CAS Application to the Construction of the Collocations and Least Residuals Method for the Solution of 3D Navier-Stokes Equations

V.P. Shapeev and E.V. Vorozhtsov

Khristianovich Institute of Theoretical and Applied Mechanics SB RAS, Novosibirsk 630090, Russia

Table of contents

2 Description of the CLR Method

CLR = **C**ollocation and **L**east **R**esiduals

Stationary Navier-Stokes equations:

$$(\mathbf{V}\cdot\nabla)\mathbf{V}+\nabla p=(1/\mathrm{Re})\Delta\mathbf{V}-\mathbf{f},\quad \mathrm{div}\,\mathbf{V}=0,\quad (x_1,x_2,x_3)\in\Omega, \tag{1}$$

Spatial region:

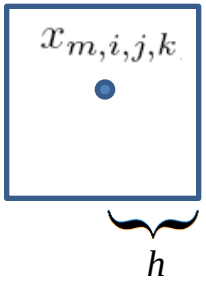
$$\Omega=\{(x_1,x_2,x_3),0\le x_i\le X,i=1,2,3\} \tag{2}$$

Local coordinates:

$$y_m=(x_m-x_{m,i,j,k})/h,\ m=1,2,3,$$

New notations for dependent variables:

$$\mathbf{u}(y_1,y_2,y_3)=\mathbf{V}(hy_1+x_{1,i,j,k},hy_2+x_{2,i,j,k},hy_3+x_{3,i,j,k}),$$
$$p(y_1,y_2,y_3)=p(hy_1+x_{1,i,j,k},hy_2+x_{2,i,j,k},hy_3+x_{3,i,j,k}).$$



NS equations in new notation:

$$\Delta u_m-\mathrm{Re}h\left(u_1\frac{\partial u_m}{\partial y_1}+u_2\frac{\partial u_m}{\partial y_2}+u_3\frac{\partial u_m}{\partial y_3}+\frac{\partial p}{\partial y_m}\right)=\mathrm{Re}\cdot h^2f_m; \tag{3}$$
$$\frac{1}{h}\left(\frac{\partial u_1}{\partial y_1}+\frac{\partial u_2}{\partial y_2}+\frac{\partial u_3}{\partial y_3}\right)=0, \tag{4}$$

where $\Delta=\partial^2/\partial y_1^2+\partial^2/\partial y_2^2+\partial^2/\partial y_3^2$, $m=1,2,3$.

The basic idea of the CLR method is to use the collocation method in combination with the Least-squares method to obtain numerical solution.

Linearization of the NS equations after Newton

$$\xi\left[\Delta u_m^{s+1}-(\mathrm{Re}\cdot h)\left(u_1^su_{m,y_1}^{s+1}+u_1^{s+1}u_{m,y_1}^s+u_2^su_{m,y_2}^{s+1}+u_2^{s+1}u_{m,y_2}^s+u_3^su_{m,y_3}^{s+1}\right.\right.$$
$$\left.\left.+\ u_3^{s+1}u_{m,y_3}^s+p_{y_m}^{s+1}\right)\right]=\xi F_m,\ m=1,2,3, \tag{5}$$

where s is the iteration number, $s=0,1,2,\ldots$, $F_m=\mathrm{Re}[h^2f_m-h(u_1^su_{m,y_1}^s+u_2^su_{m,y_2}^s+u_3^su_{m,y_3}^s)]$, $u_{m,y_l}=\partial u_m/\partial y_l$, $p_{y_m}=\partial p/\partial y_m$, $l,m=1,2,3$.

