HYBRID RELAXATION — A TECHNIQUE TO ENHANCE THE RATE OF CONVERGENCE OF ITERATIVE ALGORITHMS

R. Marek and J. Straub
Institute A for Thermodynamics, Technical University of Munich,
Arcisstrasse 21, W-8000 Munich 2, Germany

In solving multidimensional transient fluid flow and heat transfer problems, the strongly coupled conservation laws of mass, momentum, and energy require segregated iterative procedures. Derived from the SIMPLE algorithm, the fully explicit iteration scheme MAPLE (Modified Algorithm for Pressure-Linked Equations) for the pressure–velocity coupling is introduced here. A substantial speedup is gained in the iteration by utilizing hybrid relaxation, a combination of under- and overrelaxation, instead of the usual underrelaxation. Moreover, hybrid relaxation is not restricted to pressure–velocity algorithms only, but can be applied in more general iterations.

INTRODUCTION

In general, the conservation laws governing multidimensional heat transfer and fluid flow are nonlinear partial differential equations requiring a numerical approach, as analytical solutions are available only for a limited number of simplified cases. By elimination of the pressure in two-dimensional situations, the Ψ, Ω method reduces the number of dependent variables by one, with vorticity Ω and stream function Ψ being calculated instead of pressure and two velocity components. Three-dimensional problems, however, are usually treated in terms of primitive variables p, T, and $\mathbf{v}$ by means of finite differences, finite elements, finite volumes, or combinations of them. As a common feature of all these methods, the coupled partial differential conservation equations for mass, momentum, and energy are transformed into a system of linear algebraic equations, which is solved for discrete regions of the calculation domain.

Regarding time-dependent problems, the available numerical algorithms can be classified into explicit, implicit, and semiimplicit techniques. Usually, time steps of arbitrary size can be chosen with implicit and semiimplicit methods, while several stability criteria limit the increments in time with explicit procedures.

Pressure and velocity in a fluid are strongly coupled via the continuity (1) and momentum (2) equations. With both explicit and implicit solution techniques, the problem arises that, for the calculation of the flow field, the pressure of the new time level is usually unknown. A variety of iterative methods, such as SIMPLE [1–4], SIMPLER [3, 5, 6], SIMPLEC [1, 6, 7], or SIMPLEX [7] have been developed to deal with the coupling of pressure and velocity. Excellent comparisons of the different algorithms are carried out by Jang et al. [5] and Latimer and Pollard [6]. The well-known SIMPLE algorithm of Patankar and Spalding [2–4] and its enhanced successors use underrelaxation to enable convergence of the pressure–

1040-7790/93 $10.00 + .00

NOMENCLATURE

a_{nb}	velocity coefficient in the discretized continuity, Eq. (7)	ω	variable relaxation
		Ω	vorticity
b_{nb}, b_j	velocity coefficients in the discretized momentum, Eq. (8)	**Subscripts**	
c_j	pressure coefficient in the discretized momentum, Eq. (8)	adm	admissible
c_p	isobaric specific heat capacity	b	bottom
D	Diameter of the liquid column	e	eastern
d_{NB}, d_P	temperature coefficients in the discretized energy, Eq. (9)	k	coordinate (e.g., $k = x, y, z$)
		n	northern
e_{NB}, e_P	pressure coefficients in the pressure correction, Eq. (22)	nb	neighbor control volume faces (e, w, n, s, b, t)
g	vector of gravitational acceleration	NB	neighbor grid points (E, W, N, S, B, T)
H	height of the liquid column		
M	mass source	max	maximum
p	pressure	p	pressure
r	radial coordinate	P	center of a control volume
S	security factor	s	southern
t	time	t	top
T	temperature	v	velocity
$\mathbf{v}$	vector of velocity		
z	axial coordinate	**Superscripts**	
∇	nabla operator		
Δ	increment or difference	w	western
η	dynamic viscosity	0	old time level t
λ	thermal conductivity	1	new time level $t + \Delta t$
μ	convergence factor	I	previous iteration step
ϱ	density of the fluid	II	current iteration step
σ	surface tension	'	corrective term
$\bar{\bar{\tau}}$	stress tensor	*	estimated term or value of the old time level
φ	azimuthal coordinate		
Ψ	stream function	**	corrected or improved term

velocity iteration. However, underrelaxation slows down the rate of convergence due to its dampening effect. A speedup, which is generally possible by employing over-instead of underrelaxation, is highly desirable. Pure overrelaxation, on the other hand, is prone to oscillations and numerical instabilities. As a significant improvement of the SIMPLE algorithm, the SIMPLEC method of van Doormaal and Raithby [1, 6, 7] removes the need for underrelaxation in the pressure correction.

In this article, the improved explicit pressure–velocity correction algorithm MAPLE is presented utilizing the technique of hybrid relaxation, which reduces the number of necessary iterations by alternately employing under- and overrelaxation depending on the convergence or divergence of the previous iteration step.

