combinations for a variety of cycles. Although the cycling strategies used are conservative, they appear to be reasonably efficient and robust. For linear problems, multigrid theory predicts that the convergence rate is independent of the mesh size. This is approximately true in the present case as well, a fact that is somewhat surprising since the Frechet derivative for this problem is large.

2. GOVERNING EQUATIONS AND FINITE DIFFERENCE APPROXIMATION

We consider the steady-state Navier-Stokes equations for the problem of natural convection in a two-dimensional square cavity subject to differential side heating. The cavity contains a viscous, heat-conducting fluid subject to conditions for which the Boussinesq approximations may be made. The equations will be given in non-dimensional form using the scales L^2/κ, κ/L, and $\rho_0\kappa^2/L^2$ for time, velocity, and pressure. Here L is a reference length, κ is the coefficient of thermal diffusivity, and ρ_0 is a reference density. The non-dimensional temperature is defined by $T = (T^*-T_c^*)/(T_h^*-T_c^*)$, where T^* is the local fluid temperature and T_h^*, T_c^* denote the temperature at the hot and cold boundaries, respectively. In the non-dimensional spatial units, the cavity is located in the unit square [0,1] $\times$ [0,1] in the xy plane. The hot boundary is at $x = 0$, the cold boundary is at $x = 1$, and the top and the bottom are adiabatic. The x and y components of the scaled velocities are denoted by u and v; p denotes the scaled difference of the total pressure from the hydrostatic pressure. The non-dimensional conservative equations for mass, momentum, and energy take the following form:

$$u_x + v_y = 0 \tag{2.1}$$

$$-(uu)_x - (vu)_y + Pr(u_{xx}+u_{yy}) - p_x = 0 \tag{2.2}$$

$$-(uv)_x - (vv)_y + Pr(v_{xx}+v_{yy}) - p_y + RaPrT = 0 \tag{2.3}$$

$$-(uT)_x - (vT)_y + T_{xx} + T_{yy} = 0 \tag{2.4}$$

where $Pr = \nu/\kappa$ denotes the Prandtl number and $Ra = g\beta L^3(T_h^*-T_c^*)/\nu\kappa$ denotes the Rayleigh number.

Here g is the gravitational acceleration, β the coefficient of volumetric expansion, and ν the kinematic viscosity.

These equations are discretized using a hybrid finite-differencing scheme [1], which employs second-order central differences on the convection and diffusion terms when the local cell Reynolds number is less than two. However, when the local cell Reynolds number is greater than two, the scheme modifies the convective differencing procedure to a donor cell (upwind) formulation and presumes that the diffusion flux at the cell interfaces is small by comparison to the convection flux and thus can be ignored. This scheme provides reasonable accuracy for sufficiently small mesh sizes while being stable (i.e., h-elliptic) on the coarse grids.

A standard staggered mesh is overlaid on the domain. The velocities are associated with the cell faces, and the pressures and temperatures are associated with the cell centers. The mesh is uniform with cell dimensions δx and δy. Note the border of fictitious cells and the placement of the tangential velocity components at the domain boundary. If we consider the (i,j)-th cell, then the pressure associated with the cell center is denoted by p_{ij}, the x component of the velocity associated with the center of the right-hand face is denoted by $u_{i+\frac{1}{2},j}$, and the y component of the velocity associated with the top face is denoted by $v_{i,j+\frac{1}{2}}$. The resulting finite-difference equations can be written in the following form:

$$A_c^u u_{i+\frac{1}{2},j} = A_n^u u_{i+\frac{1}{2},j+1} + A_s^u u_{i+\frac{1}{2},j-1} + A_e^u u_{i+3/2,j} + A_w^u u_{i-\frac{1}{2},j} + (p_{ij}-p_{i+1,j})/\delta x \tag{2.5}$$

$$A_c^v v_{i,j+\frac{1}{2}} = A_n^v v_{i,j+3/2} + A_s^v v_{i,j-\frac{1}{2}} + A_e^v v_{i+1,j+\frac{1}{2}} + A_w^v v_{i-1,j+\frac{1}{2}} + (p_{ij}-p_{i,j+1})/\delta y + \frac{1}{2} RaPr(T_{ij} + T_{i,j+1}) \tag{2.6}$$

$$(u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j})/\delta x + (v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}})/\delta y = 0 \tag{2.7}$$

$$A_c^T T_{ij} = A_n^T T_{i,j+1} + A_s^T T_{i,j-1} + A_e^T T_{i+1,j} + A_w^T T_{i-1,j} \tag{2.8}$$

The coefficients are defined as follows. For $\phi = u$, v, and T, we have

$$A_c^\phi = A_n^\phi + A_s^\phi + A_e^\phi + A_w^\phi \tag{2.9}$$

$$A_w^\phi = \max(|C_{x^-}^\phi|, D_{x^-}^\phi) + C_{x^-}^\phi \tag{2.10}$$

$$A_e^\phi = \max(|C_{x^+}^\phi|, D_{x^+}^\phi) - C_{x^+}^\phi \tag{2.11}$$

$$A_s^\phi = \max(|C_{y^-}^\phi|, D_{y^-}^\phi) + C_{y^-}^\phi \tag{2.12}$$

$$A_n^\phi = \max(|C_{y^+}^\phi|, D_{y^+}^\phi) - C_{y^+}^\phi , \tag{2.13}$$

where the differential form of the coefficients is used to give the correct scalings across the grids. Thus we have

$$C^u_{x^-} = (u_{i+\frac{1}{2},j}+u_{i-\frac{1}{2},j})/4\delta x;\ D^u_{x^+} = Pr/\delta x^2 \tag{2.14}$$

$$C^u_{x^+} = (u_{i+\frac{1}{2},j}+u_{i+3/2,j})/4\delta x;\ D^u_{x^+} = Pr/\delta x^2 \tag{2.15}$$

$$C^u_{y^-} = (v_{i,j-\frac{1}{2}}+v_{i+1,j-\frac{1}{2}})/4\delta y;\ D^u_{y^-} = Pr/\delta y^2 \tag{2.16}$$

$$C^u_{y^+} = (v_{i,j+\frac{1}{2}}+v_{i+1,j+\frac{1}{2}})/4\delta y;\ D^u_{y^+} = Pr/\delta y^2 \tag{2.17}$$

$$C^v_{x^-} = (u_{i-\frac{1}{2},j}+u_{i-\frac{1}{2},j+1})/4\delta x;\ D^v_{x^-} = Pr/\delta x^2 \tag{2.18}$$

$$C^v_{x^+} = (u_{i+\frac{1}{2},j}+u_{i+\frac{1}{2},j+1})/4\delta x;\ D^v_{x^+} = Pr/\delta x^2 \tag{2.19}$$

$$C^v_{y^-} = (v_{i,j+\frac{1}{2}}+v_{i,j-\frac{1}{2}})/4\delta y;\ D^v_{y^-} = Pr/\delta y^2 \tag{2.20}$$

$$C^v_{y^+} = (v_{i,j+\frac{1}{2}}+v_{i,j+3/2}\delta y;\ D^v_{y^+} = Pr/\delta y^2 \tag{2.21}$$

$$C^T_{x^-} = u_{i-\frac{1}{2},j}/2\delta x;\ D^T_{x^-} = 1/\delta x^2 \tag{2.22}$$

$$C^T_{x^+} = u_{i+\frac{1}{2},j}/2\delta x;\ D^T_{x^+} = 1/\delta x^2 \tag{2.23}$$

$$C^T_{y^-} = v_{i,j-\frac{1}{2}}/2\delta y;\ D^T_{y^-} = 1/\delta y^2 \tag{2.24}$$

$$C^T_{y^+} = v_{i,j+\frac{1}{2}}/2\delta y;\ D^T_{y^+} = 1/\delta y^2\ . \tag{2.25}$$

The cavity is assumed to be solid, and no slip conditions are assumed to prevail; thus the normal and tangential components of the velocity are set to zero at the boundary. The temperature at the left-hand wall has the value one, while the temperature at the right-hand wall has the value zero. The adiabatic walls imply that the normal derivative of the temperature is zero at these walls. Note that the tangential velocities in the border cells are associated with the walls, and the temperatures in the border cells by the hot and cold walls are also associated with these walls, as indicated in Figure 1. A minor modification to the diffusion terms in the energy equation allows the temperature boundary conditions to be modelled without loss of accuracy.

3. BASIC MULTIGRID TECHNIQUES

The finite difference equations derived in the preceding section are solved by a multigrid technique. For a complete review of multigrid techniques with applications to fluid dynamics, articles by Brandt [2] and Brandt and Dinar [4] may be consulted. An introduction to the subject may be found in a review by Stuben and Trottenberg [3].

