V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

OPTIMIZATION OF THE ITERATION PARAMETERS OF THE KRYLOV SUBSPACE METHODS FOR SIMULATION OF INCOMPRESSIBLE FLOW

A. Shklyar and A. Arbel

Institute of Agricultural Engineering, Agricultural Research Organization, the Volcani Center
P.O. Box 6, Bet Dagan 50250, Israel
e-mail: shklyar@agri.gov.il, arbel@agri.gov.il

Key words: SIMPLE-like method, Krylov subspace methods, Incompressible flow, Genetic algorithm

Abstract. *In optimizing the iteration parameters of the SIMPLE-like numerical procedure, a genetic algorithm (GA) which searches for a minimum calculation time in a space of iteration numbers was developed. A methodology has been presented for the numerical solution of natural convection in a squeezed cavity at Rayleigh number of* 10^6 *and Pr number of 10.0. The pressure correction equation was employed in a conjugate gradient (CG) method with a relative incomplete factorization (RILU(0)) preconditioner. The temperature equation was solved by using preconditioned Krylov subspace methods: the generalized minimum residual (GMRES) method and the bi-conjugate gradient-stabilized (BICGSTAB) method with a RILU(0) preconditioner. The momentum equation was treated by means of the successive overrelaxation (SOR), Gauss-Seidel (GS), GMRES, and GMRES with symmetric Gauss-Seidel (SGS) preconditioner methods. We analyze computed results and we discuss how the calculation time depends on methods and on changes of iteration parameters. The calculation times were not strongly influenced by use of the RILU(0) preconditioner with optimal and close-to-optimal combinations of the perturbation and lumping coefficients, but changed sharply with changes in the pressure iteration parameter.*

1 INTRODUCTION

Since 1972, when Patankar and Spalding[1] successfully developed the SIMPLE method, several variants of SIMPLE have been proposed, to improve convergence rates. Progress of an effective iterative solution of a linearized momentum-mass-energy system depends heavily on the set of iteration parameters and the iterative procedures of the system of linear equations. In the context of values of iteration parameters of computational methods it is appropriate to refer to the iterative method for solving the unsteady viscous flow equations, according to Fletcher[2], p. 169: - "Possibly the delicate point of the method lies in the choice of parameters, since the number of iterations necessary to reach convergence is very sensitive to this choice." The number of iterations, the relaxation factors and the iteration parameters are generally chosen in the light of numerical experiments or published recommendations.

In most studies it is emphasized that the SIMPLE method has a slow convergence rate, even though Cartesian co-ordinates are used. Attempts to enhance the convergence properties of the SIMPLE method in general coordinates may be divided into two parts: enhancement of the approximation of Poisson's pressure correction equation, which includes approximation of cross-derivatives and use of enhanced linear algebraic methods, followed by choice of the set of iteration parameters. Lehnhäuser and Schäfer[3] used an approximation of the pressure derivatives based on a multi-dimensional Taylor expansion and they discussed the problem associated with the pressure-correction equation. Yen and Liu[4] proposed an explicit correction step and an optimal relaxation factor for this additional step. Optimum values of two parameters of the SIMPLE method are defined in a paper by Chatwani and Turan[5], who proposed a pressure-velocity coupling algorithm to determine the underrelaxation factor in the pressure correction. A segregated solution procedure by Tao et al.[6] for incompressible flow and heat transfer problems solves the improved pressure directly, rather than by adding a correction term; a second relaxation parameter was imputed to robustness. A segregated algorithm by Sun et al.[7] includes internal double iterative processes for pressure correction equation at each iteration level.

Thus, there are many variants of the SIMPLE algorithm, and many comparisons among them have been published (for details, see Sun et al.[7]). All variants of the SIMPLE iterative method use a set of iteration parameters, and calculation time and number of iterations depends on choice of this set. Therefore, it was correctly suggested that comparison of the iterative methods should be based on use of the optimal parameters for each method.

A study by Shklyar and Arbel[8] developed discretization on double-staggered grids (DSG), and studied the accelerated convergence of the numerical simulation of incompressible flow in general curvilinear co-ordinates. In the present study, we analyzed a segregated numerical solution of the momentum-pressure-energy equations, with optimization of the iteration parameters. This paper is organized as follows: Section 2 briefly describes the mathematical model and algorithm, and testing of numerical benchmarks of the well known problem of natural convection in a squeezed cavity. In Section 3, optimization of the iterative parameters is demonstrated for the present case by use of the GA for optimization of the solution of linear systems of equations by stationary and nonstationary numerical methods.

