Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 3159-0040
Vol. 2 Issue 9, September - 2015

Application of the Method of Collocations and Least Residuals to the Solution of the Poisson Equation in Polar Coordinates

Shapeev V.P., Vorozhtsov E.V.
Khristianovich Institute of Theoretical and Applied Mechanics, Siberian Branch of the Russian Academy of Sciences, Novosibirsk 630090, Russia
vshapeev@ngs.ru, vorozh@itam.nsc.ru

***Abstract*—A version of the method of collocations and least residuals is proposed for the numerical solution of the Poisson equation in polar coordinates on non-uniform grids. By introducing the general curvilinear coordinates the original Poisson equation is reduced to the Beltrami equation. A uniform grid is used in curvilinear coordinates. The grid non-uniformity in the plane of the original polar coordinates is ensured with the aid of functions controlling the grid stretching and entering the formulas of the passage from polar coordinates to the curvilinear ones. The method has been verified on three test problems having the exact analytic solutions. The examples of numerical computations show that if the radial coordinate origin lies outside the computational region then the proposed method has the second order of accuracy. If the computational region contains the singularity then the application of a non-uniform grid along the radial coordinate enables an increase in the numerical solution accuracy by factors from 1.7 to 5 in comparison with the uniform grid case at the same number of grid nodes[1].**

Keywords—Poisson equation; polar coodinates; the method of collocations and least residuals

I. INTRODUCTION

At the modeling of many physical processes, it is necessary to solve the Poisson equation in a circle or in an annulus between two concentric circles. In particular, when solving numerically the Navier–Stokes equations governing viscous incompressible fluid flows with the aid of projection methods one needs the solution of the Poisson equation for the pressure. The existing numerical methods for solving the Posson equations in the regions with circular boundaries (in the two-dimensional case) and in the regions with cylindrical boundaries (in the three-dimensional case) may be subdivided into two groups.

The methods, which enable the solution of the Poisson equations in the disc or in an annulus directly in Cartesian rectangular coordinates belong to the first group. In the two-dimensional case, one solves the Poisson equation

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=f(x,y), \qquad (1)$$

where x and y are the Cartesian rectangular coordinates, and $f(\cdot)$ is a given function. The immersed boundary method was proposed for the first time in [1], where the approximation of the δ-function was used for "smearing" the solution within a thin strip in the neighborhood of the computational region boundaries. A symmetric discretization of the Poisson equation was proposed in [2] for the case when the Dirichlet conditions are specified at the boundary of an irregular spatial region. According to [2], the computational region is completed to a rectangular one, and in the fictitious cells lying outside the original irregular computational region and near its boundary, the numerical solution values are computed with the aid of linear extrapolation. This enables the application of standard central differencing approximations in the entire spatial region for the Laplace operator approximation. The approximation order of the immersed boundary method described in [2] was increased in [3] up to the third and fourth orders with the aid of specifying the quantities in fictitious cells by extrapolation formulas of higher orders than the first order.

A finite volume method was presented in [4] for the numerical solution of the Poisson equation with variable coefficients in Cartesian coordinates in irregular domains with the Dirichlet boundary conditions. In addition, a multigrid algorithm was used in [4] for convergence acceleration. The Poisson equation was solved in an elliptic region in [5] by a high-order difference method with a special approximation of the boundary condition. The convection-diffusion equation was solved in [6] by the method of collocations and least squares in a region with curved boundary on an adaptive rectangular grid with irregular cells at the region boundary. A projection difference method was proposed in [7] for solving the Navier–Stokes equation on an adaptive Cartesian rectangular grid. Within the framework of this method, the Poisson equation for the pressure was solved, and a constraint was imposed on a grid that the ratio of sizes of two neighboring cells cannot exceed 2. This limitation for the size of neighboring cells was overcome in the work [8], where a finite

[1] The work was partially supported by the Russian Foundation for Basic Research (project No. 13-01-00227)

difference scheme for the numerical solution of the Poisson equation in irregular regions on an adaptive rectangular grid refining near the region boundary was presented. The refinement criterion was based on the estimation of the proximity to the irregular boundary so that the cells of the least size are located at the boundary. The data structure in the form of an octree was used in [7] to store the data on the spatial discretization, and in [8], the data structures in the form of quadtrees and octrees were employed. A shortcoming of using the data structures in the quadtree and octree form was indicated in [8]: some CPU time expenses are needed to pass along the tree from its root to the needed node of the graph.

The second group of the works devoted to the development of the numerical techniques for solving the Poisson equation in the discs or annuli is constituted by the works in which the Poisson equation in polar and cylindrical coordinates is solved in the two- and three-dimensional cases, respectively. The convenience of using the above curvilinear coordinates consists of the fact that the spatial computational region becomes a rectangle in the two-dimensional case and a parallelepiped in the three-dimensional case. The two-dimensional Poisson equation in polar coordinates was approximated in the work [9] by a finite difference scheme having a centered three-point stencil along each of the both polar coordinates. The efficient spectral-difference methods were developed later for solving the Poisson equations in polar and cylindrical coordinates by using the discrete fast Fourier transform. In the two-dimensional case, one obtains for the coefficients of the Fourier expansion a system of linear algebraic equations (SLAE), which is solved efficiently by the Thomas method, and in the three-dimensional case, the arising SLAE is solved by the matrix factorization technique. A second-order difference scheme was constructed in [10] for the Fourier coefficients. A compact fourth-order difference scheme was presented in [11] for the Fourier coefficients in the case of solving the Poisson equation in polar coordinates. The spectral-difference method of the work [10] was applied in [12] for the numerical solution of the Poisson equation for the pressure correction in cylindrical coordinates within the framework of a finite difference method [13] with the aid of which a problem of the viscous incompressible gas flow in a cylindrical casing with a rotating disc was solved numerically.

