Computers and Structures 87 (2009) 710–720

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

On pressure–velocity coupled time-integration of incompressible Navier–Stokes equations using direct inversion of Stokes operator or accelerated multigrid technique

Yuri Feldman *, Alexander Yu. Gelfgat

School of Mechanical Engineering, Faculty of Engineering, Tel-Aviv University, Ramat Aviv, Tel Aviv 69978, Israel

ARTICLE INFO

Article history:
Received 4 June 2008
Accepted 22 January 2009
Available online 26 February 2009

Keywords:
Multigrid
Free convection
Pressure–velocity coupling

ABSTRACT

Two fully pressure–velocity coupled approaches to time-integration of two- and three-dimensional Navier–Stokes equations discretized by finite volume method are proposed and verified. The first approach utilizes a direct sparse matrix solver to inverse the Stokes operator. In the second approach a multigrid iterative solver is accelerated by a modification of CLGS smoother that allows for derivation of an analytical solution for velocity and pressure corrections belonging to a whole row or column of finite volumes. Both approaches are tested by two- and three-dimensional natural convection benchmark problems. It is concluded that the analytical solution accelerated CLGS technique (ASA-CLGS) can be considered as a promising tool for solution of time-dependent three-dimensional fluid dynamics problems.

1. Introduction

Reliable and robust numerical solution of unsteady three-dimensional fluid dynamics problems at large Reynolds numbers remains a challenging task of modern computational fluid dynamics. Most of numerical approaches to time-dependent incompressible fluid dynamics utilize various forms of decoupling of pressure and velocity calculations. Discussing possible further development of CFD methods a recently published review paper [1] argued that with the increase of available computer memory pressure–velocity coupled approaches can become more effective. At the same time it was stressed that development of such approaches only started and robust fully coupled numerical procedures are yet to be developed.

In the present study we introduce two different approaches to time-integration of coupled incompressible Navier–Stokes equations discretized by the finite volume method. In particular, we are interested in calculations at parameter values close to bifurcation points, where correct results cannot be obtained without a sufficient accuracy of both discretization in space and integration in time (see, e.g. [2,3]).

The first approach is an extension of that of [3], where a direct sparse matrix solver was used for inverse of the Jacobian matrix, thus completing the most difficult part of solution of coupled steady Navier–Stokes equations by Newton iteration. Apparently, an inverse of the Stokes operator can be considered as a straightforward way to solve unsteady Navier–Stokes equations with a full coupling between pressure and velocity. However, to the best of our knowledge, such a direct approach has not been yet realized numerically. Here the direct sparse matrix solver is applied for inverse of the Stokes operator or, more precisely, for calculation of *LU* decomposition of the matrix approximating the Stokes operator. The consequent advancement in time is reduced to a series of back- and forward-substitutions, which also are carried out with utilization of the sparseness. It is shown that this approach yields a competitive efficiency for two-dimensional problems. For three-dimensional problems this approach is strongly restricted by available computer memory, which did not allow us to test it on fine enough grids.

The second approach originates from multigrid techniques proposed in [4,5] and extended in [6,7]. The Collective Line Gauss–Seidel (CLGS) approach of [6,7] considers a row (column) of finite volumes as a main block of a Gauss–Seidel-type iteration, which leads to a block tridiagonal linear equation system to be solved at each sub-iteration of a multigrid algorithm. In the present paper we show that a slight modification of the formulation of [6,7] allows one to derive an analytical solution for velocity and pressure corrections belonging to a whole row (column) of finite volumes. As a result the CPU time needed to carry out a single sub-iteration of the Gauss–Seidel type is significantly reduced, which speeds up the whole multigrid procedure and reduces the computational cost of a time step.

In the following we describe both approaches and a series of test computations that tests their efficiency. Two- and three-dimensional natural convection benchmarks are chosen to perform

* Corresponding author.
E-mail address: yurifeld@post.tau.ac.il (Y. Feldman).

0045-7949/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2009.01.013

these preliminary tests. Characteristic CPU times and consumed computer memory needed to perform the computations are reported.

2. Formulation of the problem

For a benchmark problem we consider natural convection in a three-dimensional differentially heated box of length W, height H and width D, sketched in Fig. 1. The box aspect ratio is $A = H/W$ and the width ratio is $B = D/W$. No-slip boundary conditions are posed on all the boundaries. Two opposite vertical walls of the box are maintained at different constant temperatures whereas all four remaining walls are either perfectly insulating or perfectly conducting. The flow is described by the momentum, energy and continuity equations in the Boussinesq approximation. Following Ref. [2] the dimensionless governing equations are:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\nabla p + \frac{1}{Gr^{0.5}}\nabla^2\boldsymbol{u} + \theta\vec{e_z} \tag{2}$$

$$\frac{\partial \theta}{\partial t} + (\boldsymbol{u}\cdot\nabla)\theta = \frac{1}{PrGr^{0.5}}\nabla^2\theta \tag{3}$$

where $\boldsymbol{u} = (u,v,w)$, p, t, and θ are the dimensionless velocity, pressure, time and temperature, respectively, and $\vec{e}_z$ is the unit vector in z-direction. The equations are rendered dimensionless using the scales W, $U = \sqrt{g\beta W\Delta T}$, $t = W/U$, and $P = \rho U^2$ for length, velocity, time and pressure, respectively. Here ρ is the mass density, g is the gravitational acceleration, β is the isobaric coefficient of thermal expansion, and $\Delta T = T_H - T_C$ is temperature difference between the hot and cold walls. The dimensionless temperature θ is defined as $\theta = (T - T_C)/\Delta T$. The Grashof and Prandtl numbers are $Gr = g\beta\Delta T W^3/\nu^2$ and $Pr = \nu/\alpha$, where ν is the kinematic viscosity, and α is the thermal diffusivity.

3. Numerical method

3.1. Discretization in time and space

The time derivative in the momentum and the energy equations is approximated by the second order backward finite difference:

$$\frac{\partial f^{n+1}}{\partial t} = \frac{3f^{n+1} - 4f^n + f^{n-1}}{2\Delta t} + O(\Delta t^2) \tag{4}$$

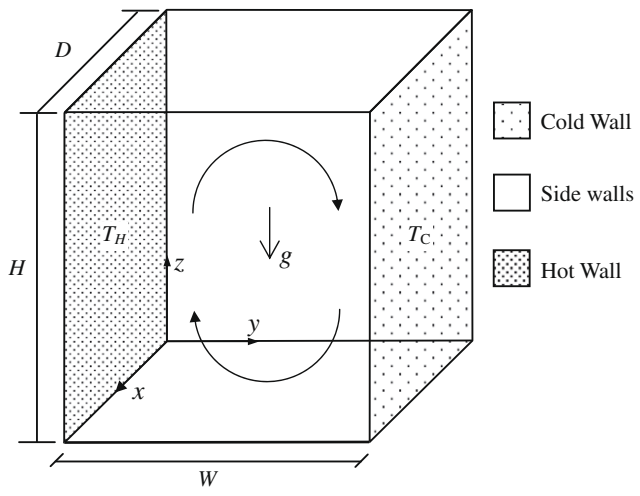

Fig. 1. Differentially heated cavity – physical model and coordinate system.

In both models diffusion and pressure terms are treated implicitly, while all other terms are treated explicitly. In this formulation the advancement of velocity in time governed by the continuity (1) and momentum Eq. (2) is reduced to the inverse of the Stokes operator. The energy Eq. (3) is decoupled from the continuity and momentum Eqs. (1) and (2). The time-integration of the whole system (1)–(3) is carried out in two separate steps:

(1) Assume initial velocity distributions and solve the energy Eq. (3) for $\theta^{(n+1)}$:

$$\frac{1}{PrGr^{0.5}}\nabla^2\theta^{(n+1)} - \frac{3}{2\Delta t}\theta^{(n+1)} = [(\boldsymbol{u}\cdot\nabla)\theta]^n + \frac{1}{2\Delta t}\left(-4\theta^n + \theta^{(n-1)}\right) \tag{5}$$

(2) Substitute the obtained temperature into the momentum equations and solve the coupled continuity and momentum Eqs. (1) and (2) for $\boldsymbol{u}^{(n+1)}$ and $p^{(n+1)}$:

$$\nabla \cdot \boldsymbol{u}^{(n+1)} = 0 \tag{6}$$

$$\frac{1}{Gr^{0.5}}\nabla^2\boldsymbol{u}^{(n+1)} - \frac{3}{2\Delta\tau}\boldsymbol{u}^{(n+1)} - \nabla p^{(n+1)} = [(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}]^n + \frac{1}{2\Delta t}\left(-4\boldsymbol{u}^n + \boldsymbol{u}^{(n-1)}\right) - \theta^{(n+1)}\vec{e}_z \tag{7}$$

Due to explicit representation of the non-linear terms, the time-integration scheme is subject to restrictions in the time step size. Thus, the time step must satisfy the usual Courant number criterion for purely explicit schemes [8]:

$$C_x = |u|\Delta t/\Delta x < 1.0, \quad C_y = |v|\Delta t/\Delta y < 1.0, \quad C_z = |w|\Delta t/\Delta z < 1.0 \tag{8}$$

and the requirement that the momentum and thermal energy must not diffuse through more than one cell in one time step:

$$(\nu,\alpha)\Delta t\left[1/(\Delta x)^2 + 1/(\Delta y)^2 + 1/(\Delta z)^2\right] \le \frac{1}{2} \tag{9}$$

A number of numerical tests showed that for grids having at least 100 grid nodes in the shortest direction the time steps $\Delta t = 1\times10^{-2}$ and $\Delta t = 1\times10^{-3}$ converge to steady states independently on initial conditions for regular and stretched grids, respectively.