2 DISCRETIZATION AND THE NUMERICAL ALGORITHM

We consider incompressible, steady-state flow. The reader may find in[9] the treatment of physical domain discretization with evaluation of metrics, discretization continuity and momentum equations, and velocity-pressure linkage a treatment which leads to generalization of the SIMPLE procedure for external and internal problems in the case of our double-staggered grid method for general curvilinear co-ordinates. Here we demonstrate discretized forms of the continuity, momentum and energy equations.

The mass flux, $\rho\boldsymbol{u}$ is integrated over a finite volume dV in physical space, bounded by lines of constant ξ and constant η.

$$\int\limits_S \rho\boldsymbol{u}\cdot\boldsymbol{n}dS = 0, \tag{1}$$

here $\boldsymbol{u}$ is a velocity vector, written as $\boldsymbol{u}$ = [u,v], with Cartesian components u, v, and unit vector $\boldsymbol{n} = [n_x, n_y]$ normal to the surface element dS. The integrations are performed by regarding all values as constant over each face of the control volume

$$U_{i+1,j} - U_{i,j} + V_{i,j+1} - V_{i,j} = 0, \tag{2}$$

The U, V in this equation, are the contravariant velocity components at point i, j of the domain mesh.

A momentum balance for finite volume dV gives:

$$\int\limits_S \rho\boldsymbol{uu}\cdot\boldsymbol{n}dS = \int\limits_S \boldsymbol{T}\cdot\boldsymbol{n}dS + \int\limits_V \rho\mathbf{g}dV, \tag{3}$$

where $\boldsymbol{g}$ is a gravity acceleration, $\boldsymbol{T}$ is the stress tensor, $\boldsymbol{T} = \boldsymbol{S}$ - $p\boldsymbol{I}$, $\boldsymbol{S}$ is the extra stress tensor

$$\mathbf{S} = \frac{2}{Re}\left[\boldsymbol{D} - \frac{1}{3}(\nabla\nu)\boldsymbol{I}\right], \tag{4}$$

here $\boldsymbol{D}$ is the rate of deformation tensor in incompressible flow.

The final form of the discretized momentum equation specific to grid point may written as (more details for staggered grid presented by Patankar and Spalding[1])

$$\begin{aligned} a_{i,j}u_{i,j} \;&=\; a_{i+1,j}u_{i+1,j} + a_{i-1,j}u_{i-1,j} + a_{i,j+1}u_{i,j+1} - \\ &\quad (p_{i,j} - p_{i-1,j})y_\eta d\eta + S_{i,j}, \end{aligned} \tag{5}$$

where $S_{i,j}$ includes second grid approximation, Shklyar and Arbel[9].

An energy balance for finite volume dS gives:

$$\int\limits_S c_p \rho \boldsymbol{u} T \cdot \boldsymbol{n} dS = \int\limits_S \boldsymbol{q} \cdot \boldsymbol{n} dS, \tag{6}$$

where T is the temperature, $\boldsymbol{q}$ is the heat flux. Discretization of (6) leads to the system of linear equations with nonsymmetry matrix.

Here we describe the numerical solution of the linear system of an incompressible flow equation by an iterative method with incomplete-block diagonal factorization preconditioner. This differs from the GCR-SIMPLE(R) method presented by Vuik et al.[10] in that the SIMPLE(R) method was seen as a distributive iterative method. The linear system (2), (6) gives rise to a saddle point problem, Benzi and Golub[11]:

$$A\mathbf{x} = \mathbf{b}, \tag{7}$$

here

$$A = \left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & 0 \end{array} \right], \quad \mathbf{x} = [\mathbf{u}, p]^T,$$

in which A_{11} is a discrete convection-diffusion operator, A_{12} and A_{21} are discrete divergence and gradient operator respectively. An approximate factorization of A, such as the incomplete LU factorization, is used for solving a linear system (7)

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}, \quad LU\mathbf{d}^{n+1} = \mathbf{r}, \quad \mathbf{x}^{n+1} = \mathbf{x} + \mathbf{d}^{n+1}, \tag{8}$$

for some vector $\mathbf{d}$ which occurs in the iterative method (we will establish the correlation between $\mathbf{d}$ and corrections in the SIMPLE method), n is an iteration number.