One should note a shortcoming of spectral-difference methods for solving the Poisson equations in polar and cylindrical coordinates: the grid along the circumferential coordinate must be uniform. The highest efficiency of the discrete fast Fourier transform is reached only in the case when the number of nodes N_θ along the circumferential coordinate has the form $N_\theta = 2^N + 1$, where N is a positive integer, $N > 1$.

As is known, at an adequate generation and use of non-uniform grids one can increase significantly the numerical solution accuracy in comparison with the use of a uniform grid with the same number of nodes [6, 14, 15]. In this connection, a number of numerical techniques were developed for solving the Poisson equation in polar coordinates on non-uniform grid. In particular, a numerical technique using the Green function was proposed in [16], where the grid was non-uniform only in the radial direction. A compact fourth-order difference scheme on non-uniform grid was proposed in [17] for the two-dimensional convection-diffusion equation in polar coordinates. Test computations have confirmed the fourth order of accuracy of the scheme.

There are many applied problems, in which it is desirable to apply the non-uniform grid along the circumferential coordinate θ. These are the problems of computing the fluid flows in the devices, for which the presence of a junction with some technological elements in the interval of the circumferential coordinate variation $0 < \theta \le 2\pi$ is typical, the size of which is much less than the external diameter of the device. The sprayers, injectors, and hydraulic drives are the examples of such devices in technologies.

A sufficiently universal applicability of the method of collocations and least residuals (CLR) for solving various initial- and boundary-value problems for partial differential equations of different types was demonstrated previously in the works [18–24]. In this connection, it appears reasonable to investigate the applicability of this method also for the numerical solution of the Poisson equation in polar coordinates.

The CLR method for the numerical solution of boundary-value problems for differential equations was developed relatively recently. It is a projection-grid technique. The solution is sought therein in each cell of a difference grid in the form of a linear combination of basis elements of some finite-dimensional functional space. One uses as the latter, in view of certain convenience, the space of polynomials. The CLR method differs from other numerical methods in that the numerical solution of the problem reduces to the solution of an overdetermined SLAE. The solution of the latter is found from the requirement of the minimization of a functional of the residual of problem equations on the numerical solution of the problem. Due to such a combination of the method of collocations with a “strong” requirement to the solution of the discrete problem, the solution properties (smoothness, accuracy) improve in comparison with the solutions obtained by a simple method of collocations. In fact, the CLR method possesses also several other improved properties as compared to the method of collocations. In particular, the residual functional minimization contributes to the suppression (damping) of various disturbances arising in the process of problem solution and speeds up the convergence of the solution at the iteration technique of its construction. A more detailed review of the bibliography on the CLR method as well as a description of the applied problems, which were solved by this method, may be found in [20, 23–25].

In the present work, we propose the versions of the CLR method for the numerical solution of the two-dimensional Poisson equation in polar coordinates on both uniform and non-uniform grids.

II. The CLR Method for the Numerical Solution of the Poisson Equation in Polar Coordinates

As a result of the passage from the Cartesian coordinates x,y to polar coordinates r, θ by the formulas $x=r\cos\theta$, $y=r\sin\theta$ the Poisson equation (1) takes the form

$$\frac{\partial^2 u}{\partial r^2}+\frac{1}{r}\frac{\partial u}{\partial r}+\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}=\overline{f}(\theta,r), \qquad (2)$$

where $\overline{f}(\theta,r)=f(r\cos\theta,r\sin\theta)$. We will omit the bar over f in the following for the sake of brevity. Equation (2) is solved in the rectangular region

$$\Omega=\{(\theta,r),\ 0\le\theta<2\pi,\ R_1\le r\le R_2\} \qquad (3)$$

under the Dirichlet boundary conditions

$$u=g_1(\theta), r=R_1;\quad u=g_2(\theta), r=R_2;\quad 0\le\theta<2\pi. \qquad (4)$$

In (3) and (4), R_1 and R_2 are the given quantities, $0 \le R_1 < R_2$. The periodicity condition is specified at the boundaries $\theta = 0$ and $\theta = 2\pi$:

$$u(0,r)=u(2\pi,r),\quad R_1\le r\le R_2. \qquad (5)$$

Let us formulate the "discrete" problem approximating the original differential boundary-value problem. In the CLR method, a computational grid is generated in the spatial computational region (3). This grid may be non-uniform along the both coordinates θ and r. Denote the r-coordinate of the jth grid node on the r-axis by r_j, and let N_r be the number of nodes of the non-uniform grid in the interval $[R_1,R_2]$. The set of grid nodes $r_1,\ldots,r_{N_r}$ must satisfy the relations $R_1 = r_1 < r_2 < \ldots < r_{N_r} = R_2$. One specifies similarly the set of grid nodes $\theta_1,\ldots,\theta_{N_\theta}$ in the interval $[0,2\pi)$ in such a way that the relations $0 = \theta_1 < \theta_2 < \ldots < \theta_{N_\theta} = 2\pi$ are satisfied, where N_θ is the number of grid nodes in the interval $[0,2\pi)$. Denote by $\Omega_{i,j}$ a subregion of region (3), which is occupied by a cell with indices i, j that is

$$\{(\theta,r),\ \theta_i\le\theta\le\theta_{i+1},\ r_j\le r\le r_{j+1}\},\quad i=1,\ldots,N_\theta-1,\quad j=1,\ldots,N_r-1. \qquad (6)$$

One often encounters in fluid dynamics problems the spatial subregions, in which the solution has large gradients. In the case of a uniform grid, such subregions may have a size of less than one grid step; in these cases, the numerical algorithm can simply "not identify" such narrow transitional regions, and this may lead to considerable errors and incorrect results of the numerical simulation. In such situations, the application of non-uniform grids clustering in the subregions of large solution gradients makes it possible to increase the accuracy of simulation.