The basic multigrid technique used in this application is the FAS (Full Approximation Storage) method which is fully discussed in [2], [3], and [4]. The basic approach can be described as follows. We define a series of uniform grids with spacing $h_k = h_{k-1}/2$ for k=1,2,...,M. In addition, we have a set of grid transfer operators I^{k-1}_k, $\hat{I}^{k-1}_k$, and I^k_{k-1}, where the first two operators map grid functions defined on grid k to functions defined on grid (k–1) (restriction) and the last operator transfers functions defined on grid k–1 to functions defined on grid k (interpolation). Starting on the finest grid $k = M$ and setting $f^k = F^k$, we wish to solve

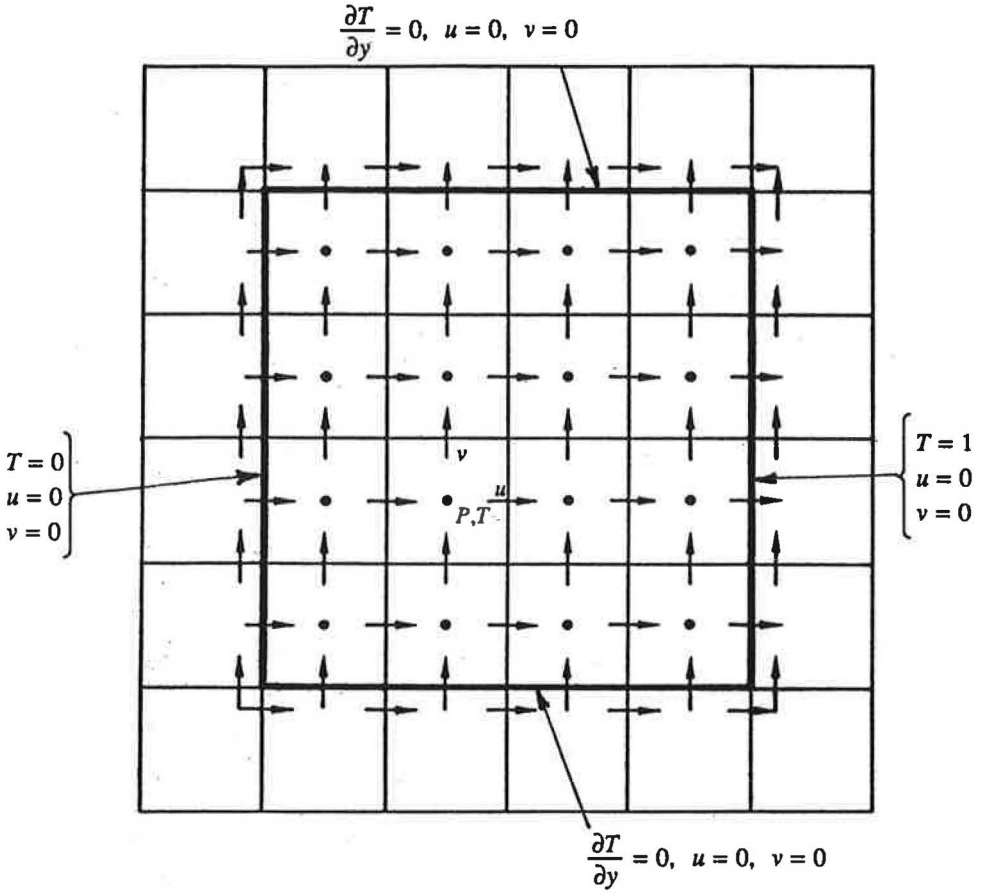


FIGURE 1. Staggered grid: arrangement of variables and boundary conditions

$$L^k(w^k) = f^k . \tag{3.1}$$

Relaxation iterations are performed generating a grid function $\tilde{w}^k$. A transfer is then made to the next coarser grid, k–1, where the following problem is posed:

$$L^{k-1}(\hat{w}^{k-1}) = f^{k-1} \equiv I_k^{k-1}(f^k - L^k(\tilde{w}^k)) + L^{k-1}(I_k^{k-1}\tilde{w}^k) . \tag{3.2}$$

Relaxation iterations are used to generate a grid function $\tilde{w}^{k-1}$. At this point a decision is made whether to restrict to the next coarser grid and repeat the above process or to interpolate back to the k-th grid and generate a new approximation to w^k with the expression

$$\tilde{w}^k_{new} = \tilde{w}^k + I^k_{k-1}(\tilde{w}^{k-1} - I_k^{k-1}\tilde{w}^k) . \tag{3.3}$$

An FAS method is not completely specified until one defines a strategy concerning when and in what direction to transfer from one grid to another. A strategy based on smoothing rates and convergence is usually referred to as adaptive FAS, while a strategy based on a fixed cyclic pattern of grid transfers and a fixed number of relaxation iterations on each grid is usually referred to as cyclic FAS, with a prefix specifying the type of cycle, e.g., V, W, F [2]. Our study is concerned with adaptive and fixed-cycle FAS methods.

4. MULTIGRID STRATEGIES

In this section we describe the adaptive strategy and the W-cycle strategy used in this study. Both strategies are described in Brandt [2] and have been used in many different investigations.

4.1. Adaptive FAS

The particular implementation of the adaptive FAS algorithm used in this study is essentially the same as that described by Brandt [2]. The process is initiated on the coarsest grid (grid number 1) where the solution of the complete nonlinear problem is sought. At this level, Newton-type iterations are used, and the resulting linear equations are solved by a direct method. The converged solution on this grid is prolongated to the next finer grid, where relaxation sweeps are performed. Since the problem is nonlinear, the coefficients are evaluated after each sweep. If the smoothing rate, as measured by the ratio of successive norms of the current residual, falls below a given threshold value η, a decision is made to transfer to the coarser mesh k–1. The residuals are transferred to grid (k–1), and one solves for grid functions w^{k-1} which are approximations to $I_k^{k-1}w^k$; the problem on grid k–1 is

$$L^{k-1}w^{k-1} = f^{k-1} = I_k^{k-1}(f^k - L^k\tilde{w}^k) + L^{k-1}I_k^{k-1}\tilde{w}^k . \tag{4.1}$$

If the grid function $\tilde{w}^{k-1}$ generated at this level is satisfactory, the correction to w^k at the k-th level is then

$$w^k_{new} = \tilde{w}^k + I^k_{k-1}(\tilde{w}^{k-1} - I_k^{k-1}\tilde{w}^k) . \tag{4.2}$$

Note that it is the correction $\tilde{w}^{k-1} - I_k^{k-1}\tilde{w}^k$ that is transferred, not the grid function $\tilde{w}^{k-1}$. Also, the relaxation sweeps for Equation (4.1) are started from the initial grid function $I_k^{k-1}\tilde{w}^k$.

At any stage there is a current finest level l; and when the convergence tolerance is met on this level, the grid function is prolongated to a finer level. Thus an adaptive FAS process is nested with many visits to the coarser grids. When the finest level ($k = M$) is solved to the desired accuracy, the overall solution cycle is terminated. Note that the tolerance level on any grid is equal to the originally prescribed value only when that grid is the current finest level l. However, when the current

level k is less than l, the tolerance ϵ_k is set to

$$\epsilon_k = \delta e_{k+1} , \tag{4.3}$$

where e_{k+1} is a norm of the residual on grid k+1, and typically $\delta = 0.2$.

When restricting from grid k to k–1, the coefficients $A_c^\phi, A_e^\phi, \ldots$ as defined in Equations (2.9)-(2.13) are initially generated from the restricted grid function $I_k^{k-1}\vec{w}^k$. For succeeding iterations, the coefficients are generated from the latest values $\vec{w}^{k-1}$.

4.2. W-Cycles

The second type of multigrid algorithms used in this study is based on the use of W-cycles (cf. [2] and [3]). Cyclic algorithms are based on a strategy of cycling through the grids in a specified pattern while performing a given number of smoothing iterations on each visit to each grid. The visitation pattern of W-cycles can be described recursively as follows. For $M = 2$, we start with relaxation iterations on the first grid ($M = 2$), restrict to the coarser grid M–1 = 1, perform a direct solution or relaxation solution, and then prolongate to the finest grid ($M = 2$) for further relaxations. Let this cycle be denoted by $W(2)$. In general, if M is given and $W(M-1)$ is defined, we generate $W(M)$ as follows. Starting on grid M, we perform relaxation iterations and then restrict to grid M–1. Next we perform two $W(M-1)$ cycles in succession, prolongate to grid M, and finish with relaxation iterations.

To complete the description of a $W(M)$ cycle, we need to specify the number of relaxation iterations performed on each grid. In this study, we specified a $W(M)$-cycle by three parameters ν_c,ν_p,ν_r, where ν_c specifies the number of Newton iterations performed on the coarsest grid ($k = 1$); for $k > 1$, ν_r specifies the number of relaxation iterations performed on a grid k when that grid is reached by a restriction from the grid (k+1); and for $k > 1$, ν_p specifies the number of relaxation iterations performed on a grid k when that grid is reached by a prolongation from grid (k–1). At local peaks in the $W(M)$-cycle (a local peak occurs when a grid is reached by prolongation and is followed by a restriction), the number of relaxation iterations is taken to be $\bar{\nu} = \max(\nu_r,\nu_p)$. To generate the initial grid function defined on the finest grid, we use a simple starting procedure consisting of performing a specified number ν_c^0 of Newton iterations on the coarsest grid ($k = 1$) and successively prolongating to the next finest grid and performing ν_p relaxation iterations, repeating this process until the finest grid is reached, at which point the W-cycle starts. The number of iterations on the finest grid is taken to be $\bar{\nu} = \nu_r + \nu_p$ where ν_p iterations come from the previous W-cycle and ν_r iterations arise from the current cycle. To avoid excessive iterations, we make the following test. On any intermediate grid $k < M$, the relaxation iterations are terminated after one additional iteration when

$$e_k < \delta\epsilon_M , \tag{4.4}$$

where e_k is a norm of the current residual, $\delta = 0.001$, and ϵ_M is the error tolerance on the finest grid. The W-cycles are repeated until convergence is achieved on the finest grid.