Incomplete block triangular factorization of the LU is:

$$LU = \left[\begin{array}{cc} A_{11} & 0 \\ A_{21} & S \end{array} \right] \left[\begin{array}{cc} I & A_{11}^{-1} A_{12} \\ 0 & I \end{array} \right],$$

here S is a Schur complement

$$S = -A_{21} A_{11}^{-1} A_{12}.$$

Thus, from the lower triangular system $L \left[\begin{array}{c} \mathbf{v}_1^{n+1} \\ \mathbf{v}_2^{n+1} \end{array} \right] = \left[\begin{array}{c} \mathbf{r}_1 \\ \mathbf{r}_2 \end{array} \right]$,

$$A_{11} \mathbf{v}_1^{n+1} = \mathbf{r}_1. \tag{9}$$

In terms of intermediate velocity $\mathbf{u}^* = \mathbf{u} + \mathbf{v}_1^{n+1}$, $\mathbf{r}_1 = \mathbf{b}_1 - A_{11}\mathbf{u} - A_{12}p$, and

$$A_{11}\mathbf{v}_1^{n+1} = \mathbf{r}_1 = \mathbf{b}_1 - A_{11}\mathbf{u} - A_{12}p = \mathbf{b}_1 - A_{11}(\mathbf{u}^* - \mathbf{v}_1^{n+1}) - A_{12}p, \tag{10}$$

equation (9) is transformed to a momentum equation for intermediate velocity:

$$A_{11}\mathbf{u}^* = \mathbf{b}_1 - A_{12}p. \tag{11}$$

Note, that conversely, approval, e.g. $\mathbf{u}^*$, $\mathbf{v}_1^{n+1} = \mathbf{u}^* - \mathbf{u}$ may be calculated from (11). The second equation from the lower triangular system is:

$$A_{21}A_{11}^{-1}A_{12}\mathbf{v}_2^{n+1} = A_{21}\mathbf{v}_1^{n+1} - \mathbf{r}_2. \tag{12}$$

We demonstrate that (12) equivalence to the pressure correction equation of the SIMPLE-like methods. First of all, from the upper triangular system $U\mathbf{d}^{n+1} = \mathbf{v}^{n+1}$ immediately should

$$\mathbf{d}_2^{n+1} = \mathbf{v}_2^{n+1} \tag{13}$$

$\mathbf{d}_2^{n+1}$ is an analog of the pressure correction p'; with $\mathbf{v}_1^{n+1} = \mathbf{u}^* - \mathbf{u}$ and $\mathbf{r}_2 = \mathbf{b}_2 - A_{21}\mathbf{u}^n$, from (12)

$$A_{21}A_{11}^{-1}A_{12}p' = A_{21}(\mathbf{u}^* - \mathbf{u}) - (\mathbf{b}_2 - A_{21}\mathbf{u}) = A_{21}\mathbf{u}^* - \mathbf{b}_2, \tag{14}$$

or,

$$A_{21}A_{11}^{-1}A_{12}p' = A_{21}\mathbf{u}^* - \mathbf{b}_2, \tag{15}$$

thus, pressure correction (p') is calculated from (15) and for $\mathbf{d}_1$ we shall cover the upper triangular system:

$$\mathbf{d}_1^{n+1} = \mathbf{v}_1^{n+1} - A_{11}^{-1}A_{12}\mathbf{d}_2^{n+1}. \tag{16}$$

In SIMPLE-like methods preconditioners D_{11}^{-1} are used for A_{11}^{-1} (for details see[10]. Rewrite (15) and (17), for velocity correction

$$\mathbf{d}_1^{n+1} = \mathbf{v}_1^{n+1} - D_{11}^{-1}A_{12}\mathbf{d}_2^{n+1}, \tag{17}$$

and

$$A_{21}D_{11}^{-1}A_{12}p' = A_{21}\mathbf{u}^* - \mathbf{b}_2. \tag{18}$$

Equation (18) is a pressure correction of the SIMPLE-like methods. Notice that use of D_{11}^{-1} and to maintain stability of the iterative procedure (8) leads to inputting of the pressure correction iterative parameter - α_p, so, that (8) transforms to

$$p^{n+1} = p + \alpha_p p'. \tag{19}$$

If we use the previous definition of $\mathbf{u}^*$, $\mathbf{d}^{n+1}$, and $\mathbf{d}_2^{n+1} = p'$, then

$$\mathbf{u}^{n+1} - \mathbf{u} = \mathbf{u}^* - \mathbf{u} - D_{11}^{-1} A_{12} p', \tag{20}$$

the conventional form for a velocity in the SIMPLE-like method is

$$\mathbf{u}^{n+1} = \mathbf{u}^* - D_{11}^{-1} A_{12} p'. \tag{21}$$

(Note that, conversely, approval, e.g. $\mathbf{d}_1^{n+1} = \mathbf{u}^{n+1} - \mathbf{u}$ may be calculated).

The single-grid DSG code was used for natural convection in a squeezed cavity. In order to preserve the practice of a five-point stencil discretization the cross-derivatives were calculated by linear interpolation and were grouped with other source terms. For the sake of brevity in the present paper we omit the mathematical treatment of the numerical method as applied to natural convection problems; we take the parameters to have been optimized. To facilitate the derivation of the finite-volume-finite-difference formulation, the physical domain in the x and y co-ordinate system is first discretized. Thus, we want to use the staggered-grid technique, which naturally offers conservation of mass, momentum and kinetic energy, and which avoids the decoupling of odd-even points.