One of the simplest techniques of controlling the grid stretching in the case of the Poisson equation (10 consists of the use of the mapping [14]

$$x=f_2(\eta)\cos f_1(\xi),\quad y=f_2(\eta)\sin f_1(\xi), \qquad (7)$$

where the monitoring functions $f_1(\xi)$, $f_2(\eta)$ enter the relations $\theta=f_1(\xi)$, $r=f_2(\eta)$ and are specified by the user with regard for the specifics of the problem to be solved. The computational region in the plane of curvilinear coordinates (ξ, η) still remains rectangular as in the case when $f_1(\xi)=\xi$, $f_2(\eta)=\eta$. Let us assume following [14,15] that the computational grid in the (ξ,η) plane is square with steps $\Delta\xi = \Delta\eta = 1$. If $f_1(\xi)\ne\xi$ or $f_2(\eta)\ne\eta$, then the computational grid in the original plane (θ, r) will be non-uniform.

Equation (1) takes the following form at the passage from the variables x, y to curvilinear coordinates ξ, η [15]:

$$\Delta_B u(\xi,\eta)=\overline{f}(\xi,\eta), \qquad (8)$$

where $\Delta_B u$ is the Beltrami operator,

$$\overline{f}(\xi,\eta)=f(f_2(\eta)\cos f_1(\xi), f_2(\eta)\sin f_1(\xi)),$$

$$\Delta_B u=\frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial\xi}\left(\frac{g_{22}u_\xi-g_{12}u_\eta}{\sqrt{g}}\right)+\frac{\partial}{\partial\eta}\left(\frac{g_{11}u_\eta-g_{12}u_\xi}{\sqrt{g}}\right)\right], \qquad (9)$$

g_{ij} (i, j=1, 2) are the scalar products of covariant tangent vectors, $g_{ij}=\vec{x}_{\xi_i}\cdot\vec{x}_{\xi_j}$, $i,j=1,2$, where $\xi_1\equiv\xi$, $\xi_2\equiv\eta$, $x_\xi=\partial x(\xi,\eta)/\partial\xi$, $y_\xi=\partial y(\xi,\eta)/\partial\xi$, etc.; $\vec{x}_\xi=(x_\xi,y_\xi)$, $\vec{x}_\eta=(x_\eta,y_\eta)$, etc., that is

$$g_{11}=x_\xi^2+y_\xi^2, g_{22}=x_\eta^2+y_\eta^2, g_{12}=g_{21}=x_\xi x_\eta+y_\xi y_\eta,\quad \sqrt{g}=x_\xi y_\eta-x_\eta y_\xi. \qquad (10)$$

The computation of quantities g_{ij} according to (10) in the specific case of the mapping (7) leads to the following expression for the Beltrami operator:

$$\Delta_B u=\frac{1}{f_2(\eta)f'_1(\xi)f'_2(\eta)}\left\{\frac{\partial}{\partial\xi}\left[\frac{f'_2(\eta)u_\xi}{f_2(\eta)f'_1(\xi)}\right]+\frac{\partial}{\partial\eta}\left[\frac{f'_1(\xi)f_2(\eta)u_\eta}{f'_2(\eta)}\right]\right\}. \qquad (11)$$

In each cell $\Omega_{i,j}$, the local coordinates y_1 and y_2 are used in the CLR method along with the global coordinates ξ and η. For the implementation of the CLR method it is convenient to introduce the local coordinates in such a way that they vary from −1 to +1 within the cell. Since the computational grid in the (ξ, η) plane has the steps $\Delta\xi = \Delta\eta = 1$, the local variables y_1 and y_2 are introduced by the formulas

$$y_1 = \frac{\xi - \xi_{i+1/2}}{0.5}, \quad y_2 = \frac{\eta - \eta_{j+1/2}}{0.5}, \qquad (12)$$

where $(\xi_{i+1/2}, \eta_{j+1/2})$ are the coordinates of the geometric center of the cell $\Omega_{i,j}$ in the (ξ, η) plane. The formulas

$$\frac{\partial}{\partial \xi} = \frac{dy_1}{d\xi} \cdot \frac{\partial}{\partial y_1} = \frac{1}{0.5} \frac{\partial}{\partial y_1} = 2 \frac{\partial}{\partial y_1}, \quad \frac{\partial}{\partial \eta} = 2 \frac{\partial}{\partial y_2} \qquad (13)$$

enable one to replace the differentiation with respect to ξ and η in (11) with the differentiation with respect to y_1 and y_2. Besides, it is necessary to replace ξ and η in $f_2(\eta), f_1'(\xi), f_2'(\eta)$ by the formulas $\xi = 0.5y_1 + \xi_{i+1/2}$, $\eta = 0.5y_2 + \eta_{j+1/2}$.

