Bull. Nov. Comp. Center, Num. Anal., 13 (2005), 13–20

Numerical solution of 3D Navier–Stokes equations on staggered grids*

Y.L. Gurieva

Abstract. One algorithm to solve the 3D mixed boundary value problem for the Navier–Stokes system of equations is presented in this paper. The 3D mixed finite volume exponential type approximations on staggered grids are used. To solve the resulting system, an algorithm based on a three-level iterative method is proposed. Results of some numerical experiments demonstrating a numerical convergence are given.

1. Problem statement

The paper deals with an algorithm of numerical solution to the stationary nonlinear system of the Navier–Stokes equations:

$$L(\vec{u})u + \frac{\partial p}{\partial x} = f_x, \quad L(\vec{u})v + \frac{\partial p}{\partial y} = f_y, \quad L(\vec{u})w + \frac{\partial p}{\partial z} = f_z, \quad \operatorname{div}\vec{u} = 0, \tag{1}$$

$$L(\vec{u}) = L^c(\vec{u}) - \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right), \quad L^c(\vec{u})s = \frac{\partial(us)}{\partial x} + \frac{\partial(vs)}{\partial y} + \frac{\partial(ws)}{\partial z}.$$

Here the unknowns are components of the velocity vector $\vec{u} = (u, v, w)$, and p is pressure. The right-hand side functions f_x, f_y, f_z are considered to be known.

A boundary value problem for these equations is considered. The solution is sought for in a bounded domain Ω composed of parallelepipedoidal blocks. The Dirichlet boundary conditions are set on the outer boundary $\partial\Omega$ of the domain. No boundary conditions are set for the pressure as those for the velocities are sufficient for defining the pressure to within a constant.

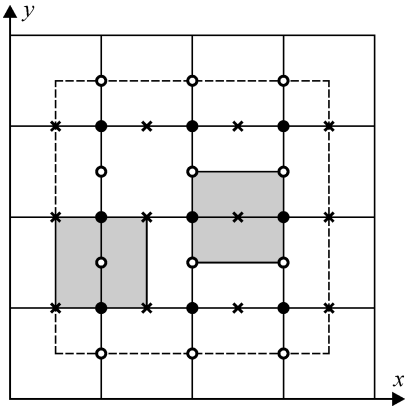
2. Numerical approaches

The exponential type finite volume approach is applied for an approximation of the equations of motion (for details, see [1]). Such an approximation is performed on a parallelepipedoidal non-uniform grid with the grid nodes (x_i, y_j, z_k). In the sequel, the node will be referred to by its grid indices (i, j, k). The coordinates of the nodes are defined via the values of given mesh steps h_i^x, h_j^y, h_k^z by the following formulas:

*Supported by the Russian Foundation for Basic Research under Grant 05-01-00487 and NWO-RFBR Grant 047.016.008.

$$x_{i+1} = x_i + h_i^x, \quad y_{j+1} = y_j + h_j^y, \quad z_{k+1} = z_k + h_k^z,$$
$$i = 1, \ldots, L-1, \quad j = 1, \ldots, M-1, \quad k = 1, \ldots, K-1.$$

A difference in the used approximation as compared to that in [1] is in the staggered grids. This approach is a way of rising the approximation order. This means that four unknowns (the velocity components and the pressure) are related to different geometrical points of the original parallelepipedoidal non-uniform grid. The nodes with the coordinates (x_i, y_j, z_k) correspond to p unknowns, and the mid-edges $(x_{i+1/2}, y_j, z_k)$, $(x_i, y_{j+1/2}, z_k)$, and $(x_i, y_j, z_{k+1/2})$ correspond to u-, v-, and w-velocity components, respectively. In the figure, u-nodes are shown by crosses, v-nodes — by empty circles, and p-nodes — by the black circles.



Unknowns and finite volumes on staggered grids

These mid-edge points are middle points of the corresponding finite volumes (the Dirichlet–Voronoi cells) used in the approximation. For example, for a pressure node with the grid indices (i, j, k), the Dirichlet–Voronoi cell is defined by the relations

$$V_{i,j,k}^p = \left\{x_{i-1/2} < x < x_{i+1/2},\ \ y_{j-1/2} < y < y_{j+1/2},\ \ z_{k-1/2} < z < z_{k+1/2}\right\}.$$

As the four types of the grid nodes are considered, the four types of the Dirichlet–Voronoi cells are built for the approximation. So, to addition to a cell for the pressure, the cells $V_{i+1/2,j,k}^u$, $V_{i,j+1/2,k}^v$, $V_{i,j,k+1/2}^w$ are used for the finite volume approximation of the equations of motion. Two finite volumes — for u- and v-unknowns — are shown in the figure.

