XX Polish Fluid Mechanics Conference,
Gliwice, 17-20 September 2012

IMPROVED EFFICIENCY OF THE PRESSURE-VELOCITY COUPLING ALGORITHM WITH WEAK-COMPRESSIBILITY TERMS

Tomasz DYSARZ
Poznan University of Life Sciences, Dep. of Hydraulic and Sanitary Eng., Poznań, Poland;
E-mail: tdysarz@gmail.com, dysarz@up.poznan.pl

Abstract

The main purpose of the presented research is to incorporate the idea of weak-compressibility into the pressure-velocity coupling algorithm in order to improve the convergence and efficiency of the flow solver. The mathematical model is a modified version of the weakly-compressible flow equations proposed by Song & Yuan (1988). The finite volume method is implemented into the staggered mesh with a dislocation of variables proposed by Tu & Aliabadi (2007). A generalized multilevel PISO scheme is used (Issa, 1986) as the pressure-velocity coupling algorithm. The MGMRES algorithm (Burkardt, 2008) is adopted to solve all linear systems in the problem. Because the fundamental element of the algorithm is a gradually vanishing term describing weak-compressibility, the models tested are called vanishing compressibility models. The developed technique is tested on two well-known cases: (1) the lid-driven cavity flow, (2) the flow over backward facing step. The assessment of efficiency is done on the basis of a comparison with the classical approach based on incompressible flow equations and the PISO scheme. The results of the simulation show that the proposed algorithm may be faster than the incompressible flow model. The results obtained also point to its better compatibility with the incompressible model than of the basic weak-compressibility approach with fixed compressibility terms. The improvement of efficiency may be explained by the changes in the structure of the main matrix of the pressure correction equations. This suggests that the improvement may be independent of the computational problem analyzed. Hence, the vanishing compressibility models may be recommended for broader use.

Key words: Navier-Stokes equations, pressure-velocity coupling, weakly-compressible flow

INTRODUCTION

The main problem analyzed in the presented paper is the efficiency of the incompressible flow simulation based on the pressure-velocity coupling algorithm. To improve the speed of convergence, the mentioned solution strategy is not used in its pure form. It is combined with the approach based on a weak-compressibility concept close to the other group of solution techniques, namely the artificial compressibility methods. Such a combination enables the control of convergence and an increase in the efficiency in the simulation of the incompressible flow. The concept and the results are presented in the paper.

In general, there are two groups of methods used to solve incompressible flow equations and to overcome the problem of a divergence-free velocity field. These are artificial compressibility methods and pressure-velocity coupling methods. The algorithms of the first group are an adaptation of density-based solvers used to solve the original compressible Navier-Stokes equations. The main idea is to add the time derivative of pressure to the continuity equation. The derivative is multiplied by a small value to approximately satisfy the

condition of a divergence-free velocity field. Such an approach was proposed by Chorin (Chorin, 1967; Drikakis & Rider, 2005). Since that time, several versions of the method have been suggested and described in books on CFD, e.g. by Drikakis & Rider (2005).

The methods of the second group may be classified as the projection method and pressure correction methods. The main idea utilized in the first sub-group is the Helmholtz-Hodge decomposition. The first very basic algorithm was proposed by Chorin (Chorin, 1968). Then the development of this approach led to the fractional step method proposed by Kim & Moin (Kim & Moin, 1985; Ferzinger & Peric, 2002). The advantage of this method is the lack of the iterative refinement, making it very suitable for transient flow simulations.

The pressure correction methods are called also the approximate projection (Drikakis & Rider, 2005), due to their close relationship with the first sub-group of pressure-velocity coupling algorithms. The most fundamental method in this area is the SIMPLE scheme (Caretto et al., 1972; Ferzinger & Peric, 2002). The name is an abbreviation of an expression '*Semi-Implicit Method for Pressure Linked Equations*'. Because of the simplification introduced, the method requires the iterative refinement. Hence, it was primarily used for steady flows. A more developed algorithm based on this concept consists of the commonly called SIMPLE-like or SIMPLE-type methods, e.g. SIMPLE Consistent (Van Doormal & Raithby, 1984; Ferzinger & Peric, 2002) called SIMPLEC, SIMPLE Revised (Patankar, 1980; Ferzinger & Peric, 2002) called SIMPLER or '*Pressure Implicit Splitting of Operators*' proposed by Issa (Issa, 1986; Ferzinger & Peric, 2002; Drikakis & Rider, 2005), which is commonly known as the PISO scheme. Any modification improves convergence, reducing the number of necessary iterations. Theoretically, an extension of the classical PISO to a multilevel correction procedure should remove the need for any such refinement.

The methods listed were proposed more than 20 years ago, but they are still widely applied and developed. For example, Nithiarasu (Nithiarasu, 2003; Nithiarasu & Liu, 2006) improved the efficiency of artificial compressibility by combining this method with a characteristic based split. Gao & Liu (2009) proposed an implementation of the fractional step method as an element of a hybrid FVM/FEM algorithm on a staggered but unstructured grid. Barton (1998) analyzed some extensions of the SIMPLE and PISO schemes as well as their linkage. Tao and co-workers (Cheng et al., 2007; Sun et al., 2008; Sun et al., 2009) presented the CLEARER and IDEAL schemes, which are a mixture of SIMPLER and SIMPLE algorithms. They were successfully implemented in the steady flow problems and they showed a better efficiency than the fastest SIMPLER and PISO schemes.

The research in this area is still being conducted because there are still unresolved problems related to the implementation of the incompressible flow solvers. The artificial compressibility method may be applied with relatively fast ODE solvers, but a control of stability is needed in such a case. Although the solutions ensured by this method converge with the solutions of real incompressible flow equations, the velocity filed is never perfectly solenoidal. The reason for that are the ODE solvers requiring a non-zero term with the time derivative of pressure in continuity equations. The fractional step method is a relatively fast solver, but its non-iterative nature is related to the form of a numerical scheme used. Hence, the simplifications introduced in the method formulation also carry some errors. On the other hand, the pressure correction methods are very robust, but require iterative refinement. The basic SIMPLE scheme is slowly convergent and requires under-relaxation. The more developed SIMPLEC and SIMPLER are faster, but they are still recommended mainly for steady flows. In case of transient flows, the most common approach is to use a slightly inaccurate fractional step method or the iterative PISO scheme. The last method seems to be the most effective algorithm in the group of pressure correction methods, which is applied to steady as well as transient flows.

The most time consuming part of the pressure-velocity coupling algorithm is a single or multiple solution of the Poisson equation. This element is used for two purposes. In the fractional step method, current pressure values are determined that way. The corrections of the approximated pressures are taken from the solution of the Poisson equation in SIMPLE-type schemes. The choice of the linear solver implemented in this problem is crucial for the efficiency of the whole algorithm (Barrett et al., 1994). The classical stationary solvers, such as Jacobi, the Gauss-Siedel method or the SOR algorithm, seem to be too slow for incompressible flow simulations. More efficient non-stationary methods based on conjugate gradients should be used carefully, due to a number of constraints imposed on their performance, e.g. symmetry, stability, diagonal domination, etc. Today, a more general and the most effective linear solver for serial computations seems to be the GMRES method (Schafer, 2006). The main idea of the method is an application of the Arnoldi iterations (Barrett et al., 1994). The method is implemented in many codes available through the Internet, e.g. the MGMRES algorithm (Burkardt, 2008).

The mentioned problems with the solution of incompressible flow equations lead to alternative approaches. One of them is the so-called low Mach number flow or weakly-compressible flow (e.g. Song & Yuan, 1988). The basis of this approach is a simplification of compressible flow equations, but not as deep as the one applied in case of the incompressible model. Instead of constant density, a relatively simple relationship between pressure and density is used as the state equation. The flow model is constructed on the basis of an introduction of this relationship and an assumption of a low Mach number. A simplified system of conservative equations still includes time derivatives in continuity as well as momentum equations. This approach may be treated as a physical interpretation of the artificial compressibility method. The structure of the model enables an application of the ODE solvers (e.g. Bajantri et al. (2007) as well as pressure-velocity coupling algorithms (Munz et al., 2003).

The main purpose of the research is to improve the efficiency of incompressible flow simulation in a transient regime. The approach presented is based on the incorporation of weakly-compressible terms into the equations of the model. The algorithm is built as a multilevel PISO scheme. The weak-compressibility terms vanish during subsequent restarts, so that the obtained results converge with the results of an incompressible flow model. The analysis is done on the basis of simulations for two cases described later. The proposed algorithm is compared with incompressible and weakly-compressible flow models.

The algorithm and results are described in the following sections. The next section presents details on the mathematical model and basic numerical schemes used. The third section explains the idea of the pressure-velocity coupling algorithm applied. The fourth section shows the obtained results and discussion. The last section includes conclusions.

THE BASICS OF MATHEMATICAL AND NUMERICAL MODEL

Continuity and momentum equations

The model described by Song & Yuan (1988) is adopted here. It was implemented successfully in case of the problem of spillway flow simulation by Song & Zhou (1999) and later by Bajantri et al. (2007). The model is derived form the basic form of Navier-Stokes equations completed with continuity equation written for real compressible fluid. Song & Yuan (1988) assumed that the state equation is a very basic definition of speed sound a which may be written as follows

$$dp = a^2 \, d\rho \tag{1}$$

In equation (1), dp is pressure increment dependent on density increment $d\rho$.

The implementation of this assumption with the assumption of low Mach number flow leads to a system of PDEs called a weakly-compressible flow model. The equations written in the Einstein notation are as follows:

$$\frac{\beta}{a^2}\frac{\partial p}{\partial t}+\rho\frac{\partial u_i}{\partial x_i}=0 \qquad \rho\frac{\partial u_i}{\partial t}+\rho\frac{\partial}{\partial x_j}\left(u_i u_j-\nu\frac{\partial u_i}{\partial x_j}\right)=-\frac{\partial p}{\partial x_i} \tag{2a,b}$$

The equation (2a) is a single continuity equation. The second formula, (2b), represents two momentum equations in a 2D case. It is worth noting that the Einstein notation is used by default in the paper for sums exceeding spatial dimensions. Other sums in formulas, e.g. a sum over cell faces, are written explicitly in order to avoid errors and misunderstandings.