Eqs. (5)–(7) are discretized using conservative second order finite volume schemes [9] on a staggered mesh (Fig. 2). Following notations of [9] we represent the finite volume discretization of Eqs. (5)–(7) as

$$\left(\frac{1}{PrGr^{0.5}}a_P^\theta - \frac{3}{2\Delta t}\right)\theta_P^{n+1} + \frac{1}{PrGr^{0.5}}\sum_{nb}a_{nb}^\theta\theta_{nb}^{n+1} = RHS_\theta^n \tag{10}$$

$$\left(u_e^{n+1} - u_w^{n+1}\right)/(x_e - x_w) + \left(v_n^{n+1} - v_s^{n+1}\right)/(y_n - y_s) + \left(w_u^{n+1} - w_d^{n+1}\right)/(z_u - z_d) = 0 \tag{11}$$

$$\left(\frac{1}{Gr^{0.5}}a_e^u - \frac{3}{2\Delta t}\right)u_e^{n+1} + \frac{1}{Gr^{0.5}}\sum_{nb}a_{nb}^u u_{nb}^{n+1} - \frac{p_E^{n+1} - p_P^{n+1}}{x_E - x_P} = RHS_u^n \tag{12.1}$$

$$\left(\frac{1}{Gr^{0.5}}a_n^v - \frac{3}{2\Delta t}\right)v_n^{n+1} + \frac{1}{Gr^{0.5}}\sum_{nb}a_{nb}^v v_{nb}^{n+1} - \frac{p_N^{n+1} - p_P^{n+1}}{y_N - y_P} = RHS_v^n \tag{12.2}$$

$$\left(\frac{1}{Gr^{0.5}}a_d^u - \frac{3}{2\Delta t}\right)w_u^{n+1} + \frac{1}{Gr^{0.5}}\sum_{nb}a_{nb}^v w_{nb}^{n+1} - \frac{p_U^{n+1} - p_P^{n+1}}{z_U - z_P} = RHS_w^n \tag{12.3}$$

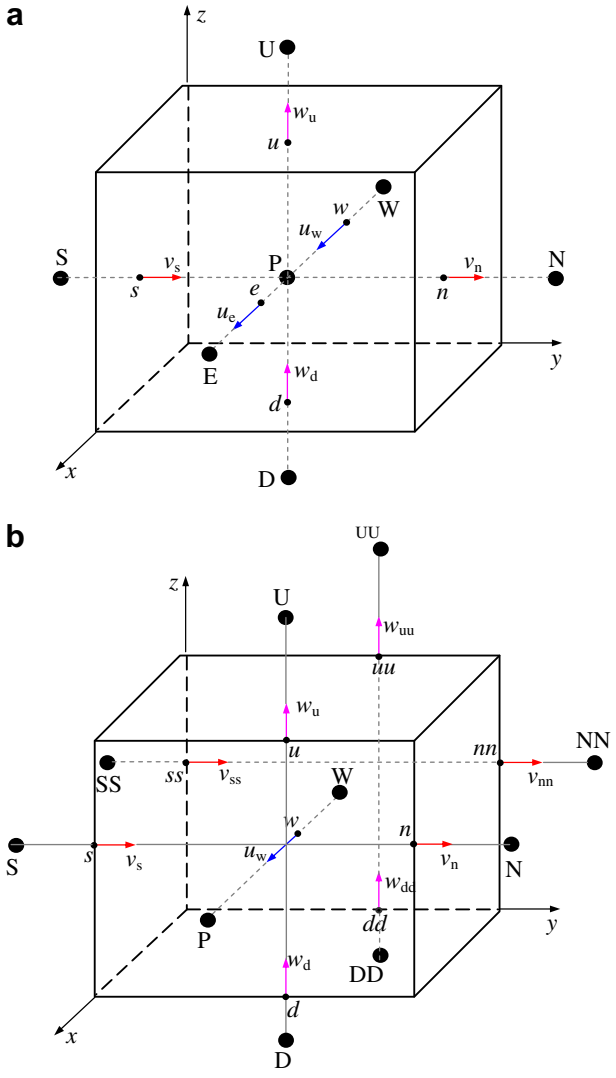

Fig. 2. Staggered grid arrangement (a) control volume for scalar fields and (b) control volume for u velocity component.

Expressions for the coefficients a_p, a_e, a_n, a_d, a_{nb} for each unknown function are derived from the finite volume discretization of the Laplace operator, while expressions for the right hand sides $RHS^n_\theta, RHS^n_u, RHS^n_v, RHS^n_w$ contain discretization of all other terms and are calculated using functions computed at a previous time step. For the reasons described in [3] the conservative approximation of convective terms is used. At the same time, the two approaches described below can be applied for any upwinding or higher-order scheme approximating the convective terms. Note, that for a fully coupled solution on a staggered mesh one does not need to know pressure values at the boundaries, which removes the well-known problem of pressure boundary conditions.

3.2. The full pressure coupled direct (FPCD) solution

This approach is based on *LU*-factorization of the Stokes operator defined on the whole computational domain. The matrix approximating the Stokes operator is comprised from left hand sides of Eqs. (11) and (12). Formally, to advance in a single time step it is necessary to multiply the r.h.s of Eqs. (11) and (12) by the inverse of this matrix. To evaluate the latter matrix-vector product we compute the *LU*-factorization of the matrix in the beginning of the time-integration process. The computation is carried out by a direct multifrontal sparse solver (we use the MUMPS[1] package). The advancement of the velocity and pressure fields in one time step is then reduced to one backward and one forward substitution procedures, which are also realized for sparse triangular L and U matrices. Due to effective utilization of the matrix sparseness both *LU*-factorization and backward/forward-substitutions are relatively fast. The characteristic CPU times and memory requirements needed for the *LU*-factorization are studied in Ref. [3]. The characteristic times for the whole time-integration process are given below.

The decoupled energy Eq. (3) was solved by the Bi-CGStab algorithm [10]. In this way we decrease computer memory usage in about 20% comparing with solving the same equation by the MUMPS package. It was found that for the energy equation the iterative Bi-CGSTAB procedure and the direct solution using the MUMPS package consume almost same CPU time. For two-dimensional simulations the characteristic CPU time needed for a single time step per one node and per one CPU is of order 5×10^{-3} m s on a single AMD 2.4 GHz processor, which is close to the best result among 32 studies reported during a special MIT2001 session [11]. It should be noted, however, that being very effective for two-dimensional problems, the FPCD algorithm is extremely memory demanding for three-dimensional calculations. Thus, for 32-bit integers it is restricted to approximately 40^3 grids, which is insufficient for obtaining quantitatively reliable results [3]. Nevertheless, taking into account a rapid increase of availability of computer memory it should not be immediately disregarded.

3.3. The multigrid solution

Following previous studies cited below we consider Eqs. (11) and (12) for corrections of pressure and velocity, that are defined as difference of the corresponding values at two consequent time steps and are denoted by prime.

The multigrid approach used in the present work is a modification of the method originally developed and tested by Vanka for square [4] and cubic [5] lid-driving cavities. Linearity of the Stokes operator allows us to use the correction scheme (CS) of Ref. [12] and the Coupled Line Gauss–Seidel smoother (CLGS) developed in [6]. An additional description of this method can be found in [7]. A typical V-cycle technique [13] is used for the multigrid iterations. Contrary to so called self-controlling algorithms detailed in [4,5] the V-cycle technique is controlled only by rate of convergence and does not require any additional accommodative criteria for switching from one grid to another [12]. Therefore it is expected to be more efficient for analysis of unsteady flows where small time steps are necessarily used, so that variations between two sequential time steps are small. The rate of convergence on each grid level has been defined as $|(\|R_{m+1}\|_{L2} - \|R_m\|_{L2})/\|R_{m+1}\|_{L2}| > \eta$ where R_{m+1} and R_m are residuals at iterations $m+1$ and m, respectively, and η is a convergence threshold. In the present study the convergence rate at a specific grid level is supposed to be slow if maximal value of all threshold rates obtained for both the continuity and the momentum equations is greater than $\eta = 0.5$. To define the algorithm stopping criterion on the finest grid the point-wise absolute difference between iterations $m+1$ and m is firstly obtained at each point of the computational domain. The difference is defined by $\left(|v^M_{m+1} - v^M_m|/|v^M_m|\right)_{(i,j)}$ if $v^M_m > 0.1$ and $\left(|v^M_{k+1} - v^M_k|\right)_{(i,j)}$ otherwise. Such definition yields representative values of the iterations difference for the wide range of the v^M. At a given time step the overall multigrid iteration process is stopped when a maximal value of all obtained local differences is less then 10^{-7}.