In the case of non-staggered grids, the boundary $\partial\Omega$ of the domain Ω coincides with the boundary of the discrete computational domain Ω^h. In the case of the staggered grids, only velocity unknowns lie on different parts of the boundary $\partial\Omega$. We define the computational domain Ω^h in such a way that $\partial\Omega^h$ is situated away from $\partial\Omega$, and a distance between them is $h^x/2$, $h^y/2$, and $h^z/2$ for the boundary parts parallel to x-, y-, and z-axes, respectively. To illustrate this, the boundary $\partial\Omega^h$ is shown in the figure (for simplicity, this is a two-dimensional example) by the solid line as well as the pressure grid lines, and the boundary $\partial\Omega$ is shown by the dashed line.

As the grid lines are referred to by the grid indices $i = 1, \ldots, L$, $j = 1, \ldots, M$, $k = 1, \ldots, K$, the grid indexes for the unknowns (including the unknowns on the boundary $\partial\Omega$) will be the following:

$$
\begin{aligned}
u: &\quad i = 1, \ldots, L-1, \quad j = 2, \ldots, M-1, \quad k = 2, \ldots, K-1,\\
v: &\quad i = 2, \ldots, L-1, \quad j = 1, \ldots, M-1, \quad k = 2, \ldots, K-1,\\
w: &\quad i = 1, \ldots, L-1, \quad j = 1, \ldots, M-1, \quad k = 2, \ldots, K-1,\\
p: &\quad i = 2, \ldots, L-1, \quad j = 2, \ldots, M-1, \quad k = 2, \ldots, K-1.
\end{aligned}
$$

Taking the integral of any of the equations of motion (1) for different functions $s = u$, $s = v$ or $s = w$ over the corresponding cell $V_{i,j,k}$, we obtain the balance relation

$$\int_{S_{i,j,k}} J^n ds = \int_{V_{i,j,k}} g_s \, dV,$$

where $S_{i,j,k}$ is the surface of $V_{i,j,k}$, g_s dependent on one of the right-hand side functions and one of the pressure gradient components, and $\vec{u}_n$, $J^n = \dfrac{\partial s}{\partial n} + \vec{u}_n s$ are the velocity component and the flux density in the direction of the outer normal to $S_{i,j,k}$.

For example, the left-hand side of the balance relation for the flux, say, over the cell surface perpendicular to x-axis after the approximation by simple quadrature formulas has the following form: $I^{x,h}_{0,3} = S_x J^{x,h}_{0,3} = a^x_{0,0} s_0 - a_{0,3} s_3$. Here the indices 0 and 3 correspond to the points with $x = x_i$ and $x = x_{i+1}$, respectively, $S_x = h^y_j h^z_k/4$, and the entries of the local balance matrix are introduced as

$$
a^x_{0,0} = \frac{S_x u_{\frac{0+3}{2}}}{1 - c^{-1}_{0,3}} = a_{0,3} c_{0,3}, \qquad a^x_{0,3} = \frac{S_x u_{\frac{0+3}{2}}}{c_{0,3} - 1},
$$

$$
u_{\frac{0+3}{2}} = \frac{u_0 + u_3}{2}, \qquad c_{0,3} = \exp\bigl(h^x_i u_{\frac{0+3}{2}}\bigr)
$$

with the element-by-element approach for obtaining the global matrix. In the same manner the rest entries of the local balance matrix are defined.

The approximation is done with the help of an element-by-element approach when the local balance matrices and the local additives to the right-hand side are computed for the grid elements. According to the staggered grids, there are four types of the grid elements corresponding to the four groups of unknowns. Note that u finite volume is simultaneously v grid element and vice versa.

For the fixed velocity components which are the coefficients of (1), the assembling procedure gives the global non-symmetric balance matrix and the global right-hand side vector of the final system of linear equations. The velocity global balance matrix is a 3×3 block-diagonal matrix. Each block is a seven-diagonal matrix and corresponds to the unknowns of one velocity component.

Some words should be said about taking into account the Dirichlet boundary conditions.