The independent variables in (2a) and (2b) are time t and two spatial dimensions, namely x_1 and x_2. The dependent variables are pressure p and two velocity components u_i ($i = 1, 2$). ρ and ν are fluid density and kinematic viscosity coefficient, respectively. a is the speed of sound mentioned earlier. One of the main assumptions made here is that the coefficients ρ, ν and a are constants and they are specific for a particular fluid. A more detailed description of the model may be found in papers cited above or references given there.

Another element in (2a) not discussed yet is the β coefficient. The coefficient is absent in the approach described by Song & Yuan (1988). It was introduced here for the purpose of this research. The β coefficient describes the "level of weak-compressibility". If its value equals one, the model is weakly-compressible as derived by Song & Yuan (1988). For β equaling zero, the model becomes fully incompressible. In general, this coefficient is used to handle the level of weak-compressibility in the simulation. Then, its range of variability is [0, 1]. As it is discussed further, such a construction enables the control of a computational convergence.

The boundary conditions

The Robin's boundary conditions are used for each variable. They may be written in a general form as follows:

$$\omega_\phi \phi+\lambda_\phi\frac{\partial \phi}{\partial n}=\Psi_\phi \tag{3}$$

where ϕ represents one of the variables defined in boundaries: pressure (p), normal velocity (v_n) or tangential velocity (v_s). The ω_ϕ, λ_ϕ and Ψ_ϕ coefficients are specific for a particular variable and the conditions imposed in a particular boundary. Such a form is a well-known generalization of Dirichlet, Neumann and the linear Hankel conditions.

The normal and tangential velocity components are linked to Cartesian velocities u_i ($i = 1, 2$) represented by the following formulas

$$v_n=u_i n_i \qquad v_s=u_i s_i \tag{4a,b}$$

where n_i and s_i ($i = 1, 2$) are the components of normal and tangential unit vectors defined for a boundary. The normal vector is outward to the flow area and the tangential vector indicates the counterclockwise direction along the boundary.

The staggered mesh approach

The staggered grid with a dislocation of variables in the mesh points is implemented here. Such an approach is used to overcome common problems with the chessboard pressure patterns in pressure-velocity coupling algorithms. The scheme of variable dislocation is adopted from Tu & Aliabadi (2007) and Gao & Liu (2009). The approach used here differs slightly from the research quoted in methods applied. In both mentioned papers, the hybrid FVM/FEM approach is used. The finite volume method (FVM) is implemented to solve momentum balance equations and finite element method (FEM) is applied to Poisson equation for pressure corrections. In this research, the FVM with the splitting of second order derivatives is used to solve both PDEs. Such approach is simpler and good enough for tested

cases. However, the conclusions relating to the combination of incompressible and weakly-compressible model presented here may also be extended to a more complex algorithm with a hybrid FVM/FEM approach.

The pattern of variable location is shown in fig. 1a. The velocity components u_i ($i = 1, 2$) are located in cell centers. The finite volume used in the approximation of momentum equation is schematically shown in fig. 1b. F_c is an area of such a cell-centered volume. The faces of the volume are the lines connecting grid nodes. Hence, the fluxes through faces are calculated on the basis of the values in two cells sharing the same face. The pressure corrections are located in mesh nodes. The finite volume used for approximation of pressure correction equations is node-centered. It is illustrated in fig. 1c. F_n is an area of such a control volume. The faces of the volume are the lines connecting the centers of neighboring cells. It is worth noticing that the velocity components are easily calculated in the faces of the pressure correction volume, because their values are determined at the ends of such a face. Another important thing is that the pressure gradients are easily calculated in the velocity volumes on the basis of the values determined in the volume corners.

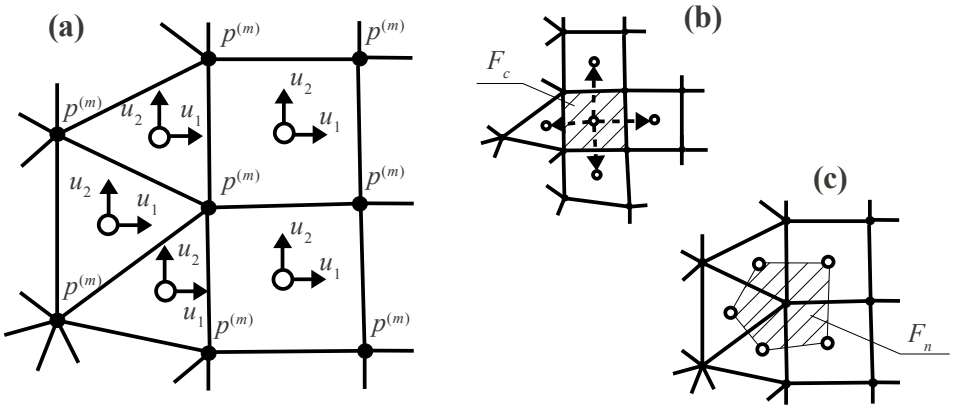

Fig. 1a,b,c The main elements of an implemented staggered grid: (a) location of variables, (b) cell-centered volume for momentum equations, (c) node-centered volume for pressure correction equations

The approximation of momentum equation

The FVM is used for the approximation of momentum equations. The first step is the integration of equations (2b) in a cell-centered volume. Afterwards, the divergence theorem is applied and the following equation is obtained:

$$\int_{F_c} \rho \frac{\partial u_i}{\partial t} dF + \int_{\sigma_c} \rho \left(u_i u_j - \nu \frac{\partial u_i}{\partial x_j} \right) n_j d\sigma = -\int_{F_c} \frac{\partial p}{\partial x_i} dF \tag{5}$$

The first term relates to the time derivative of the velocity component. The second term includes convective and diffusive fluxes integrated over the boundaries of a cell-centered volume. This boundary is denoted as σ_c. n_j ($j = 1, 2$) are the components of an outward normal vector to the boundary. The last term is the integral of pressure gradient components constructed on the basis of pressure values in cell corners.

The standard approach for the approximation of elements present in the equation (5) is used. A detailed description of basic numerical techniques applied here may be found in many books (e.g. Ferzinger & Peric, 2002; Drikakis & Rider, 2005; Schafer, 2006), papers (e.g. Ubbink & Issa, 1999) or other publications (e.g. Jasak, 1996). Hence, only the main steps and elements of the presented algorithm are briefly described below.

The average value of the velocity components $(U_i)_c$ ($i = 1, 2$) in cell-centered volume is introduced. It is assumed that the changes in the velocity components in each finite volume are linear. Hence, the average value is treated as a value in the cell center. The blended flux

approach is used for the approximation of convective fluxes in cell faces (Schafer, 2006). Such a method is a weighted sum of upwind difference and higher order scheme. In the presented case, central difference is used as the second scheme. The diffusive fluxes are approximated by splitting them into the orthogonal and non-orthogonal part according to the formulas explained by Jasak (1996). The average pressure gradient in a cell-centered volume is determined by a linear approximation of pressure on the basis of node values. Finally, the semi-implicit scheme with time step Δt and weighted parameter θ is used for time integration. The result of the above mentioned procedure is a set of two algebraic equations written for each cell-centered volume as follows:

$$A_c \left(U_i\right)_c^{k+1} + \sum_f A_f \left(U_i\right)_f^{k+1} = \left(r_i\right)_c^k + \left(r_i\right)_c^{k+1} - \frac{\theta \Delta t}{\rho} \left(\frac{\partial p}{\partial x_i}\right)_c^{k+1} \tag{6}$$

In equation (6), $(U_i)_c^{k+1}$ is the i-th velocity component determined in a cell center for the k+1 time level, where $(U_i)_f^{k+1}$ are similar values determined in the neighboring cells or boundary face. The Σ sign means a summation over cell faces. A_c and A_f are derived coefficients. What is important, the A_c coefficient is greater than the sum of A_f coefficients and the main matrix of the momentum system is diagonally dominant. The first term on the right-hand side, namely $(r_i)_c^k$, is the term including all the elements calculated for the previous time level k. The second element $(r_i)_c^{k+1}$ includes terms resulting from convective flux linearization and diffusive flux splitting. The last element in (6) is a pressure gradient approximated in the cell for time level k+1. This element is multiplied by specific coefficient composed of time scheme parameter θ, time step Δt and density ρ.

The system of equations composed on the basis of (6) have to be completed with properly approximated boundary conditions (3).

THE PRESSURE-VELOCITY COUPLING WITH VANISHING COMPRESSIBILITY

The multilevel PISO scheme

The implementation of the weakly-compressible flow equations (2a) and (2b) with pressure-velocity coupling algorithm is the main concern in this paper. For this purpose, the PISO concept (Issa, 1986, Ferzinger & Peric, 2002; Drikakis & Rider, 2005) is adopted and extended. This idea is briefly described below.

The first step in the derivation of the pressure correction equation is time integration of the continuity equation (2a). Semi-implicit scheme with the θ_0 weighted parameter is used for this purpose.

$$\frac{\beta}{a^2 \Delta t}\left(p^{k+1} - p^k\right) + \left(1-\theta_0\right)\rho \left(\frac{\partial u_i}{\partial x_i}\right)^k + \theta_0 \rho \left(\frac{\partial u_i}{\partial x_i}\right)^{k+1} = 0 \tag{7}$$

The θ_0 parameter may be equal to the value of the primary time scheme parameter θ used for the approximation of the momentum equation. However, in general, the θ_0 value may differ.