[1] See http://www.enseeiht.fr/apo/MUMPS/ or http://grall.ens-lyon.fr/MUMPS/.

3.3.1. Restriction and prolongation

Definitions of the restriction I_{k+1}^{k} and prolongation I_{k}^{k+1} operators on uniform staggered grids are given in [4,5]. In the present study these definitions are generalized in terms of coordinate differences between fine and coarse grids that make them applicable for the non-uniform and strongly stretched grids characterized by an unfixed mesh ratio between coarse and fine grid steps. For prolongation of the pressure corrections at the nodes adjacent to domain boundaries we require the pressure derivative in the normal to the boundary direction to vanish. It is emphasized that for the finest grid of the V-cycle, for which the actual solution is being sought, no prolongation and no boundary conditions for pressure are needed.

3.3.2. The line-wise smoother

We start the description of the smoother that we propose here from recalling the characteristic matrix for the coupled pressure–velocity corrections for the Symmetric Coupled Gauss–Seidel smoother (SCGS) developed in [4]. This matrix is illustrated in Fig. 3 and for the discretized Eqs. (10)–(12) its coefficients are $A_1=\frac{1}{Gr^{0.5}}, a_w^u-\frac{3}{2\Delta t}, A_3=\frac{1}{Gr^{0.5}}a_e^u-\frac{3}{2\Delta t}, A_5=\frac{1}{Gr^{0.5}}a_w^u-\frac{3}{2\Delta t}, A_9=\frac{1}{Gr^{0.5}}a_n^v-\frac{3}{2\Delta t}, A_7=-1/(x_e-x_w), A_8=-1/(y_n-y_s), A_2=-1/(x_P-x_W), A_4=1/(x_E-x_P), A_6=-1/(y_P-y_S), A_{10}=1/(y_N-y_P)$.

It can be inversed analytically, so that required corrections are easily computed and are immediately added to the values of the current solution. Under-relaxation is implemented by adding a fraction of the calculated correction to the current variables.

The cell-wise implementation described above is known to have poor convergence characteristics when grids are stretched or the flow is largely unidirectional [7]. An improvement can be achieved by an update of variables for the entire line (column or row). This concept, called Collective Lines Gauss–Seidel smoother (CLGS), was proposed by Zeng and Wesseling [6] who used a block-tridiagonal structure of the matrix assembled from the matrices corresponding to single lines of finite volumes. The corresponding linear equations system was solved in [6] by the standard backward–forward substitution. Performance of CLGS and SCGS smoothers was later compared by Paisley [7] who showed that the CLGS smoother is more efficient. However, this approach did not attract much attention, possibly because the pressure–velocity decoupling methods were still more efficient, as it was shown for the SIMPLE algorithm in [7].

To describe our modification of CLGS smoother we recall the corresponding system of algebraic equations assembled for pressure and velocity corrections belonging to a column of finite volumes. For brevity only 2D case is described. Assuming that pressure values are located in the nodes (i,j), while the vertical and horizontal velocities are located in the nodes $(i,j+1/2)$, and $(i+1/2,j)$, respectively, the correction equations are considered in the following form

$$
\begin{aligned}
&\cdots\\
&A^{(x)}_{i+1/2,j}u'_{i+1/2,j}+B^{(x)}_{i+1/2,j}p'_{i,j}=R^{(x)}_{i+1/2,j}\\
&A^{(x)}_{i-1/2,j}u'_{i-1/2,j}-B^{(x)}_{i-1/2,j}p'_{i,j}=R^{(x)}_{i-1/2,j}\\
&A^{(y)}_{i,j+1/2}v'_{ij+1/2}-B^{(y)}_{i,j+1/2}\left(p'_{i,j+1}-p'_{i,j}\right)=R^{(y)}_{i,j+1/2}\\
&A^{(x)}_{i,j}\left(u'_{i+1/2,j}-u'_{i-1/2,j}\right)+A^{(y)}_{i,j}\left(v'_{i,j+1/2}-v'_{i,j-1/2}\right)=0\\
&\cdots
\end{aligned}
\tag{13}
$$

All the coefficients in Eqs. (13) are derived from the discretized Eqs. (11) and (12). The vertically arranged neighbor volumes are connected through the values $v'_{i,j+1/2}$ and $p'_{i,j+1}$, so that the resulting system is block-tridiagonal and can be inversed by, e.g. block *LU*-factorization [6]. However, numerical experiments of [7] showed that this algorithm performs slower than SIMPLE. To make the matrix inverse faster we propose to modify the Eq. (13) by assignment of the term with $p'_{i,j+1}$ to the right hand side. This reads

$$
\begin{aligned}
&\cdots\\
&A^{(x)}_{i+1/2,j}u'_{i+1/2,j}+B^{(x)}_{i+1/2,j}p'_{i,j}=R^{(x)}_{i+1/2,j}\\
&A^{(x)}_{i-1/2,j}u'_{i-1/2,j}-B^{(x)}_{i-1/2,j}p'_{i,j}=R^{(x)}_{i-1/2,j}\\
&A^{(y)}_{i,j+1/2}v'_{ij+1/2}+B^{(y)}_{i,j+1/2}p'_{i,j}=\tilde{R}^{(y)}_{i,j+1/2}\\
&A^{(x)}_{i,j}(u'_{i+1/2,j}-u'_{i-1/2,j})+A^{(y)}_{i,j}(v'_{i,j+1/2}-v'_{i,j-1/2})=0\\
&\cdots
\end{aligned}
\tag{14}
$$

where $\tilde{R}^{(y)}_{i,j+1/2}=R^{(y)}_{i,j+1/2}+B^{(y)}_{i,j+1/2}p'_{i,j+1}$ and the value of $p'_{i,j+1}$ is taken from the previous iteration. The resulting equations for the column of finite volumes remain linked by the values of $v'_{i,j+1/2}$, but now one can derive a Thomas-like algorithm for calculation of an analytical solution of this equation system. This algorithm is illustrated in Fig. 4 and is described below.

For a 2D problem a typical column (row) consists of L finite volumes denoted by V_I, $I=1,2,\ldots,L$ (Fig. 4). Each volume contains 5 unknown correction values (four for velocity components and one for pressure). In total, the entire column contains k unknowns. It is assumed that the boundary conditions yield the values of v'_1 and v'_k. We start from equations written for the upper volume V_L (Fig. 4). The value of v'_k is known and the values of u'_{k-2}, u'_{k-3} can be expressed via the value of p'_{k-1} using first two equations of the system (14). Then the expressions for v'_k, u'_{k-2}, u'_{k-3} are substituted into the last equation (the continuity), which yields a linear link between p'_{k-1} and v'_{k-4}. Then p'_{k-1} can be expressed via v'_{k-4} in the following form

$$
p'_{k-1}=(c_1^L v'_{k-4}+R^L_{k-1}+c_2^L R^L_{k-2}+c_3^L R^L_{k-3}+c_4^L R^L_{k-4})/c_5^L \tag{15}
$$

Now, assuming that v_{k-4} is known we can repeat this procedure for the volume L-1 and to continue till the volume 1. At the volume 1 the value of v'_1 is known and therefore the value p'_4 can be calculated using equation similar to (15). Then, the values of v'_5, u'_2 and u'_3 are readily expressed via p'_4 as

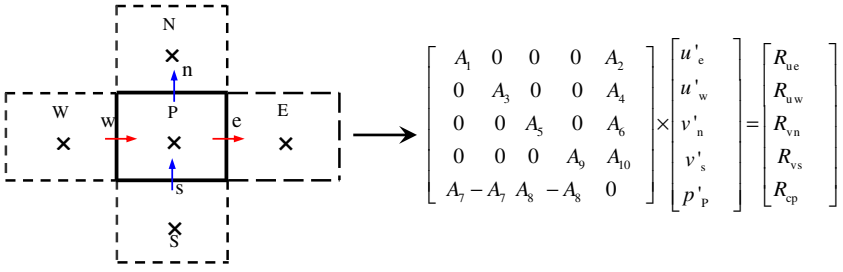

Fig. 3. Matrix corresponding to a single finite volume for Symmetrical Coupled Gauss–Seidel (SCGS) smoother [4].