In the works devoted to the application of the CLR method for solving various boundary-value problems for partial differential equations, which were mentioned above in the Introduction, the polynomials in local coordinates were used for the solution approximation in each computational grid cell. In the present work, the polynomial representation of the solution of the Poisson equation in each cell $\Omega_{i,j}$ is also employed. Let $U(y_1,y_2)$ be an approximate solution in the cell of the polynomial form. In order to have the possibility of approximating the second derivatives the polynomial must have at least the second order in variables y_1 and y_2. Therefore, we employ in the following the second-order polynomial of the form

$$U(y_1,y_2) = a_1 + a_2 y_1 + a_3 y_2 + a_4 y_1^2 + 2a_5 y_1 y_2 + a_6 y_2^2. \qquad (14)$$

The derivatives $f_1'(\xi), f_2'(\eta), f_1''(\xi), f_2''(\eta)$ enter formula (11). These derivatives were approximated at the center of the cell $\Omega_{i,j}$ with the second order of accuracy. Let us illustrate the procedure for calculating these derivatives by the example of the derivatives $f_2'(\eta), f_2''(\eta)$. The central differences were used for their approximation in internal cells [14, 15]:

$$f_2'(\eta_{j+1/2}) = r_{j+1} - r_j, \; f_2''(\eta_{j+1/2}) = r_{j+3/2} - 2r_{j+1/2} + r_{j-1/2}. \qquad (15)$$

In the left boundary cell $\Omega_{i,1}$, we apply the right one-sided differences:

$$f_2'(\eta_{3/2}) = (1/2)(4r_{5/2} - 3r_{3/2} - r_{7/2}),$$
$$f_2''(\eta_{3/2}) = r_{7/2} - 2r_{5/2} + r_{3/2}. \qquad (16)$$

In the right boundary cell Ω_{i,N_r-1}, we apply the left one-sided differences:

$$f_2'(\eta_{N_r-1/2}) = (1/2)(r_{N_r-1/2} - 4r_{N_r-3/2} + r_{N_r-5/2}),$$
$$f_2''(\eta_{N_r-1/2}) = r_{N_r-1/2} - 2r_{N_r-3/2} + r_{N_r-5/2}. \qquad (17)$$

It is to be noted that at the application of the CLR method for solving any problems, it is important that the equations of the overdetermined system, which play equal role in the approximate solution, have approximately equal weight coefficients. Denote by

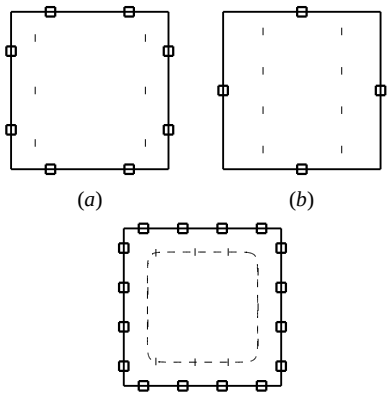

Fig. 1. *Versions of the specification of collocation and matching points: (a) N_c = 6, N_m = 2; (b) N_c = 8, N_m = 1; (c) N_c = 11, N_m = 4.*

Δ_{B,y_1,y_2} the Beltrami operator in local variables y_1 and y_2. Not that the factor $1/\left[f_2'(\eta)\right]^2$ enters the Beltrami operator (8). This factor has the order of smallness $1/O\left(h_r^2\right)$ in the uniform grid case, where h_r is the grid step in the interval [R_1, R_2]. And the coefficients of the equations obtained from the boundary condition have the order of smallness $O(1)$. To ensure the same orders of smallness for the coefficients of all equations of the algebraic system for $a_1,\ldots,a_6$ it is enough to multiply the both sides of the Beltrami equation by a quantity of the order $O\left(h_r^2\right)$. One can ensure this by multiplying the equation by the quantity $\left[f_2'(\eta)\right]^2$:

$$\left[f_2'(\eta)\right]^2 \Delta_{B,y_1,y_2} U = \left[f_2'(\eta)\right]^2 F(y_1,y_2), \qquad (18)$$

where $F(y_1,y_2) = \bar{f}\left(0.5y_1 + \xi_{i+1/2}, 0.5y_2 + \eta_{j+1/2}\right)$. This results in some improvement of the numerical solution accuracy.

The number of collocation points N_c in each cell $\Omega_{i,j}$ and their location inside the cell are specified by the user, and this can be done in different ways. More than 20 versions of the specification of the local coordinates ($y_{1,i,m}$, $y_{2,j,m}$) of collocation points were implemented in the given study. For the values N_c = 6 and N_c = 8, two different techniques of placing the collocation points inside the cell were implemented. In the first technique, at N_c = 6 the coordinates of collocation points are as follows: $\left(\pm\omega, -\frac{2}{3}\right)$, $(\pm\omega, 0)$, $\left(\pm\omega, \frac{2}{3}\right)$, where ω is a value specified by the user in the interval $0 < \omega < 1$, see Fig. 1, (*a*). The collocation points are shown in Fig. 1 by dark circles, and $\omega = 0.7$. At N_c = 8, the local coordinates of collocation points were computed in the first technique by the formulas $\left(\pm\omega, -\frac{3}{4}\right), \left(\pm\omega, -\frac{1}{4}\right), \left(\pm\omega, \frac{1}{4}\right), \left(\pm\omega, \frac{3}{4}\right)$, see Fig. 1, (*b*).

Thus, at N_c = 6 and N_c = 8, a larger number of rows of collocation points were specified along the polar axis r in the cell than along the θ axis. This was done for the purpose of a more accurate computation of the solution near the boundaries $r = R_1$ and $r = R_2$, where the boundary layers with large solution gradients may be available.