The Dirichlet boundary condition for the first velocity component $u = C$ can be easily implemented on the left $x = x_{1+1/2}$ and the right $x = x_{L-1/2}$ sides of the domain Ω as these boundaries run through the corresponding nodes for unknowns (dash lines of $\partial\Omega$ run via crosses on the left and on the right, see Figure 1). If this condition is given on the other outer boundaries, e.g., on the boundary $y = y_{1+1/2}$, then we have

$$u_{i,1+1/2,k} \equiv (u_{i,1,k} + u_{i,2,k})/2 = C,$$

and hence, $u_{i,1,k} = 2C - u_{i,2,k}$. Then by substituting this into the resulting algebraic equation for the unknown $u_{i,2,k}$ of the form

$$\begin{aligned} a^0_{i,j,k}u_{i,j,k} - a^1_{i,j,k}u_{i-1,j,k} - a^2_{i,j,k}u_{i,j-1,k} - a^3_{i,j,k}u_{i+1,j,k} - \\ a^4_{i,j,k}u_{i,j+1,k} - a^5_{i,j,k}u_{i,j,k-1} - a^6_{i,j,k}u_{i,j,k+1} = g_{i,j,k} \end{aligned} \tag{2}$$

with $j = 2$, we arrive at

$$(a^0_{i,j,k} + a^2_{i,j,k})u_{i,j,k} - a^1_{i,j,k}u_{i-1,j,k} - a^3_{i,j,k}u_{i+1,j,k} - \ldots = g_{i,j,k} + 2Ca^2_{i,j,k}$$

and as we have excluded the link with the unknown $u_{i,1,k} \equiv u_{i,j-1,k}$, set $a^2_{i,j,k} = 0$ after the presented modification of the diagonal coefficient and the corresponding right-hand side of the unknown $u_{i,j,k}$.

Similarly, such a condition for u-component is taken into account on three other faces of the domain:

$$\begin{aligned} y = y_M: &\quad a^0_{i,j,k} = a^0_{i,j,k} + a^4_{i,j,k}, \quad g_{i,j,k} = g_{i,j,k} + 2Ca^4_{i,j,k}, \quad a^4_{i,j,k} = 0; \\ z = z_1: &\quad a^0_{i,j,k} = a^0_{i,j,k} + a^5_{i,j,k}, \quad g_{i,j,k} = g_{i,j,k} + 2Ca^5_{i,j,k}, \quad a^5_{i,j,k} = 0; \\ z = z_K: &\quad a^0_{i,j,k} = a^0_{i,j,k} + a^6_{i,j,k}, \quad g_{i,j,k} = g_{i,j,k} + 2Ca^6_{i,j,k}, \quad a^6_{i,j,k} = 0. \end{aligned}$$

So, the unknowns on the Dirichlet boundaries are excluded from the resulting algebraic system.

After the discretization, system (1) can be written down in a well-known block form

$$\begin{pmatrix} A^h(\vec{u}) & B \\ B^t & 0 \end{pmatrix} \begin{pmatrix} \vec{u}^h \\ p^h \end{pmatrix} = \begin{pmatrix} f^h \\ 0 \end{pmatrix}, \tag{3}$$

where B is a difference gradient operator and B^t is a difference divergence.

The difference divergence is defined at the pressure nodes via the known values of the velocities and can be presented after integrating the corresponding equation over the finite volume $V^p_{i,j,k}$ by the following formula:

$$\begin{aligned} \operatorname{div}^h \overline{u}_{i,j,k} = \big(s_x(u_{i+1/2,j,k} - u_{i-1/2,j,k}) + s_y(v_{i,j+1/2,k} - v_{i,j-1/2,k}) + \\ s_z(w_{i,j,k+1/2} - w_{i,j,k-1/2})\big)/V_{i,j,k}. \end{aligned}$$

Here s_x, s_y, and s_z are the areas of the cell surfaces perpendicular to x-, y-, and z-directions, respectively, at the pressure nodes, and $V_{i,j,k}$ is the volume of the pressure cell.

Let us note that the dimension of the system is the following: $N^u + N^v + N^w$ unknowns for the velocities and N_p for the pressure. In a general case, $N^u \neq N^v \neq N^w$, each being defined by the number of the corresponding mid-edges except those on the Dirichlet boundaries.