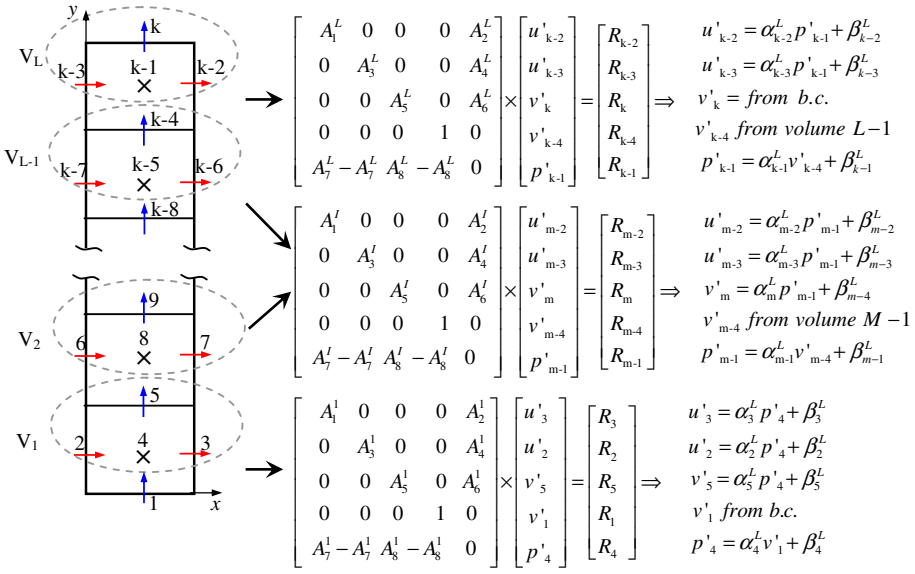

Fig. 4. A schematic description of ASA-CLGS smoother.

$$\begin{bmatrix} v'_5 \\ u'_2 \\ u'_3 \end{bmatrix} = \begin{bmatrix} c_6^1 \\ c_7^1 \\ c_8^1 \end{bmatrix} \times p'_4 + \begin{bmatrix} c_9^1 R_5^L \\ c_{10}^1 R_2^L \\ c_{11}^1 R_3^L \end{bmatrix} \tag{16}$$

Now, having the value of v'_5 we calculate the pressure and velocity correction values belonging to the volume 2 and continue to the volume L.

To finalize, the proposed algorithm consists of two stages. At the first stage we calculate the coefficients c_i^l and at the second stage obtain all the current corrections. It is emphasized that for a constant time step the coefficients c_i^l remain unchanged for all the iterations and time steps, which makes the algorithm considerably fast.

An exact analytical estimation of the developed algorithm overall complexity is barely possible since, as for any iterative algorithm, its convergence rate is problem dependent. Nevertheless, a quantitative estimation of the algorithm complexity can be obtained by applying it to representative benchmark problems, known to have periodical solutions (see Section 4.5 for more details). Here we present some general remarks illustrating advantages of the developed smoother. Since all the coefficients c_i are calculated only once at the beginning of the process the CPU time consumed for their calculation is negligible. To proceed with Eqs. (15) and (16) we need 11 multiplications and divisions and 7 summations to calculate 4 corrections for a single volume. Note that the fifth correction is yielded by the result obtained for the previous finite volume or from a boundary condition. Thus, we need about $5N$ operations to compute N corrections for a single finite volume and thus the same order of operations amount for the whole line of volumes. Note that the assembled matrix corresponding to the whole line of volumes is six-diagonal. The calculation of its solution, however, needs the same amount of operations as Thomas algorithm applied to a three-diagonal matrix. It should be emphasized, that the above estimate does not take into account additional mathematical operations needed to obtain residuals R_i^l (Fig. 4), which must be updated at each sub-iteration and will inevitably increase the overall complexity of the algorithm. This, however, is a disadvantage of all multigrid algorithms based on the Gauss–Seidel smoother.

The developed solver is easily extended to three-dimensional geometry by treating a line of three-dimensional finite volumes. Consider a single column of L finite volumes. For a finite volume I ($I = 1,2,\ldots,L$) we use notations of Fig. 2. Then the pressure and velocity corrections belonging to this volume are calculated as a solution of:

$$\begin{aligned} &\ldots \\ &A_e^{(x)} u'_e + B_e^{(x)} p'_p = R_e^{(x)} \\ &A_w^{(x)} u'_w - B_w^{(x)} p'_p = R_w^{(x)} \\ &A_n^{(y)} v'_n + B_n^{(y)} p'_p = R_n^{(x)} \\ &A_s^{(y)} u'_s - B_s^{(y)} p'_p = R_s^{(x)} \\ &A_u^{(z)} w'_u + B_u^{(z)} p'_p = \tilde{R}_u^{(z)} \\ &A_p^{(x)} (u'_e - u'_w) + A_p^{(y)} (v'_n - v'_s) + A_p^{(z)} (w'_u - w'_d) = 0 \\ &\ldots \end{aligned} \tag{17}$$

A similar solution procedure yields

$$p'_p = \left(c_1^l w'_d + R_p^l + c_2^l R_e^l + c_3^l R_w^l + c_4^l R_n^l + c_5^l R_s^l + c_6^l R_d^l \right) / c_7^l \tag{18}$$

$$\begin{bmatrix} w'_d \\ v'_s \\ v'_n \\ u'_w \\ u'_e \end{bmatrix} = \begin{bmatrix} c_8^l \\ c_9^l \\ c_{10}^l \\ c_{11}^l \\ c_{12}^l \end{bmatrix} \times p'_p + \begin{bmatrix} c_{13}^l R_d^l \\ c_{14}^l R_s^l \\ c_{15}^l R_n^l \\ c_{16}^l R_w^l \\ c_{17}^l R_e^l \end{bmatrix} \tag{19}$$

As in the two-dimensional case all the coefficients c_i are calculated only once at the beginning of the process. To calculate 6 corrections for a single volume we need 17 multiplications and divisions and 11 summations, which results in about $5N$ operations to compute N corrections located in one column (row). Again, the seventh correction is yielded by the finite volume proceeded before or by a boundary condition. Thus, both for two- and three-dimensional problems the number of operations needed for a single sub-iteration grows proportionally to the number of unknowns belonging to one column or row of finite volumes. Note that in a two-dimensional formulation the number of columns to be processed is of the order N, while in a three-dimensional formulation the number of columns is of the order of N^2. Therefore three-dimensional computations remain significantly more demanding.

Apparently, the described solution procedure can be reformulated for a row of finite volumes, which yields an additional line-wise smoother for 2D problems and two additional smoothers in 3D case. Considering convection test problems we observed that for a two-dimensional configuration the line-wise smoother utilizing columns (vertically arranged lines) was about 30% faster than that utilizing the rows. Similarly, for the three-dimensional configurations the columns arrangement was about 35% faster than two other possibilities of rows arrangement. This observation can be explained by a slow convergence of the pressure corrections, which is characterized by a vertical stratification forming inside the differentially heated cavity. This shows that a fast propagation of the pressure correction changes through the computational domain improve the convergence, which, for the test problems considered, is achieved by choosing a line-wise smoother directed along maximal pressure variations. Apparently, the choice of the optimal direction for the line-wise smoother is problem-dependent.

The approach described above is in fact a modified CLGS smoother proposed in [6] and applied in [7]. The modification introduced here allows for an analytical solution at the lowest level of block Gauss–Seidel iteration, which speeds up the whole process. We call this approach an Analytical Solution Accelerated Coupled Line Gauss–Seidel (ASA-CLGS) smoother.

4. Results and discussion

The two approaches, FPCD and ASA-CLGS, were tested for convection of air in two- and three-dimensional rectangular containers. In particular, we considered time-periodic solutions of the MIT2001 benchmark dealing with 8:1 thermally driven cavity and steady state solution of convection in a laterally heated cubical enclosure as two representative 2D and 3D examples, respectively. Convergence to steadiness is assumed to be reached when the maximal point-wise relative difference for each function at two consecutive time steps is less than 10^{-5}.

4.1. Thermally driven rectangular cavity of aspect ratio 8:1

The results obtained using the FPCD and ASA-CLGS multigrid approaches are compared with time-periodic solutions obtained by Xin and Le Quéré [2] and by Guo and Bathe [14] for the differentially heated cavity of aspect ratio 8:1 at $Ra = 3.4 \times 10^5$

Table 1
Comparison of the results of FPCD approach with the independent solutions [2,14]. $Ra = 3.4105 \times 10^5$, $Pr = 0.71$, uniform grid, calculations performed for the time interval of 700 dimensionless units.

Quantity	Average	Amplitude	Period	Average	Amplitude	Period
	48×180 pseudo-spectral Chebyshev modes steps per period 2024.37 [2]			Stretched grid, 40×120 elements steps per period 34 [14]		
u_1	5.6345×10^{-2}	5.4768×10^{-2}	3.4115	5.6100×10^{-2}	5.2940×10^{-2}	3.422
v_1	0.46188	7.7125×10^{-2}	3.4115	0.4620	7.5120×10^{-2}	3.422
θ_1	0.26548	4.2690×10^{-2}	3.4115	0.2654	4.1340×10^{-2}	3.425
ΔP_1	-1.8536×10^{-3}	2.0355×10^{-2}	3.4115	-2.0030×10^{-3}	2.0040×10^{-2}	3.421
ΔP_{51}	−0.53486	2.2442×10^{-2}	3.4115	−0.53490	2.2080×10^{-2}	3.422
ΔP_{35}	0.53671	1.0057×10^{-2}	3.4115	0.53690	0.9976×10^{-2}	3.422
Nu_{x0}	−4.57946	7.0921×10^{-3}	3.4115	−4.57950	6.8900×10^{-3}	3.422
Nu_{x1}	−4.57946	7.0921×10^{-3}	3.4115	−4.57950	6.8920×10^{-3}	3.422
	Uniform grid, 100×800 steps per period 1707.32 ($\Delta t = 0.002$)			Uniform grid, 100×800 steps per period 3414.63 ($\Delta t = 0.001$)		
u_1	5.7490×10^{-2}	6.2236×10^{-2}	3.4146	5.6095×10^{-2}	5.2232×10^{-2}	3.4146
v_1	0.46165	8.5893×10^{-2}	3.4146	0.46177	7.3937×10^{-2}	3.4146
θ_1	0.26565	4.8475×10^{-2}	3.4146	0.26561	4.090×10^{-2}	3.4146
ΔP_1	-1.7841×10^{-3}	2.2811×10^{-2}	3.4146	-1.85305×10^{-3}	1.9522×10^{-2}	3.4146
ΔP_{51}	−0.53575	2.4893×10^{-2}	3.4146	−0.53546	2.1564×10^{-2}	3.4146
ΔP_{35}	0.53753	1.1393×10^{-2}	3.4146	0.53732	0.9764×10^{-2}	3.4146
Nu_{x0}	−4.579232	6.7021×10^{-3}	3.4146	−4.59220	6.9334×10^{-3}	3.4146
Nu_{x1}	−4.579232	6.7021×10^{-3}	3.4146	−4.59220	6.9334×10^{-3}	3.4146

Table 2
Comparison of the of FPCD approach with the independent solution [2,14]. $Ra = 3.4105 \times 10^5$, $Pr = 0.71$, stretched grid, calculations performed for the time interval of 700 dimensionless units.