Another algorithm for specifying the collocation points has also been implemented in the same computer code at $N_c \geq 2$. In this algorithm, the collocation points were set at the same angular distance from one another on the closed curve

$$\left(\frac{y_1}{\omega}\right)^M + \left(\frac{y_2}{\omega}\right)^M = 1, \qquad (19)$$

where M is a user-specified even number, $M \geq 2$. Fig. 1, (*c*) shows the example of specifying 11 collocation points by the given technique; the dashed line shows curve (19) at M = 12. It is to be noted that the collocation points are located in the cell $\Omega_{i,j}$ asymmetrically with respect to the straight lines $y_1 = 0$, $y_2 = 0$ at odd N_c (see Fig.1, (*c*)), which may deteriorate to some extent the accuracy of the solution obtained by the CLR method. It is, therefore, desirable to use the even values of parameter N_c.

The substitution of expression (14) in (18) leads to an algebraic equation, which is linear in the coefficients $a_1,\ldots,a_6$. The coordinates of N_c collocation points ($y_{1,i,m}$, $y_{2,j,m}$), m = 1,…,N_c are then substituted in this linear equation. As a result, one obtains N_c collocation equations. Since the number of unknown coefficients $a_1,\ldots,a_6$ in (14) is equal to six, it is desirable to specify in each cell six or more collocation points.

As in [**Error! Reference source not found.**], one specifies on the sides of each cell the conditions for matching the solution therein with the solutions in neighboring cells. These conditions ensure the unique piecewise polynomial solution. The requirements of the continuity of a linear combination of the values of the approximate solution and its derivative along a normal to the wall have been taken here as such conditions:

$$\sigma_1 h \partial U^+ / \partial n + \sigma_2 U^+ = \sigma_1 h \partial U^- / \partial n + \sigma_2 U^-. \qquad (20)$$

One takes in the left-hand side of these relations the solution in the current cell, and in the right-hand side, one takes the solution in the neighboring cell. The points at which equations (20) are written are called the matching points. Here $n = (n_1, n_2)$ is the external normal to the cell side, and U^+, U^- are the limits of the function U as its arguments tend to the cell side from within and outside the cell; σ_1 and σ_2 are the non-negative weight parameters, which affect to some extent the condition number of the obtained system of linear algebraic equations (SLAE) and the solution convergence rate [26].

The quantity h in (20) is specified as follows: on the side $r = r_{j+1}$ of the cell $\Omega_{i,j}$ we assume h = ½ according to (12). Then $h\partial U^+/\partial n = h \cdot dy_2/dn \cdot dU^+/dy_2 = (1/2)\,\partial U^+/\partial y_2$. We have similarly on the side $\theta = \theta_{i+1}$: $h_\theta \partial U^+/\partial n = h_\theta \cdot dU^+/d\theta = \partial U^+/\partial y_1$, h_θ = ½. Denote by N_m the number of matching points on each cell side. Since the number of cell sides is equal to four, we obtain $4N_m$ matching conditions in each cell (the matching points are shown by small squares in Fig. 1).

If the cell side on which r = const belongs to the boundary of the Ω region, then one writes the boundary conditions $U(y_1,y_2) = g_1$ or $U(y_1,y_2) = g_2$ according to (4) instead of the matching conditions on this side at the points, to which on the cell sides lying inside the region the points of assigning the matching conditions correspond.

In the matching conditions (20), the periodicity conditions (5) were taken into account along the θ coordinate in the boundary cells $\Omega_{1,j}$ and $\Omega_{N_\theta-1,j}$, j = 1,…,N_r – 1. Consider at first the cell $\Omega_{1,j}$. The side θ_1 = 0 of this cell is simultaneously the side $\theta_1 = 2\pi$ of the cell $\Omega_{N_\theta-1,j}$. Therefore, equality (20) was implemented in the cell $\Omega_{1,j}$ as follows:

$$\left[\sigma_1 \frac{\partial U(y_1,y_2)}{\partial y_1} + \sigma_2 U(y_1,y_2)\right]_{i=1,y_1=-1} = \left[\sigma_1 \frac{\partial U(y_1,y_2)}{\partial y_1} + \sigma_2 U(y_1,y_2)\right]_{i=N_\theta-1,y_1=1}. \qquad (21)$$

In a similar way, the equation

$$\left[\sigma_1 \frac{\partial U(y_1,y_2)}{\partial y_1} + \sigma_2 U(y_1,y_2)\right]_{i=N_\theta-1,y_1=1} = \left[\sigma_1 \frac{\partial U(y_1,y_2)}{\partial y_1} + \sigma_2 U(y_1,y_2)\right]_{i=1,y_1=-1}. \qquad (22)$$

was included in the SLAE when assembling it for the cell $\Omega_{N_\theta-1,j}$.

In the version of the method implemented here, the numerical solution of the global problem is found iteratively in the so-called Gauss–Seidel process. In this process, all cells of the region are scanned sequentially at each global iteration after the initial guess has been assigned to the solution in each cell. One solves in each cell a SLAE, which determines a “local” piece of the global solution. In the implemented version, the solution in the cell was found by an orthogonal method using the reduction of the matrix of the overdetermined SLAE by the Givens method to the upper triangular form. In doing so the solution in a current cell is matched with the aid of matching conditions with the solutions available in neighboring cells. If the current cell belongs to the region boundary, the boundary conditions of the problem are then realized therein because their approximation has been included in the SLAE determining the solution in this cell. The global iterations are terminated at the satisfaction of the criteria (27), (28) for stopping the iterations.