It is assumed that the reader is familiar with the basic ideas of the SIMPLE algorithm, which is described in detail in [2–4], and also in [1] and [6]. For consistency, Patankar's notation for the grid points and coefficients [3] is used throughout.

THE CONSERVATION LAWS

Transient multidimensional fluid flow and heat transfer are governed by the conservation laws [8] for

Mass:

$$\frac{\partial \varrho}{\partial t} + \nabla \cdot (\varrho \mathbf{v}) = 0 \tag{1}$$

Momentum:

$$\frac{\partial(\varrho \mathbf{v})}{\partial t} = -\nabla \cdot (\varrho \mathbf{v}\mathbf{v}) - \nabla p - \nabla \bar{\bar{\tau}} + \varrho \mathbf{g} \tag{2}$$

Energy (work of viscous and pressure forces neglected):

$$\frac{\partial(\varrho T)}{\partial t} = -\nabla \cdot (\varrho T \mathbf{v}) + \frac{\lambda}{c_p} \nabla^2 T \tag{3}$$

EXPLICIT VERSUS IMPLICIT METHODS

In transient heat transfer and fluid flow calculations, implicit and semiimplicit methods are often preferred to explicit schemes, as the size of the time steps is not limited. However, implicit algorithms are sometimes abstruse, and exhibit a greater computational effort than explicit methods, as systems of linear equations have to be solved. This handicap can be partially compensated for by employing sufficiently large time steps, which may, however, be disadvantageous when a fine resolution of the time scale is required in unsteady-state cases. On the other hand, only a few problems of numerical instability or divergence have been reported with implicit methods.

As a superior feature of explicit schemes, the unknown variables of the new time level (superscript 1) are calculated directly from the old values (superscript 0), starting with the values defined by the initial conditions. Consequently, explicit methods can easily be implemented into a computer code due to their computational simplicity and their conceptual clarity. Even more beneficial for three-dimensional time-dependent simulations are the smaller requirements of computer memory compared to implicit schemes. Especially when the focus if on transient development of a flow, explicit schemes are preferable.

On the other hand, for the choice of the time step Δt, both diffusive and convective stability criteria have to be obeyed. If not, instabilities in the numerical solution or physically unrealistic results occur. The diffusive (heat conduction) stability criterion derives from the second law of thermodynamics [9, 10], while the convective limit is given by the Courant-Friedrichs-Lewy (CFL) condition [11, 12]. The admissible time step Δt_{adm} is the minimum time step resulting from both criteria [10]. Additionally, we employ a security factor S for the choice of the time increment:

$$\Delta t = \frac{1}{S}\Delta t_{adm} \qquad (4)$$

The security factor S or the time step, respectively, significantly determines the number of iterations in the pressure–velocity algorithm and the consumed CPU time. A small time step Δt causes frequent calls of the subroutines for momentum, energy, the boundary conditions, the pressure–velocity iteration, and of some other auxiliary routines. On the other hand, the variations between two successive time steps in the dependent variables pressure, temperature, and velocities are insignificant for small time steps. As a consequence, the iteratively calculated corrections for pressure and velocities are small, too, and only a few iteration steps are necessary. Decreasing the size of the time step shifts the computational effort from the pressure–velocity algorithm toward other routines. In the performed calculations, a reasonable compromise between the number of iterations and the totally consumed computing time was obtained for a security factor of $S = 4$.

In our opinion, the benefits of explicit methods completely make up for the limitation of the time step by stability criteria and other disadvantages. Our experience of several years with explicit finite-difference and finite-volume schemes [13–16] favors their application.

A FULLY EXPLICIT PRESSURE–VELOCITY ALGORITHM: MAPLE

Subsequently, the explicit procedure MAPLE (Modified Algorithm for Pressure-Linked Equations) is utilized to uncouple the discretized equations for mass, momentum, and energy. The basic ideas of MAPLE are similar to those of Patankar's SIMPLE algorithm, while the semiimplicit calculations of the pressure corrections is replaced by a fully explicit evaluation. Although this modification increases the number of iterations, the overall computational expenditure is still smaller than in the semiimplicit case.

The Discretized Conservation Laws in Explicit Form

Frequently, the so-called Boussinesq approximation [17], originally proposed by Overbeck [18], is employed for the density of the fluid and other properties. In our study, we use an extended Boussinesq approximation that takes into account the temperature dependence of the density not only in the buoyancy term, but throughout the conservation equations. The density ϱ_P^1 of the central grid point P at the new time level $t + \Delta t$ is a function of the temperature T_P^1 of this node, and is unknown, as T_P^1 has to be calculated first:

$$\varrho_P^1 = \varrho(T_P^1) \qquad (5)$$

However, ϱ_P^1 influences the temperature T_P^1 itself via the conservation equations. In order to avoid an iterative calculation of temperature and density, the density difference between two successive time steps is neglected for the evaluation of temperatures and velocities [13]:

$$\varrho_P^1 \approx \varrho_P^0 \qquad (6)$$

This approximation is certainly valid for the small time steps of an explicit method. As soon as the new temperature T_P^1 is available, the density ϱ_P^1 is updated using Eq. (5). With the simplification in Eq. (6), the conservation Eq. (1)–(3) are discretized as follows:

Mass:

$$0 = \sum_{\text{nb}} a_{\text{nb}} v_{k,\text{nb}}^0 \qquad (7)$$

Momentum:

$$v_{k,j}^1 = b_j v_{k,j}^0 + \sum_{\text{nb} \neq j} b_{\text{nb}} v_{k,\text{nb}}^0 + c_j \Delta p_j^0 + \Delta t g_k \qquad (8)$$

Energy

$$T_P^1 = d_P T_P^0 + \sum_{\text{NB} \neq P} d_{\text{NB}} T_{\text{NB}}^0 \qquad (9)$$

In Eqs. (7)–(9), the index k represents the coordinate (e.g., x, y, z), the index j represents the face of the control volume, and the indices nb and NB represent neighbor control volume faces and neighbor points.

The Mass Source

To obtain the flow field of the new time level from the discretized momentum Eq. (8), only the pressure of the old time level p^0 is available. Hence, the velocities $v_{k,j}^* = v_{k,j}^1$ calculated in such a manner do, in general, not satisfy the continuity Eq. (7). A residual representing a nonsolenoidal flow with an imaginary mass source M remains [3]:

$$M = \sum_{\text{nb}} a_{\text{nb}} v_{k,\text{nb}}^* \qquad (10)$$

A stepwise calculation of pressure and velocities is required, usually performed by iterative algorithms, such as SIMPLE, SIMPLER, SIMPLEC, or SIMPLEX. For this purpose, corrected values of pressure and velocities, denoted by a double asterisk, are defined:

$$v_{k,j}^{**} = v_{k,j}^* + v_{k,j}' \qquad (j = \text{e, w, n, s, b, t}) \qquad (11)$$

$$p_J^{**} = p_J^* + p_J' \qquad (J = \text{P, E, W, N, S, B, T}) \qquad (12)$$

Corrected or improved values are achieved by adding the primed corrective terms to the starting values characterized by a single asterisk.

The Velocity Correction

The starting velocities $v^*_{k,j}$ are calculated from the discretized momentum Eq. (8), while the pressure differences Δp^*_j are evaluated for the old time level or the initial condition at $t = 0$, respectively:

$$v^*_{k,j} = b_j v^0_{k,j} + \sum_{\mathrm{nb} \neq j} b_{\mathrm{nb}} v^0_{k,\mathrm{nb}} + c_j \Delta p^*_j + \Delta t g_k \tag{13}$$

$$p^*_j = p^0_j \tag{14}$$

The improved velocities satisfy the momentum Eq. (8), as well:

$$v^{**}_{k,j} = v^*_{k,j} + v'_{k,j} = b_j v^0_{k,j} + \sum_{\mathrm{nb} \neq j} b_{\mathrm{nb}} (v^0_{k,\mathrm{nb}} + v'_{k,\mathrm{nb}}) + c_j(\Delta p^*_j + \Delta p'_j) + \Delta t g_k \tag{15}$$

Subtracting Eq. (13) from Eq. (15) yields an implicit relation for the corrective velocities:

$$v'_{k,j} = \sum_{\mathrm{nb} \neq j} b_{\mathrm{nb}} v'_{k,\mathrm{nb}} + c_j \Delta p'_j \tag{16}$$

Due to the iterative nature of the SIMPLE method, Patankar [2–4] suggests neglecting the corrective terms deriving from the neighbor velocities:

$$\sum_{\mathrm{nb} \neq j} b_{\mathrm{nb}} v'_{k,\mathrm{nb}} \approx 0 \tag{17}$$

By means of this simplification, the velocity corrections $v'_{k,j}$ can be calculated explicitly from the pressure corrections $\Delta p'_j$:

$$v'_{k,j} \approx c_j \Delta p'_j \tag{18}$$

Equation (18) reduces the evaluation of appropriate velocity corrections to the problem of finding corresponding pressure corrections.

The Pressure Correction

Both the corrections $v'_{k,j}$ and $\Delta p'_j$ have to be determined in a way that finally the improved velocities $v^{**}_{k,j}$ satisfy the continuity Eq. (7) exactly:

$$0 = \sum_{\mathrm{nb}} a_{\mathrm{nb}} v^{**}_{k,\mathrm{nb}} \tag{19}$$

This condition is equivalent to $M = 0$. Due to the great computational effort, a zero mass source is not desirable in practical calculations. The iteration is terminated, rather, when the mass source M is smaller than an admissible value M_{adm}:

$$M \leq M_{\mathrm{adm}} \tag{20}$$

Hence, the aim of the pressure–velocity correction is to annihilate the mass source M in a step-by-step procedure by successively updating the corrective terms $v'_{k,\mathrm{nb}}$ and $\Delta p'_j$. Substituting Eqs. (10) and (11) into Eq. (19) yields another relation for the corrective velocities $v'_{k,\mathrm{nb}}$:

$$0 = M + \sum_{\mathrm{nb}} a_{\mathrm{nb}} v'_{k,\mathrm{nb}} \tag{21}$$

With the help of Eq. (18), the velocity corrections $v'_{k,\mathrm{nb}}$ are eliminated from Eq. (21), providing an implicit relation for the corrective pressure:

$$e_P p'_P = \sum_{\mathrm{NB} \neq P} e_{\mathrm{NB}} p'_{\mathrm{NB}} - M \tag{22}$$