2.1 Convergence criteria

During the numerical procedure, we monitored three norms of residuals; convergence of momentum was controlled by the residual norm,

$$||Residual_u||_2 = (\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |u_{i,j}^{n+1} - u_{i,j}^{n}|^2)^{1/2}, \tag{22}$$

the corresponding norm for the temperature field was

$$||Residual_T||_2 = (\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |T_{i,j}^{n+1} - T_{i,j}^{n}|^2)^{1/2}, \tag{23}$$

in which $N_x \times N_y$ is a grid dimension. The velocity fields which are obtained must be divergence-free at steady state (div$\boldsymbol{u}$=0), therefore it is natural to control the convergence of the continuity equation in its discretized form:

$$div\boldsymbol{u} = U_{i+1,j} - U_{i,j} + V_{i,j+1} - V_{i,j}, \tag{24}$$

this condition gives rise to the norm of velocity divergence

$$||Residual_{div\boldsymbol{u}}||_1 = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |div\boldsymbol{u}|. \tag{25}$$

2.2 Validation

The use of the methodology will be illustrated by means of numerical comparison with a benchmark solution. The numerical algorithm was tested with double precision on an IBM server, CPU X355, 2.66GHz. The control test case involved buoyancy-driven cavity flows. Appropriate benchmark solutions were described by Demirdžić et al.[12] (Figure 1(a)). Non-orthogonal grids were set up by inclining the sidewalls or by squeezing the cavity (Figure 1(b)).

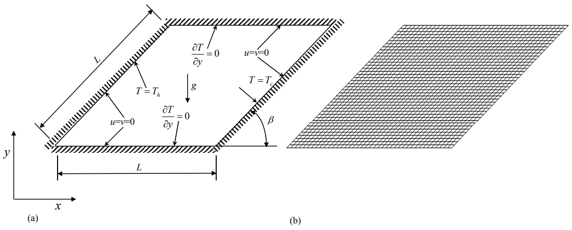
L
$\frac{\partial T}{\partial y}=0$
u=v=0
$T=T_h$
g
$T=T_c$
u=v=0
$\frac{\partial T}{\partial y}=0$
β
y
L
x
(a)
(b)

Figure 1: Squeezed cavity test case: (a) geometry and boundary conditions; (b) example of a coarse uniform grid.

The inclined walls were kept at constant temperatures T_h and T_c, respectively; the horizontal walls were assumed to be adiabatic; the inclination angle β was set to $\pi/4$. Scaling for the flow was characterized by the following reference quantities: $L = H$ for length; $V = (k_T/H)Ra^{0.5}$ for velocity, where k_T is a thermal conductivity, Ra is the Rayleigh number; and the temperature was scaled by $(T - T_c)/(T_h - T_c)$.

In the present study the steady-state numerical model was solved on a staggered grid by a finite-volume finite-difference method; the second-order central differences were used for space derivatives, as well as within the cavity. The solutions of this problem velocities, temperature, and rates of heat transfer have been obtained at Ra=10^6 and number of Pr=10. Non-uniform grids, symmetrically expanding from all walls towards the center line from, were applied by Demirdžić et al.[12], e.g., on the finest grid, with 224×192 control volumes, the smallest values δx=$L/467$ and δy=$L/454$. In our computation interpolation between the two boundaries was provided by:

$$x(\xi,\eta) = (1-s)x_{AB}(q_{AB}) + sx_{DC}(q_{DC}), \tag{26}$$

$$y(\xi,\eta) = (1-s)y_{AB}(q_{AB}) + sy_{DC}(q_{DC}), \tag{27}$$

in which AB and DC are boundary curves, and s and q are stretching functions. Non-uniform, symmetrical expansion from all walls towards the centerline, was implied by the stretching functions $q = s = 1 + \tanh(P(\eta - 0.5))/\tanh(P/2)/2$, in which P=2.95 for all grids, e.g., on the grid with a 512^2 control volume, the smallest δx was 0.00030397 (L=1.0), or 1/3289. Velocity boundary conditions were v=u=0 at the walls. The Nusselt number at the walls was calculated as:

$$Nu = \sqrt{\frac{x_\eta^2 + y_\eta^2}{J}} T_\xi, \tag{28}$$

variable derivatives at the wall (u_η, v_ξ, T_η and T_ξ) were calculated by one-sided second-order accurate formulas, such as, temperature derivative at the left(hot) wall is

$$T_\xi|_{\xi=0} = -y_\eta(3T_1 - T_2/3 - 8/3)/N_x/J, \tag{29}$$

and that at the right (cold) wall is

$$T_\xi|_{\xi=1} = -y_\eta(T_{N_x-1}/3 - 3T_{N_x})/N_x/J, \tag{30}$$

in which J is a Jacobian of the transformation. The values of the Nusselt number presented in Table 1, are more sensitive to the number of grid cells at the extreme points than to that at the middle points. The grid 512^2 with stretching function (26) and (27) permits displacement of 22 cells in the interval [1,0.9984] on the η coordinate (0.9984 is the extreme top η coordinate in which[12] presents their calculated results, in accordance with[12] this η coordinate is approximately the coordinate of the middle cell closest to 1.0).