At the practical implementation of the CLR method, the solution is found in the cells $\Omega_{i,j}$ in the direction of the increasing indices *i*, *j*. Therefore, at the SLAE assembly in the cell $\Omega_{1,j}$, the solution in the cell $\Omega_{N_\theta-1,j}$ is not known yet. In this connection, we have implemented the computation with the use of the alternating Schwarz method [28]. According to this method, the values known at the moment of the solution in the given cell were taken as U^- in (20) and (21). Let *k* be the iteration number, *k* = 0,1,2,…. Condition (21) was then implemented as follows:

$$\left[\sigma_1\frac{\partial U^{k+1}(y_1,y_2)}{\partial y_1}+\sigma_2 U^{k+1}(y_1,y_2)\right]_{i=1,y_1=-1}=$$
$$\left[\sigma_1\frac{\partial U^{k}(y_1,y_2)}{\partial y_1}+\sigma_2 U^{k}(y_1,y_2)\right]_{i=N_\theta-1,y_1=1}.$$

And on the right-hand side of equation (22), one can take the values of $U(y_1,y_2)$ and $\partial U(y_1,y_2)/\partial y_1$ at the (*k*+1)th iteration because at the computation in the direction of the increasing index *i*, the values of the coefficients $a_1,\ldots,a_6$ in (. (14) are already known by the moment when the computational process reaches the boundary cell $\Omega_{N_\theta-1,j}$.

The initial guess $u^0(\theta, r)$ was specified with regard for the boundary conditions (5) by a linear interpolation of the values $g_1(\theta)$ and $g_2(\theta)$:

$$u^0(\theta,r)=\frac{\left[g_1(\theta)-g_2(\theta)\right]r+g_2(\theta)R_1-g_1(\theta)R_2}{R_1-R_2}.$$

As a result, one obtains in each cell a system involving $N_c + 4N_m$ equations, where $N_c \geq 6$, $N_m \geq 1$, by including in the SLAE the collocation equations and the matching conditions. By virtue of the fact that $N_c + 4N_m \geq 10$, the SLAE for finding six unknown coefficients $a_1,\ldots,a_6$ in (. (14) is overdetermined. The Givens method [23, 29] was applied for the numerical solution of this SLAE.

The Poisson equation (2) contains a singularity at point *r* = 0. The solution itself is regular if the right-hand side of the Poisson equation and the boundary conditions are sufficiently smooth. In the spectral-difference methods [10, 11], the singularity problem was solved by using a uniform grid on the *r* axis, which was shifted by a half-step from the point *r* = 0, and the symmetry conditions of the coefficients of the expansion into the Fourier series.

There is no singularity problem in the proposed CLR method at finite grid step values. The collocation points are specified inside the cell, therefore, always $r = r_{j,m} > 0$ ($j = 1,\ldots,N_r - 1$; $m = 1,\ldots,N_c$). There is no division by *r* in the matching conditions (20) that is they have no singularity.

To derive all needed formulas of the numerical algorithm for problem solution by the CLR method all the above analytic calculations were implemented in the system of symbolic computations *Mathematica*. All the arithmetic operators were translated into the FORTRAN operators with the aid of this system and were included in the corresponding places of the computer code for the numerical solution of the problem.

III. Computational Results

To investigate the accuracy of the above-proposed version of the CLR method the following test solutions of the Poisson equation (1) were taken [10, 16]:

$$u(x,y)=3e^{x+y}\left(x-x^2\right)\left(y-y^2\right)+5, \quad (23)$$

$$u(x,y)=\frac{e^x+e^y}{1+xy}, \quad (24)$$

$$u(x,y)=\left((x+1)^{5/2}-(x+1)\right)\left((y+1)^{5/2}-(y+1)\right). \quad (25)$$

The corresponding right-hand sides *f*(*x*, *y*) are easily obtained by substituting solutions (23)–(25) into the left-hand side of equation (1). Then one finds the expression for the function $\bar{f}(\theta,r)$ in (2). Note that at the use of test (24), it is necessary to specify $R_2 < \sqrt{2}$ in (3) because at $R_2 = \sqrt{2}$ and θ = 3π/4, the denominator in (24) vanishes. One can also note that solution (25) possesses the property: *u*(*x*,*y*) = *u*(*y*,*x*).

The computations by the CLR method were done on both uniform and non-uniform grids along the *θ* and *r* axes. The non-uniform grids were generated along each axis by the same algorithm described in [14]. Let us briefly describe the algorithm for obtaining the non-uniform grid in the interval $R_1 \leq r \leq R_2$. In this algorithm, one must at first specify the grid steps $r_2 - r_1$ and $r_{N_r} - r_{N_r-1}$ by the formulas: $r_2 - r_1 = \lambda_{r,L}\cdot h_r$, $r_{N_r} - r_{N_r-1} = \lambda_{r,R}\cdot h_r$, where h_r is the uniform grid step in the interval $R_1 \leq r \leq R_2$, the grid has N_r nodes that is $h_r = (R_2 - R_1)/(N_r - 1)$; $\lambda_{r,L}$ and $\lambda_{r,R}$ are the coefficients specified by the user, $0 < \lambda_{r,L}, \lambda_{r,R} \leq 1$. If $\lambda_{r,L} < 1$, $\lambda_{r,R} = 1$, then one obtains along the *r* axis a grid clustering near the boundary $r = R_1$; if $\lambda_{r,L} < 1$ and $\lambda_{r,R} < 1$, then the grid is refined near the both boundaries $r = R_1$ and $r = R_2$; if $\lambda_{r,L} = 1$, $\lambda_{r,R} < 1$, then the grid is refined near the boundary $r = R_2$; and, finally, at $\lambda_{r,L} = \lambda_{r,R} = 1$, one obtains a uniform grid. The function sinh (*ζ*) is involved in the computations of the coordinates of grid node coordinates in this algorithm.