We now present the approximate solution in each cell $\Omega_{i,j,k}$ as a linear combination of the basis vector functions φ_l

$$(u_1^s,u_2^s,u_3^s,p^s)^T=\sum_l b_{i,j,k,l}^s\varphi_l, \tag{6}$$

Table 1. The form of basis functions φ_l

l	1	2	3	4	5	6	7	8	9	10
φ_l	1	0	0	y_1	0	0	y_2	0	0	y_3
	0	1	0	$-y_2$	y_1	0	0	y_2	0	0
	0	0	1	0	0	y_1	0	$-y_3$	y_2	0
	0	0	0	0	0	0	0	0	0	0
l	11	12	13	14	15	16	17	18	19	20
φ_l	0	y_1^2	0	0	$-2y_1y_2$	0	$-2y_1y_3$	0	y_1^2	y_2^2
	y_3	$-2y_1y_2$	y_1^2	0	y_2^2	0	0	y_1y_3	0	0
	0	0	0	y_1^2	0	y_1y_2	y_3^2	0	$-2y_1y_3$	0
	0	0	0	0	0	0	0	0	0	0
l	21	22	23	24	25	26	27	28	29	30
φ_l	0	0	y_2y_3	0	y_3^2	0	0	0	0	0
	y_2^2	0	0	$-2y_2y_3$	0	y_3^2	0	0	0	0
	$-2y_2y_3$	y_2^2	0	y_3^2	0	0	0	0	0	0
	0	0	0	0	0	0	1	y_1	y_2	y_3

Collocation equations

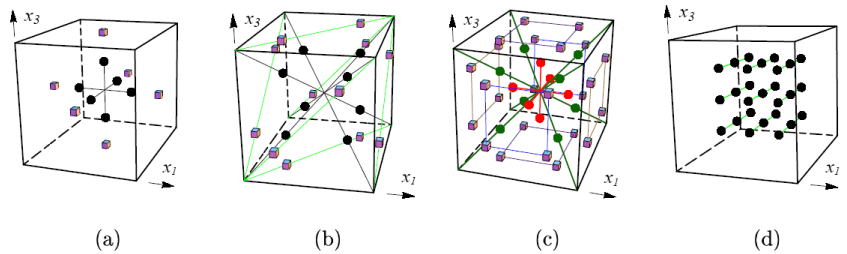


Fig. 1. The variants of the specification of the collocation and matching points: (a) $N_c = 6$, $N_m = 6$; (b) $N_c = 8$, $N_m = 12$; (c) $N_c = 14$, $N_m = 24$; (d) $N_c = 27$.

Substituting the coordinates of collocation points in equations (5) we obtain $3N_c$ equations of collocations:

$$a^{(1)}_{\nu,m} \cdot b^{s+1}_m = f^s_\nu, \quad \nu = 1, \ldots, 3N_c; \; m = 1, \ldots, 30. \tag{7}$$

Matching conditions

$$\begin{aligned} &h\partial(u^+)^n/\partial n + \eta_1(u^+)^n = h\partial(u^-)^n/\partial n + \eta_1(u^-)^n; \\ &h\partial(u^+)^{\tau_1}/\partial n + \eta_2(u^+)^{\tau_1} = h\partial(u^-)^{\tau_1}/\partial n + \eta_2(u^-)^{\tau_1}; \\ &h\partial(u^+)^{\tau_2}/\partial n + \eta_2(u^+)^{\tau_2} = h\partial(u^-)^{\tau_2}/\partial n + \eta_2(u^-)^{\tau_2}; \\ &p^+ = p^-. \end{aligned} \tag{8}$$

Let us write these equations in the form

$$a^{(2)}_{\nu,m} \cdot b^{s+1}_m = g^{s,s+1}_\nu, \quad \nu = 1, \ldots, 3N_m + 6 + \delta^1_i\delta^1_j\delta^1_k; \; m = 1, \ldots, 30. \tag{9}$$

Let us introduce the matrix $A_{i,j,k}$, which unites the matrices of systems (7) and (9), as well as the column vector of the right-hand sides $\boldsymbol{f}^{s,s+1}_{i,j,k}$. The following SLAE is then solved in each cell:

$$A_{i,j,k} \cdot \boldsymbol{b}^{s+1} = \boldsymbol{f}^{s,s+1}_{i,j,k}, \tag{10}$$

where $\boldsymbol{b}^{s+1} = (b^{s+1}_{i,j,k,1}, \ldots, b^{s+1}_{i,j,k,30})^T$.