Table 1: Profile of the local Nusselt number along the cold wall as predicted on various grids, Ra=10^6, Pr=10, inclination angle is $\pi/4$.

	Grids					
η	64×64	128×128	256×256	512×512	1024×1024	Demirdžić et al.[12]
1.4309×10^{-3}	0.049	0.142	0.173	0.181	0.18180	0.19757
6.5813×10^{-2}	8.219	8.137	8.116	8.111	8.11093	8.12495
9.6025×10^{-2}	10.746	10.687	10.671	10.668	10.667	10.674
1.2916×10^{-1}	12.089	12.058	12.050	12.129	12.048	12.051
1.5610×10^{-1}	12.452	12.433	12.428	12.426	12.426	12.428
1.7028×10^{-1}	12.491	12.475	12.471	12.470	12.470	12.471
1.8497×10^{-1}	12.461	12.447	12.444	12.443	12.443	12.445
2.3220×10^{-1}	12.092	12.084	12.082	12.082	12.082	12.083
3.0334×10^{-1}	11.210	11.207	11.207	11.207	11.207	11.208
5.2954×10^{-1}	8.106	8.110	8.111	8.111	8.1113	8.1111
8.1999×10^{-1}	4.315	4.320	4.322	4.322	4.3220	4.3212
9.6406×10^{-1}	2.016	2.032	2.036	2.037	2.0375	2.0338
9.7969×10^{-1}	1.871	1.895	1.902	1.904	1.9049	1.8995
9.9219×10^{-1}	2.209	2.147	2.163	2.175	2.1790	2.1651
9.9844×10^{-1}	4.445	4.425	3.743	3.433	3.4748	4.2622

3 RESULTS AND DISCUSSION

Linearization of the set of momentum-mass-energy conservation equations by SIMPLE-like methods gives rise to two equations for velocity components (u,v), a pressure-correction equation and an equation for temperature. One of the distinguishing feature of the SIMPLE-like method is the insertion of an iteration parameter for pressure. The number of iteration parameters depends also on the method of solution of the linearized system of equations. When methods with iteration parameters are used for the solution g.e., successive overrelaxation (SOR), or symmetric SOR (SSOR) methods, there are eight parameters in a 2D SIMPLE method: N_p - iteration number of pressure correction equation; N_u - iteration number of the u velocity component; α_u - u iteration parameter; N_v - iteration number of the v velocity component; α_v - v iteration parameter; N_t - iteration number of the temperature equation; ω_t - temperature iteration parameter; α_p from (19); and however in the bi-conjugate stabilized (BiCGSTAB) iterative procedure for the temperature equation, the temperature iteration parameter is not used, but in the preconditioned BiCGSTAB iterative method by means of a modified incomplete factorization (MILU(0)) a two additional parameters may be input and optimized.

A (GA) algorithm searches for a global minimum calculation time in a space of iteration numbers and parameters of the linear system; one which provides a means for unbiased comparison of computational methods. Unfortunately, a genetic algorithm takes a long time to search for a global minimum - a time that encompasses calculation time and the number of external iterations - for a grid containing more than $\approx$ 100k elements ($\approx$320×320 cells in a 2D case), therefore, in order to make an estimate of the numbers of iterations and iteration parameters needed for general cases, including a GA, we will examine at the convergence at various iteration parametres.

This section is organized as follows: in subsection 3.1 we briefly describe a genetic algorithm of the optimization of the search for the iteration parameters of the SIMPLE-like method. The subsequent subsections present the results of numerical computation of natural convection in a squeezed cavity by various iterative procedures: conjugate gradient (CG) with MILU(0) preconditioner for pressure correction equation; SOR and Gauss-Seidel (GS) methods applied to the momentum equation; BiCGSTAB with MILU(0) preconditioner for the temperature equation.