The next step is splitting the variables into approximated values, namely p^* and u_i^* (i = 1, 2), and corrections, $p^{(m)}$ and $u_i^{(m)}$ (i = 1, 2), where $m \geq 1$ is the correction number.

$$p^{k+1} = p^* + p^{(1)} + p^{(2)} + \ldots \qquad u_i^{k+1} = u_i^* + u_i^{(1)} + u_i^{(2)} + \ldots \tag{8a,b}$$

The small letters mean that the values in (8a) and (8b) may be calculated in arbitrary locations. If they are originally determined in cell centers or nodes, the transfer to other points is done by interpolation. In a basic SIMPLE-like scheme, e.g. SIMPLE, SIMPLEC, the value of m is one. The simplifications introduced into such methods require the use of several restarts during a single time step. In a more developed PISO scheme m is equal to 2. The restarts may be done, but in some cases the method is accurate enough after a single run during the time step. In general, the concept may be extended into a number of corrections as well as the number o restarts.

The splitting of variables (8a) and (8b) enables a decomposition of momentum balance equations shown below

$$A_c\left(U_i\right)_c^{*}+\sum_f A_f\left(U_i\right)_f^{*}=\left(r_i\right)_c^{k,*}-\frac{\theta\,\Delta t}{\rho}\left(\frac{\partial p}{\partial x_i}\right)_c^{*} \qquad A_c\left(U_i\right)_c^{(1)}=-\frac{\theta\,\Delta t}{\rho}\left(\frac{\partial p}{\partial x_i}\right)_c^{(1)} \tag{9a,b}$$

$$A_c\left(U_i\right)_c^{(m)}+\sum_f A_f\left(U_i\right)_f^{(m-1)}=-\frac{\theta\,\Delta t}{\rho}\left(\frac{\partial p}{\partial x_i}\right)_c^{(m)},\ m=2,3,\ldots \tag{9c}$$

The formula (9a) is a momentum balance equation based on approximated values of pressure gradient $(\partial p/\partial x_i)_c^{*}$ and velocity components $(U_i)_c^{*}$ (i = 1, 2). The term $(r_i)_c^{k,*}$ is a linkage of two right hand side terms from equation (6). A standard approach derived from SIMPLE-like schemes is to assume the values of pressure first, then to determine the velocity components from (9a). The equation (9b) is a relationship between the first corrections of velocity components $(U_i)_c^{(1)}$ (i = 1, 2) and the gradient of pressure correction $(\partial p/\partial x_i)_c^{(1)}$. The equation (9c) is the same relation written for subsequent corrections. The important element is the dependence of equation (9c) on previous corrections $(U_i)^{(m-1)}$ (i = 1, 2) seen as second term on the left-hand side.

For the sake of simplicity, the following parameters are introduced:

$$\left(w_p\right)_c=\frac{\theta\,\Delta t}{\rho\,A_c} \qquad \left(\tilde{U}_i\right)_c^{(m-1)}=\frac{1}{A_c}\sum_f A_f\left(U_i\right)_f^{(m-1)} \qquad a_p=\frac{\beta}{a^2\,\Delta t} \tag{10a,b,c}$$

The first two are calculated in the cell centers, but may be easily interpolated in nodes or faces. The third is constant in the whole domain. The above mentioned approach enables us to write the corrections in arbitrary location as follows:

$$u_i^{(1)}=-w_p\left(\frac{\partial p}{\partial x_i}\right)^{(1)} \qquad u_i^{(m)}=-\tilde{u}_i^{(m-1)}-w_p\left(\frac{\partial p}{\partial x_i}\right)^{(m)} \tag{11a,b}$$

The corrections are inserted into the continuity equation (7), which is split then. The results are subsequent equations for pressure corrections:

$$a_p\,p^{(m)}-\theta_0\,\rho\frac{\partial}{\partial x_i}\left[w_p\left(\frac{\partial p}{\partial x_i}\right)^{(m)}\right]=-r_0^{(m-1)} \qquad m=1,2,\ldots \tag{12}$$

In the first level m = 1, right-hand side $r_0^{(0)}$ includes all elements depending on values from previous time step, u_i^{k} (i = 1, 2) and p^{k}, or approximated solutions, u_i^{*} (i = 1, 2) and p^{*}. During all next levels $m \geq 2$, $r_0^{(m-1)}$ represent all the values calculated on the basis of previous corrections, $u_i^{(m-1)}$ (i = 1, 2) and $p^{(m-1)}$.

The FVM approximation of (12) may be written as follows:

$$a_p\int_{F_n} p^{(m)}\,dF-\theta_0\,\rho\int_{\sigma_n} w_p\left(\frac{\partial p}{\partial x_i}\right)^{(m)} n_i\,d\sigma=-\int_{F_n} r_0^{(m-1)}\,dF \tag{13}$$

The equation (13) involves the integration of a single equation for pressure correction. The integration is done over node-centered volume schematically shown in fig. 1c. The node has to be located inside the domain. For boundary nodes, proper equations are constructed from boundary conditions. The divergence theorem is used to change the area integral of pressure correction laplacian into the curvilinear integral of pressure correction gradient. The fluxes in node-centered values are calculated by splitting into orthogonal and non-orthogonal parts.

After approximations, the single algebraic equation may be written for each node:

$$A_n\,P_n^{(m)}-\sum_{af} A_{af}\,P_{af}^{(m)}=RHS_n^{(m-1,m)} \tag{14}$$

Small letter n denotes the node number. The A_n and A_{af} coefficients are derived from (13). The subscript "*af*" means the face between two node-centered volumes. Such denotation is used to differentiate it from the "*f*" used in (6). $P_n^{(m)}$ is the value of the m-th pressure correction in node n. The values $P_{af}^{(m)}$ are the m-th pressure correction in neighboring nodes. $RHS_n^{(m-1,m)}$ is

right-hand side calculated on the basis of two different elements. These are: (1) the values from the previous iterations, (2) the values arising from the splitting of second order derivatives in faces.

It is important to notice the structure of the main diagonal element shown below:

$$A_n = a_p F_n + \sum_{af} A_{af} \tag{15}$$

It is a sum including off-diagonal elements and one additional term related to weak-compressibility according to (10c). It means that the main system matrix is diagonally dominant if coefficient β is zero, representing the basic incompressible flow. If coefficient β is non-zero, the main matrix is strictly diagonally dominant.

The main elements of the algorithm

The algorithm is composed of the elements described above. Several linear systems have to be solved during a single time step. The iterative algorithm MGMRES is used for each such problem. The algorithm was prepared by Lily Ju in C programming language and translated into Fortran 90 by John Burkardt (Burkardt, 2008). The algorithm is based on the well-known GMRES method (Barrett et al., 1994). The method is fast and easy for implementation. The code is distributed under the GNU LGPL license.

In general, there are two types of linear systems in the presented problem. The first is a momentum system composed of equations (9a) with boundary conditions. The system consists of $2(N_c+N_b)$ equations, where N_c is the number of cells and N_b is the number of boundary lines. The unknowns are velocity components $(U_1, U_2)_c$ for each cell c ($c = 1, 2, \ldots N_c$) and velocity components $(v_n, v_s)_b$ for each boundary b ($b = 1, 2, \ldots N_b$). The second type of linear system is prepared for determination of pressure corrections. This system is composed of equations (14). Such system consists of N_n equations, where N_n is a number of nodes, including internal as well as boundary nodes. The unknowns are pressure corrections $P_n^{(m)}$ for each node n ($n = 1, 2, \ldots N_n$).

The main elements of the simulation process are shown in a simplified form in fig. 2. Only the steps necessary for the explanation of computations are presented there. The pre-processing, results saving, convergence monitoring as well as post-processing are omitted. The basic elements are marked as figures with solid edges and they are discussed first. The optional processes marked as figures with discontinuous edges are explained in the next section.

Before the time loop starts, the initial condition is calculated on the basis of previously loaded data. k is time level counter equaling zero for the initial condition. The initial condition includes velocity components in cells $(U_1^k, U_2^k)_c$, velocity components in boundaries $(v_n^k, v_s^k)_b$, and pressure in nodes P_n^k.

The time loop includes several elements. The t variable represents simulation time. Δt is a computational step. The main elements of time loop are: (1) the determination of the approximated pressure P^*, (2) the restart loop and (3) the stop criterion used to finish the simulation. For the first step, the values from the previous time level P^k are taken. The restart loop with counter r begins after that. N_{res} is the number of required restarts or the maximum number of restarts. The difference between these two values is explained below. After the restart loop, the condition for the continuation of computations is checked. If the time horizon $T_{\max}$ is not reached, the next time level starts. Before, the time loop counter k is increased.

The restart loop includes the solution to several equations systems. The first is a momentum system (9a) with the boundary conditions. This block implicitly includes the preparation of main matrix, the calculation of right-hand side and the additional processes if they are necessary. The additional processes are related to the non-linearity of convective

terms and diffusive flux splitting. The main element of this block is the implementation of the MGMRES algorithm. The result is the set of approximated velocities $(U_1^*, U_2^*)_c$ and $(v_n^*, v_s^*)_b$.