5. RELAXATION TECHNIQUES

The choice of an efficient relaxation (smoothing) operator is of primary importance for the success of the multigrid technique. The choice of a relaxation procedure is somewhat problem dependent, and there is a tradeoff between a robust technique with a larger operation count and a less robust but simpler technique with a lower operation count. Of course, the primary objective in the design of a

relaxation procedure is to achieve the best possible smoothing rate.

In this study, the relaxation technique is a modification of the procedure introduced by Vanka [5]. The temperature, momentum, and continuity equations are relaxed in a coupled manner. In this scheme the temperature, four velocities, and one pressure associated with one finite-difference cell are simultaneously updated by solving a 5×5 set of equations with special structure. Thus the velocities on all four sides of a cell are updated together. This type of procedure is referred to as a symmetrical coupled Gauss-Seidel (SCGS) procedure. The details of the procedure when applied to the natural convection problem are as follows.

For any given grid level k, consider a staggered mesh centered at cell (i,j), which we take to be an interior cell. We are given a set of grid functions $\tilde{T}_{ij}$, $\tilde{U}_{i+\frac{1}{2},j}$, $\tilde{V}_{i,j+\frac{1}{2}}$, $\tilde{P}_{ij}$, and a set of right-hand side grid functions S^T_{ij}, $S^u_{i+\frac{1}{2},j}$, $S^v_{i,j+\frac{1}{2}}$, S^c_{ij} which are generated from residual and variable transfers as indicated by the right-hand side of Equation (4.1). The variables $\tilde{T}_{ij}$, $\tilde{U}_{i+\frac{1}{2},j}$,... are used to generate the finite difference coefficients (A^{ϕ}_c),(A^{ϕ}_e),... as specified in Equations (2.9)-(2.13). With these coefficients defined over the entire mesh, our task is to solve Equation (4.1). We write this equation in block form as follows. We order the mesh cells lexicographically, and with each mesh cell (i,j) (cf. Figure 4) we group the following set of six variables as a unit to determine the block structure:

$$(T_{ij},U_{i-\frac{1}{2},j},U_{i+\frac{1}{2},j},V_{i,j-\frac{1}{2}},V_{i,j+\frac{1}{2}},P_{ij}) \ . \tag{5.1}$$

With this blocking and ordering of the mesh cells, Equation (4.1) has the following form:

$$AX = S \ , \tag{5.2}$$

where

$$A = D-L-U \ , \tag{5.3}$$

D is the block diagonal matrix found from the grouping of the six variables in the (i,j)-th cell, and L,U are block lower, upper triangular matrices relative to this ordering. The particular form of relaxation used in this study is motivated by the following considerations. Standard block Gauss-Seidel relaxation applied to Equation (5.2) would take the following form:

$$D\hat{X} = LX^{(1)} + UX^{(0)} + S \tag{5.4a}$$

$$X^{(1)} = w\hat{X} + (1-w)X^{(0)} \ , \tag{5.4b}$$

where $X^{(0)}$ is some initial estimate, as is a given parameter, and $X^{(1)}$ is the new estimate generated by the procedure. This relaxation procedure can be written in the following form by setting

$$R = S + LX^{(1)} + UX^{(0)} - DX^{(0)} = S - (DX^{(0)}-LX^{(1)}-UX^{(0)}) \tag{5.5}$$

and then observing that

$$D(\hat{X}-X^{(0)}) = R \ . \tag{5.6}$$

Combining this result with 5.4b, we find

$$\frac{1}{w} D(X^{(1)}-X^{(0)}) = R \ . \tag{5.7}$$

Equation (5.7) is the basis for the relaxation procedure used in this study. We have modified this procedure by using the factor $1/w$ only on the diagonal elements of D rather than on all the elements of D.

As indicated in Equation (5.7), we will solve for the corrections; thus, for example, we write

$$T_{ij}^{(1)} = T_{ij}^{(0)} + T'_{ij} \tag{5.8}$$

with similar expressions for the other variables. In Equation (5.5), set

$$Q = DX^{(0)} - LX^{(1)} - UX^{(0)} . \tag{5.9}$$

We defined the following quantities associated wtih the (i,j)-th cell:

$$Q_{ij}^{T} = (A_c^T)_{ij}T_{ij}^{(0)} - (A_e^T)_{ij}T_{i+1\,j}^{(0)} - (A_n^T)_{ij}T_{i\,j+1}^{(0)} - (A_w^T)_{ij}T_{i-1\,j}^{(1)} - (A_s^T)T_{i\,j-1}^{(1)} \tag{5.10}$$

$$Q_{i+\frac{1}{2}j}^{u} = (A_c^u)_{i+\frac{1}{2}j}U_{i+\frac{1}{2}j}^{(0)} - (A_e^u)_{i+\frac{1}{2}j}U_{i+\frac{3}{2}j}^{(0)} - (A_n^u)_{i+\frac{1}{2}j}U_{i+\frac{1}{2}j+1}^{(0)} \tag{5.11}$$

$$- (A_w^u)_{i+\frac{1}{2}j}U_{i-\frac{1}{2}j}^{(1)} - (A_s^u)_{i+\frac{1}{2}j}U_{i+\frac{1}{2}j-1}^{(1)} - \frac{1}{\delta x}\,P_{i+1\,j}^{(0)}$$

$$Q_{i-\frac{1}{2}j}^{u} = (A_c^u)_{i-\frac{1}{2}j}U_{i-\frac{1}{2}j}^{(0)} - (A_n^u)_{i-\frac{1}{2}j}U_{i-\frac{1}{2}j+1}^{(0)} - (A_e^u)_{i-\frac{1}{2}j}U_{i+\frac{1}{2}j}^{(0)} \tag{5.12}$$

$$- (A_w^u)_{i-\frac{1}{2}j}U_{i-\frac{3}{2}j}^{(1)} - (A_s^u)_{i-\frac{1}{2}j}U_{i-\frac{1}{2}j-1}^{(1)} + \frac{1}{\delta x}\,P_{i-1\,j}^{(1)}$$

$$Q_{i\,j+\frac{1}{2}}^{v} = (A_c^v)_{i\,j+\frac{1}{2}}V_{i\,j+\frac{1}{2}}^{(0)} - (A_e^v)_{i\,j+\frac{1}{2}}V_{i+1\,j+\frac{1}{2}}^{(0)} - (A_n^v)_{i\,j+\frac{1}{2}}V_{i\,j+\frac{3}{2}}^{(0)} \tag{5.13}$$

$$- (A_s^{v)_{i\,j+\frac{1}{2}}}V_{i\,j-\frac{1}{2}}^{(1)} - (A_w^v)_{i\,j+\frac{1}{2}}V_{i-1\,j+\frac{1}{2}}^{(1)} - \frac{1}{\delta y}\,P_{i\,j+1}^{(0)} + \frac{1}{2}\,RaPrT_{i\,j+1}^{(0)}$$

$$Q_{i\,j-\frac{1}{2}}^{v} = (A_c^v)_{i\,j-\frac{1}{2}}V_{i\,j-\frac{1}{2}}^{(0)} - (A_e^v)_{i\,j-\frac{1}{2}}V_{i+1\,j-\frac{1}{2}}^{(0)} - (A_w^v)_{i\,j-\frac{1}{2}}V_{i-1\,j-\frac{1}{2}}^{(1)} \tag{5.14}$$

$$- (A_n^v)_{i\,j-\frac{1}{2}}V_{i\,j+\frac{1}{2}} - (A_s^v)_{i\,j-\frac{1}{2}}V_{i\,j-3/2}^{(1)}$$

$$Q_{ij}^{c} = 0 . \tag{5.15}$$

Then form R by setting

$$R_{ij}^{T} = S_{ij}^{T} - Q_{ij}^{T} \tag{5.16a}$$

$$R_{i\pm\frac{1}{2}j}^{u} = S_{i\pm\frac{1}{2}j}^{u} - Q_{i\pm\frac{1}{2}j}^{u} \tag{5.16b}$$

$$R_{i\,j\pm\frac{1}{2}}^{v} = S_{i\,j\pm\frac{1}{2}}^{v} - Q_{i\,j\pm\frac{1}{2}}^{v} \tag{5.16c}$$

$$R_{ij}^{c} = S_{ij}^{c} . \tag{5.16d}$$

At this juncture we have calculated the right-hand side for Equation (5.7). As mentioned earlier, we have modified the relaxation procedure described by Equation (5.7). Actually, we have modified the procedure in two ways:

(i) The factor $1/w$ will not multiply the entire 6×6 block matrix, but instead just the diagonal entries.