3.1 Genetic algorithm

GA mimics the biological evolutionary process and determine an optimal value, in parallel with a multi-point search procedure, based on crossover and mutation in genetics, Holland[13]. The search procedure for determining the optimal iteration parameters by means of the GA is as follows: the initial solution (population) is generated at random; some solutions (100 samples) are added to the original set of solutions from another set (200 samples); genetic operations, crossover and mutation, are applied to those individuals. As a results of the crossover, some individuals are newly created according to the crossover rate (70%), and other sorts of individuals are then newly generated according to the mutation rate (50%). As a results of these operations, new individuals are created; an optimal value can be obtained by repeating these procedures. In the GA, we considered $\|Residual_{div\boldsymbol{u}}\|_1 \leq 10^{-10}$, $\|Residual_u\|_2 \leq 10^{-5}$, $\|Residual_T\|_2 \leq 10^{-5}$.

3.2 Iterative procedures for the momentum equation

Application of the SIMPLE-like method to a non-linear momentum equation leads to an appropriate linear system with a non-symmetry, strictly diagonally dominant matrix A_{11} in (11). In the calculation procedure we assumed: the values of iteration numbers N_u of the linearized equation for u-component velocity were the same for number N_v of v-component velocity; and the iteration parameter α_u was equal to α_v.

Momentum equation in segregated algorithms of incompressible flow simulation is most commonly solved by the SOR and GS iterative methods. Table 2(a)-(d) presents the optimization results obtained by application of these two methods to the momentum equation. Optimal values of α_u found by GA in applying the SOR iterative method to the linearized momentum equation were always less then 1 (thus it is more accurate to describe this iterative procedure as successive underrelaxation, SUR) and N_u was changed from 5 to 6 (Table 2(a)). Table 2(b) shows how the calculation time increased by ≈2.5-3 times under the condition N_u=1. The optimal value of α_u was found be close to 1. Changing α_u to 1 leads to the Gauss-Seidel iterative method, because A_{11} is a strictly diagonally dominant, therefore the GS iteration converges for any initial guess x_0 of x. Table 2(c) shows that the calculation time with this value of α_u and iteration numbe r N_u=1 was slightly greater than that obtained with the SOR iterative procedure, Table 2(b). Optimization of the GS method for the momentum equation with N_u=var enables us to achieve the same calculation time as the SOR method, Table 2(b).

GMRES and GMRES with SGS preconditioner methods, Saad[14], were also applied to the momentum equation. The preconditioned GMRES method calculated the momentum equation with only one internal iteration but calculation time was more than twice that obtained with the GS method. GMRES without preconditioner gave the same internal iterations that were required with the GS method, but the calculation time was the same as that with the GMRES with SGS preconditioner method.

3.3 Implementation of the ILU(0), MILU(0) and RILU(0) preconditioners in the optimization process

Preconditioned CG and Bi-CGSTAB iterative procedures were applied to the pressure correction and the temperature equations, respectively. In this subsection we present some specific examples of a preconditioner that enables optimization of the calculation procedures.

The general problem of finding a preconditioner for a linear system (7) is to find a matrix M (preconditioner) with properties that ensure that M is a good approximation.

A suitable choice of M can significantly accelerate the convergence of the method. We used only the popular variants with explicit optimization opportunities. Let A correspond to the following 5-point difference stencil in the usual natural row-wise or column-wise ordering:

The entries in the above stencil are exactly the non-zero elements on the row of the matrix A corresponding to the *(i,j)*-th grid point. In what follows, it is convenient to compute an incomplete LD^{-1}U(0) or a modified incomplete factorization (MILU(0)) of system matrix, where L is the lower triangular matrix, D is the diagonal matrix and U is the upper triangular matrix and they correspond to the following difference stencils (for details see Chan and Van der Vorst[15]):

$$A = \begin{pmatrix} & t_{i,j} & \\ c_{i,j} & a_{i,j} & b_{i,j} \\ & s_{i,j} & \end{pmatrix}, L = \begin{pmatrix} & & \\ f_{i,j} & d_{i,j} & \\ & g_{i,j} & \end{pmatrix}, D = \begin{pmatrix} & & \\ & d_{i,j} & \\ & & \end{pmatrix}, U = \begin{pmatrix} & p_{i,j} & \\ & d_{i,j} & q_{i,j} \\ & & \end{pmatrix}.$$

Off-diagonal entries of the LU factors were calculated with the following simple formulas:

$$f_{i,j} = c_{i,j}, g_{i,j} = s_{i,j}, p_{i,j} = t_{i,j}, q_{i,j} = b_{i,j}. \tag{31}$$

Diagonal elements were calculated by the so-called *interpolated* (RILU(0)) version between ILU(0) and MILU(0):

$$d_{i,j} = a_{i,j} - c_{i,j}(b_{i,j} + \chi t_{i,j})/d_{i-1,j} - s_{i,j}(t_{i,j-1} + \chi b_{i,j-1})/d_{i,j-1}, \tag{32}$$

For χ=0 we have the standard ILU(0) factorization, whereas for χ=1 we have the modified MILU(0) factorization. The row sum lumping parameter method, which groups together all the elements that were dropped in the elimination process and adds them to the diagonal D, was also optimized. Furthermore, it was found (Chan and Van der Vorst[15], that an additional small term ch^2 might improve the convergence rate for elliptic problems:

$$d_{i,j} = a_{i,j} - c_{i,j}(b_{i,j} + \chi t_{i,j})/d_{i-1,j} - s_{i,j}(t_{i,j-1} + \chi b_{i,j-1})/d_{i,j-1} + ch^2, \tag{33}$$

A perturbation parameter c was also included in the list of optimized parameters.