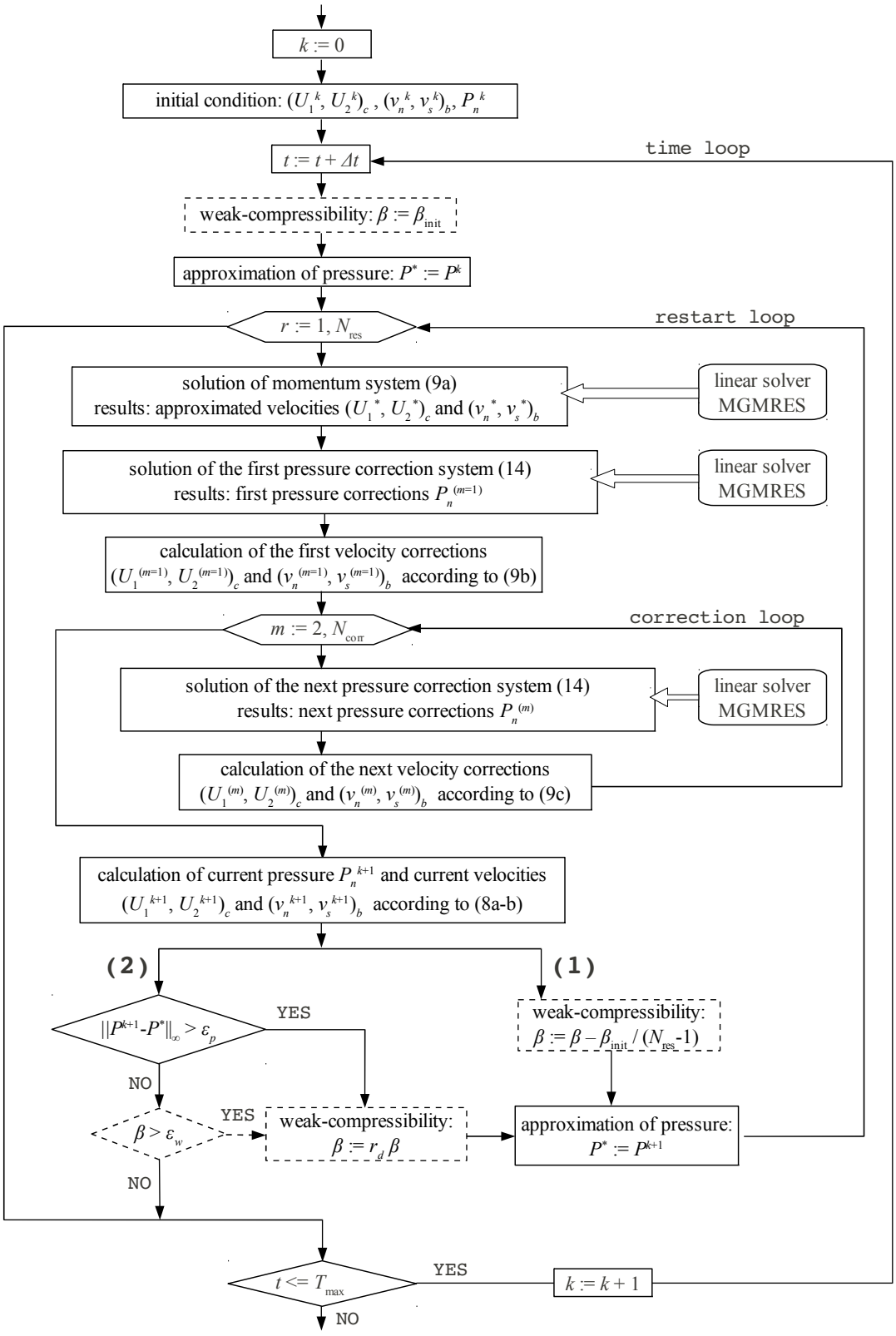

Fig. 2 The main elements of the simulation process (see: description in the text)

Solving the system (14) in order to indicate the first pressure corrections is the next step of the restart loop process. In this step, the general matrix for all pressure correction computations is prepared. Also the right-hand side for (14) is determined. The block may include iterative refinement of the right-hand side due to the splitting of second derivatives in the Poisson equation (14). In the analyzed cases, only the orthogonal meshes are used, hence there is no need for such additional iterations. The MGMRES algorithm is implemented here, too. The result is the set of pressure corrections $P_n^{(m=1)}$ determined for each node. The first correction level is completed by a calculation of velocity corrections $(U_1^{(m=1)}, U_2^{(m=1)})_c$ and $(v_n^{(m=1)}, v_s^{(m=1)})_b$.

The loop for next corrections is active only if the number of required corrections N_{corr} exceeds one. The correction counter is m. The loop includes only two steps. The first is a solution of an equation system (14) for the m-th pressure correction. It includes the calculation of the right-hand side for the m-th pressure corrections, but it does not include a preparation of the main matrix. It is done in the preceding calculation of the first pressure corrections. This block may also include iterative refinement of the right-hand side due to the splitting of second derivatives as it is mentioned above. The MGMRES algorithm is used once again and the solution is the set of pressure corrections $P_n^{(m)}$. The next element is a calculation of the (m-th) velocity corrections $(U_1^{(m)}, U_2^{(m)})_c$ and $(v_n^{(m)}, v_s^{(m)})_b$.

When the loop over corrections is finished, the current pressure P_n^{k+1} and current velocities $(U_1^{k+1}, U_2^{k+1})_c$ and $(v_n^{k+1}, v_s^{k+1})_b$ are determined according to (8a-b). Some under-relaxation may be used in this step like in the SIMPLE scheme. Then the current values of variables are corrected according to

$$p^{k+1} = p^* + \alpha \sum_m p^{(m)} \qquad u_i^{k+1} = u_i^* + \alpha \sum_m u_i^{(m)} \tag{16a,b}$$

where $\alpha \leq 1$ is the under-relaxation parameter.

Before the next restart, the new approximated pressure P^* is assumed as equal to the current pressure P^{k+1}. If such a restart is necessary, the procedure comes back to the solution of momentum system (9a). Otherwise, the current time level is finished.

In the very basic form of the algorithm, the number of restarts N_{res} is fixed. This form is represented by path (1) in fig. 2. A more extended version of the algorithm allows for the adaptive adjustment of the restart number. It is presented as path (2) in fig. 2. Adaptation is based on the monitoring of total pressure corrections in the previous restart. The results of numerical experiments conducted proved that this is the most crucial element in the whole procedure. The restarts are stopped if following condition is satisfied:

$$\|P^{k+1} - P^*\|_\infty \leq \varepsilon_p \tag{17}$$

where ε_p is the tolerance to total pressure correction. The norm in the left side of (17) means the maximum sum of corrections from m levels in a single node. If adaptation is applied, the parameter N_{res} plays the role of the upper limit for restarts.

It is worth noticing that the number of restarts N_{res} and the number of correction N_{corr} are also used to control the algorithm performance and meaning. If N_{corr} is 1 and several restarts are required ($N_{\text{res}}>1$), the algorithm becomes a SIMPLE scheme. In such a case, the under-relaxation is recommended. If the number of corrections N_{corr} is 2 and N_{res} is 1, we obtain a classical PISO scheme. It is visible that the construction of the algorithm allows for the implementation of a more general method with the combination of several restarts and corrections.

The application of vanishing compressibility

There are additional elements in fig. 2, which are not discussed in the previous section. These are blocks with dashed edges including elements relating to the weak-compressibility coefficient β. They are not active in the basic incompressible or weakly-compressible flow

simulation, because β is constant in such a run. These blocks represent the main idea of the presented paper, because they contain instructions for the control of the weak-compressibility value.

At the beginning of the restart loop, the weak-compressibility coefficient β is equal to the initial value denoted as β_{init}. The default value is one but it may also be loaded as the input data. After each restart, the value of weak-compressibility coefficient β is decreased. This is done in two ways marked as path (1) and (2) in fig. 2. In path (1), the number of restarts N_{res} is fixed. In such case, the decrement of weak-compressibility is a constant value equal to $\beta_{\text{init}} / (N_{\text{res}} - 1)$. In such a way, the value of β is zero at the end of the restarting process. Path (2) represents a case involving an adaptive number of N_{res} restarts. For such a problem, the r_d ratio of decrease is defined. This parameter is in the range [0, 1). The weak-compressibility β is multiplied by r_d after each restart, which gives a geometric decrease. The restarts are stopped if condition (17) is satisfied and the weak-compressibility is small enough. In mathematical terms, this requirement is written as follows:

$$\beta \leq \varepsilon_w \tag{18}$$

where ε_w is tolerance to weak-compressibility.

The idea presented here utilizes the idea of multiple restarts used in the SIMPLE-like schemes. The whole procedure is similar to computational methods applied as the artificial compressibility method (ACM), but the use of incompressible flow solver allows for stepping further without the loss of stability. The final computations are done for fully incompressible flow equations with coefficient β equaling zero in path (1) or almost equaling zero in path (2).

Because the value of β is smoothly decreasing from the initial value to zero during the subsequent restarts, the term “vanishing compressibility” is used to describe such a procedure.

NUMERICAL EXPERIMENTS

The description of chosen test cases

The results of computational tests are presented in this section. They are described to prove the usefulness and efficiency of the proposed concept. There are two chosen test cases: (a) the lid-driven cavity flow, (a) the flow over the backward facing step. They are schematically shown in fig. 3a-b. Several tests are described in literature, where such examples are successfully used, e.g. Barton (1998), Tu & Alibadi (2007), Sun et al. (2009), Gao & Liu (2009). The fluid with constant physical properties is chosen. The parameters of the fluid are presented in tab. 1. The first column is the name of the parameter, where the second presents its symbol. The dimensions in third column are presented independently of the unit system. M means the mass unit, L is the length unit, and T is time. This convention is also used to present other values. The values of the parameters are given in the last column.

Tab. 1
Fluid parameters applied in all tests

parameter	symbol	unit	value
density	ρ	ML^{-3}	1000
viscosity	v	L^2T^{-1}	1
speed of sound	a	LT^{-1}	1500

The domain for the simulation of the lid-driven cavity flow is a basic square. The size of the square as shown in fig. 3a is $L = 1$ [L]. The monitoring point used in the further analyses is marked there. Its location is determined by the distances from the domain boundaries, which are as follows: $l_1 = l_2 = L/2$. The flow is induced by the tangential movement in the

upper boundary. The rest of the boundaries are non-moving walls. There is no inflows and outflows. The initial condition is lack of movement.