Several words about numeration and a storage scheme. Unknowns are numerated as follows: firstly, N^u unknowns by natural ordering in x-, y-, and z-directions, then N^v, N^w, and N^p unknowns in the same order. As the order of the final system is $N \times N$, and every equation contains no more than seven unknowns according to (2), the final matrix is a sparse one. To save the storage space, the global matrix is stored in the form of a special row-wise sparse format [4]. Only non-zero entries of the final matrix are stored. Moreover, the representation $A^h = D - L - U$ is used for storage, where U is the upper triangular part of the matrix, and L is the lower triangular part of the matrix. For each ith row of the matrix U, the number NE(i) of the non-zero entries $a_{i,j}$, $j > i$, is stored. For each non-zero entry, its column number j and its value $a_{i,j}$ are stored in the arrays NEIB(NU) and AU(NU), respectively, where NU is the total number of non-zero entries in the matrix U. The matrix L is similarly stored.

Analysis of iterative algorithms to solve such a system is given in [2]. One of them is a widespread Uzawa method. This approach is used in the presented algorithm and consists in the following two steps — a velocity step and a pressure step:

$$\begin{aligned} \vec{u} &= (A^h)^{-1}(f - Bp), \\ Ap \equiv B^t(A^h)^{-1}Bp &= B^t(A^h)^{-1}f \equiv g. \end{aligned} \tag{4}$$

The pressure step is solved (under fixed velocities) by a minimal residual method ([5], p. 76–77). It has the following stages: given p_0, compute $r_0 = g - Ap_0 = B^t(A^h)^{-1}f - B^t(A^h)^{-1}Bp_0$, then perform the computations

$$\begin{aligned} \tau_n &= \frac{(Ar_{n-1}, r_{n-1})}{(Ar_{n-1}, Ar_{n-1})}, \\ p_n &= p_{n-1} + \tau_n r_{n-1}, \quad r_n = r_{n-1} - \tau_n A r_{n-1}. \end{aligned}$$

A system with the matrix A^h is a linearized system. The process to compute the velocities is a low relaxation procedure with a parameter $0 < \omega < 1$:

$$\vec{u}^{m+1} = \omega\hat{\vec{u}}^{m+1} + (1-\omega)\vec{u}^m, \quad m = 0, 1, \ldots$$

Thus, A is $A^m = A(\vec{u}^m)$ in (4) and we have one more iterative level.

A solution of the block linear non-symmetric system with the matrix A^h is found by a preconditioned biconjugate (multi-step) residual method (BiCR). In this algorithm, this procedure is applied three times — two times to compute r_0, and one time inside the pressure minimal residual solver.

The pseudo-code for such an algorithm (4) is the following:

$$
\begin{array}{l}
\text{Set initial guesses } \vec{u}_0,\ p_0 \\
\vec{u} = \vec{u}_0,\ p = p_0 \\
\text{do while } (e_{\text{vel}} > \varepsilon_{\text{vel}}) \\
\quad \text{compute } A^h(\vec{u}),\ f \\
\quad \text{compute } r_0 \\
\quad \text{do while } (e_p > \varepsilon_p) \\
\qquad \text{solve } p \text{ by (4)} \\
\quad \text{end do} \\
\quad \text{relaxation of } \vec{u} \\
\text{end do}
\end{array}
\tag{5}
$$

Here the values of $\varepsilon_p,\ \varepsilon_{vel}$ are the given tolerances. The left sides of the stopping criteria are some residual functions:

$$e_{vel} = \left(\frac{\sum (\vec{u}_n - \vec{u}_{\text{exact}})^2}{\sum \vec{u}^2_{\text{exact}}} \right)^{1/2}$$

if the exact solution is known (the sum here means the sum of the vector components), and

$$e_p = \left(\frac{(r_n, r_n)}{(B^t L^{-1} B f, B^t L^{-1} B f)} \right)^{1/2},$$

where r_n is a pressure residual on iteration number n.

So, this process is a three-level iterative procedure.

3. Numerical experiments

The aim of the experiments is to check the convergence of the algorithm presented and the truncation error for the tests with the given exact analytical solution.

The results of the numerical experiments are presented in the tables below. The values of the truncation errors $\delta_s = \max_{ijk} \|s(x_i, y_j, z_k) - s^h_{i,j,k}\|$ are given for different grid sizes.

The relaxation parameter $\omega = 1$ was taken in all the experiments.