Details of the numerical algorithm

The overdetermined system of equations (10) was solved numerically by the method of rotations with column pivoting. It is known that the application of this method to an overdetermined system is equivalent to the minimization of scalar product $(Ax\text{-}f, Ax\text{-}f)$, where A is the matrix of system and f is the vector of right-hand sides.

On the face $y_1 = 1$, two matching points are specified as $(1,\zeta,\zeta)$ and $(1,-\zeta,-\zeta)$, and on the face $y_1 = -1$, two matching points are specified as $(-1,-\zeta,\zeta)$ and $(-1,\zeta,-\zeta)$. It was found by numerical experiments that the value ζ=0.6 was generally better than the value ζ=0.5 in terms of convergence acceleration.

The local coordinates of collocation points were specified as (ω, ω, ω). It was found by numerical experiments that the value ω=0.6 was generally better than the value ω=0.5 in terms of convergence acceleration.

3 The Multigrid Algorithm

1°. **Prolongation.** Let us illustrate the prolongation algorithm by the example of the velocity component $u_1(y_1, y_2, y_3, b_1, \ldots, b_{30})$. Let $h_1 = h$, where h is the half-step of the coarse grid, and let $h_2 = h_1/2$ be the half-step of a fine grid on which one must find the decomposition of the function u_1 over the basis.
Step 1. Let X_1, X_2, X_3 be the global coordinates of the center of a coarse grid cell. We make the following substitutions in the polynomial expression for u_1:

$$y_l = (x_l - X_l)/h_1, \quad l = 1, 2, 3. \tag{11}$$

As a result we obtain the function

$$U_1(x_1, x_2, x_3, b_1, \ldots, b_{30}) = u_1\left(\frac{x_1 - X_1}{h_1}, \frac{x_2 - X_2}{h_1}, \frac{x_3 - X_3}{h_1}, b_1, \ldots, b_{30}\right). \tag{12}$$

Step 2. Let $(\tilde{X}_1, \tilde{X}_2, \tilde{X}_3)$ be the global coordinates of the center of any of the eight cells of the fine grid, which lie within the coarse grid cell. We make in (12) the substitution $x_l = \tilde{X}_l + \tilde{y}_l \cdot h_2$, $l = 1, 2, 3$. As a result we obtain the function

$$\tilde{U}_1 = \tilde{b}_1 + \tilde{b}_4\tilde{y}_1 + \tilde{b}_{12}\tilde{y}_1^2 + \tilde{b}_{19}\tilde{y}_1^2 + \tilde{b}_7\tilde{y}_2 - 2\tilde{b}_{15}\tilde{y}_1\tilde{y}_2 + \tilde{b}_{20}\tilde{y}_2^2 + \tilde{b}_{10}\tilde{y}_3 - 2\tilde{b}_{17}\tilde{y}_1\tilde{y}_3 + \tilde{b}_{23}\tilde{y}_2\tilde{y}_3 + \tilde{b}_{25}\tilde{y}_3^2.$$

The analytic expressions for coefficients $\tilde{b}_1, \ldots, \tilde{b}_{30}$ were found efficiently with the aid of the *Mathematica* function `Coefficient[...]`. To reduce the length of the obtained expressions for the above coefficients we have applied a number of transformation rules as well as the Mathematica function `FullSimplify[...]`. As a result, the length of the final expressions for $\tilde{b}_1, \ldots, \tilde{b}_{30}$ proved to be five times shorter than the length of the original expressions. It turns out that the coordinates X_1, X_2, X_3 and $\tilde{X}_1, \tilde{X}_2, \tilde{X}_3$ enter the $\tilde{b}_l$ $(l = 1, \ldots, 30)$ only in the form of combinations $\delta x_l = (X_l - \tilde{X}_l)/h_1$. For example, $\tilde{b}_4 = (h_2/h_1) \cdot (b_4 - 2b_{12}\delta x_1 - 2b_{19}\delta x_1 + 2b_{15}\delta x_2 + 2b_{17}\delta x_3)$. In accordance with (11) the quantity $-\delta x_l = (\tilde{X}_l - X_l)/h_1$ has the meaning of the local coordinate on the coarse grid of the coordinate $\tilde{X}_l$ of the center of the fine grid cell.