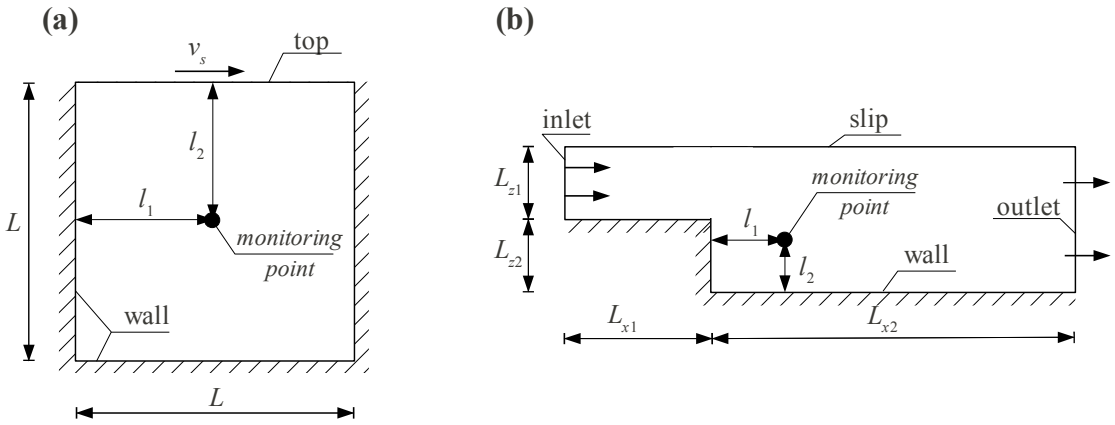

Fig. 3a,b The schematic view of chosen test cases:
(a) the lid-driven cavity flow, (b) the backward facing step

The coefficients ω_ϕ, λ_ϕ and Ψ_ϕ in (3) are specified in such a way that the boundary conditions shown in tab. 2 are imposed. u_0 is imposed velocity in the top. The value of u_0 depends on the test preformed. The tests are described below and their parameters are presented in tab. 4.

Tab. 2
Boundary conditions for the lid-driven cavity flow

boundary	normal velocity	tangential velocity	pressure
top	$v_n = 0$ [LT^{-1}]	$v_s = u_0$ [LT^{-1}]	$p = 0$ [ML^{-1}T^{-2}]
wall	$v_n = 0$ [LT^{-1}]	$v_s = 0$ [LT^{-1}]	$\partial p/\partial n = 0$ [ML^{-2}T^{-2}]

The second example is a case with inflow and outflow sections as well as some asymmetries in geometry. The dimensions shown in fig. 3b are as follows: $L_{x1} = 4$ [L], $L_{x2} = 10$ [L], $L_{z1} = L_{z2} = 2$ [L]. The distances of the monitoring point from boundaries are $l_1 = 2$ [L] and $l_2 = 1.8$ [L]. The initial condition is imposed in the same way as in the previous example.

The ω_ϕ, λ_ϕ and Ψ_ϕ coefficients in (3) are specified in such a way that the boundary conditions shown in tab. 3 are imposed. This time, u_0 is the velocity imposed in the inlet. Like previously, the value of u_0 depends on the test preformed.

Tab. 3
Boundary conditions for the backward facing step

boundary	normal velocity	tangential velocity	pressure
inlet	$v_n = u_0$ [LT^{-1}]	$v_s = 0$ [LT^{-1}]	$\partial p/\partial n = 0$ [ML^{-2}T^{-2}]
outlet	$\partial v_n/\partial n = 0$ [T^{-1}]	$\partial v_s/\partial n = 0$ [T^{-1}]	$p = 0$ [ML^{-1}T^{-2}]
wall	$v_n = 0$ [LT^{-1}]	$v_s = 0$ [LT^{-1}]	$\partial p/\partial n = 0$ [ML^{-2}T^{-2}]
slip	$\partial v_n/\partial n = 0$ [T^{-1}]	$\partial v_s/\partial n = 0$ [T^{-1}]	$\partial p/\partial n = 0$ [ML^{-2}T^{-2}]

Three computational tests are performed for each case. They are denoted as test-X, where X is the number of the test. The Gmsh package (Geuzaine & Remacle, 2009) is used as pre-processor and mesh generator. The structured grid was prepared for each case. The dimension of the grid cell is h. The main parameters of each test are time step Δt, mesh resolution h^2 and the imposed velocity u_0. The computations are performed until steady state is achieved.

Tab. 4
Specific parameters selected for tests

parameter	symbol	unit	test-1	test-2	test-3
time step	Δt	T	10^{-4}	10^{-4}	10^{-4}
mesh resolution	h^2	L^2	0.02×0.02	0.02×0.02	0.04×0.04
imposed velocity	u_0	LT^{-1}	10	100	10

To assess the differences between the tests performed, three dimensionless numbers are used. These are Reynolds number *Re*, Mach number *Ma* and Courant-Friedrichs-Levy condition *CFL*. The adopted estimation of their values is presented below

$$Re = \frac{u_0 L_c}{\nu} \qquad Ma = \frac{u_0}{a} \qquad CFL = \frac{u_0 \Delta t}{h} \tag{19a,b,c}$$

The first two, *Re* and *Ma*, relate to the physical parameters of a fluid and flow characteristics. The L_c in the definition of Reynolds number *Re* is a characteristic length of the problem. In case of the lid-driven cavity flow, the L size of the domain is used. In the second case, the L_{z2} height of the step is applied. The Courant – Friedrichs – Levy condition *CFL* is a numerical measure providing the information on method convergence and stability.

The Mach number *Ma* does not exceed the value 10^{-01} in all tests performed . It means that the condition of low Mach number flow is satisfied. The Reynolds number *Re* is equal 10 in test-1 and test-2 for lid-driven cavity flow. This number is equal 20 in the same tests for flow over backward facing step. The Reynolds number *Re* is greater for test-2 for both lid-driven cavity flow and flow over backward facing step, equaling 100 and 200, respectively. All the tested cases are examples of laminar flow. The *CFL* condition varies from 0.025 in test-3 to 0.5 in test-2. This criterion is equal to 0.05 in test-1. As it is visible, the small values of *CFL* condition guarantee the stability of simulations presented.

The simulation results

Four models are tested. The basic incompressible flow model with a classical PISO scheme is used as a reference model for all comparisons. The choice of the scheme guarantees the greatest efficiency reported today (e.g. Ferzinger & Peric, 2002). This model is denoted as the ICF model. On the opposite end, there is the weakly-compressible flow model denoted as the WCF model. In this case, the pressure-velocity coupling with the same PISO scheme is implemented. In both cases, the elements with changes of weak-compressibility ratio shown in fig. 2 are inactive.

The idea of vanishing compressibility is incorporated in the structure of next two models. They are denoted by the VC letters relating to the 'vanishing compressibility' expression. Both models are used with fixed and adaptive numbers of restarts. The difference between them is the number of correction levels used. So, the models denoted as VC 2x1 or VC Rx1, are vanishing compressibility models with one correction level. The number of restarts is fixed and equals 2 in the first case, or it is the adaptive R in the second. The next model is denoted as VC 2x2 or VC Rx2, what means that two correction levels are used. The number of restarts applied is the same as in the previous case.

The examples of results obtained from flow simulations are shown in fig. 4-9. They are presented as streamlines. The differences between such results of the four tested models are slightly visible in the scale used in this paper. Hence, only the results of the ICF and WCF models are shown as examples. The results of vanishing compressibility models differ from the IFC model too insignificantly to present them.

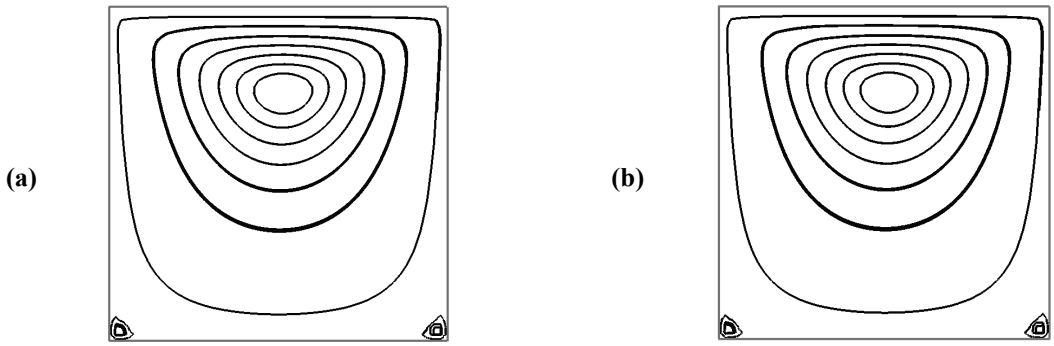

Fig. 4a,b The results of test-1 for the lid-driven cavity flow: (a) the ICF model, (b) the WCF model

Fig. 4 shows the results of test-1 for the lid-driven cavity flow. The drawing on the left presents results from the ICF model simulations. On the right, the results of the WCF model are seen. We can observe a specific fluid circulation in the flow domain. The imposed velocity u_0 is small ($Re = 10$), hence the whole picture seems to be symmetric. In the bottom corners, the small circulation zones are observed.

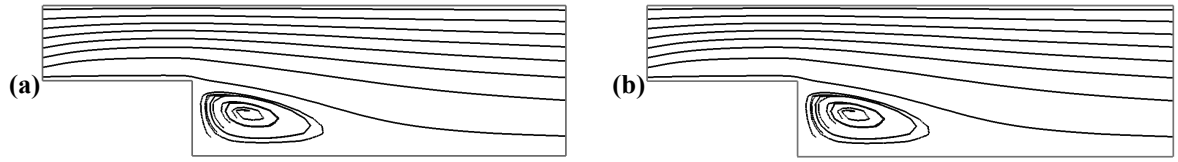

Fig. 5a,b The results of test-1 for the backward facing step: (a) the ICF model, (b) the WCF model

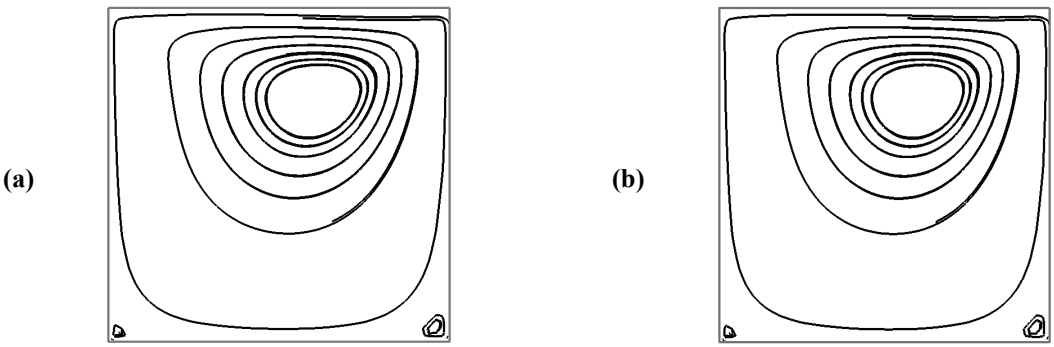

Fig. 6a,b The results of test-2 for the lid-driven cavity flow: (a) the ICF model, (b) the WCF model

In a similar way, the results of test-1 for the flow over the backward facing step are presented in fig. 5. On the left, the results of the ICF model simulation are shown. On the right, the WCF model is presented. The characteristic circulation zone below the step is shown. Because the inflow velocity u_0 is relatively small ($Re = 20$), the circulation zone is not long. The rest of the fluid flow over the zone in the part of the domain located downstream of the step. The fluid gradually extends to the whole accessible width.