(ii) Since the problem is nonlinear, a local approximation is made to the Jacobian to bring in the effect of velocity on the temperature. The effect is to modify the block diagonal matrix D in Equation (5.7).

Thus Equation (5.4) is replaced by the nonlinear Gauss-Seidel relaxation scheme:

$$D(\hat{X})\hat{X} = L(X^{(1)})X^{(1)} + U(X^{(0)})X^{(0)} + S \tag{5.17a}$$

$$X^{(1)} = w\hat{X} + (1-w)X^{(0)} . \tag{5.17b}$$

Consider the (i,j)-th cell, and let Φ denote the group of six variables defined in Equation (5.1) which are associated with that cell. Then for this cell we wish to solve the vector equation (of dimension 6)

$$D(\hat{\Phi})\hat{\Phi} = F , \tag{5.18}$$

where F is the right-hand side of (5.17a) restricted to the (i,j)-th cell, and $D(\hat{\Phi})$ is the 6×6 matrix associated with this cell. So for this cell, the task is to solve the vector equation

$$G(\hat{\Phi}) = F - D(\hat{\Phi})\hat{\Phi} = 0 . \tag{5.19}$$

Using Newton's method with $\Phi^{(0)}$ as the initial estimate would lead to the linear system

$$H\Phi' = G(\Phi^{(0)}) = F - D(\Phi^{(0)})\Phi^{(0)} = R . \tag{5.20}$$

Note that R is that portion of the vector R appearing in Equation (5.7) which is associated with the (i,j)-th cell. Here H is the negative of the Jacobian of $G(\Phi)$ given by

$$H_{pq} = D_{pq}(\Phi^{(0)}) + \sum_{s=1}^{6} \Phi_s^{(0)} \frac{\partial}{\partial \Phi_q} D_{ps}(\Phi^{(0)}) . \tag{5.21}$$

The simplest approximation to H_{pq} is $D_{pq}(\Phi^{(0)})$ (frozen coefficient approximation). In this study, we have incorporated some of the terms from the summation appearing in (5.21).

With the variables associated with the (i,j)-th cell grouped as indicated in (5.1), the 6×6 matrix D_{pq} which is formed from the frozen finite-difference coefficients is defined by the following expression:

$$D(\Phi^{(0)}) = \begin{bmatrix} (A_c^T)_{ij} & 0 & 0 & 0 & 0 & 0 \\ 0 & (A_c^u)_{i-\frac{1}{2},j} & 0 & 0 & 0 & 1/\delta x \\ 0 & 0 & (A_c^u)_{i+\frac{1}{2},j} & 0 & 0 & -1/\delta x \\ \frac{1}{2}RaPr & 0 & 0 & (A_c^v)_{i,j-\frac{1}{2}} & 0 & 1/\delta y \\ \frac{1}{2}RaPr & 0 & 0 & 0 & (A_c^v)_{i,j+\frac{1}{2}} & -1/\delta y \\ 0 & -1/\delta x & 1/\delta x & -1/\delta y & 1/\delta y & 0 \end{bmatrix} \tag{5.22}$$

Here

$$\Phi^{(0)} = (T_{ij}^{(0)}, U_{i-\frac{1}{2},j}^{(0)}, U_{i+\frac{1}{2},j}^{(0)}, V_{i,j-\frac{1}{2}}^{(0)}, V_{i,j+\frac{1}{2}}^{(0)}, P_{ij}^{(0)})^T , \tag{5.23}$$

and the coefficients $A_c^T, A_c^u, \ldots$ are evaluated using $\Phi^{(0)}$.

To form the H_{pq} used in this study, we have used only the additional terms in Equation (5.21) that give the velocity contribution in the temperature equation. Thus we set $p = 1$ and observe that

$$\frac{\partial}{\partial \Phi_s} D_{1s}(\Phi^{(0)}) = 0, \quad s=2,3,\ldots,6 . \tag{5.24}$$

Hence

$$H_{1q} = D_{1q} + \Phi_1^{(0)} \frac{\partial}{\partial \Phi_q} D_{11}(\Phi^{(0)}), \quad q=1,2,\ldots,6 . \tag{5.25}$$

Recall that $\Phi_1^{(0)} = T_{ij}^{(0)}$, $D_{11}(\Phi^{(0)}) = (A_c^T)_{ij}$, and from Equation (2.9)

$$A_c^T = A_e^T + A_w^T + A_s^T + A_n^T . \tag{5.26}$$

Here it is understood that we are dealing with the (i,j)-th cell, so this subscript is omitted for simplicity. The coefficients A_e^T, A_w^T, A_s^T, A_n^T are defined by Equations (2.10)-(2.13) and Equations (2.22)-(2.25). Consider, for example,

$$\frac{\partial}{\partial \Phi_2^{(0)}} D_{11}(\Phi^{(0)}) = \frac{\partial}{\partial U_{i-\frac{1}{2},j}} (A_c^T)_{ij} = \frac{\partial}{\partial U_{i-\frac{1}{2},j}} (A_w^T)_{ij} , \tag{5.27}$$

since only A_w^T depends on $U_{i-\frac{1}{2},j}$. From the definition of A_w^T we have

$$A_w^T = \max(|C_{x^-}^T|, D_{x^-}^T) + C_{x^-}^T \tag{5.28}$$

with

$$C_{x^-}^T = \frac{U_{i-\frac{1}{2},j}}{2\delta x}, \; D_{x^-}^T = \frac{1}{\delta x^2} . \tag{5.29}$$

Define

$$\sigma_{x^-} = \max(0, \operatorname{sign}(U_{i-\frac{1}{2},j})) . \tag{5.30}$$

Then

$$H_{12} = T_{ij}^{(0)} \frac{\partial}{\partial U_{i-\frac{1}{2},j}} A_w^T = \begin{cases} T_{ij}^{(0)} \dfrac{\sigma_{x^-}}{\delta x} & \text{if } |U_{i-\frac{1}{2},j}|\delta x \geq 2 \\ \dfrac{T_{ij}^{(0)}}{2\delta x} & \text{if } |U_{i-\frac{1}{2},j}|\delta x < 2 \end{cases} \tag{5.31}$$

In a similar manner, we define

$$\sigma_{x^+} = \max(0, \operatorname{sign}(-U_{i+\frac{1}{2},j})) , \tag{5.32a}$$

$$\sigma_{y^-} = \max(0, \operatorname{sign}(V_{i,j-\frac{1}{2}})) , \tag{5.32b}$$

$$\sigma_{y^+} = \max(0, \operatorname{sign}(V_{i,j+\frac{1}{2}})) . \tag{5.32c}$$

Then, we find

$$H_{13} = T_{ij}^{(0)} \frac{\partial}{\partial U_{i+\frac{1}{2},j}} (A_e^T) = \begin{cases} \dfrac{1}{\delta x} \sigma_{x^+} & \text{if } |U_{i+\frac{1}{2},j}|\delta x > 2 \\ \dfrac{-1}{2\delta x} & \text{else} \end{cases} \tag{5.33}$$

$$H_{14} = T_{ij}^{(0)} \frac{\partial}{\partial V_{i,j-\frac{1}{2}}} A_s^T = \begin{cases} \dfrac{1}{\delta y} \sigma_{y^-} & \text{if } |V_{i,j-\frac{1}{2}}|\delta y > 2 \\ \dfrac{1}{2\delta y} & \text{else} \end{cases} \tag{5.34}$$

$$H_{15} = T_{ij}^{(0)} \frac{\partial}{\partial V_{i,j+\frac{1}{2}}} A_n^T = \begin{cases} \dfrac{1}{\delta y} \sigma_{y^+} & \text{if } |V_{i,j+\frac{1}{2}}|\delta y > 2 \\ \dfrac{-1}{2\delta y} & \text{else} \end{cases} \tag{5.35}$$

$$H_{16} = 0 \tag{5.36}$$

If we use the factor $1/w$ only on the diagonals and use the elements H_{1q} as defined above, the 6×6 matrix approximation to the matrix H which appears in Equation (5.20) has the following form:

$$\tilde{H} = \begin{bmatrix} \frac{1}{w}(A_c^T)_{ij} & H_{12} & H_{13} & H_{14} & H_{15} & 0 \\ 0 & \frac{1}{w}(A_c^u)_{i-\frac{1}{2}j} & -(A_e^u)_{i-\frac{1}{2}j} & 0 & 0 & \frac{1}{\delta x} \\ 0 & -(A_w^u)_{i+\frac{1}{2}j} & \frac{1}{w}(A_c^u)_{i+\frac{1}{2}j} & 0 & 0 & \frac{-1}{\delta x} \\ \frac{1}{2}RaPr & 0 & 0 & \frac{1}{w}(A_c^v)_{ij-\frac{1}{2}} & -(A_n^v)_{ij-\frac{1}{2}} & \frac{1}{\delta y} \\ \frac{1}{2}RaPr & 0 & 0 & -(A_s^v)_{ij+\frac{1}{2}} & \frac{1}{w}(A_c^v)_{ij+\frac{1}{2}} & \frac{-1}{\delta y} \\ 0 & \frac{-1}{\delta x} & \frac{1}{\delta x} & \frac{-1}{\delta y} & \frac{1}{\delta y} & 0 \end{bmatrix} \tag{5.37}$$