To determine the error of the method on a specific spatial computational grid the grid analogs of the error norms were computed with the use of the norms of the L_p spaces ($p \geq 1$) by the formula

$$\left\|\delta u^k\right\|_p=\left[\frac{1}{\pi\left(R_2^2-R_1^2\right)}\sum_{i=1}^{N_\theta-1}\sum_{j=1}^{N_r-1}\left(u^k_{i+\frac{1}{2},j+\frac{1}{2}}-u^{ex}_{i+\frac{1}{2},j+\frac{1}{2}}\right)^p\times r_{j+\frac{1}{2}}\left(r_{j+1}-r_j\right)\left(\theta_{i+1}-\theta_i\right)\right]^{\frac{1}{p}}, \quad (26)$$

TABLE I. ERROR $\|\delta U\|_2$ AND CONVERGENCE RATE νU_2 ON A SEQUENCE OF GRIDS,

$$u(x,y) = 3e^{x+y}\left(x-x^2\right)\left(y-y^2\right)+5, \quad 0.5 \le r \le 1.$$

$N_\theta - 1$	$N_r - 1$	$\|\delta u\|_2$	ν_2
Uniform grid			
75	6	5.0539E−04	
100	8	2.8206E−04	2.03
150	12	1.2498E−04	2.01
200	16	7.0317E−05	2.00
250	20	4.5033E−05	2.00
Non-uniform grid along the r axis ($\lambda_{r,L}$ = 0.93, $\lambda_{r,R}$ = 1.0) and uniform grid along the θ axis			
75	6	5.6142E−04	
100	8	3.0738E−04	2.09
150	12	1.3454E−04	2.04
200	16	7.5413E−05	2.01
250	20	4.8216E−05	2.00
Uniform grid along the r axis and non-uniform grid along the θ axis ($\lambda_{\theta,L} = \lambda_{\theta,R}$ = 0.6)			
75	6	4.7726E−04	
100	8	2.6860E−04	2.00
150	12	1.1989E−04	1.99
200	16	6.7660E−05	1.99
250	20	4.3404E−05	1.99

TABLE II. ERROR $\|\delta U\|_\infty$ AND CONVERGENCE RATE νU_∞ ON A SEQUENCE OF GRIDS,

$$u(x,y) = 3e^{x+y}\left(x-x^2\right)\left(y-y^2\right)+5, \quad 0.5 \le r \le 1.$$

$N_\theta - 1$	$N_r - 1$	$\|\delta u\|_\infty$	ν_∞
Uniform grid			
75	6	1.1629E−03	
100	8	6.6402E−04	1.95
150	12	2.9861E−04	1.97
200	16	1.6847E−04	1.99
250	20	1.0805E−04	1.99
Non-uniform grid along the r axis ($\lambda_{r,L}$ = 0.93, $\lambda_{r,R}$ = 1.0) and uniform grid along the θ axis			
75	6	1.1213E−03	
100	8	6.8916E−04	1.69
150	12	3.2792E−04	1.83
200	16	1.9028E−04	1.89
250	20	1.2396E−04	1.92
Uniform grid along the r axis and non-uniform grid along the θ axis ($\lambda_{\theta,L} = \lambda_{\theta,R}$ = 0.6)			
75	6	1.1430E−03	
100	8	6.6327E−04	1.89
150	12	3.0050E−04	1.95
200	16	1.6884E−04	2.00
250	20	1.0842E−04	1.99

where $u^{ex}_{i+1/2,\,j+1/2}$ and $u^{k}_{i+1/2,\,j+1/2}$ are, respectively, the exact solution and the approximate solution by the CLR method, which have been computed at the center of the cell $\Omega_{i,j}$.

The convergence rate ν_p of the CLR method on a sequence of grids at the grid refinement was computed by the formula known in numerical analysis

$$\nu_p = \frac{\log\left(\left\|\delta u^k\left(h_{m-1}\right)\right\|_p / \left\|\delta u^k\left(h_m\right)\right\|_p\right)}{\log\left(h_{m-1}/h_m\right)},$$

where h_m, m = 2,3,… are some values of steps h_r and h_θ such that $|h_{r,m-1} - h_{r,m}| + |h_{\theta,m-1} - h_{\theta,m}| > 0$.

Let $a^k_{i,j,l}$ (k = 0,1,…; l = 1,…,6) be the value of the coefficient a_l in (. (14) in the cell $\Omega_{i,j}$ at the kth iteration.

The following condition was used for the termination of iterations by the Schwarz's alternating method:

$$\left\|\delta a^{k+1}\right\| < \varepsilon, \qquad (27)$$

where

$$\left\|\delta a^{k+1}\right\| = \max_{i,j}\left(\max_{1\le l\le 6}\left|a^{k+1}_{i,j,l} - a^{k}_{i,j,l}\right|\right), \qquad (28)$$

ε is a user-specified small positive number,

TABLE III. ERROR $\|\delta U\|_2$ AND CONVERGENCE RATE νU_2 ON A SEQUENCE OF GRIDS,

$$u(x,y) = 3e^{x+y}\left(x-x^2\right)\left(y-y^2\right)+5, \quad 0 \le r \le 1.$$

$N_\theta - 1$	$N_r - 1$	$\|\delta u\|_2$	ν_2
Uniform grid			
75	12	3.4861E−03	
100	16	2.3369E−03	1.39
150	24	1.3983E−03	1.27
200	32	1.0222E−03	1.09
250	40	8.2012E−04	0.99
Non-uniform grid along the r axis ($\lambda_{r,L}$ = 0.4, $\lambda_{r,R}$ = 0.998) and uniform grid along the θ axis			
75	12	2.0722E−03	
100	16	1.2641E−03	1.72
150	24	6.2044E−04	1.76
200	32	3.7129E−04	1.78
250	40	2.4852E−04	1.80

$$\varepsilon \le \left[\min_{i,j}\left\{\left(\theta_{i+1}-\theta_i\right),\left(r_{j+1}-r_j\right)\right\}\right]^2.$$