Quantity	Average	Amplitude	Period	Average	Amplitude	Period
	Stretched Grid, 40×120 elements steps per period 34 [14]			48×180 pseudo-spectral Chebyshev modes steps per period 2024.37 [2]		
u_1	5.6100×10^{-2}	5.2940×10^{-2}	3.4115	5.6345×10^{-2}	5.4768×10^{-2}	3.422
v_1	0.4620	7.5120×10^{-2}	3.4115	0.46188	7.7125×10^{-2}	3.422
θ_1	0.2654	4.1340×10^{-2}	3.4115	0.26548	4.2690×10^{-2}	3.425
ΔP_1	-2.0030×10^{-3}	2.0040×10^{-2}	3.4115	-1.8536×10^{-3}	2.0355×10^{-2}	3.421
ΔP_{51}	−0.53490	2.2080×10^{-2}	3.4115	−0.53486	2.2442×10^{-2}	3.422
ΔP_{35}	0.53690	0.9976×10^{-2}	3.4115	0.53671	1.0057×10^{-2}	3.422
Nu_{x0}	−4.57950	6.8900×10^{-3}	3.4115	−4.57946	7.0921×10^{-3}	3.422
Nu_{x1}	−4.57950	6.8920×10^{-3}	3.4115	−4.57946	7.0921×10^{-3}	3.422
	Stretched Grid, 100×800 Steps per period 3414.63 ($\Delta t = 0.001$)			Stretched Grid, 100×800 Steps per period 1707.32 ($\Delta t = 0.002$)		
u_1	5.7985×10^{-2}	5.7985×10^{-2}	3.4146	5.7535×10^{-2}	7.4267×10^{-2}	3.4146
v_1	0.46119	8.1097×10^{-2}	3.4146	0.46220	8.4425×10^{-2}	3.4146
θ_1	0.26547	4.5300×10^{-2}	3.4146	0.26524	4.9038×10^{-2}	3.4146
ΔP_1	-1.8967×10^{-3}	2.1381×10^{-2}	3.4146	-1.8602×10^{-3}	2.4578×10^{-2}	3.4146
ΔP_{51}	−0.53527	2.3609×10^{-2}	3.4146	−0.53528	2.5371×10^{-2}	3.4146
ΔP_{35}	0.53769	1.0762×10^{-2}	3.4146	0.53735	1.1157×10^{-2}	3.4146
Nu_{x0}	−4.57923	6.7021×10^{-3}	3.4146	−4.57957	8.0646×10^{-3}	3.4146
Nu_{x1}	−4.57923	6.7021×10^{-3}	3.4146	−4.57957	8.0646×10^{-3}	3.4146

Table 3
Comparison of the results obtained by the ASA-CLGS multigrid approach with the independent solution [2]. [14] $Ra = 3.4105 \times 10^5$, $Pr = 0.71$, uniform grid, calculations performed for the time interval of 700 dimensionless units.

Quantity	Average	Amplitude	Period	Average	Amplitude	Period
	48×180 pseudo-spectral Chebyshev modes steps per period 2024.37 [2]			Stretched grid, 40×120 elements steps per period 34 [14]		
u_1	5.6345×10^{-2}	5.4768×10^{-2}	3.4115	5.6100×10^{-2}	5.2940×10^{-2}	3.422
v_1	0.46188	7.7125×10^{-2}	3.4115	0.4620	7.5120×10^{-2}	3.422
θ_1	0.26548	4.2690×10^{-2}	3.4115	0.2654	4.1340×10^{-2}	3.425
ΔP_1	-1.8536×10^{-3}	2.0355×10^{-2}	3.4115	-2.0030×10^{-3}	2.0040×10^{-2}	3.421
ΔP_{51}	−0.53486	2.2442×10^{-2}	3.4115	−0.53490	2.2080×10^{-2}	3.422
ΔP_{35}	0.53671	1.0057×10^{-2}	3.4115	0.53690	0.9976×10^{-2}	3.422
Nu_{x0}	−4.57946	7.0921×10^{-3}	3.4115	−4.57950	6.8900×10^{-3}	3.422
Nu_{x1}	−4.57946	7.0921×10^{-3}	3.4115	−4.57950	6.8920×10^{-3}	3.422
	Uniform grid, 100×800 steps per period 1707.32 ($\Delta t = 0.002$)			Uniform grid, 100×800 steps per period 3414.63 ($\Delta t = 0.001$)		
u_1	5.7846×10^{-2}	6.4487×10^{-2}	3.4146	5.64156×10^{-2}	5.4671×10^{-2}	3.4146
v_1	0.46168	8.8484×10^{-2}	3.4146	0.46179	7.6954×10^{-2}	3.4146
θ_1	0.26558	5.0139×10^{-2}	3.4146	0.26558	4.2729×10^{-2}	3.4146
ΔP_1	-1.8054×10^{-3}	2.3524×10^{-2}	3.4146	-1.84985×10^{-3}	2.0334×10^{-2}	3.4146
ΔP_{51}	−0.53565	2.5613×10^{-2}	3.4146	−0.53539	2.2402×10^{-2}	3.4146
ΔP_{35}	0.53745	1.1698×10^{-2}	3.4146	0.53725	1.0121×10^{-2}	3.4146
Nu_{x0}	−4.58996	8.0165×10^{-3}	3.4146	−4.58969	7.0867×10^{-3}	3.4146
Nu_{x1}	−4.58995	8.3944×10^{-3}	3.4146	−4.58969	7.2008×10^{-3}	3.4146

Table 4
Comparison of the results obtained by the ASA-CLGS multigrid approach with the independent solution [2,14]. $Ra = 3.4105 \times 10^5$, $Pr = 0.71$, stretched grid, calculations performed for the time interval of 700 dimensionless units.

Quantity	Average	Amplitude	Period	Average	Amplitude	Period
	48×180 pseudo-spectral Chebyshev modes steps per period 2024.37 [2]			Stretched grid, 40×120 elements steps per period 34 [14]		
u_1	5.6345×10^{-2}	5.4768×10^{-2}	3.4115	5.6100×10^{-2}	5.2940×10^{-2}	3.422
v_1	0.46188	7.7125×10^{-2}	3.4115	0.4620	7.5120×10^{-2}	3.422
θ_1	0.26548	4.2690×10^{-2}	3.4115	0.2654	4.1340×10^{-2}	3.425
ΔP_1	-1.8536×10^{-3}	2.0355×10^{-2}	3.4115	-2.0030×10^{-3}	2.0040×10^{-2}	3.421
ΔP_{51}	−0.53486	2.2442×10^{-2}	3.4115	−0.53490	2.2080×10^{-2}	3.422
ΔP_{35}	0.53671	1.0057×10^{-2}	3.4115	0.53690	0.9976×10^{-2}	3.422
Nu_{x0}	−4.57946	7.0921×10^{-3}	3.4115	−4.57950	6.8900×10^{-3}	3.422
Nu_{x1}	−4.57946	7.0921×10^{-3}	3.4115	−4.57950	6.8920×10^{-3}	3.422
	Stretched grid, 100×800 steps per period 1707.32 ($\Delta t = 0.002$)			Stretched grid, 100×800 steps per period 3414.63 ($\Delta t = 0.001$)		
u_1	5.7326×10^{-2}	6.3663×10^{-2}	3.4146	5.5869×10^{-2}	5.3652×10^{-2}	3.4146
v_1	0.46123	8.7693×10^{-2}	3.4146	0.46128	7.5847×10^{-2}	3.4146
θ_1	0.26552	4.9546×10^{-2}	3.4146	0.26547	4.1990×10^{-2}	3.4146
ΔP_1	-1.8137×10^{-3}	2.3258×10^{-2}	3.4146	-1.88024×10^{-3}	1.9981×10^{-2}	3.4146
ΔP_{51}	−0.53553	2.5521×10^{-2}	3.4146	−0.53526	2.2116×10^{-2}	3.4146
ΔP_{35}	0.53735	1.1578×10^{-2}	3.4146	0.53714	0.9984×10^{-2}	3.4146
Nu_{x0}	−4.57949	8.1449×10^{-3}	3.4146	−4.57925	6.9967×10^{-3}	3.4146
Nu_{x1}	−4.57950	8.4378×10^{-3}	3.4146	−4.57925	7.1579×10^{-3}	3.4146