Hitherto, the presented ideas of the MAPLE method are nearly identical with those of the SIMPLE algorithm. However, in the SIMPLE procedure the implicit pressure correction Eq. (22) is solved directly, which requires the solution of a set of linear equations. This, in turn, gives rise to an additional computational effort. Wengle [19] proposed a similar simplification as given for the corrective velocities by Eq. (17). In his opinion, the influence of the mass source is more decisive than that of the pressure corrections of the neighbor grid points, which are therefore neglected:

$$\sum_{\mathrm{NB} \neq P} e_{\mathrm{NB}} p'_{\mathrm{NB}} \approx 0 \tag{23}$$

Finally, the corrective pressure p'_P can be calculated explicitly in the MAPLE algorithm:

$$p'_P \approx -\frac{M}{e_P} \tag{24}$$

Neglecting the neighbor pressure corrections according to Eq. (23) slightly increases the number of iterations necessary to satisfy the stopping criterion (20). On the other hand, our recent calculations [16] indicate that this additional expenditure easily compensates for the numerical effort of solving Eq. (22) directly.

The Iteration Loop

The main steps of the MAPLE algorithm can be summarized as follows:

1. Obtain the starting velocities from the discretized momentum Eq. (13) using a pressure field according to Eq. (14).
2. Evaluate the mass source for all control volumes by means of Eq. (10).

3. Calculate the pressure corrections for all grid points from Eq. (24).
4. Update the pressure field by employing Eq. (12).
5. Calculate the velocity corrections from Eq. (18).
6. Update the velocity according to Eq. (11).
7. Calculate a new mass source from Eq. (10) with the corrected velocities of step 6.
8. Terminate the iteration, if the stopping criterion Eq. (20) is satisfied. Otherwise continue with step 2 using the mass source calculated in step 7.

THE HYBRID RELAXATION

Due to problems of convergence in the SIMPLE algorithm, Patankar [2–4] uses the technique of relaxation for the calculation of the corrected pressure and velocities:

$$v_{k,j}^{**} = v_{k,j}^{*} + \omega_v v'_{k,j} \qquad (j = \mathrm{e, w, n, s, b, t}) \tag{25}$$

$$p_J^{**} = p_J^{*} + \omega_p p'_J \qquad (J = \mathrm{P, E, W, N, S, B, T}) \tag{26}$$

In contrast to Eqs. (11) and (12), the corrective terms are now weighted with relaxation parameters ω_v and ω_p. For semiimplicit schemes, Patankar [3] recommends relaxation factors of $\omega_p \leq 0.8$ and $\omega_v \leq 0.5$ to enable convergence of the pressure–velocity iteration.

Underrelaxation ($\omega < 1$) is a very powerful tool for dampening out oscillations in iterative algorithms, as it slows down the variations in the dependent variable from iteration step to iteration step. However, this stabilizing effect is paid for with an increase in the number of necessary iterations. Overrelaxation ($\omega > 1$), on the other hand, is frequently employed to speed up iterative procedures. With $\omega > 1$, the dependent variable is overcorrected. The next iteration step has to provide a correction in the opposite direction, thus inducing the amplifying oscillations.

The question arises whether these two relaxation techniques can be joined in a way that both the stabilizing effect of underrelaxation and the acceleration of overrelaxation can be utilized. This combination is possible, indeed, and is called hybrid relaxation with reference to Spalding's hybrid finite-difference scheme [20].

Generally, the mass source M is different from every control volume. However, there exists at least one control volume in the computational domain where M has a maximum value, $M_{\max}$. The stopping criterion Eq. (20) can be checked for every control volume separately, or it can be applied only to the control volume with the maximum mass source. The latter is computationally easier, and guarantees that Eq. (20) is satisfied for all control volumes in the calculation domain. It was therefore employed in our studies. Additionally, the maximum mass source is an indication for the convergence or divergence of the pressure–velocity iteration. If the maximum mass source gradually decreases, the iteration process converges; otherwise it diverges.

As a criterion for the convergence of an iteration step, a convergence factor μ is introduced as the ratio of the maximum mass sources of two successive iterations:

$$\mu = \frac{M_{\max}^{\mathrm{II}}}{M_{\max}^{\mathrm{I}}} \tag{27}$$

To distinguish the iteration step level from the time level, Roman numerals are used as superscripts; II denotes the current iteration step, while the previous one is indicated by I. For $\mu < 1$, the performed iteration was convergent, otherwise it diverged.