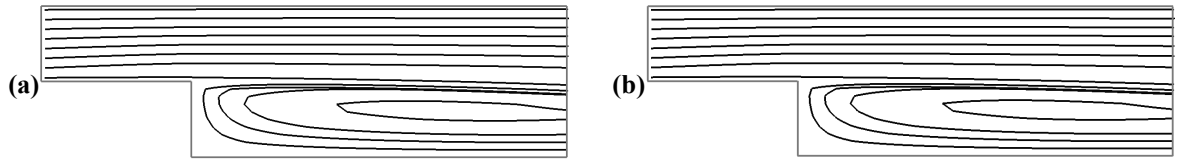

Fig. 7a,b The results of test-2 for the backward facing step: (a) the ICF model, (b) the WCF model

The same method is used to present the results of test-2 for the lid-driven cavity flow in fig. 6. This time, the imposed velocity u_0 is much greater. The asymmetry in streamlines and pressure contour lines is clearly visible. The main circulation zone is shifted to the right. The right bottom circulation zone is significantly greater that the left one.

Fig. 7 shows the results of test-2 for the flow over the backward facing step. The circulation zone below the step is much greater then previously.

The results for the test-3 are not shown here due to their similarity to the results of test-1. The coarse grid is used in this test. The only differences are sharper shapes of streamlines in both tested cases.

It is worth noticing that all the result post-processing is done in the Gmsh package (Geuzaine & Remacle, 2009).

The assessment of efficiency

The applied concept of vanishing compressibility is used to improve the convergence of the incompressible flow solver. Hence, the assessment of efficiency is based on the comparison of the results obtained thanks to the application of this concept with the results from the simulation of the ICF model described earlier. The results from the WCF model trials are also presented for a better recognition of the problem.

Two measures are used for the comparison of the results. The first one is the number of the MGMRES iterations for the solution of the Poisson equations for pressure corrections. It is reported in literature (e.g. Schafer, 2006) and observed by the author of the paper that this element of flow simulation is the most time-consuming. The system of equations for approximated velocities (9a) is greater, but simpler to solve. The number of the MGMRES iterations implemented in this case is much smaller that the same number in the determination of pressure corrections. Hence, the convergence speed is assessed on the basis of the total MGMRES iteration needed for the solution of pressure correction systems in a single time level.

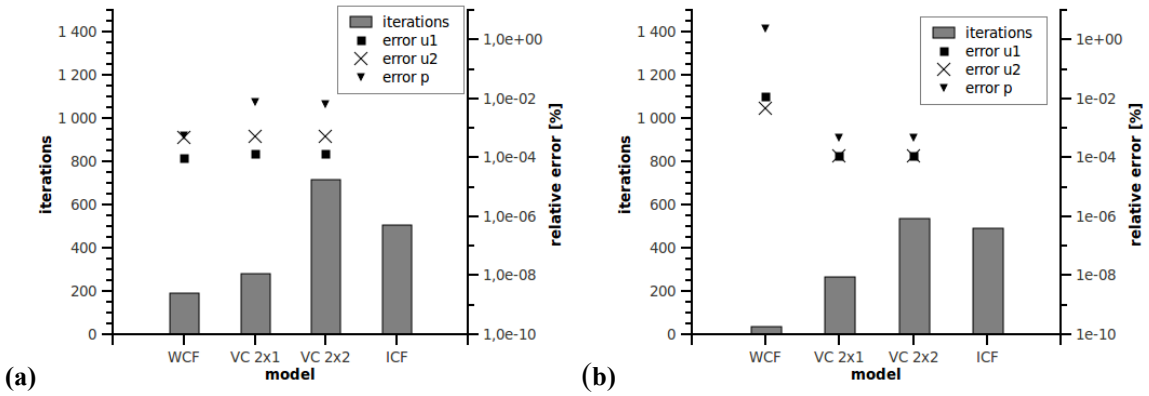

Fig. 10a,b Numbers of iterations and relative errors δ_ϕ for test-1:
(a) the lid-driven cavity flow, (b) the flow over the backward facing step

The second measure is used to assess the accuracy of the tested model. Due to the purposes of the research, the term 'accuracy' is understood as a deviation from the ICF model results. The deviations are calculated for monitoring points marked in fig. 3a and fig. 3b.The definitions shown below are used

$$\Delta_\phi = \frac{1}{K_s}\left[\sum_{k=1}^{K_s} |\phi^k_{mod} - \phi^k_{ICF}|\right] \qquad \delta_\phi = \frac{\Delta_\phi}{R_\phi}\cdot 100\,\% \tag{20a,b}$$

Δ_ϕ is the average deviation for a single time step, where ϕ is one of velocity components u_1, u_2 or pressure p. K_s is a number of time steps from the beginning of the simulation to simulation horizon such that steady state is achieved. $\phi^k{}_{mod}$ is the result form a particular model '*mod*' in time level k, when $\phi^k{}_{ICF}$ is the same result from the ICF model. The second definition (20b) is the ratio of deviation from the ICF model results Δ_ϕ and the total range of variability R_ϕ for a particular variable ϕ. This value is referred to as the relative error δ_ϕ and is expressed as percentage of the variability range R_ϕ. This range is defined as the difference between the maximum and minimum value calculated in the ICF model run:

$$R_\phi = \max_{k=1,K_s}\left(\phi^k_{ICF}\right) - \min_{k=1,K_s}\left(\phi^k_{ICF}\right) \tag{21}$$

The obtained results are presented in graphs (fig. 10-15). In each graphs horizontal axis represent model tested. These are weakly-compressible model WCF, two vanishing compressibility models VC with different numbers of correction levels and incompressible model ICF. The left vertical axis represent average number of MGMRES iteration discussed above. The right axis represents values of δ_ϕ defined by (20b).

The results for test-1 with a fixed number of restarts are presented first (fig. 10). If the vanishing compressibility is implemented, the computations are done according to path (1) in fig. 2. In both cases, the fastest one is the WCF model. It occurs 62.21 % faster than the ICF model in the case of the lid-driven cavity flow and even 92.80 % faster for the second case. In the same cases, the VC 2x1 model is 44.31 % and 46.31 % faster, respectively. The slowest model in these tests is the VC 2x2. It is 41.37 % slower than the ICF model for the lid-driven cavity flow and 8.44 % slower for the backward facing step. The differences between the ICF model and other models show the advantage of vanishing compressibility models over the WCF model for the second case. However, the results are not so unequivocal in case of the lid driven cavity flow. The nature of this problem and a fixed number of restarts may be an explanation for such unexpected values.

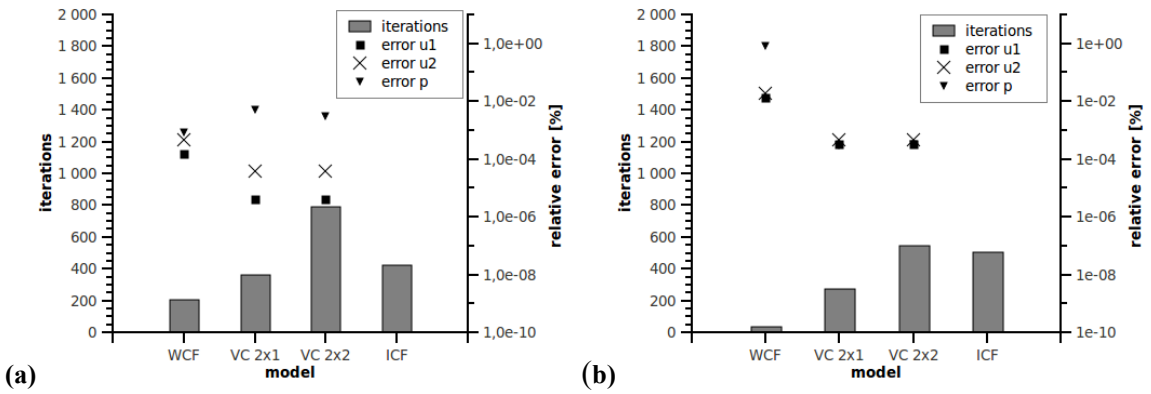

Fig. 11a,b Numbers of iterations and relative errors δ_ϕ for test-2:
(a) the lid-driven cavity flow, (b) the flow over the backward facing step

Similar tendencies are observed for test-2 (fig. 11) and test-3 (fig. 12) with a fixed number of restarts. In test-2, the WCF model is 51.73 % and 92.93 % faster than ICF model. In test-3

its speed is 53.84 % and 90.51 % better. The second fastest model is the VC 2x1 one. Its convergence is 14.31 % and 45.87 % better than that of the ICF model in test-2. For test-3, the results of the VC 2x1 model are obtained 27.83 % and 44.88 % faster. The slowest model is the VC 2x2 once again. The differences from the ICF model results are smaller for the vanishing compressibility models than for the WCF model in case of the flow over the backward facing step. In case of the lid-driven cavity flow, slightly better results are observed for weakly compressible flow model.