($Ra = GrPr$). The comparison of present results obtained by both algorithms using uniform and stretched grids with benchmark solution [2] and reference solution [14] is presented in Tables 1–4. The average Nusselt number reported in the tables defines the dimensionless heat flux through a plane $x = const$ per unit width:

$$Nu_x = \int_0^1 -\frac{\partial \theta}{\partial x} dy \tag{20}$$

Nu_{x0} and Nu_{x1} are the Nusselt numbers obtained for the hot and the cold vertical boundaries, respectively. The solutions obtained by both approaches are in a good agreement with the benchmark-quality solution of [2]. Noticeable discrepancies between the present results and those of [14] can be explained by a large time step, which in [14] was two orders of magnitude larger than in the present work, as well as by rather coarse spatial resolution used in [14]. Our results show that the accuracy tends to increase with the decrease of the time step (Tables 1–4). The grid stretching slightly improves the accuracy for the large time step but its effect weakens when the time step decreases. The hot and cold wall Nusselt number amplitudes obtained by the multigrid solution slightly differ from each other that should not occur. The possible reason for such behavior is inaccuracy introduced by an insufficient numerical convergence of the temperature field, which makes the Nusselt number the most sensitive quantity. Note, that such an inequality of Nusselt number amplitudes is observed also in other studies, e.g. in [14], where quadrilateral finite elements were used for the spatial discretization. This shows that the instantaneous heat balance is a common problem for different numerical methods, which worth to be studied in more detail. The average values of both Nusselt numbers, however, are equal (Tables 1–4), which proves the conservation of heat throughout the cavity. It is expected that the observed inaccuracy will decrease by utilizing semi-implicit discretization of the convection terms, which is the issue of our future research. In this case the FAS-FMG (full approximation storage-full multigrid) algorithm [12], applicable for the non-linear problems should be used.

4.2. Laterally heated cubic box with four perfectly insulating walls

A comparison with independent results [15–18] for convection in three-dimensional box is shown in Table 5. Stretched grid was used for $Ra = 10^6$ and 10^7 to resolve thin boundary layers adjacent

Table 5
Laterally heated cubic box with four perfectly insulating walls: present and previously published results comparison.

	Quantity	Present[a]	Ref. [15][b]	Ref. [16][c]	Ref. [17][d]	Ref. [18][e]
$Ra = 10^4$	$u_{ymax}(y, z)$ at $x = 0.5$	(0.5146,0.8229) 0.1975	(0.5196,0.8250) 0.1984	(0.5250,0.8250) 0.1983		
	$\lvert u_{zmin}(y, z)\rvert$ at $x = 0.5$	(0.8894,0.5240) 0.2211	(0.8870,0.5219) 0.2217	(0.8875,0.5225) 0.2217		
	$u_{y\ max}(Z)$ at $x, y = 0.5$	(0.8250) 0.1970			(0.8370) 0.2065	(0.8250) 0.1984
	$\lvert u_{z\ min}(y)\rvert$ at $x, z = 0.5$	(0.8800) 0.2202			(0.8870) 0.2207	(0.8823) 0.2216
	$Nu_{y=0}$	2.0547	2.0540	2.0550	2.0850	2.0624
$Ra = 10^5$	$u_{ymax}(y,z)$ at $x = 0.5$	(0.3125,0.8894) 0.1633	(0.3135,0.8848) 0.1616	(0.2946,0.8889) 0.1605		
	$\lvert u_{zmin}(y,z)\rvert$ at $x = 0.5$	(0.9371,0.5143) 0.2466	(0.9368,0.5100) 0.2456	(0.9371,0.5000) 0.2452		
	$u_{ymax}(y)$ at $x, y = 0.5$	(0.8500) 0.1434			(0.8640) 0.1490	(0.8500) 0.1416
	$\lvert u_{zmin}(y)\rvert$ at $x, z = 0.5$	(0.9370) 0.2464			(0.9350) 0.2398	(0.9323) 0.2464
	$\lvert Nu_{y=0}\rvert$	4.3349	4.3370	4.3370	4.3780	4.3665
$Ra = 10^6$	$u_{ymax}(y, z)$ at $x = 0.5$	(0.1859,0.9374) 0.1471	(0.5000,0.9366) 0.1465	(0.1920,0.9371) 0.1457		
	$\lvert u_{zmin}(y, z)\rvert$ at $x = 0.5$	(0.9629,0.5391) 0.2592	(0.9638,0.5353) 0.2590	(0.9658,0.5500) 0.2582		
	$u_{ymax}(z)$ at $x, y = 0.5$	(0.8605) 0.0802				(0.8603) 0.0811
	$\lvert u_{zmin}(y)\rvert$ at $x, z = 0.5$	(0.9663) 0.2575				(0.9677) 0.2583
	$\lvert Nu_{y=0}\rvert$	8.6461	8.6400	8.6400		8.6973
$Ra = 10^7$	$u_{ymax}(y, z)$ at $x = 0.5$	(0.1186,0.9674) 0.1410	(0.1223,0.9662) 0.1440			
	$\lvert u_{zmin}(y, z)\rvert$ at $x = 0.5$	(0.9797,0.5391) 0.2626	(0.9794,0.5354) 0.2621			
	$\lvert Nu_{y=0}\rvert$	16.4644	16.3427			

[a] 103^3 uniform grid for $Ra = 10^4$,10^5 and 103^3 non-uniform grid for $Ra = 10^6$,10^7.
[b] 81^3 uniform grid for $Ra = (10^4 \div 10^6)$ and 111^3 uniform grid for $Ra = 10^7$.
[c] 81^3 uniform grid for $Ra = 10^4$ and 81^3 non-uniform grid for $Ra = 10^5$ and $Ra = 10^6$.
[d] $61 \times 45 \times 45$ non-uniform grid for $Ra = 10^4$ and $91 \times 45 \times 45$ non-uniform grid for $Ra = 10^5$.
[e] Richardson extrapolation using 80^3 and 120^3 uniform grids.

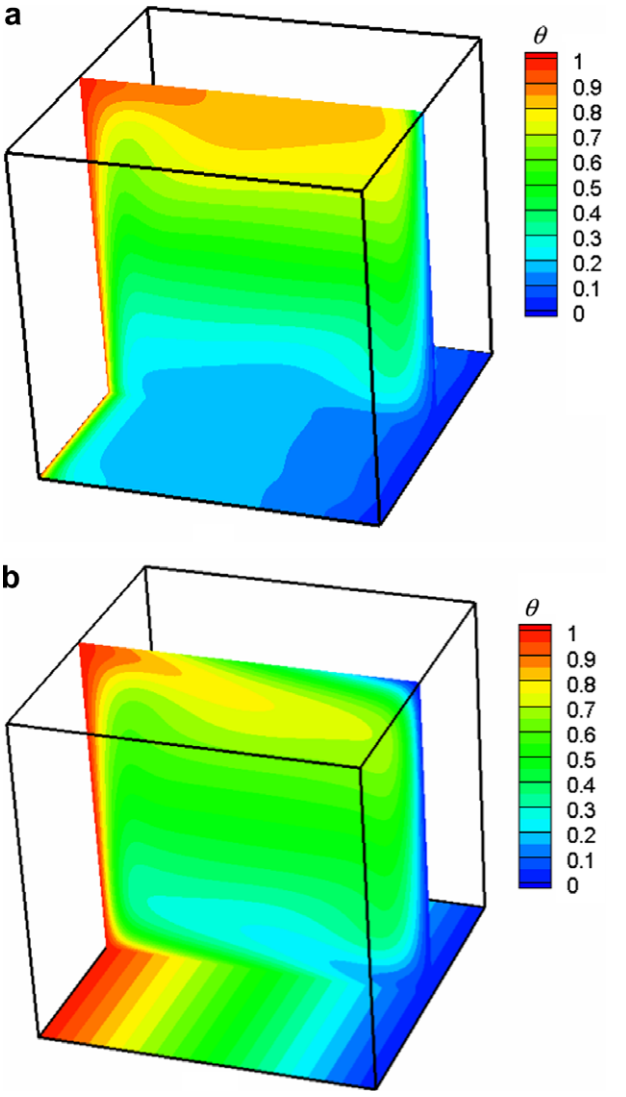

Fig. 5. Temperature distribution in a differentially heated cubic box for $Ra = 10^6$ with: (a) four perfectly insulating walls and (b) four perfectly conducting walls.

to the hot and cold boundaries (see Fig. 5a). An average Nusselt number determining the thermal flow rate trough the plane $y = const$ is defined as:

$$Nu_y = -\int_0^1 \int_0^1 \frac{\partial \theta}{\partial y} dxdz \tag{21}$$

The calculated Nusselt numbers at the hot and cold walls were equal, thus verifying the heat balance inside the cavity. In the following only results for $Nu_{y=0}$ are reported. The obtained benchmark results are well compared with the previously published data for the entire range of Ra numbers. In particular, we would like to stress the agreement with results of Tric et al. [15], whose spatial resolution is believed to be better than 0.02%, as well as of Wakashima and Saitoh [18] who utilized a fourth order finite-differences scheme on a 120^3 grid. The velocity maxima reported in [15] for the mid-plane ($x = 0.5$) agree to within 0.04% for $Ra = 10^4$, 10^5 and 10^6, and to within 0.08% for $Ra = 10^7$ (Table 5). The calculated Nusselt numbers agree to within 0.034%, 0.048%, 0.07% and 0.74% for the above Rayleigh numbers, respectively. Comparison of velocities calculated at two centerlines (0.5, 0.5, z) and (0.5, y, 0.5) agree with results of [18] to within 1.26% for $Ra = 10^4$ and 10^5, and to within 1.1% for $Ra = 10^6$ (Table 5). A possible reason for such relatively large discrepancies may arise from vorticity-vector potential formulation utilized in [18] instead of using primitive variables formulation as in the present study and in [15]. The corresponding Nusselt numbers agree to within 0.37%, 0.72%, 0.59% for these Rayleigh numbers, respectively.