3.4 Iterative procedures of pressure correction and temperature equations

The matrix A of the system (12) for the pressure correction equation is symmetry and strictly diagonally dominant, and therefore, in this short article we address the application of a more (for this case) effective preconditioned CG iterative procedure. Oliveira and Issa[16] gave one version of the CG solver with incomplete-Cholesky preconditioner for computations of buoyancy-driven flows in a rectangular cavity, the number of internal iterations N_p they required to solve a set of linear equations varied according to the specified tolerance for relative decay of the residuals.

In contrast to the system (12) for the pressure correction equation, matrix A of the temperature equation is a non-symmetry but also strictly diagonally dominant. We describe here the results of calculation of the preconditioned BiCGSTAB iterative procedure for the temperature equation.

In parallel with the impact of the choices of the iterative procedure and of the iteration parameters of the momentum equation on the calculated time, Table 2 demonstrates dependencies of the lumping parameter χ and a perturbation parameter c on the RILU(0) preconditioner. Attempts to perform the calculation with c=0 and χ=1, MILU(0) preconditioner failed. The minimum calculation time archived by inputting a small perturbational parameter, $c \approx 0.1$, Table 2(c). All the sub-tables of this table show that, as one would expect, the waste time, were cases in which ILU(0) preconditioners was used.

It has been argued sometimes that the methods of chosen to solve a pressure-correction equation or a temperature equation are critical from the point of view of convergence or calculation time. Use of the SIMPLE-like iterative procedure with optimal parameters will enable an understanding of the role of the choice of the iterative methods. The calculation time of the pressure-correction equation with the MILU(0)-CG method, and the calculation time of the temperature equation with the MILU(0)-BiCGSTAB method amount, respectively, to 18-17% of the overall calculation time in the iteration procedure which, the metric calculation time, in tern, for the strongly non-uniform grid, amounts to 30% of the total time. This finding supports the view that the calculation with optimal parameters by a SIMPLE-like method does not depend strongly on the choice of the solution methods for a set of linear equations, or for the pressure or temperature equations.

4 CONCLUSION

A methodology has been presented for optimization of the numerical solution of two-dimensional convection-diffusion problems. Use of this methodology was illustrated by comparison with the benchmark solution for natural convection in a squeezed cavity. A genetic algorithm used to solve problems of numerical optimization of calculation time provides a means of making an unbiased comparison in among a variety of iterative procedures for a large linear system. For the momentum equation it was found that a few iterations in the Gauss-Seidel method gave a similar results to the SOR method; calculation time with the preconditioned GMRES method applied to the momentum equation was more than twice that obtained with the GS method. The temperature equation was solved by the bi-conjugate gradient-stabilized method with MILU(0) preconditioner and the same preconditioner was used with a conjugate gradient iterative procedure for the pressure correction equation. Use of a MILU(0) preconditioner with various and optimal combinations of the perturbation constant c and lumping coefficient χ did not influence the calculation time strongly.

REFERENCES

[1] S. V. Patankar and D. B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *Int. J. Heat and Mass Transfer*, **15**, 1787–1806 (1972).

[2] C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics*, *Springer-Verlag*, Berlin (1988).

[3] M. Schäfer T. Lehnhäuser, Efficient discretization of pressure-correction equations on non-orthogonal grids, *Int. J. Numer. Meth. Fluids*, **42**, 211–231 (2003).

[4] R. H. Yen and C. H. Liu, Enahancement of the simple algorithm by an additional explicit corrector step, *Numer. Heat Tr., Part B*, **24**, 127–141 (1993).

[5] A. U. Chatwani and A. Turan, Improved pressure-velocity coupling algorithm based on minimization of global residual norm, *Numer. Heat Tr., Part B*, **20**, 115–123 (1991).

[6] W. Q. Tao, Z. Q. Qu, and Y. L. He, A novel segregated algorithm for incompressible fluid and heat transfer problems-CLEAR (coupled and linked equations algorithm revized) part 1: mathematical formulation and solution procedure, *Numer. Heat Tr., Part B*, **45**, 1–17 (2004).