The principle of hybrid relaxation is to choose the relaxation factor ω for the next iteration step according to the convergence factor μ of the current iteration. If the completed iteration diverged, the following iteration step has to be underrelaxed, and vice versa:

$$\begin{aligned} \mu > 1 &\Rightarrow \omega < 1 \\ \mu < 1 &\Rightarrow \omega > 1 \end{aligned} \tag{28}$$

As long as the iteration converges, overrelaxation, is used to fully utilize its fast rate of convergence. As soon as divergence occurs, the hybrid relaxation algorithm switches to underrelaxation and thus dampens out the beginning instabilities. When convergence is reached again, the iteration is continued with overrelaxation. Step 4 of the iteration loop is performed with Eq. (26) instead of Eq. (12), and step 6 Eq. (25) instead of Eq. (11).

The history of convergence is shown in Fig. 1 for different but constant relaxation factors for the liquid column described below with a grid of 14 × 8 × 14 nodes. The plain iteration ($\omega = 1$), which is feasible for the proposed explicit MAPLE algorithm, can be taken as a reference for the rate of convergence. Overrelaxation with $\omega = 1.2$ exhibits a fast convergence in the very early stage; however, after a few iterations, it becomes unstable and diverges. On the other hand, underrelaxation with $\omega = 0.8$ decelerates the convergence, but guarantees a stable iteration. The combination of under- ($\omega = 0.8$) and overrelaxation ($\omega = 1.2$) according to Eq. (28) remarkably reduces the number of iterations.

The optimum relaxation parameters are variable and depend on the convergence factor μ itself. In a lengthy series of calculations, we found optimum values of the relaxation parameters for our test problem:

$$\begin{aligned} \omega(\mu > 1) &= 0.6 \\ \omega(\mu < 1) &= \frac{2.2}{1 + \sqrt{1 - \mu}} \end{aligned} \tag{29}$$

Nevertheless one should not conclude that these values are the optimum ones for all problems; usually they are problem-dependent, and have to be found out by a

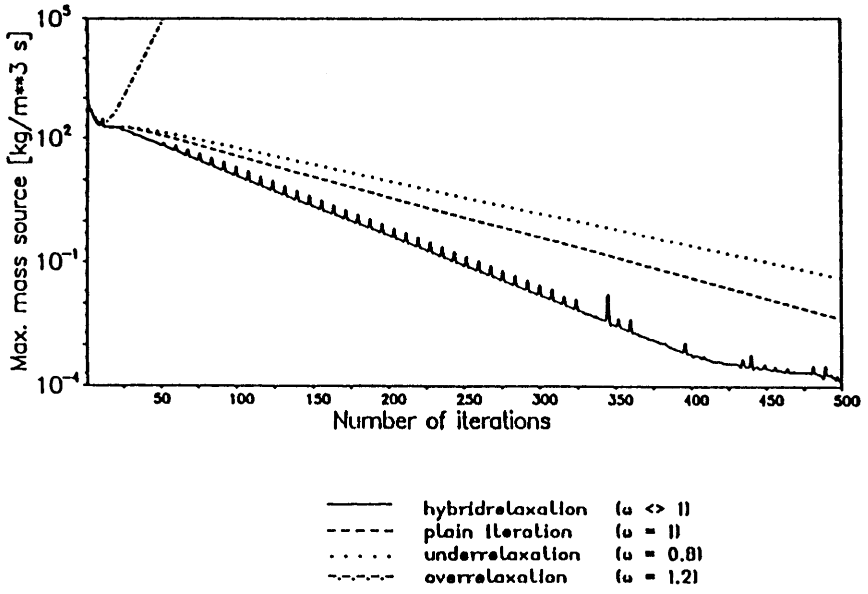

Fig. 1 Rate of convergence for various relaxation methods.

trial-and-error search for every special problem. Thus, the factors given by Eq. (29) can rather be regarded as starting values for their iterative evaluation.

The wavy convergence curve for the hybrid relaxation is due to alternately switching between under- and overrelaxation. In all the performed calculations, no global divergence occurred; only peaks in the maximum mass source were observed. To satisfy the stopping criterion (20), it is not the transient development of the mass source but its final value that is decisive. Therefore the peaks in the convergence curve do not matter.

For a variety of test problems, an average speedup of about 30–40% was obtained by hybrid relaxation. When the grid is refined, the speedup is slightly reduced, as the differences in the dependent variables between the old and the new time levels, and also between different grid points, are smaller than with coarser grids. Nevertheless, an acceleration by the MAPLE algorithm can be noticed even with fine grids. Although the basic concept of hybrid relaxation is presented for an explicit method, it is applicable to implicit iteration algorithms as well.