Test 1 has the following analytical velocity components with the property $\operatorname{div} \vec{u} = 0$ under any constant pressure and zero volume force:

$$u = \sin^2 \pi x \sin \pi y \sin 2\pi z, \quad v = \sin \pi x \sin^2 \pi y \sin 2\pi z,$$
$$w = -(\sin 2\pi x \sin \pi y + \sin \pi x \sin 2\pi y) \sin^2 \pi z.$$

The right-hand side functions f_u, f_v, f_w are analytically calculated from equations (1). The boundary conditions for the velocity components are zero Dirichlet boundary conditions.

The boundary value problem is solved in a cube domain $\Omega = [0, 1]^3$ whose computational domain has a size depending on the grid: $\Omega^h = [-h/2, 1 + h/2]^3$, where $h = 1/N$ and N is a number of grid steps on the unit segment. So, Ω^h contains $N + 1$ steps.

The computational domain is discretized by the uniform grids with the number of steps $N = 5$, 9, 17, 33.

The Stokes problem is considered, i.e., $L^c(\vec{u})s = 0$ in (1). In this case, the matrix $L^h(\vec{u})$ is symmetric and positive definite. No iterations on velocities is done, and so the code gives a two-level iterative process.

The test has zero initial velocity guess. As for the pressure initial guess, three its types were tested: $p_0 = 0$, $p_0 = \sin x$, $p_0 = \sin \pi x$.

The tolerance for the minimal residual procedure is $\varepsilon_p = 10^{-7}$. For the BiMR procedure, the tolerance is 10^{-7}, also.

In the tables below, the number of the minimal residual iterations n_{MR}, of the BiMR iterations n_{BiMR}, and the values of the truncation errors for the pressure and the velocities are given. In the second column, the first number is for $p_0 = 0$ and, in the parenthesis, there are the numbers of iterations for $p_0 = \sin x$ and $p_0 = \sin \pi x$.

In this test, $\delta_u = \delta_v$, so only δ_u is given in Table 1.

Table 1. The results of Test 1, zero pressure initial guess

N	n_{MR}	n_{BiMR}	δ_p	δ_u	δ_w
5	8 (29, 17)	7	0.4858	0.2251	0.4502
8	11 (39, 27)	12	0.3387	0.08153	0.1382
16	16 (50, 39)	19	0.1234	0.02206	0.03425
32	21 (59, 51)	28	0.03756	0.005607	0.008738

Hence it follows that all the truncation errors of the velocities and the pressure have order $O(h^2)$ on the uniform grids.

Test 2. Table 2 gives the results for the same exact solution as in Test 1 but for the Navier–Stokes problem. The test has zero initial guess for the velocity components and the constant initial pressure guess: $p_0 = 10$.

The number of non-linear velocity iterations equals five for all the runs. As the number of nonlinear velocity iterations is greater than 1, the numbers of the pressure iterations n_{MR} on every nonlinear velocity step of the solution process and the final truncation errors are given.

Table 2. The results of Test 2

N	δ_p	δ_u	δ_w	n_{MR}
4	0.5583	0.2266	0.4505	24, 23, 11, 4, 0
8	0.3899	0.08187	0.1386	34, 36, 21, 9, 2
16	0.1378	0.02215	0.03436	40, 47, 30, 16, 5
32	0.04104	0.005630	0.008769	45, 57, 38, 23, 10

It follows from this table that the truncation errors for the pressure and velocity components have the errors of order $O(h^2)$ as in the previous test.

As one can see from the last table, the number of the pressure iterations is the bottleneck of the presented approach. Its future development is in some acceleration of the pressure solver.

References

[1] Gurieva Y.L. On one numerical solution of 3D Navier–Stokes equations // Proceedings of the International Conference on Computational Mathematics ICCM-2004. Part II. — Novosibirsk, 2004. — P. 852–857.

[2] Chizhonkov E.V. Relaxation Methods of Solving Saddle Problems. — Moscow: Institute of Numerical Mathematics of RAS, 2002 (In Russian).

[3] Il'in V.P. The Finite Difference and Finite Volume Methods for Elliptic Equations. — Novosibirsk: ICM&MG, 2001 (In Russian).

[4] Andreeva M.Yu., Il'in V.P., Itskovich E.A. Two solvers for non-symmetric SLAEs // NCC Bulletin. Series Numerical Analysis. — Novosibirsk: NCC Publisher, 2003. — Issue 12. — P. 1–12.

[5] Il'in V.P. Incomplete Factorization Methods for Solving Algebraic Linear Systems. — Moscow: Fizmatlit, 1995 (In Russian).