For simplicity, the four elements $(A_e^u)_{i-\frac{1}{2}j}$, $(A_w^u)_{i+\frac{1}{2}j}$, $(A_n^v)_{ij-\frac{1}{2}}$, and $(A_s^v)_{ij+\frac{1}{2}}$ are neglected. Recall that the limiting coupling for this problem is the temperature-velocity, and omitting these terms does not affect this. Effectively we are decomposing $\tilde{H} = H_0 + \hat{H}$, where $\hat{H}$ contains the neglected terms, and then writing (5.20) in the form $H_0\Phi' = \hat{H}\Phi' + R$ and performing our iteration with the initial connection $\Phi' = 0$. In each mass control volume the task is to solve

$$H_0\Phi' = R\ . \tag{5.38}$$

The cells are swept over in a lexicographic ordering, which means that the interior velocity components are updated twice. The increased rate of convergence compensates for this extra arithmetic, and somewhat greater robustness is achieved. Set

$$\xi^T = (H_{12}, H_{13}, H_{14}, H_{15}, 0) \tag{5.39a}$$

$$\zeta^T = (0, 0, \tfrac{1}{2}RaPr, \tfrac{1}{2}RaPr, 0) \tag{5.39b}$$

$$\alpha = \frac{1}{w}(A_c^T)_{ij} \tag{5.39c}$$

$$\tau = \Phi'_1 \tag{5.39d}$$

$$\Psi^T = (\Phi'_2, \Phi'_3, \Phi'_4, \Phi'_5, \Phi'_6) \tag{5.39e}$$

$$f = R_1 \tag{5.39f}$$

$$E^T = (R_2, R_3, R_4, R_5, R_6) \tag{5.39g}$$

$$B = \begin{bmatrix} \frac{1}{w}(A_c^u)_{i-\frac{1}{2},j} & 0 & 0 & 0 & \frac{+1}{\delta x} \\ 0 & \frac{1}{w}(A_c^u)_{i+\frac{1}{2},j} & 0 & 0 & \frac{-1}{\delta x} \\ 0 & 0 & \frac{1}{w}(A_c^v)_{i,j-\frac{1}{2}} & 0 & \frac{+1}{\delta y} \\ 0 & 0 & 0 & \frac{1}{w}(A_c^v)_{i,j+\frac{1}{2}} & \frac{-1}{\delta y} \\ \frac{-1}{\delta x} & \frac{+1}{\delta x} & \frac{-1}{\delta y} & \frac{+1}{\delta y} & 0 \end{bmatrix} \tag{5.40}$$

Then Equation (5.38) has the bordered matrix form

$$\begin{bmatrix} \alpha & \xi^T \\ \zeta & B \end{bmatrix} \begin{bmatrix} \tau \\ \underline{\Psi} \end{bmatrix} = \begin{bmatrix} f \\ \underline{F} \end{bmatrix} . \tag{5.41}$$

This equation implies

$$\begin{bmatrix} \tau \\ \underline{\Psi} \end{bmatrix} = \begin{bmatrix} \beta & q^T \\ \underline{r} & P \end{bmatrix} \begin{bmatrix} f \\ \underline{F} \end{bmatrix} , \tag{5.42}$$

where the elements of the inverse are given by

$$\beta^{-1} = \alpha - \xi^T B^{-1} \zeta \tag{5.43a}$$

$$\underline{r} = -\beta B^{-1} \zeta \tag{5.43b}$$

$$q^T = -\beta \xi^T B^{-1} \tag{5.43c}$$

$$P = B^{-1} + \beta B^{-1} \zeta \xi^T B^{-1} . \tag{5.43d}$$

Thus, if we define $\underline{z}$ and $\underline{y}$ by

$$B\underline{z} = \underline{F} \tag{5.44a}$$

$$B\underline{y} = \zeta , \tag{5.44b}$$

the system (5.41) is solved by

$$\beta = (\alpha - \xi^T \underline{y})^{-1} , \tag{5.45a}$$

$$\tau = \beta(f - \xi^T \underline{z}) , \tag{5.45b}$$

$$\underline{\Psi} = \underline{z} - \tau \underline{y} , \tag{5.45c}$$

and therefore the correction vector for the (i,j)-th cell is

$$\Phi' = \begin{bmatrix} \tau \\ \underline{\Psi} \end{bmatrix} . \tag{5.46}$$

In this study, we scaled the vector ξ^T in Equation (5.39a) with a specified parameter, and the parameter w was always used as an under-relaxation factor.

As noted earlier, the block Gauss-Seidel type scheme described is an adaptation of the SCGS scheme introduced by Vanka [5]. A further discussion of that scheme may be found in that reference. It should be noted that SCGS-type schemes differ substantially from Brandt's Distributive

Gauss-Seidel (DGS) scheme. The SCGS-type schemes employ a simultaneous update of all the variables; whereas DGS is a decoupled procedure which would correct all the momentum equations first and then go back to recorrect the velocities, pressure, and temperature to satisfy the remaining equations.

6. RESTRICTION AND PROLONGATION

Recall that restriction procedures are used for transferring fine grid values to a coarse grid; thus the operation of transferring from grid k to grid $k-1$ is denoted by I_k^{k-1}. In this study, the variables are restricted in the same manner as the residual; hence $\hat{I}_k^{k-1} = I_k^{k-1}$ in Equations (3.3) and (3.4). The prolongation operator I_{k-1}^k is used to transfer variables from a coarse grid to a fine grid and generally involves an interpolation procedure.

In this study, the restriction operators are defined by the simplest average of nearby values. Let (ic,jc) and (if,jf) denote coarse and fine grid indices corresponding to grid $k-1$ and grid k, respectively. In this context, let $u_{i+\frac{1}{2},j}$ be referred to as $u_{i,j}$ and $u_{i-\frac{1}{2},j}$ as $u_{i-1,1}$ with similar notation for v. Then $if = 2(ic)-1$, $jf = 2(jc)-1$, and

$$u^c(ic,jc) = (\frac{1}{2})[u^f(if,jf) + u^f(if,jf-1)] , \tag{6.1a}$$

$$v^c(ic,jc) = (\frac{1}{2})[v^f(if,jf) + v^f(if-1,jf)] , \tag{6.1b}$$

$$p^c(ic,jc) = (\frac{1}{4})[p^f(if,jf) + p^f(if-1,jf) + p^f(if,jf-1) + p^f(if-1,jf-1)] , \tag{6.1c}$$

and

$$T^c(ic,jc) = (\frac{1}{4})[T^f(if,jf) + T^f(if-1,jf) + T^f(if,jf-1) + T^f(if-1,jf-1)] . \tag{6.1d}$$

Temperature and continuity equation residuals are associated with mesh centers, so that boundary values do not occur for these residuals. Momentum residuals are zero on the boundary since we have no slip and solid walls. Thus there is no fine-to-coarse transfer of residuals associated with the boundary. The restriction of boundary values of each variable requires separate consideration. The velocities are specified on the boundary, so no fine-to-coarse transfer is needed for boundary velocities. The pressures are associated with cell centers, and no boundary conditions are imposed; thus no fine-to-coarse boundary transfer is needed for the pressure. On the walls when the temperature is specified, no transfer is needed, and on the adiabatic walls the temperature in the border cells is set equal to the temperature in the adjacent cell. Note that in this study none of the restriction operators is of the full weighting type (cf. [2]).

A crucial consEquence of Equation (6.1c), which is also used to restrict the continuity residuals, is that the continuity residual for a coarse-grid mass control volume is proportional to the sum of the fine-grid continuity residuals contained in the coarse-grid control volume. Thus satisfaction of the discrete analogue of (2.1) on the fine grid forces satisfaction of this condition on all grids. This is necessary for a solution of the coarse grid equations to exist. (Recall that the operators are singular.)