Similarities with the SIMPLEC Algorithm

In the SIMPLEC procedure of van Doormaal and Raithby [1], the term $\Sigma_{nb \neq j}\, b_{nb}\, v'_{k,j}$ is subtracted from both sides of Eq. (16), and a simplification different from Eq. (17) is made:

$$\sum_{nb \neq j} b_{nb}(v'_{k,nb} - v'_{k,j}) \approx 0 \tag{30}$$

The velocity corrections $v'_{k,j}$ derive immediately form the pressure corrections $\Delta p'_j$:

$$v'_{k,j} \approx \frac{c_j}{1 - \sum\limits_{nb \neq j} b_{nb}} \Delta p'_j \tag{31}$$

In contrast to Eq. (18), c_j is now divided by $1 - \Sigma_{nb \neq j} b_{nb}$. All other steps and equations in the SIMPLEC procedure are the same as in the SIMPLE algorithm. With the velocity corrections $v'_{k,j}$ and $v'_{k,nb}$ being of the same order, Eq. (30) introduces a smaller error in the iteration than Eq. (17). therefore, the rate of convergence increases compared to the SIMPLE algorithm. Underrelaxation is necessary only in the velocity correction and not for the pressure correction, as in the SIMPLE method.

Nevertheless, calculating pressure and velocity corrections with the SIMPLEC method involves the solution of a set of algebraic equations. For this purpose, van Doormaal and Raithby [1] recommend the use of a TDMA (tridiagonal matrix algorithm) solver.

RESULTS

The proposed MAPLE algorithm has been tested extensively in a series of different simulations for two- and three-dimensional recirculating laminar and turbulent flows [13–16]. However, these investigations focused more on heat transfer and fluid flow characteristics, and parameter studies, while a detailed description of the theoretical background of the MAPLE algorithm is given here.

Thermocapillary Convection in a Liquid Column

Temperature gradients along a free surface induce a convective flow termed Marangoni [21], thermocapillary, or surface tension-driven convection. As a test problem for the MAPLE algorithm with hybrid relaxation, the interaction of buoyancy and thermocapillary flow in a vertical, cylindrical liquid column held between two rigid circular plates is studied under various gravity levels [14]. The configuration under consideration and the corresponding boundary and initial conditions are sketched in Fig. 2. The bottom and top walls are held at different but constant temperatures, while the free lateral face is adiabatic. The nonslip condition applies for the rigid bottom and top walls. The lateral wall is a nondeformable free surface subject to the Marangoni boundary conditions [14]. Furthermore, a linear temperature profile in the axial direction deriving from pure heat conduction is initially applied, and the fluid is at rest. A hydrostatic pressure distribution prevails in the beginning.

The three-dimensional transient computer code in terms of primitive variables employs a staggered grid as proposed by Harlow and Welch [22] and a hybrid finite-volume scheme in cylindrical coordinates [3]. The temperature dependence of density is considered in a extended Boussinesq approximation, while all other properties of the fluid are assumed to be constant. For the treatment of the cylinder's axis, the grid arrangement suggested by Schneider [15] is used. The

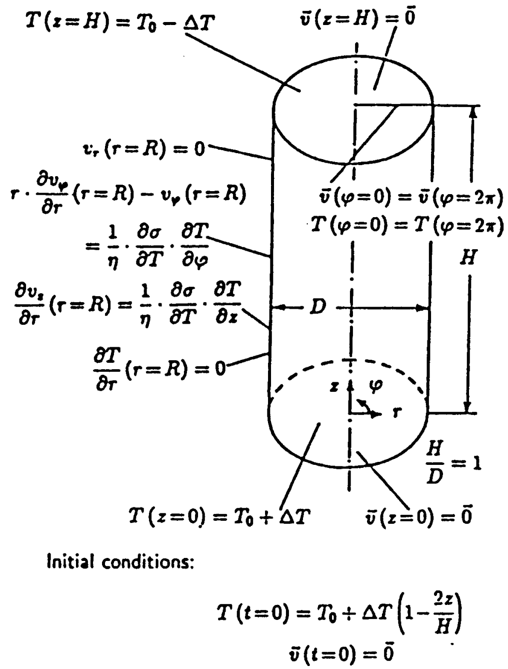
$T(z=H) = T_0 - \Delta T$
$\vec{v}(z=H) = \vec{0}$
$v_r(r=R) = 0$
$r \cdot \frac{\partial v_\varphi}{\partial r}(r=R) - v_\varphi(r=R) = \frac{1}{\eta} \cdot \frac{\partial \sigma}{\partial T} \cdot \frac{\partial T}{\partial \varphi}$
$\frac{\partial v_z}{\partial r}(r=R) = \frac{1}{\eta} \cdot \frac{\partial \sigma}{\partial T} \cdot \frac{\partial T}{\partial z}$
$\frac{\partial T}{\partial r}(r=R) = 0$
$\vec{v}(\varphi=0) = \vec{v}(\varphi=2\pi)$
$T(\varphi=0) = T(\varphi=2\pi)$
H
D
z
φ
r
$\frac{H}{D} = 1$
$T(z=0) = T_0 + \Delta T$
$\vec{v}(z=0) = \vec{0}$
Initial conditions:
$T(t=0) = T_0 + \Delta T\left(1 - \frac{2z}{H}\right)$
$\vec{v}(t=0) = \vec{0}$

Fig. 2 Geometry of the configuration with initial and boundary conditions.

numerical grid is shown in Fig. 3. Further details not related directly to the MAPLE algorithm can be found in [14].