4 New version of the Krylov's algorithm

The purpose of this algorithm is the acceleration of convergence of the iterations in nonlinearity in the CLR method.

We at first unite the equations for
from all cells into a single big algebraic system

$$AX^{n+1} = f. \tag{12a}$$

Let us assume that the iteration process for solving (1) converges and denote by X the converged solution. Then

$$AX = f. \tag{12b}$$

Let us rearrange (12b) in the equivalent form

$$X = TX + f. \tag{14}$$

We now write the iteration process solving (14) as

$$X^{n+1} = TX^n + f. \tag{13}$$

Let us now introduce the residual

$$r^{n+1} = X^{n+1} - X^n$$

Then $r^{n+1} = Tr^n$. Let

$$Z^n = X - X^n.$$

The basic idea is to search for the error Z^{n+1} in the form

$$Y^{n+1} = \sum_{i=1}^{k} \alpha_i r^i$$

as an approximate value of the error that is this vector is sought in the Krylov's subspace

$$\mathcal{K}_k(\vec{r}^1, \mathrm{T}) = \mathrm{span}\{\vec{r}^1, \mathrm{T}\vec{r}^1, \ldots, \mathrm{T}^{k-1}\vec{r}^1\},$$

where $\mathrm{span}\{\vec{v}^1, \ldots, \vec{v}^k\}$ is a linear span of vectors $\vec{v}^1, \ldots, \vec{v}^k$.

Then it is easy to show by using (13) and (14) that the solution error satisfies the equation $Z^{n+1} = TZ^n$.

We now use the relation $r^n = Z^n - Z^{n+1}$.
Then

$$Z^n = Z^{n+1} + r^n,$$

or

$$(T^{-1} - E)Z^{n+1} = r^n.$$

Substituting in this equation the representation

$$Z^{n+1} = \sum_{i=1}^{k} \alpha_i r^i \qquad (6)$$

we obtain the following system for determining the α_i at $n = k$:

$$(T^1 - r^0)\alpha_1 + \ldots + (T^k - r^{k-1})\alpha_k = -r^k. \qquad (7)$$

The system of equations (7) was solved by the method of rotations.
After that the new iteration X^{n+1} is computed as

$$X^{*n+1} = X^{n+1} + Z^{n+1}.$$

The overdetermined system for the coefficients of the solution expansion over the solenoidal basis was solved by the same method.

One of the advantages of the above method for convergence acceleration is that it can easily be applied to already programmed iteration processes. To this end, it is sufficient to introduce in the existing computer code a small procedure for correction computation.

5 Numerical results

Analytic test

$$u_1 = -\cos(x_1)\sin(x_2)\sin(x_3), \quad u_2 = 0.5\sin(x_1)\cos(x_2)\sin(x_3),$$
$$u_3 = 0.5\sin(x_1)\sin(x_2)\cos(x_3), \qquad (11)$$
$$p = \cos(x_1) + \cos(x_2) + \cos(x_3) - (3/X)\sin(X).$$