[7] D. L. Sun, Z. G. Qu, Y. L. He, and W. Q. Tao, Perfomance analysis of IDEAL algorithm for three-dimensional incompressible fluid flow and heat transfer problems, *Int. J. Numer. Meth. Fluids*, **61**, 1132–1160 (2009).

[8] A. Shklyar and A. Arbel, Accelerated convergence of the numerical simulation of incompressible flow in general curvilinear co-ordinates by discretizations on the double-staggered grids, *Int. J. Numer. Meth. Fluids*, **57**, 205–238 (2008).

[9] A. Shklyar and A. Arbel, Numerical method for calculation of the incompressible flow in general curvilinear co-ordinates with double staggered grid, *Int. J. Numer. Meth. Fluids*, **41**, 1273–1294 (2003).

[10] C. Vuik, A. Saghir, and G. P. Boerstoler, The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces, *Int. J. Numer. Meth. Fluids*, **33**, 1027–1040 (2000).

[11] M. Benzi and G. H. Golub, Numerical solution of saddle point problems, *Acta Numerica* **14**, 1–137 (2005).

[12] I. Demirdžić, Ž. Lilek, and M. Perić, Fluid flow and heat transfer test problems for non-orthogonal grids: benchmark solutions, *Int. J. Numer. Meth. Fluids*, **15**, 329–354 (1992).

[13] J. H. Holland, Genetics algorithms, *Sci. Am.*, **7**, 66–72 (1992).

[14] Y. Saad, *Iterative method for sparse linear system*, SIAM, (2003).

[15] T. C. Chan and H. A. Van der Vorst, Approximate and incomplete factorization. In *Parallel Numerical Algorithms, edited by D.E. Keys, A. Sameh, and V. Venkatakrishnan, ICASE/LaRC Interdisciplinary Series in Science and Engineering*, Kluwer Academic, Dordrecht, 167–202 (1997).

[16] P. J. Oliveira and R. I. Issa, An improved PISO algorithm for the computation of buoyancy-driven flows, *Numer. Heat Tr., Part B*, **40**, 473–493 (2001).

Table 2: Optimal iteration parameters of the iteration procedures; RILU(0)-CG iterative method for pressure correction equation with various combinations of c_p and χ_p; SOR and GS iterative methods for the momentum equation; RILU(0)-BiCGSTAB iterative method for the temperature equation with various combination of c_T and χ_T; C_t is a computing time (s).

(a) SOR iterative method for momentum equation, (128^2 cells)

C_t	N_{ex}	N_p	N_v	N_t	ω_v	α_p	c_p	χ_p	c_T	χ_T
21	348	11	5	3	0.936	0.108	0	0	0	0
15	346	5	6	1	0.867	0.105	0.116	1	0.096	1
21	325	14	5	3	0.923	0.112	0.276	0	0.761	0
14	338	4	5	1	0.953	0.115	0.054	0.993	0.042	0.998

(b) SOR iterative method for momentum equation, (N_v=1, 128^2 cells)

C_t	N_{ex}	N_p	N_t	ω_v	α_p	c_p	χ_p	c_T	χ_T
69	1421	6	1	0.999	0.427	0	0	0	0
56	1284	5	1	1.094	0.261	0.355	1	0.955	1
49	850	9	2	1.195	0.171	0.572	0	0.433	0
46	1088	4	1	0.997	0.475	0.016	0.996	0.263	0.843

(c) GS iterative method for momentum equation, (N_v=1, ω_v=1, 128^2 cells)

C_t	N_{ex}	N_p	N_t	α_p	c_p	χ_p	c_T	χ_T
69	1420	6	1	0.475	0	0	0	0
60	1265	5	1	0.422	0.179	1	0.903	1
69	1455	6	1	0.446	0.397	0	0.732	0
58	1173	7	1	0.198	0.783	0.775	0.413	0.861

(d) GS iterative method for momentum equation, (ω_v=1, c_p=c_T=$const$, χ_p=χ_T=1.0, 128^2 cells)

C_t	N_{ex}	N_p	N_v	N_t	α_p	c_p	c_T
21	354	12	4	1	0.128	0.3	0.3
19	382	6	4	1	0.123	0.2	0.2
18	354	7	4	1	0.112	0.1	0.1
24	368	11	4	2	0.131	0.01	0.01

(e) GS iterative method for momentum equation, (c_p=c_T=0.1, χ_p=χ_T=1.0)

Grid	C_t	N_{ex}	N_p	N_v	N_t	α_p
64^2	2	206	4	3	2	0.248
128^2	18	357	6	4	1	0.113
256^2	144	671	7	1	8	0.068
512^2	1504	1403	5	1	4	0.054