For illustration purposes, the typical transient development of the flow and temperature fields under microgravity conditions is displayed in Fig. 4 in an axial cross section of the cylinder.

Some More Test Problems for MAPLE

Heiss [13] successfully simulated three-dimensional laminar and turbulent natural convection in a cubical box with an explicit finite-difference scheme on a nonequidistant grid using the simplifications in the pressure–velocity iteration proposed by Wengle [19].

Schneider [15] investigated numerically the influence of gravity perturbations on natural-convection fluid flow and the transfer in inclined cylindrical enclosures of different aspect ratios. The transient three-dimensional computer code is based on hybrid finite differences and the MAPLE algorithm.

More recently, Straub et al. [16] studied Marangoni convection around a floating gas bubble in a rectangular container. The power law scheme [3] and the MAPLE pressure–velocity correction with hybrid relaxation were employed to develop the transient two-dimensional numerical solution algorithm with explicit time steps. The fully vectorized code was run on CRAY Y-MP 4/432 supercomputer. Moreover, searching for an efficient pressure–velocity procedure, a comparison between the SIMPLE [1,4] and SIMPLEC [1,5,6] algorithms with TDMA solver, on the one hand, and the MAPLE iteration with hybrid relaxation, on the other hand, was performed. In the very early stage of the flow, the algorithms with TDMA solver consumed less computing time; however, later on, their computational effort increased enormously. Thus, the MAPLE algorithm was found to be the best choice.

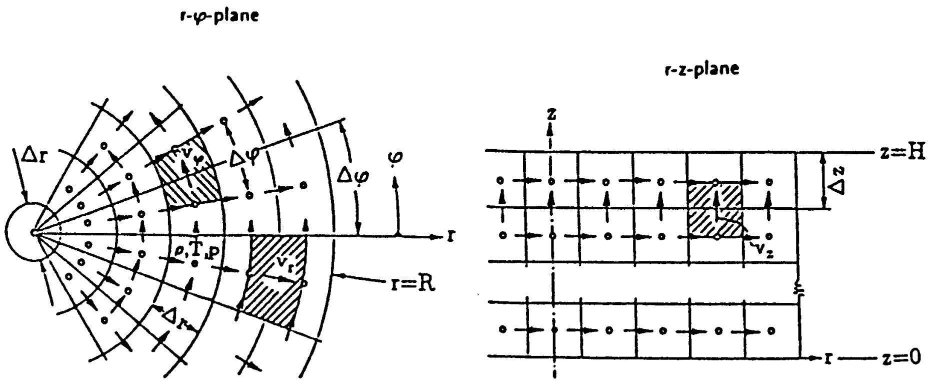

Fig. 3 Employed staggered grid in cylindrical coordinates.

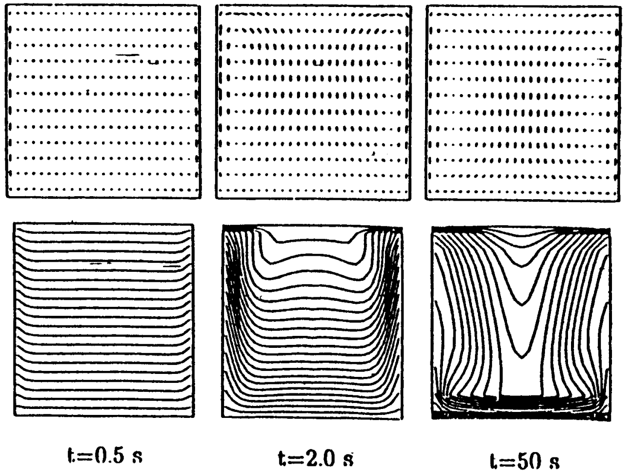

Fig. 4 Transient development of fluid flow and heat transfer in an axial cross section of the investigated configuration.

These test problems indicate that the MAPLE algorithm works for both two- and three-dimensional problems of different geometries. In all test cases, substantial savings in CPU time could be registered. The MAPLE algorithm can be extended to other geometries and flow regimes, such as turbulent or swirl flows.