4.3. Laterally heated cubic box with four perfectly conducting walls

Comparison with the independent experimental [19] and numerical [16] results for this problem is presented in Table 6. As above, the stretched grid was used for $Ra = 10^6$ to resolve thin boundary layers developed close to the hot and cold boundaries (see Fig. 5b). The Nusselt numbers obtained in the present study are in a good agreement with both experimental and numerical data of [19,16] for the entire range of Ra numbers: the maximal deviation between the corresponding Nusselt numbers does not exceed 1.1%. The calculated velocity maxima at the middle plane of the cube ($x = 0.5$) agree to within 0.12% for $Ra = 10^4$, 0.3% for $Ra = 10^5$ and 0.11% for 10^6 with the results of [16] (Table 6).

Table 6
Characteristic quantities for free convection in laterally heated cubic box with four perfectly conducting walls: comparison of present results with previously published data.

Source	$u_{y\ max}(y,z)$ at $x = 0.5$		$\lvert u_{z\ min}(y,z)\rvert$ at $x = 0.5$		$\lvert Nu_{y=0}\rvert$
$Ra = 10^4$					
Ref. [19][a]	–	–	–	–	1.52 ± 0.015
Ref. [16][b]	(0.5250,0.8375)	0.2504	(0.8750,0.5125)	0.2503	1.5060
Present[c]	(0.5240,0.8317)	0.2501	(0.8798,0.5144)	0.2500	1.5158
$Ra = 10^5$					
Ref. [19][a]	–	–	–	–	3.097 ± 0.028
Ref. [16][b]	(0.4250,0.8750)	0.2356	(0.9375,0.5250)	0.2717	3.1180
Present[c]	(0.4279,0.8798)	0.2363	(0.9375,0.5144)	0.2721	3.1191
$Ra = 10^6$					
Ref. [19][a]	–	–	–	–	6.383 ± 0.070
Ref. [16][b]	(0.2380,0.9371)	0.2142	(0.9612,0.5500)	0.2833	6.512
Present[c]	(0.2260,0.9375)	0.2141	(0.9664,0.5721)	0.2830	6.584

[a] Experimental data.
[b] 81^3 uniform grid for $Ra = 10^4, 10^5$ and 81^3 non-uniform grid for $Ra = 10^6$.
[c] 103^3 uniform grid for $Ra = 10^4, 10^5$ and 103^3 non-uniform grid for $Ra = 10^6$.

4.4. Flow visualization

The three-dimensional flows were visualized by plotting isosurfaces of the temperature (Fig. 5) and trajectories of four particle passing through the main diagonal of the mid-plane $x = 0.5$ (Fig. 6). All the results are plotted for $Ra = 10^6$. Both configurations with perfectly insulating or perfectly conducing walls are characterized by thin thermal boundary layers located near the hot and cold walls. The graphical representation of the streamlines at the cavity mid-plane can be found, e.g. in [16]. It is seen that in both configurations the trajectories leave the mid-plane, thus indicating on fully three-dimensional flow pattern. The three-dimensional velocity and temperature fields preserve the symmetry with respect to a 180°-rotation around the x-axis. At the same time there exists a pressure difference between the mid-plane and vertical conducting or insulating boundaries, which leads to a non-zero velocity in the x-direction. This emphasizes the qualitative difference between flows in three-dimensional boxes with no-slip boundaries and similar two-dimensional problems.

In the configuration with four perfectly insulating walls the trajectories are almost two-dimensional close to the cavity mid-plane. After leaving the mid-plane they exhibit a complicated three-dimensional pattern (Fig. 6a). In the configuration with four perfectly conducting walls the trajectories are smoother than in the previous case (Fig. 6b). It is seen that these trajectories remain inside the planes that are close and parallel to the mid-plane for some time. After leaving these planes they rather quickly approach the insulated vertical wall and immediately turn back.

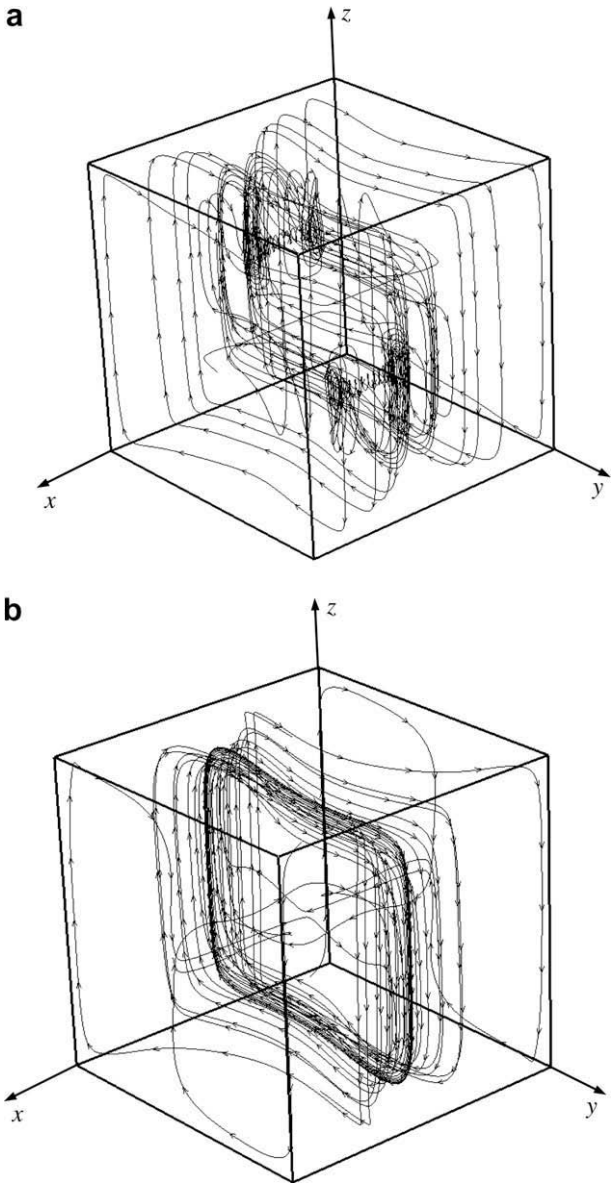

Fig. 6. A particle trajectory in a differentially heated cubic box for $Ra = 10^6$ with: (a) four perfectly insulating walls and (b) four perfectly conducting walls.

4.5. CPU time and memory consumption

In this section we give the characteristic CPU times and memory requirements for our multigrid-based approach. It should be noted that computing characteristic CPU times for any iterative algorithm is a rather complicated task since its convergence rate might change during the whole solution depending on several factors such as mesh and time step size, over or under-relaxation parameter values, and closeness of an initial guess to a current solution. Needless to say, it is also strongly problem-dependent. Among all, the closeness to a solution could be crucial and its effect on the characteristic CPU time may result in reduction of number of iterations in several times. Two benchmarks, known to have periodical solutions: 8:1 differentially heated cavity with perfectly insulating horizontal walls at $Ra = 3.4 \times 10^5$ [11] and differentially heated cubic box with perfectly conducting bottom and top and insulated left and right walls (see Fig. 1) at $Ra = 3.3 \times 10^6$ [20] have been chosen for CPU time measurements in two- and three-dimensional geometry, respectively. For both geometries the time step was $\Delta t = 10^{-3}$ and after a series of tests the under-relaxation parameter was chosen to be 0.6. Memory consumption and characteristic CPU times were measured for uniform grids only.