The coarse-to-fine (prolongation) operators I_{k-1}^k are based on bilinear interpolation. Thus the coarse-to-fine transfer of u-velocity values is defined as follows:

$$u^f(if,jf) = (\frac{1}{4})(3u_1^c + u_2^c) \tag{6.2a}$$

$$u^f(if,jf+1) = (\frac{1}{4})(u_1^c + 3u_2^c) \tag{6.2b}$$

$$u^f(if+1,jf) = (\frac{1}{8})(3u_1^c + u_2^c + 3u_3^c + u_4^c) \tag{6.2c}$$

$$u^f(if+1,jf+1) = (\frac{1}{8})(u_1^c + 3u_2^c + u_3^c + 3u_4^c)\ , \tag{6.2d}$$

where

$$u_1^c = u^c(ic,jc);\ \ u_2^c = u^c(ic,jc+1); \tag{6.3a}$$

$$u_3^c = u^c(ic+1,jc);\ \ u_4^c = u^c(ic+1,jc+1)\ . \tag{6.3b}$$

At the top and bottom mesh cells, the above formulas are modified to reflect the fact that we associate the velocities in the border mesh cells with the wall values. The fine-grid v-velocities are defined by analogous expressions. Since the pressure and the temperature values are associated with the mesh cell centers, the interpolation formulas have a different weighting from the velocities. At interior mesh cells, the coarse-to-fine temperature transfers are given as follows:

$$T^f(if,jf) = (\frac{1}{16})(9T_1^c + 3T_2^c + 3T_3^c + T_4^c) \tag{6.4a}$$

$$T^f(if,jf+1) = (\frac{1}{16})(3T_1^c + T_2^c + 9T_3^c + 3T_4^c) \tag{6.4b}$$

$$T^f(if+1,jf) = (\frac{1}{16})(3T_1^c + 9T_2^c + T_3^c + 3T_4^c) \tag{6.4c}$$

$$T^f(if+1,jf+1) = (\frac{1}{16})(T_1^c + 3T_2^c + 3T_3^c + 9T_4^c)\ , \tag{6.4d}$$

where

$$T_1^c = T^c(ic,jc);\ \ T_2^c = T^c(ic+1,jc); \tag{6.5a}$$

$$T_3^c = T^c(ic,jc+1);\ \ T_4^c = T^c(ic+1,jc+1)\ . \tag{6.5b}$$

At the top and bottom walls a zero-derivative condition holds; thus, for example, bilinear interpolation for the top mesh cells would give

$$T^f(if,jf) = (\frac{1}{4})(3T_1^c + T_2^c)\ , \tag{6.6a}$$

$$T^f(if+1,jf) = (\frac{1}{4})(T_1^c + 3T_2^c)\ , \tag{6.6b}$$

where

$$T_1^c = T^c(ic,jc);\ \ T_2^c = (ic+1,jc)\ , \tag{6.7}$$

with corresponding expressions for the bottom mesh cells. At the left and right walls, the temperature is specified; thus, for example, bilinear interpolation for the left mesh cells would give

$$T^f(if,jf) = (\frac{1}{8})(3T_1^c + 3T_2^c + T_3^c + T_4^c) \tag{6.8a}$$

$$T^f(if,jf+1) = (\frac{1}{8})(T_1^c + T_2^c + 3T_3^c + 3T_4^c)\,, \tag{6.8b}$$

where

$$T_1^c = T^c(ic,jc) = \text{boundary value};\ \ T_2^c = T^c(ic+1,jc); \tag{6.9a}$$

$$T_3^c = T^c(ic,jc+1) = \text{boundary value};\ \ T_4^c = T^c(ic+1,jc+1) \tag{6.9b}$$

with the corresponding expressions for the rightmost mesh cells. The interpolations in the corner mesh cells are handled in the obvious way. The pressure is treated in a manner similar to the temperature except that a zero-derivative condition is assumed to hold on all boundaries. This approximation is used only in the implementation of the prolongation operator; since the continuity equation is satisfied in the relaxation phase, pressure boundary conditions are unnecessary.

Notice that the prolongation operator I_{k-1}^k which appears in Equation (3.4) is acting on what is essentially a correction to the solution on the $(k-1)$-st grid. This is the operator we have been discussing in this section. In the present study we have used the same prolongation operator for the corrections as for the solutions.

It should be emphasized that in an FAS-type algorithm, the values on the coarse grid are not directly prolongated (see Equation (3.4)); rather, the changes from previously restricted values are prolongated. That is,

$$\tilde{w}_{new}^k = \tilde{w}_{old}^k + I_{k-1}^k \delta w^{k-1} \tag{6.10a}$$

$$\delta w^{k-1} = \tilde{w}^{k-1} - I_k^{k-1} w_{old}^k\,. \tag{6.10b}$$

The operators defined in Equations (6.1)-(6.9) are applied to $\delta\tilde{w}^{k-1}$.

7. COARSEST GRID SOLUTION

The relaxation sweeps described in Section 5 are used on every grid level except the coarsest grid $k = 1$. On this grid, where the dimension of the system represented in Equation (4.1) is small, we solve this system using a Newton-Raphson type procedure combined with a direct solution of the resulting linear system. The matrix for the linear system is generated from the Jacobian of the operator appearing in Equation (4.1) on the coarsest grid. (As usual, the pressure non-uniqueness is eliminated by defining a reference pressure at one point. This does not affect the rate of convergence of the multigrid cycle since it is applied only on the coarsest grid.)

We will illustrate the nature of the Jacobian in the case of the temperature equation. Recall from Section 2 that the finite difference equation for the temperature has the following form when considered at the (i,j) mesh cell:

$$F_c^T = A_w^T(T_c - T_w) + A_e^T(T_c - T_e) + A_s^T(T_c - T_s) + A_n^T(T_c - T_n) = R_c\,, \tag{7.1}$$

where

$$T_c = T(i,j);\ \ T_w = T(i-1,j);\ \ T_e = T(i+1,j);\tag{7.2}$$

$$T_s = T(i,j-1);\ \ T_n = T(i,j+1),$$

and

$$A_w^T = (A_w^T)_{ij}\ ,\ \text{etc.}$$

Setting

$$U_c = U(i+\tfrac{1}{2},j);\ \ U_w = U(i-\tfrac{1}{2},j);\ \ U_e = U(i+\tfrac{3}{2},j);\tag{7.3}$$

$$U_s = U(i+\tfrac{1}{2},j-1);\ \ U_n = U(i+\tfrac{1}{2},j+1)$$

with similar notation for the v-velocities, we can write the defining relations for the coefficients as follows:

$$A_w^T = A_w^T(U_w) = \max(1/\delta x^2, |U_w|/2\delta x) + U_w/2\delta x\ ,\tag{7.4a}$$

$$A_e^T = A_e^T(U_c) = \max(1/\delta x^2, |U_c|/2\delta x) - U_c/2\delta x\ ,\tag{7.4b}$$

$$A_s^T = A_s^T(V_s) = \max(1/\delta y^2, |V_s|/2\delta y) + V_s/2\delta y\ ,\tag{7.4c}$$

$$A_n^T = A_n^T(V_c) = \max(1/\delta y^2, |V_c|/2\delta y) - V_s/2\delta y\ .\tag{7.4d}$$

Thus we have the following dependencies:

$$F_c^T = F^T(T_c, T_w, T_e, T_s, T_n, U_w, U_c, V_s, V_c)\ .\tag{7.5}$$

The Jacobian is generated by using the partial derivatives of F^T with respect to each of these variables. In the case of the temperature dependence, we have, for example,

$$\frac{\partial F^T}{\partial T_c} = A_w^T + A_e^T + A_s^T + A_n^T = A_c^T\ .\tag{7.6}$$

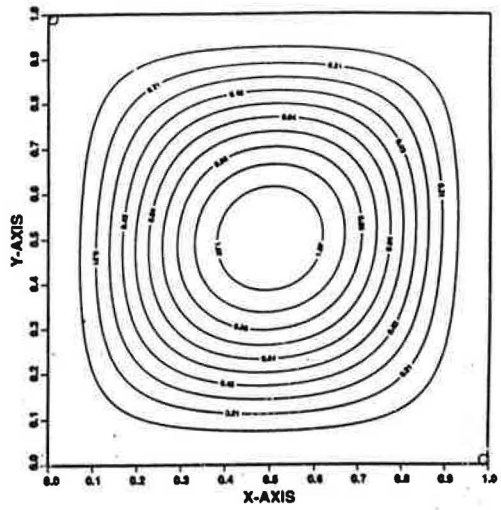
In the case of velocity dependence, for example U_w, we have

$$\frac{\partial F^T}{\partial U_w} = (T_c - T_w)\frac{\partial}{\partial U_w} A_w^T = (T_c - T_w)\begin{cases} \left\{\begin{array}{ll} 1/\delta x & \text{if} \quad U_w > 0 \\ 0 & \text{else} \end{array}\right\} & \text{if}\ \ \delta x|U_w| > 2 \\ 1/2\delta x & \text{if}\ \ \delta x|U_w| < 2 \end{cases}\tag{7.7}$$

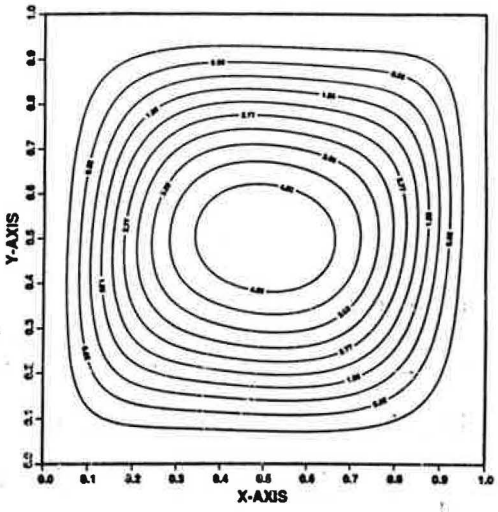
with similar expressions corresponding to the other variables. Then, as mentioned earlier, the resulting linear system is solved by a direct method.