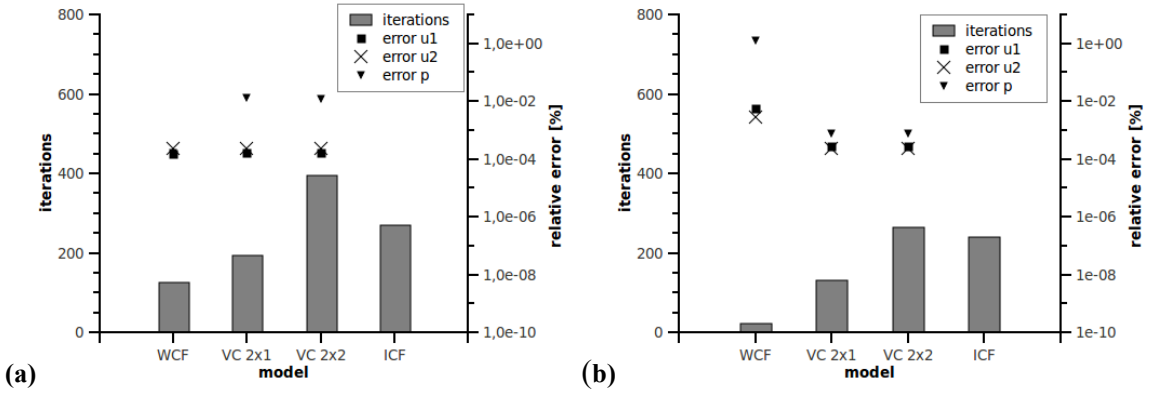

Fig. 12a,b Numbers of iterations and relative errors δ_ϕ for test-3:
(a) the lid-driven cavity flow, (b) the flow over the backward facing step

Slightly different results are obtained for tests with an adaptive number of restarts. They are shown in fig. 13-15. Although the number of iterations is greater, the WCF model is still the fastest one, in general. An increase in the convergence speed varies from 50.88 % to 95.55 % in comparison with the ICF model. The best results are obtained for the flow over the backward facing step in test-2, which is the most difficult one. The second in this competition is the VC Rx1 model. The speed of convergence is from 20.43 % to 58.95 % better that of ICF model. In one case, the VC Rx1 occurred faster than the WCF model, but it happened in test-3 for the lid-driven cavity flow. In this test, the mesh resolution is coarse and the results of the simulation are inaccurate. The results of the VC Rx2 are a little bit surprising. This model is not slower, but faster than the ICF model in all tests with adaptive number of restarts. The vanishing compressibility models are much better than the WCF model if the differences with the IFC model are taken into account. In some test with adaptive number of restarts, a perfect match of the VC models with the ICF model is achieved.

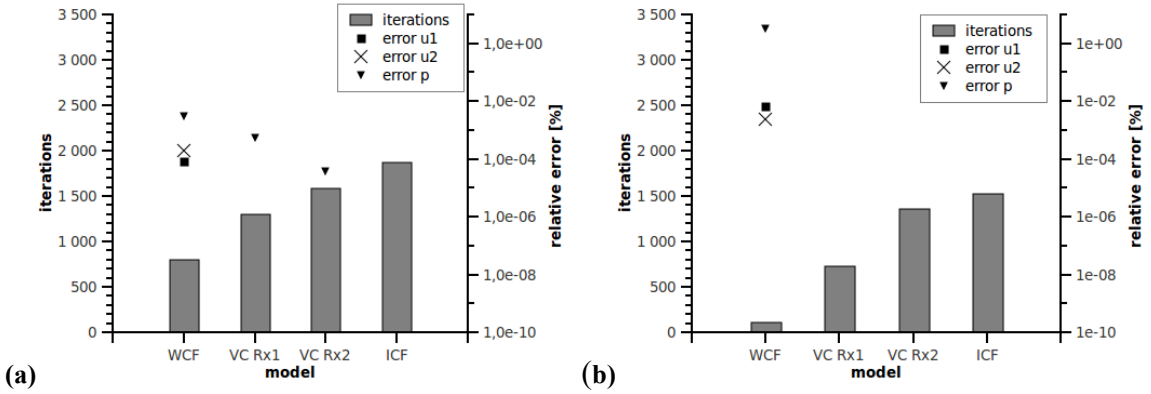

Fig. 13a,b Numbers of iterations and relative errors δ_ϕ for test-1 with an adaptive number of restarts:
(a) the lid-driven cavity flow, (b) the flow over the backward facing step

It is also observed that the number of restarts in the VC Rx2 model is smaller than the number of restarts in the VC Rx1 model, in general. For example, the number of restarts in the VC Rx1 run for test-2 of the lid-driven cavity flow oscillates around 25 per time level. For the same case, the run of VC Rx2 gives about 17 restarts in a single time level. Hence, the number of total iterations for the pressure correction system becomes closer in these two models.

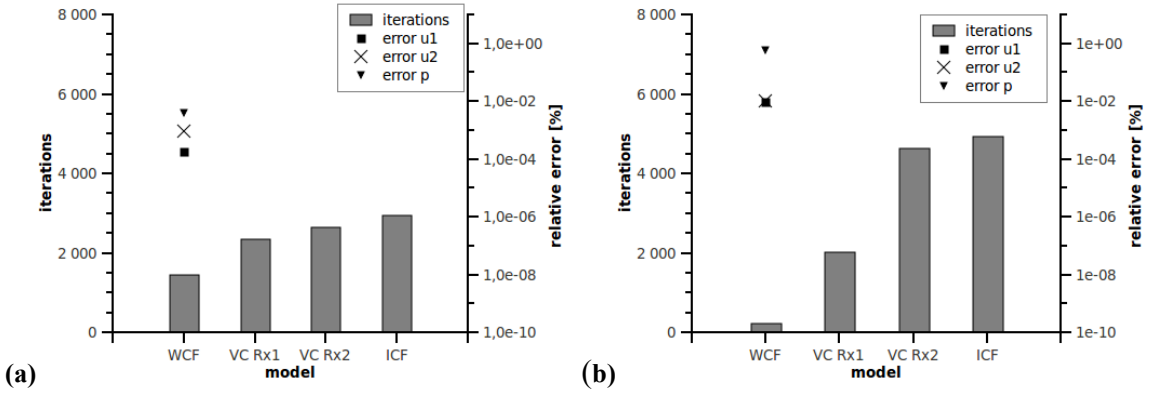

Fig. 14a,b Numbers of iterations and relative errors δ_ϕ for test-2 with an adaptive number of restarts: (a) the lid-driven cavity flow, (b) the flow over the backward facing step

The convergence of vanishing compressibility models depends strongly on two parameters of the algorithm. The first is under-relaxation α shown in (22a) and (22b). This parameter is used only in the VC Rx1 model. The values used in tests vary from 0.6 to 0.9. The second is a ratio of weak-compressibility decrease r_d presented in fig. 2 and explained earlier. This parameter is important in both the VC Rx1 and the VC Rx2 models. The values which provided the best results presented here are relatively small, in the range between 10^{-03} to 10^{-02}. Hence, the weak-compressibility decreases quickly in tests with an adaptive number of restarts.

Discussion

The acceleration of convergence by linking the vanishing compressibility with the pressure-velocity coupling may be explained by two elements in the structure of the proposed algorithm. These are (1) the change in the diagonal domination of the main matrix in the pressure correction system and (2) the determination of a better guess for the iterative refinement of velocity and pressure corrections. The first element is clearly visible in formula (16). The term a_p including weak-compressibility may change the diagonally dominated matrix into a strictly diagonally dominant matrix. Although the GMRES method implemented in the presented algorithm is not limited to the systems with diagonally dominant matrices, it is shown that the method works better if a_p is non zero. A stricter diagonal domination is the only element which marks the difference between the weakly-compressible and the incompressible models.

In case of the vanishing compressibility models, the strict diagonal domination is replaced by a basic domination during subsequent restarts. However, the convergence is still good due to the next element. It is obvious that a better first guess in the convergent iterative process decreases the number of necessary steps. In the analyzed cases, the better first guess is provided by the first solution of the system with a strictly dominant matrix. These two elements complement each other.

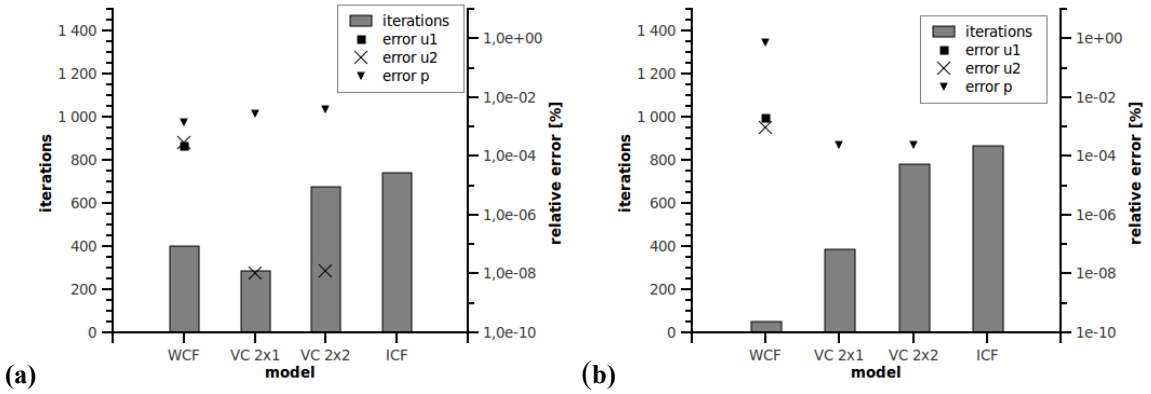

Fig. 15a,b Numbers of iterations and relative errors δ_ϕ for test-3 with an adaptive number of restarts: (a) the lid-driven cavity flow, (b) the flow over the backward facing step

The deeper analysis of computations suggests some further improvements of the convergence speed. The quick decrease of the weak-compressibility ratio in tests with an adaptive number of restarts shows that the effects related to weak-compressibility are needed only at the beginning of the iterative process. It leads to a remark that a time-consuming solution of the linear systems for approximated velocity components and pressure correction may be replaced by the application of an ODE solver. Such approach is closer to the solution of the so-called density-based systems and a simulation of compressible or artificially compressible fluids. It adapts the structure of the basic equations (2a) and (2b) to their time derivatives.