Computer memory and the characteristic CPU time per one node per one CPU consumed for periodical solution in two- and three-dimensional configurations are shown in Fig. 7. The calculations were performed on Intel Pentium 2.4 GHz processor with 2 GB RAM. Grid resolutions and total nodes number used for the analysis are detailed in Table 7. Note that the dependence of memory and CPU time consumption for both configurations versus the total nodes number is almost linear that is typical for the multigrid method [13]. Note that both memory and CPU consumption growth rates are almost equal for both configurations considered (Figs. 7). Roughly, the CPU time consumption per one node and per one CPU for a three-dimensional problem is about 2 times larger than that needed for a two-dimensional problem. The memory consumption for a three-dimensional problem is about 1.5 times

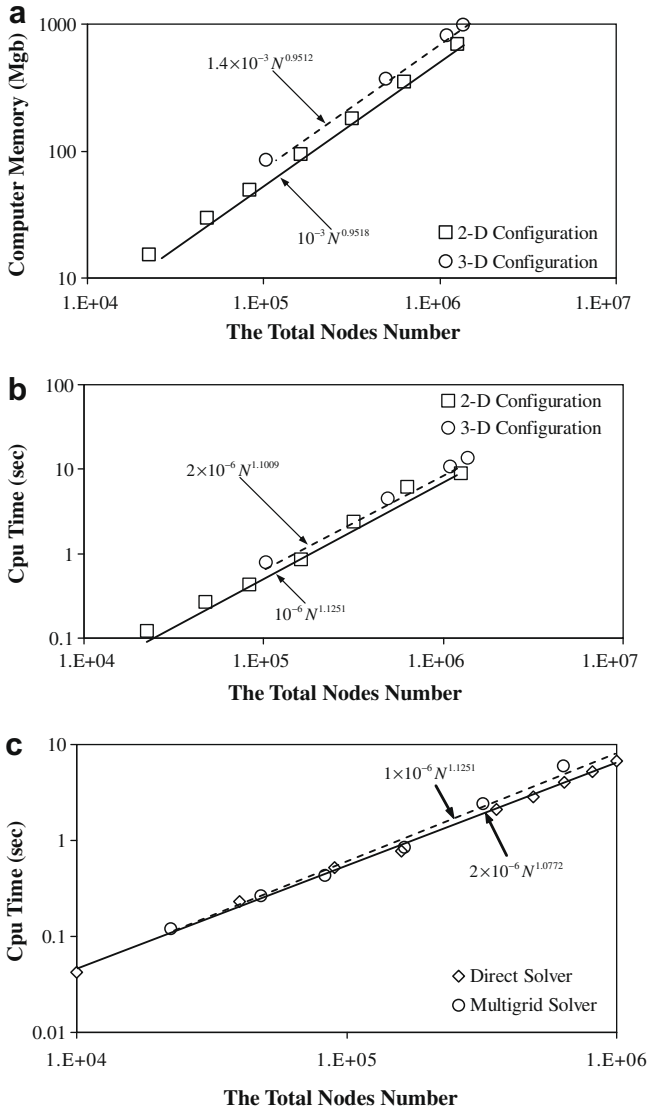

Fig. 7. Computer memory (a), CPU time per one node and per one CPU consumed for periodical solution for 2D and 3D configurations (b), and consumed by FPCD and ASA-CLGS for the 2D configuration (c).

Table 7
Grid resolutions and the total nodes number of two- and three-dimensional configurations used for computation of memory and CPU time consumption.

2-D configuration		3-D configuration	
Grid resolution	Total nodes number	Grid resolution	Total nodes number
55 × 407	22,385	47^3	103,823
79 × 607	47,953	79^3	493,039
103 × 807	83,121	103^3	1,092,727
135 × 1207	162,945	111^3	1,367,631
199 × 1607	319,793		
263 × 2407	633,041		
391 × 3207	1,253,937		

larger than that needed for a two-dimensional problem with the same number of grid nodes. Finally, in Fig. 7c we compare the CPU time consumption for our FPCD and ASA-CLGS approaches measured for the two-dimensional problem. We observe that at large number of nodes the FPCD approach becomes faster. This indicates again on its possible attractiveness assuming that the computer memory restrictions are removed.

5. Concluding remarks

Two approaches to time-integration of the coupled incompressible Navier–Stokes equations are proposed and verified. The preliminary test calculations described above show that the ASA-CLGS multigrid solver can be considered as a reliable and robust tool for performing time-dependent computations on fine three-dimensional finite volume grids. This approach needs relatively small amount of computer memory and can become attractive for 3D calculations on fine grids.

The FPCD approach, utilizing the *LU* decomposition of the Stokes operator, shows competitive computational times for two-dimensional problems, but remains restricted by the available computer memory when is applied to three-dimensional models. Assuming that this restriction will be removed in near future this approach should not be immediately neglected. An additional advantage of the FPCD approach is a constant and a priori known CPU time consumed at each time step. Apparently it is not a case for any iterative solver.

It seems us obvious that both approaches can be easily parallelized. The corresponding studies of scalability are yet to be done. Additionally the ASA-CLGS approach can be extended to semi-implicit and fully-implicit formulations, which can improve the robustness of time-integration, as well as will allow one to extend Newton iteration based steady state solvers and stability solvers of [3,21] to three-dimensional geometries.

An additional advantage of the ASA-CLGS approach is a possibility to extend it to higher-order schemes formulated on larger grid stencils. To do that all the terms not allowing for an analytical solution similar to (18), (19) must be put in the right hand sides of the pressure and velocity corrections equations. Standard linear interpolation functions were applied in the finite volume formulation used for the above test computations. It would be interesting to use higher-order interpolation functions, for example those proposed in [22,23] for the finite element formulation, and to verify convergence of such a finite volume higher-order formulation.

Acknowledgements

This study was supported by Ministry of Science, Sport and Culture, State of Israel, Grant No. 3-4293.

References

[1] Acharya S, Baliga BR, Karki K, Murthy JY, Prakash C, Vanka SP. Pressure-based finite-volume methods in computational fluid dynamics. J Heat Transf 2007;129:407–24.
[2] Xin S, Le Quéré P. An extended Chebyshev pseudo-spectral benchmark for the 8:1 differentially heated cavity. Int J Num Meth Fluids 2002;40:981–98.
[3] Gelfgat A.Yu. Stability of convective flows in cavities: solution of benchmark problems by a low-order finite volume method. Int J Num Meth Fluids 2007;53:485–506.
[4] Vanka SP. Block-implicit multigrid solution of Navier–Stokes equations in primitive variables. J Comput Phys 1986;65:138–58.
[5] Vanka SP. A calculation procedure for three-dimensional steady recirculating flows using multigrid methods. Comp Meth Appl Mech Eng 1986;55:321–38.
[6] Zeng S, Wesseling P. Numerical study of a multigrid method with four smoothing methods for the incompressible Navier–Stokes equations in general coordinates. In: Melson ND, Montueffel TA, McCormick SF, editors. Sixth copper mountain conference on multigrid methods. NASA Conference Publication 3224, 1993. p. 691–708.
[7] Paisley MF. Multigrid solution of the incompressible Navier–Stokes equations for three-dimensional recirculating flow: coupled and decoupled smoothers compared. Int J Num Meth Fluids 1999;30:441–59.
[8] Anderson DA, Tannehill JC, Pletcher RH. Computational fluid mechanics and heat transfer. 2nd ed. New York: Hemisphere Publishing Corporation; 1997.
[9] Patankar SV. Numerical heat transfer and fluid flow. New York: McGraw-Hill; 1980.

[10] van der Vorst HA. Iterative Krylov methods for large linear systems. Cambridge University Press; 2003.
[11] Christon MA, Gresho PM, Sutton SB. Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution). Int J Num Meth Fluids 2002;40:953–80.
[12] Brandt A. Multi-level adaptive solutions to boundary-value problems. Math Comput 1977;31(138):333–90.
[13] Trottenberg U, Oosterlee C, Schüller A. Multigrid. London: Academic Press; 2001.
[14] Guo Y, Bathe K-J. A numerical study of natural convection flow in a cavity. Int J Num Meth Fluids 2002;40:1045–57.
[15] Tric E, Labrosse G, Betrouni M. A first inclusion into the 3D structure of natural convection of air in a differentially heated cavity, from accurate numerical solutions. Int J Heat Mass Transf 1999;43:4043–56.
[16] Bennet BAV, Hsueh J. Natural convection in a cubic cavity: implicit numerical solution of two benchmark problems. Num Heat Transf Part A 2006;50:99–123.
[17] Peng Y, Shu C, Chew YT. A 3D incompressible thermal Lattice Boltzmann Model and its application to simulate natural convection in a cubic cavity. J Comput Phys 2003;193:260–74.
[18] Wakashima S, Saitoh TS. Benchmark solutions for natural convection in a cubic cavity using the high-order time-space method. Int J Heat Mass Transf 2004;47:853–64.
[19] Leong WH, Hollands KGT, Brunger AP. Experimental Nusselt numbers for a cubical-cavity benchmark problem in natural convection. Int J Heat Mass Transf 1999;42:1979–89.
[20] Jones DN, Briggs DG. Periodic two-dimensional flow: effect of linear horizontal thermal boundary condition. J Heat Transf 1989;111:86–91.
[21] Tuckerman LS. Numerical methods for bifurcation problems. In: Descalzi O, Martinez J, Rica S, editors. Instabilities and non-equilibrium structures IX. Kluwer, Dordecht; 2004.
[22] Kohno H, Bathe K-J. A flow-condition-based interpolation finite element procedure for triangular grids. Int J Num Meth Fluids 2006;51:673–99.
[23] Banijamali B, Bathe K-J. The CIP method embedded in finite element discretization of incompressible fluid flow. Int J Num Meth Eng 2007;71:66–80.