8. RESULTS

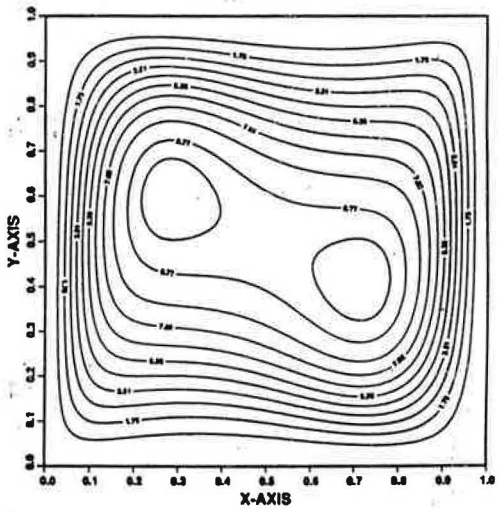
The algorithm described in Sections 3-7 has been applied to the laminar double-glazing problem. Streamlines and isotherms are shown in Figures 2 and 3. The complex flow structures, which are introduced by the nonlinear coupling of the equations, can be seen clearly. Moreover, the fine detail present at the highest Rayleigh number means that small mesh spacing has to be used to resolve the boundary layers. The present version of our code uses uniform meshes. This is not the most



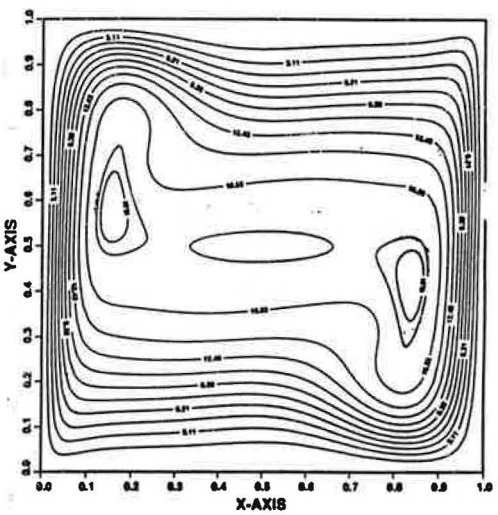
Rayleigh number 10^3



Rayleigh number 10^4

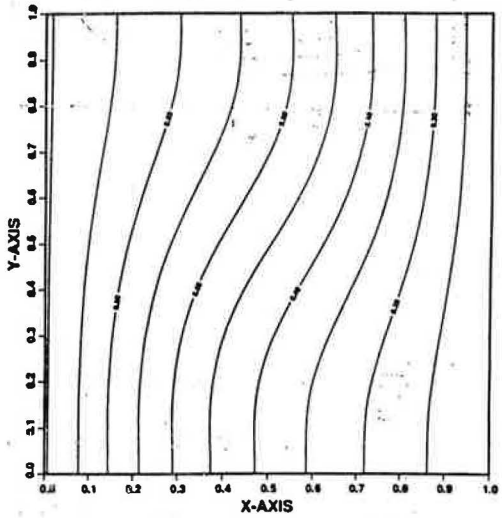


Rayleigh number 10^5

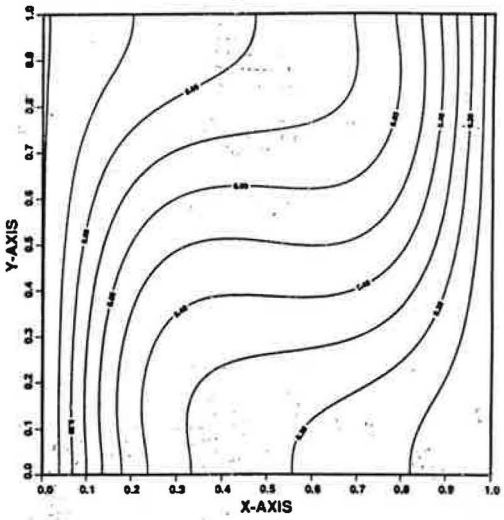


Rayleigh number 10^6

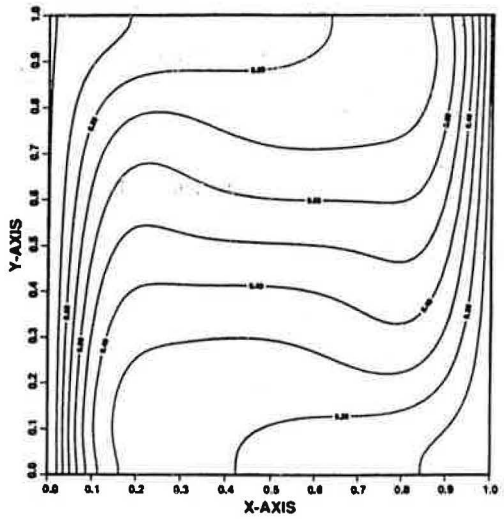
FIGURE 2. Contour maps of the stream function



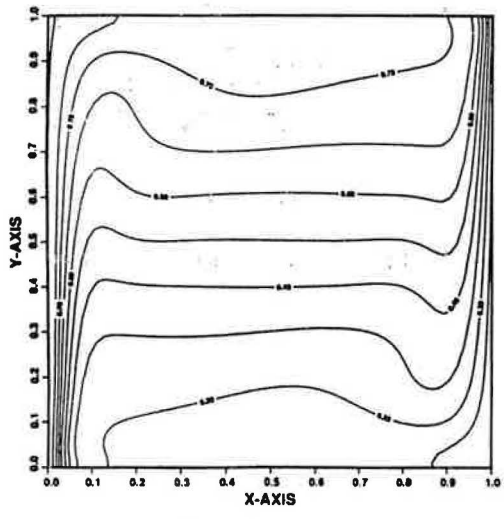
Rayleigh number 10^3



Rayleigh number 10^4



Rayleigh number 10^5



Rayleigh number 10^6

FIGURE 3. Contour maps of the temperature

efficient form to apply to this class of problem, but it does allow demonstration of the effectiveness of the algorithm on this system of equations.

First, we consider the accuracy of solution method. We concentrate on the case with the highest Rayleigh number, which proved to be the most difficult. These results are typical. Table 1 compares two parameters from our calculations with those obtained by Kessler and Oertel [11] and de Vahl Davis [10] from the benchmarking exercise mentioned earlier. (These solutions were judged particularly accurate [12].) The parameters are the maximum horizontal velocity component in the vertical mid-plane (with the vertical position shown in the second row) and the corresponding quantities in the horizontal mid-plane. Clearly, our results agree closely with those from the earlier studies. These quantities are those demanded in the original statement of the double-glazing problem and are reasonably sensitive to any errors in the solutions. Thus we have some confidence that a closer comparison would reveal no anomalies.

Providing some sort of error estimate is extremely useful for all numerical solution techniques. The additional information available with multigrid algorithms facilitates this process. The simplest way (assuming an exact solution is not known) is to use the defect (τ_h^H) which approximates the local truncation error. In Figure 4 we plot τ_h^H for each grid level in the 256×256 calculation. On the finest grids the slope shows that we are in the asymptotic regime and that the errors are behaving as $O(h)$. On the coarse grids the errors are actually increasing. In fact, the thickness of the vertical boundary layer is about 1/30, so the discrete problem is a very poor approximation to the continuous one for $h \leq 1/32$. To obtain estimates of the actual errors, we must solve some extra equations. This is beyond the scope of the present work.

Table 2 shows the average rate of convergence and the times per cycle for the various Rayleigh-number/Prandtl-number combinations for F(2,2) cycles. This is a conservative cycling strategy, but it appears to be reasonably efficient. Moreover, experiments with different values of ν_r and ν_p show only small changes in total CPU time. As can be seen from Table 2, it proved necessary to introduce some under-relaxation to force convergence at the highest Rayleigh numbers (i.e., the problems with largest Frechet derivative); this procedure was done as indicated in Equation (5.40). Moreover, for the $Ra = 10^6$ case, it was necessary to increase the size of the coarsest grid from 2×2 (which was adequate for the lower Rayleigh numbers) to 4×4. This situation was disappointing; a parameter-free algorithm would have been preferable. The convergence data displayed in Tables 2, 3, and 4 indicate that this is a problem associated with our treatment of the nonlinearities, and various techniques are being considered to ameliorate the problem. However, it seems likely that more sophisticated handling of the nonlinearities will have a major overhead in computing time, and it is not clear that the new technique will prove worthwhile.