It is to be noted here that the solution (11) satisfies the continuity equation (2). We now write down the right-hand sides f_1, f_2, f_3 of equations (1):

$$f_1 = (3\cos(x_1)\sin(x_2)\sin(x_3) + \mathrm{Re}\sin(x_1)(1 + \cos(x_1)(0.5\cos^2(x_3)\sin^2(x_2) + (0.5\cos^2(x_2) + \sin^2(x_2))\sin^2(x_3))))/\mathrm{Re},$$
$$f_2 = (-1.5\cos(x_2)\sin(x_1)\sin(x_3) + \mathrm{Re}\sin(x_2)(1 + \cos(x_2) \times (-0.25\cos^2(x_3)\sin^2(x_1) + (0.5\cos^2(x_1) + 0.25\sin^2(x_1))\sin^2(x_3))))/\mathrm{Re},$$
$$f_3 = (\mathrm{Re}\sin(x_3) + \cos(x_3)(-1.5\sin(x_1)\sin(x_2) + 0.5\mathrm{Re}\cos^2(x_1)\sin^2(x_2)\sin(x_3) + \mathrm{Re}\sin^2(x_1)(-0.25\cos^2(x_2) + 0.25\sin^2(x_2))\sin(x_3)))/\mathrm{Re}.$$

RMS errors

To determine the absolute numerical errors of the method on a specific uniform grid with half-step h we have computed the following root mean square errors:

$$\delta\mathbf{u}(h) = \left[\frac{1}{3M^3}\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{\nu=1}^{3}(u_{\nu,i,j,k} - u^{ex}_{\nu,i,j,k})^2\right]^{0.5},$$

$$\delta p(h) = \left[\frac{1}{M^3}\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}(p_{i,j,k} - p^{ex}_{i,j,k})^2\right]^{0.5},$$

Convergence orders

$$\nu_u = \frac{\log[\delta\mathbf{u}(h_{m-1})] - \log[\delta\mathbf{u}(\mathbf{h_m})]}{\log(h_{m-1}) - \log(h_m)}, \quad \nu_p = \frac{\log[\delta p(h_{m-1})] - \log[\delta p(h_m)]}{\log(h_{m-1}) - \log(h_m)},$$

Table 1. The errors $\delta\mathbf{u}, \delta p$ and the convergence orders ν_u, ν_p on a sequence of grids, Re = 100

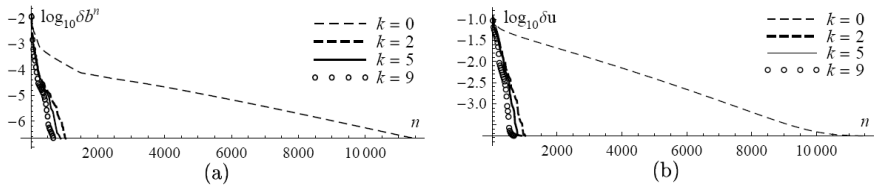
M	$\delta\mathbf{u}$	δp	ν_u	ν_p
10	$0.364 \cdot 10^{-3}$	$0.585 \cdot 10^{-2}$		
20	$0.852 \cdot 10^{-4}$	$0.232 \cdot 10^{-2}$	2.10	1.33
30	$0.247 \cdot 10^{-4}$	$0.131 \cdot 10^{-2}$	3.05	1.41

Table 2. The errors $\delta\mathbf{u}, \delta p$ and the convergence orders ν_u, ν_p on a sequence of grids, Re = 1000

M	$\delta\mathbf{u}$	δp	ν_u	ν_p
10	$0.462 \cdot 10^{-3}$	$0.341 \cdot 10^{-2}$		
20	$0.189 \cdot 10^{-3}$	$0.240 \cdot 10^{-2}$	1.29	0.51
30	$0.112 \cdot 10^{-3}$	$0.155 \cdot 10^{-2}$	1.29	1.08

The definition of the pseudo-error:

$$\delta b^{s+1} = \max_{i,j,k} \left(\max_{1 \le l \le 30} \left| b^{s+1}_{i,j,k,l} - b^{s}_{i,j,k,l} \right| \right)$$



Influence of quantity *k* in (6) on convergence rate of the CLR method at Re = 1000: (a) the logarithm of the pseudo-error; (b) the logarithm of the error in velocity.