Unfortunately, in such an approach, there are two problems, which may be shortly called (1) stability and (b) complexity. The ODE solver used should provide stable results, which is not so simple in the fluid flow simulations. It leads to the application of implicit ODE solvers or more sophisticated explicit methods, such as the Runge-Kutta stepping. And this causes the second problem, which is a complexity of the algorithm. The application of such a method may increase the convergence slightly, but increases the complexity of the algorithm significantly. Hence, the approach described is not recommended, supposedly.

Another element seen in computations is the difference between the number of restarts in two vanishing compressibility models. Also the better results of the VC Rx2 model in these tests suggest that such a model may be recommended for more complex cases. There is a temptation to increase the number of correction levels. Very basic tests done by the author of the paper with the number of corrections greater than two showed that such an approach does not improve anything. This remark is consistent with the results reported in literature, e.g. Ferzinger & Peric (2002).

Another opportunity for the improvement of convergence seems to be a better approximation of corrections in the last correction level. This concept is an extension of the SIMPLEC scheme, which could be applied only for the last correction level in the presented algorithm. Although the simulations reported in the literature suggest a faster convergence of the SIMPLEC method in comparison with the SIMPLE scheme, e.g. Schafer (2006), Ferzinger & Peric (2002), it is difficult to estimate the possible effects observed in the cases analyzed. The implementation of this idea requires a small change in the calculation of the main matrix coefficient pressure correction systems. As it is noticed, the important thing is the impact of the change on the diagonal domination in the matrix. It also requires a double calculation of that matrix. The first one is prepared for all the correction levels but for the last, the second is used only in the last level. It means that one of the advantages of the presented method based on a very simple preparation of the pressure correction system is lost. As a result, the complexity of the algorithm increases. The achieved improvement in the

convergence speed expressed in the MGMRES iterations may be lost by more computations for the preparation stages. Hence, the idea is not implemented here.

Other opportunities for improvements may lead to a different linear solver, parallel computing or adaptive adjustment of algorithm parameters. All these ideas may be implemented, but the complexity of the algorithm increases in almost all cases. If such techniques are used, the assessment of effects caused by week-compressibility could be more complex. Hence, the incorporation of this element into a relatively basic algorithm with a serial solver seems to be the optimal choice for the purposes of the presented research. The interpretation of the results is clear, but the opportunities for future improvements are still open.

CONCLUSIONS

In the paper, the construction of the proposed algorithm is presented. Basic elements are discussed. The title of the paper and its main ideas are explained on that basis. The pressure-velocity coupling is applied in the weakly-compressible model. The model is modified to control the convergence of the computations. The vanishing compressibility mechanism is introduced instead of fixed and constant weak-compressibility. The algorithm is prepared in two versions, namely with a fixed and adaptive number of restarts.

The idea is tested with the use of standard numerical tests. These are the lid-driven cavity flow and flow over backward facing step used. The results are provided for the vanishing compressibility models as well as two models used for comparisons, namely the incompressible and weakly-compressible flow models. In the tests mentioned, different flow conditions and mesh resolutions are used. The results obtained for the analysed cases are typical. A comparison with the results in literature does not show any significant differences. It leads to a conclusion that algorithms are prepared in a correct way.

The application of the pressure-velocity coupling in the weakly-compressible flow equations is not a standard procedure. It requires special treatment of the term with the derivative of pressure in a continuity equation. The procedure based on a multilevel PISO scheme is implemented. It proves that such a flow model may be used independently of the complexity of the pressure-velocity coupling algorithm. As it is shown, the presented approach provides stable results with very fast convergence. The efficiency of the weakly-compressible flow model is at its best when compared with other tested models. It is much faster than the standard incompressible flow model with the classical PISO scheme. However, the results obtained differ from the incompressible flow model results.

The vanishing compressibility models seem to be well suited for the problems presented. They are not as fast as the weakly-compressible model, but much faster than the incompressible one. The reasons for such an improvement are explained and they seem to be independent of the flow problem analyzed. The results provided by the vanishing compressibility models slightly differ from or equal the results of the incompressible flow model. So, if the incompressible flow model is well suited for some problems, the vanishing compressibility models may be used successfully in such a case, too. The difference in results is not significant, but the convergence is much faster. It means a lot of computational time to be saved.

The vanishing compressibility models presented may be considered a combination of the pressure-velocity coupling method with the artificial compressibility method. Such a combination gives a stable algorithm with the results of a fully incompressible model. Additionally, the weakly-compressible concept provides a physical interpretation of parameters used to model artificial compressibility.

In general, the idea of vanishing compressibility may be recommended for broader use. The concept is consistent with the theories and ideas implemented in an incompressible flow

simulation. The observed improvement of the algorithm efficiency is significant. Although the structure of the presented algorithm is optimal for the current applications, there are also some opportunities for further development.

ACKNOWLEDGEMENTS

The research presented is a part of a scientific project called 'Analysis and modeling of preliminary sedimentation part in small reservoirs' founded by the National Center for Science in Poland, grant no. 2967/B/P01/2011/40. Several open-source tools are used in this work. The most important are: (1) gfortran – Fortran 90/95 compiler which is a part of the GNU Compilers Collection (GCC), (2) Gmsh – a pre- and post-processing package for 2D and 3D numerical simulations, (3) the MGMRES – a procedure for the solution of linear systems prepared on the basis of the GMRES method.

REFERENCES

Bajantri, M.R., Eldho, T.I., Deolalikar, P.B. (2007): *Modeling hydrodynamic flow over spillway using weakly compressible flow equations* , J. Hydra. Res., 45 (6), 844–852

Barrett, R., Berry, M., Chan, T.C., Demmel, J., Donato, J.M., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., Van der Vorst, H. (1994): *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA, http://www.netlib.org/templates/Templates.html

Barton, I.E. (1998): *Comparison of SIMPLE- and PISO-type algorithms for transient flows*, Int. J. Numer. Meth. Fluids, 26, 459-483

Burkardt, J. (2008): *MGMRES, Restarted GMRES solver for sparse linear systems,* http://people.sc.fsu.edu/~jburkardt/f_src/mgmres/mgmres.html

Caretto, L.S., Gosman, A.D., Patankar, S.V., Spalding, D.B. (1972): *Two calculation procedures for steady, three-dimensional flows with recirculation*, Proc. Third Int. Conf. Numer. Meth. Fluid Dyn., Paris

Cheng, Y.P., Lee, T.S., Low, H.T., Tao, W.Q (2007): *An efficient and robust numerical scheme for the simpler algorithm on non-orthogonal curvilinear coordinates: CLEARER*, Numer. Heat Transfer, Part B, 51, 433–461

Chorin, A.J. (1967): *A numerical method for solving incompressible viscous flow problems*, J. Comp. Physics, 2, 12-26

Chorin, A. J. (1968): *Numerical Solution of the Navier-Stokes Equations*, Math. Comp., 22, 745–762

Drikakis, D., Rider, W. (2005): *High-Resolution Methods for Incompressible and Low-Speed Flows*, Springer-Verlag, Berlin, Heidelberg

Ferzinger, J.H., Peric, M. (2002): *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, Heidelberg, New York

Gao, W., Liu , R. (2009): *A hybrid finite volume/finite element method for incompressible generalized Newtonian fluid flows on unstructured triangular meshes* , Acta Mech Sin, 25, 747–760

Geuzaine, C., Remacle, J.-F. (2009): *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*, Int. J. Numer. Meth. Eng., 79 (11), 1309-1331

Issa, R.I. (1986): *Solution of the implicitly discretized fluid-flow equations by operator-splitting*, J. Comp. Physics, 62, 40–65

Jasak, H. (1996): *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*, Ph.D. dissertation, The Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine

Kim, J., Moin, P. (1985): *Application of a fractional step method to incompressible Navier-Stokes equations*, J. Comp. Physics, 59, 308-323

Munz, C.-D., Roller, S., Klein, R. , Geratz , K.J. (2003): *The extension of incompressible flow solvers to the weakly compressible regime*, Comp. & Fluids, 32, 173–196

Nithiarasu, P. (2003): *An efficient artificial compressibility (AC) scheme based on the characteristic based split (CBS) method for incompressible flows*, Int. J. Numer. Meth. Engng, 56,1815–1845

Nithiarasu, P., Liu, C.-B. (2006): *An artificial compressibility based characteristic based split (CBS) scheme for steady and unsteady turbulent incompressible flows*, Comput. Methods Appl. Mech. Engrg., 195, 2961–2982

Patankar, S.V. (1980): *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York

Schafer, M. (2006): *Computational Engineering: Introduction to Numerical Methods*, Springer-Verlag, Berlin, Heidelberg

Song, C.C.S., Yuan, M., (1988): A *weakly compressible flow model and rapid convergence method*, J. Fluid Engrg., 110, 441–445

Song, C.C.S., Zhou, F. (1999): *Simulation of free surface flow over spillway*, J. Hydra. Eng., 125 (9), 959-967

Sun, D.L., Qu, Z.G., He, Y.L., Tao, W.Q. (2008): *An efficient segregated algorithm for incompressible fluid flow and heat transfer problems—IDEAL (inner doubly iterative efficient algorithm for linked equations) Part I: Mathematical formulation and solution procedure*, Numer. Heat Transfer, Part B, 53, 1–17

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

Tu, S., Aliabadi , S. (2007): *Development of a hybrid finite volume/element solver for incompressible flows*, Int. J. Numer. Meth. Fluids, 55, 177-203

Ubbink, O., Issa, R.I. (1999): *A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes*, J. Comp. Physics, 153, 26–50

Van Doormal, J.P., Raithby, G.D. (1984): *Enhancements of the SIMPLE method for predicting incompressible fluid flows*, Numer. Heat Transfer, 7, 147-163