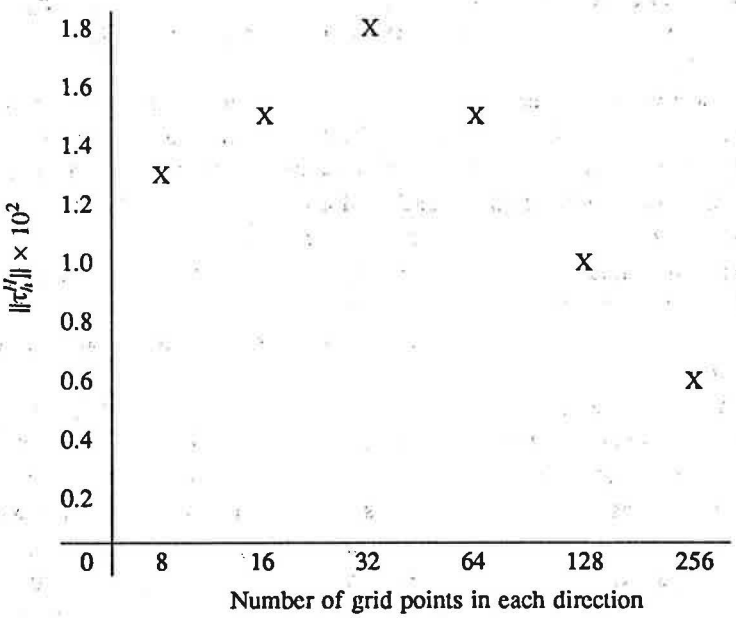
The data presented in Table 3 show that the rate of convergence is sensitive to the under-relaxation factor and the cycle type. The behavior of the algorithm is complex because of the high degree of nonlinearity and the fact that hybrid differencing has been used. This difference scheme uses stable, upwind-differencing at high mesh-Reynolds numbers and more accurate, central differencing at low mesh-Reynolds numbers. Thus, the coarse grid operators are only fair approximations to the fine grid ones. Moreover, at the highest Rayleigh numbers there is a large domain where very low velocities allow central differencing even on the coarse grids. The accuracy of the coarse grid correction is, therefore, somewhat varied and affects the convergence in a complex fashion. The amount of work that should be spent on the different grids is extremely difficult to predict because of this. Our experiments show that F-cycles are the most efficient. It is clear that V-cycles do not provide sufficiently good corrections. W-cycles are marginally better than F-cycles, in terms of convergence rates for the same under-relaxation parameter (Table 3(b)), although the

TABLE 1. Accuracy of results - Double glazing problem

Maximum horizontal velocity in vertical mid-plane (with location)
Maximum vertical velocity in the horizontal mid-plane (with location)
(Rayleigh number: 10^6, Prandtl number: 0.71)

Grid	32^2	64^2	128^2	256^2	Benchmark Results	
$U_{max}(x=0.5)$ $y =$	66.18 0.86	65.43 0.85	64.99 0.85	64.88 0.85	64.63[1] 0.850	65.21[2] 0.854
$V_{max}(y=0.5)$ $x =$	202.4 0.047	221.6 0.039	217.9 0.043	220.8 0.037	219.36 0.0379	220.4 0.039

Note: (1) de Vahl Davis [10]
(2) Kessler and Oertel [11]



Rayleigh number: 10^6, Prandtl number: 0.71;
thickness of vertical boundary layer approximately 1/30

FIGURE 4. Local truncation error estimates for double-glazing problem

TABLE 2(a). Convergence rate as a function of Rayleigh number
(Prandtl number: 0.71, F(2,2) cycles, 64×64 grid)

Ra	1.0	10	10^2	10^3	10^4	10^5	10^6
$\omega = 1.0$	0.084	0.085	0.099	0.091	0.12	---	---
$\omega = 0.395$	0.26	0.295	0.306	0.315	0.283	0.352	0.45

TABLE 2(b). Convergence rate as a function of Prandtl number
(Rayleigh number: 10^4, F(2,2) cycles, 64×64 grid)

Pr	0.1	0.5	0.71	5.0
$\omega = 1.0$	0.47	0.15	0.12	0.12
$\omega = 0.395$	0.47	0.32	0.28	0.32

TABLE 3(a). Rate of convergence as a function of cycle type and under-relaxation factor
(Rayleigh number: 10^6, Prandtl number: 0.71, grid: 64^2)
(Calculations terminated when $\|\text{Residual}\| < 10^{-4}$)

F(2,2) cycles		V(2,2) cycles		W(2,2) cycles	
ω	# cycles	ω	# cycles	ω	# cycles
0.3	31	0.15	73	0.30	31
0.35	27	0.17	71	0.31	30
0.36	26	0.18	74	0.32	29
0.37	25	0.19	72	0.33	28
0.38	24	0.20	72	0.34	27
0.39	24	0.21	75	0.345	28
0.395	23				
0.40	24				
0.4014	26				

TABLE 3(b). Rate of convergence as a function of cycle type

(Prandtl number: 0.71, grid: 64^2)

Rayleigh number 10^4, $\omega = 1.0$		Rayleigh number 10^6, $\omega = 0.2$	
F(2,2)	0.124	F(2,2)	0.68
V(2,2)	0.427	V(2,2)	0.77
W(2,2)	0.106	W(2,2)	0.67

benefits are slight because of the hybrid differencing. Even in this case, the extra arithmetic in W-cycles makes them less efficient. Somewhat surprisingly, F-cycles have significantly better stability properties, allowing larger under-relaxation parameters and better convergence rates.

In Table 4 we show the rate of convergence for the most difficult problem ($Ra = 10^6$) for a range of grid sizes. Linear multigrid theory predicts a rate independent of mesh size, and we see the same sort of behavior here. This fact is very encouraging: we are observing proper multigrid behavior and rates of convergence that are extremely good for the lower Rayleigh number cases and that are still reasonable for the high Rayleigh number case, even though under-relaxation has been introduced.

TABLE 4. Rate of convergence as a function of grid size

(Prandtl number: 0.71)

Rayleigh number 10^6			Rayleigh number 10^4		
Grid size	ω	Rate of convergence	Grid size	ω	Rate of convergence
32^2	0.35	0.5	32^2	1.0	0.17
64^2	0.395	0.45	64^2	1.0	0.12
128^2	0.395	0.367	128^2	1.0	0.085
256^2	0.395	0.5	256^2	1.0	0.077

It is possible that the initial goal of finding a parameter-free algorithm that would be highly efficient over a very wide range of nonlinearities was overambitious. Almost all computational fluid dynamics codes employ such techniques, and the multigrid convergence rates that have been achieved are one or two orders of magnitude better than corresponding single-grid algorithms.

9. CONCLUSIONS

A novel multigrid algorithm has been presented for buoyancy-induced flows. The relaxation scheme avoids the introduction of Brunt-Vasaila oscillations which limit the performance of classical, segregated approaches. The new method appears to be reasonably efficient and robust, converging over a range of physical parameters from zero initial guess.

It is necessary to use some under-relaxation at the highest Rayleigh numbers. The reasons are being investigated.

ACKNOWLEDGMENTS

The authors thank Judy Beumer for carefully typing this manuscript, Gail Pieper for her editorial assistance, and the referee whose comments improved this paper.

REFERENCES

1. D. B. Spalding, "A Novel Finite-Difference Formulation for Differential Expressions Involving Both First and Second Derivatives," *International Journal for Numerical Methods in Engineering*, Vol. 4, pp. 551-559, 1972.

2. A. Brandt, "Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics," *Computational Fluid Dynamics, Lecture Series 1984-04*, von Karman Institute, Belgium, 1984.

3. K. Stuben and U. Trottenberg, "Multigrid Methods, Fundamental Algorithms, Model Problem Analysis and Applications," *Multigrid Methods, Lecture Notes in Mathematics*, Vol. 960, Springer-Verlag, Berlin, 1982.

4. A. Brandt and N. Dinar, "Multigrid Solutions to Elliptic Flow Problems," *Numerical Methods for Partial Differential Equations*, Academic Press, New York, pp. 53-148, 1979.

5. S. P. Vanka, "Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables," *J. Comp. Phys.*, Vol. 65, No. 1, pp. 138-158, 1986.

6. J. R. Kightley and C. P. Thompson, "On the Performance of Some Rapid Elliptic Solvers on a CRAY-1," *SIAM J. on Sci. Stat. Comp.* (to appear) (also AERE Report CSS 186, 1986).

7. C. P. Thompson, N. S. Wilkes, and I. P. Jones, "Numerical Studies of Buoyancy Driven Turbulant Flow in a Rectangular Cavity," *Int. J. Num. Meth. Eng.*, Vol. 24, 1987 (to appear).

8. I. P. Jones, "The Convergence of Predictor-Corrector Methods Applied to the Prediction of Strongly Stratified Flows," Proc. Conf. on Numerical Methods in Laminar and Turbulent Flow, Swansea, Pineridge Press, pp. 733-740, 1985.

9. P. F. Galpin and A. D. Raithby, "Numerical Solution of Problems in Incompressible Fluid Flow: Treatment of Temperature-Velocity Coupling," *Num. Heat Transf.*, Vol. 10, pp. 105-129, 1986.

10. I. P. Jones and C. P. Thompson, "Numerical Solutions for a Comparison Problem on Natural Convection in an Enclosed Cavity," *AERE Report 9955*, 1981.

11. R. Kessler and H. Oertel Jr. in [10] (supplement 1).

12. G. de Vahl Davis and I. P. Jones, "Natural Convection in a Square Cavity: A Comparison Exercise," *Int. J. Num. Meth. in Fluids*, Vol. 3, pp. 227-248, 1983.