The number of iterations needed for satisfying the inequality $\delta b^n < 2 \cdot 10^{-7}$ was less than without the application of the Krylov's algorithm by the factors of 11, 13, and 17, respectively, at $k = 2, k = 5$, and $k = 9$.

Flow in the Lid-Driven Cavity

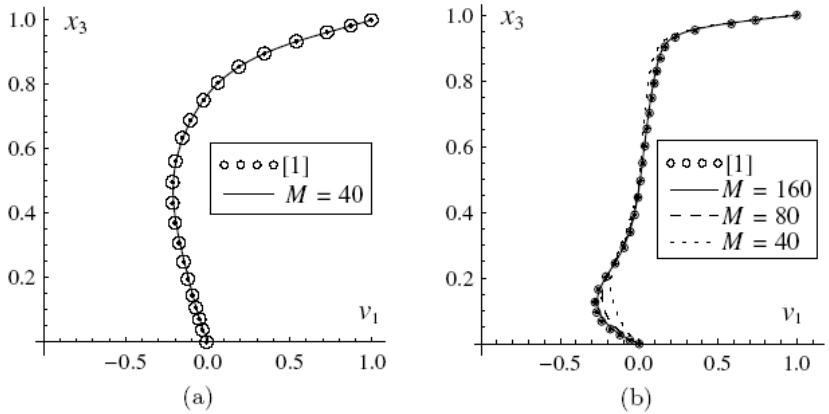


Fig. 2. Profiles of the velocity component v_1 on the central line $x_1 = x_2 = 0.5$ for Re = 100 (a) and Re = 1000 (b)

1. Albensoeder, S., Kuhlmann, H.C.: Accurate three-dimensional lid-driven cavity flow. J. Comp. Phys. 206, 536–558 (2005)

Pseudo-streamlines in different sections of the cubic cavity at Re = 100 (see CASC 2012 Proc.)

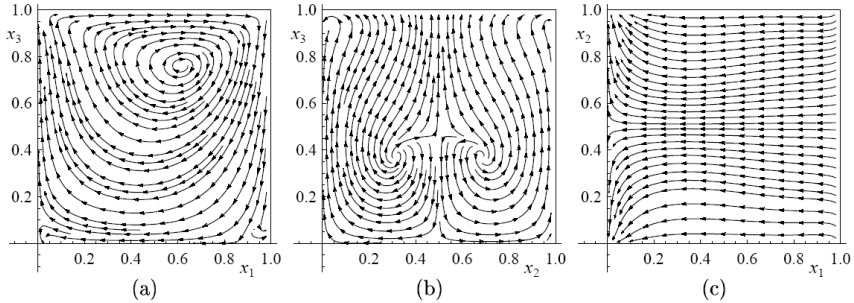


Fig. 5. Pseudo-streamlines in different sections of the cubic cavity at Re = 100: (a) section $x_2 = 0.5$; (b) section $x_1 = 0.5$; (c) section $x_3 = 0.5$.