CONCLUSION

Although explicit methods are subject to some restrictions regarding the choice of the time step, their conceptual simplicity and computational efficiency clearly emphasizes their usefulness. Combining the techniques of under- and overrelaxation, a fully explicit, fast, and powerful pressure–velocity iteration, the MAPLE algorithm, has been proposed. With the implementation of the hybridrelaxation, speedups of 30–40% on an average compared to the plain iteration ($\omega = 1$) were obtained for recirculating-flow problems. The MAPLE algorithm has been applied successfully to a variety of fluid flow and heat transfer problems of different geometries and boundary conditions [13–16]. As the pressure–velocity correction is independent of the flow pattern itself, the MAPLE algorithm is easily applicable for both laminar and turbulent flows in different geometries. Refined comparisons of the MAPLE algorithm with well-established methods such as SIMPLE, SIMPLER, SIMPLEC, or SIMPLEX are in progress.

The proposed method of hybrid relaxation is not restricted to MAPLE or similar algorithms only. In addition, it can easily be extended to other iterative procedures, when the slow-converging underrelaxation is necessary for stability, and overrelaxation alone would produce divergence or oscillatory instability. The values of the optimum relaxation parameters have to be determined as the case may be. However, a first approximation for them is given in this article.

REFERENCES

1. J. P. van Doormaal and G. D. Raithby, Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows, *Numer. Heat Transfer*, vol. 7, pp. 147–163, 1984.
2. S. V. Patankar and D. B. Spalding, A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows, *Int. J. Heat Mass Transfer*, vol. 15, pp. 1787–1806, 1972.
3. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, London, 1980.
4. S. V. Patankar, A Calculation Procedure for Two-Dimensional Elliptic Situations, *Numer. Heat Transfer*, vol. 4, pp. 409–425, 1981.
5. D. S. Jang, R. Jetli, and S. Acharya, Comparison of the PISO, SIMPLER and SIMPLEC Algorithms for the Treatment of the Pressure–Velocity Coupling in Steady Flow Problems, *Numer. Heat Transfer*, vol. 10, pp. 209–228, 1986.
6. B. R. Latimer and A. Pollard, Comparison of Pressure–Velocity Coupling Solution Algorithms, *Numer. Heat Transfer*, vol. 8, pp. 635–652, 1985.
7. G. D. Raithby and G. E. Schneider, Elliptic Systems: Finite Difference Method II, in W. J. Minkowycz, E. M. Sparrow, G. E. Schneider, and R. H. Pletcher (eds.), *Handbook of Numerical Heat Transfer*, chap. 7, pp. 241–291, John Wiley, New York, 1988.
8. B. R Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*, John Wiley, New York, London, 1960.

9. U. Grigull and H. Sandner, *Wärmeleitung*, 2d ed., Springer Verlag, Berlin, 1990.
10. P. J. Roache, *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, NM, 1976.
11. R. Courant, K. Friedrichs, and H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, *Math. Ann.*, vol. 100, pp. 32–74, 1928.
12. C. Hirsch, *Numerical Computational of Internal and External Flows*, vol. 1, John Wiley, Chichester, New York, Brisbane, Toronto, Singapore, 1988.
13. A. Heiss, Numerische und experimentelle Untersuchungen der laminaren und turbulenten Konvektion in einem geschlossenen Behälter, dissertation, Technical University of Munich, 1987.
14. R. Marek and J. Straub, Three-Dimensional Transient Simulation of Marangoni Flow in a Cylindrical Enclosure Under Various Gravity Levels, *Microgravity Sci. Technol.*, vol. 4, no. 2, pp. 153–154, 1991.
15. S. Schneider, Laminare freie Konvektion in einem Zylinder bei konstanter und zeitveränderlicher Schwerkraft (*g*-jitter), dissertation, Technical University of Munich, 1990.
16. J. Straub, J. Betz, and R. Marek, Enhancement of Heat Transfer by Thermocapillary Convection Around Bubbles—A Numerical Study, submitted to *Numer. Heat Transfer, Part A. Appl.*, 1991.
17. J. V. Boussinesq, *Théorie analytique de la chaleur*, vol. 2, Gauthier-Villard, Paris, 1903.
18. A. Oberbeck, Über die Wärmeleitung der Flüssigkeiten bei Berücksichtigung der Strömungen infolge von Temperaturdifferenzen, *Ann. Phy. Chem., Neue Folge*, vol. 7, pp. 271–292, 1879.
19. H. Wengle, Diskretisierung der dreidimensionalen und zeitabhängigen Navier-Stokes-Gleichungen für ein stark dichteveränderliches Fluid, Rep. no. I/82/1, Hochschule der Bundeswehr München, Neubiberg, 1983.
20. D. B. Spalding, A Novel Finite Difference Formulation for Differential Expressions Involving Both First and Second Derivatives, *Int. J. Numer. Meth. Eng.*, vol. 4, pp. 551–559, 1972.
21. C. G. M. Marangoni, Ueber die Ausbreitung der Tropfen einer Flüssigkeit auf der Oberfläche einer anderen, *Ann. Phys. Chem.*, vol. 143, no. 7, pp. 337–354, 1871.
22. F. H. Harlow and E. J. Welch, Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, *Phys. Fluids*, vol. 8, no. 12, pp. 2182–2189, 1965.

Received 6 February 1992
Accepted 15 December 1992