Flow in the Lid-Driven Cavity, Re = 1000

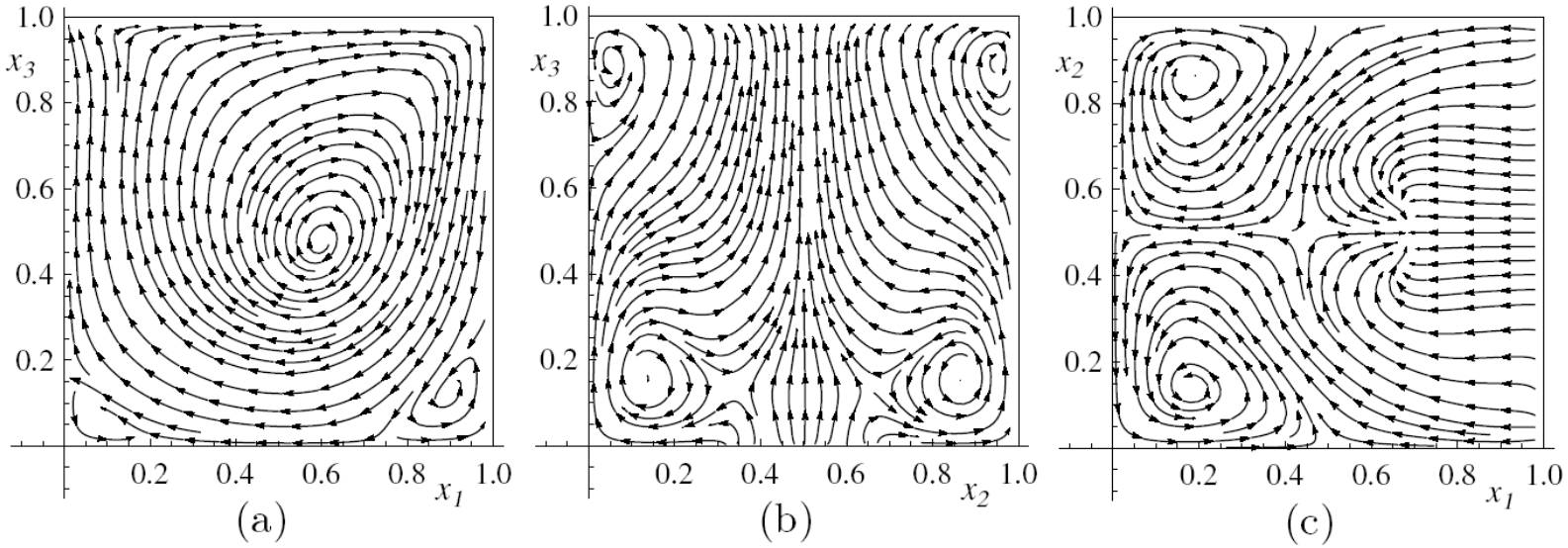


Fig. 3. Pseudo streamlines in different sections of the cubic cavity at Re = 1000: (a) section $x_2 = 0.5$; (b) section $x_1 = 0.5$; (c) section $x_3 = 0.5$

5 Convergence acceleration in 2D by a combined use of the multigrid and Krylov's algorithm

Table 3

Influence of the application of grids sequence $5 \to 10 \to 20 \to 40$ cells along each coordinate and the Krylov's algorithm on the convergence rate of the CLR method at Re = 1000

Method	N_{it}	CPU time, sec.	Acceleration factor	$\delta\mathbf{u}$	δp
$K_{mgr} = 1, k = 0$	132090	5824		$0.6961 \cdot 10^{-3}$	$0.6094 \cdot 10^{-3}$
$K_{mgr} = 4, k = 0$	50520	1640	3.55	$0.6961 \cdot 10^{-3}$	$0.6094 \cdot 10^{-3}$
$K_{mgr} = 1, k = 10$	5138	212	22.28	$0.6961 \cdot 10^{-3}$	$0.6092 \cdot 10^{-3}$
$K_{mgr} = 4, k = 10$	1925	58.6	99.39	$0.6961 \cdot 10^{-3}$	$0.6092 \cdot 10^{-3}$

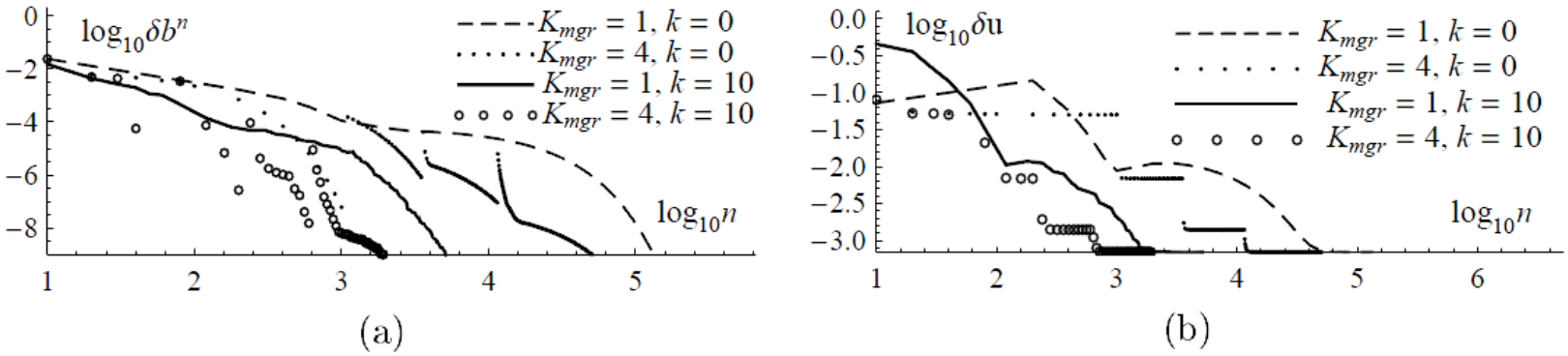


Fig. 7. The influence of the use of a sequence of grids and the quantity k in (16) on the convergence rate of the CLR method at Re = 1000: (a) the logarithm of pseudo-error δb^n; (б) the logarithm of error $\delta\mathbf{u}$

6 Conclusions

The computer algebra system *Mathematica* has been applied for constructing a new version of the method of collocations and least residuals (CLR) for solving the 3D Navier–Stokes equations. A large amount of symbolic computations, which arose in the work, was done efficiently with *Mathematica*. It is very important that the application of CAS has facilitated greatly this work, reduced at all its stages the probability of errors usually introduced by the mathematician-numerist at the development of a new algorithm.

The verification of the method accuracy by solving the well-known benchmark problem of the lid-driven cubic cavity flow and comparison with the most accurate published solutions of this problem [1], which were obtained by other researchers, have shown a high accuracy of the constructed method. This has confirmed additionally the efficiency and benefit of using the CASs for constructing new analytic-numerical methods.

Note

A more detailed presentation of the above material may be found in the following recently published paper:

306 ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ И ПРОГРАММИРОВАНИЕ. 2013. Т. 14

УДК 519.63.4:532.51.5

МЕТОД КОЛЛОКАЦИЙ И НАИМЕНЬШИХ НЕВЯЗОК ДЛЯ ТРЕХМЕРНЫХ УРАВНЕНИЙ НАВЬЕ–СТОКСА

В. П. Шапеев[1], Е. В. Ворожцов[2], В. И. Исаев[3], С. В. Идимешев[4]

V.P. Shapeev, E.V. Vorozhtsov, V.I. Isaev, S.V. Idimeshev. The method of collocations and least residuals for three-dimensional Navier-Stokes equations. Vychislitel'nye metody i programmirovanie. Vol. 14, P. 306-322 (2013)

Abstract

The method of collocations and least residuals, which was previously proposed for the numerical solution of two-dimensional Navier-Stokes equations, is extended here for the three-dimensional case. In the implemented version of the method, the solution is sought in the form of the expansion in the basis solenoidal functions. An overdetermined system of linear algebraic equations is obtained in each cell of the computational grid. This system is solved by the method of rotations. To accelerate the iteration process convergence a new algorithm is proposed, which is based on the Krylov's subspaces. The results of the verification of the method confirm its second order of convergence for the velocity vector components. The results of solving the benchmark problem of the lid-driven cubic cavityflow for the Reynolds numbers Re = 100 and Re = 1000 are presented. It is shown that the obtained results are very close to the most accurate results obtained by other authors with the aid of different numerical high-accuracy methods.

http://num-meth.srcc.msu.ru/zhurnal/tom_2013/pdf/v14r134.pdf