Numerical results presented in Tables I, II, III, and IV were obtained by using the value ω = 0.7 at the specification of collocation points; $\sigma_1 = \sigma_2$ = 1 in (20). The value $\varepsilon = 10^{-9}$ was taken in criterion (27) for

TABLE IV. ERROR $\|\Delta U\|_\infty$ AND CONVERGENCE RATE νU_∞ ON A SEQUENCE OF GRIDS,

$$u(x,y) = 3e^{x+y}\left(x-x^2\right)\left(y-y^2\right)+5, \quad 0 \le r \le 1.$$

$N_\theta - 1$	$N_r - 1$	$\|\delta u\|_\infty$	ν_∞
Uniform grid			
75	12	2.5131E−02	
100	16	2.0583E−02	0.69
150	24	1.5271E−02	0.74
200	32	1.2481E−02	0.70
250	40	1.1284E−02	0.45
Non-uniform grid along the r axis ($\lambda_{r,L}$ = 0.4, $\lambda_{r,R}$ = 0.998) and uniform grid along the θ axis			
75	12	1.1213E−03	
100	16	7.1873E−03	1.30
150	24	4.3878E−03	1.22
200	32	3.0676E−03	1.24
250	40	2.3191E−03	1.25

TABLE V. ERROR $\|\Delta U\|_2$ AND CONVERGENCE RATE νU_2 ON A SEQUENCE OF GRIDS,

$$u(x,y) = \left(e^x + e^y\right)/(1+xy), \quad 0 \le r \le 1.$$

$N_\theta - 1$	$N_r - 1$	$\|\delta u\|_2$	ν_2
Uniform grid			
75	12	1.2773E−03	
100	16	1.0188E−02	0.79
150	24	7.3624E−03	0.80
200	32	5.8088E−03	0.82
250	40	4.8153E−03	0.84
Non-uniform grid along the r axis ($\lambda_{r,L}$ = 0.3, $\lambda_{r,R}$ = 1.0) and uniform grid along the θ axis			
75	12	8.9070E−03	
100	16	5.9090E−03	1.43
150	24	3.3825E−03	1.38
200	32	2.2881E−03	1.36
250	40	1.6888E−03	1.36

computation termination. Six collocation points were used in each cell, equation (20) with M = 2 was used for specifying their coordinates. Four matching points were specified on each side of the cell $\Omega_{i,j}$.

Tables I and II present the numerical results obtained by the CLR method in the annulus between two concentric circles with radii R_1 = 0.5 and R_2 = 1.0, that is the singularity r = 0 is outside the computational region. One can see from these tables that the convergence rate ν_2 is very close to the second one; the convergence rate ν_∞ is also close to the second one, although it is slightly lower than 2.

As one can see further from Tables I and II, in the case of solving the Poisson equation in the annulus $0.5 \le r \le 1$, the application of a non-uniform grid along the r-axis does practically give no gain in accuracy. If

TABLE VI. ERROR $\|\Delta U\|_\infty$ AND CONVERGENCE RATE νU_∞ ON A SEQUENCE OF GRIDS,

$$u(x,y) = \left(e^x + e^y\right)/(1+xy), \quad 0 \le r \le 1.$$

$N_\theta - 1$	$N_r - 1$	$\|\delta u\|_\infty$	ν_∞
Uniform grid			
75	12	7.7051E−02	
100	16	6.9087E−02	0.38
150	24	5.6739E−02	0.49
200	32	4.8329E−02	0.56
250	40	4.2320E−02	0.60
Non-uniform grid along the r axis ($\lambda_{r,L}$ = 0.3, $\lambda_{r,R}$ = 1.0) and uniform grid along the θ axis			
75	12	4.9344E−02	
100	16	3.8715E−02	0.84
150	24	2.8247E−02	0.78
200	32	2.2689E−02	0.76
250	40	1.9107E−02	0.77

one uses a non-uniform grid along the θ-axis, then there is some improvement in solution accuracy.

If the singular point r = 0, however, a boundary point of the computational region, then the application of a non-uniform grid along the r-axis leads to a significant increase in accuracy, see Tables III and IV. The errors $\|\delta u\|_2$ and $\|\delta u\|_\infty$ drop, respectively, by 1.7–3.3 times and 2–5 times as compared to the uniform grid case.

One can see from the comparison of Tables I, II and III, IV that if the point r = 0 is a computational region boundary (that is R_1 = 0), then there occurs a considerable (by one or two decimal orders) increase in the errors $\|\delta u\|_2$ and $\|\delta u\|_\infty$ in comparison with the case when R_1 = 0.5. One can explain this effect as follows by the example of a uniform grid with steps h_r and h_θ along the r and θ axes, respectively. Consider the cells $\Omega_{i,j}$ lying near the point r = 0. According to (2), the derivative $\partial u/\partial r$ is divided by r. In the cell $\Omega_{i,1}$, the coordinates $r_{j,m}$ of collocation points satisfy the inequalities $0 < r_{j,m} < h_r$. This leads to the fact that in the cell under consideration, the order of the approximation error of the term $(1/r)\partial u/\partial r$ decreases. If $R_1 = O(1)$, then no such loss of the numerical solution accuracy is observed.

In the case of the test solution (24), the errors of the numerical solution obtained by the CLR method are higher than in the case of test (23), compare Tables I, II, III, IV with Tables V and VI. A similar effect of the numerical solution loss in the case of test (24) as compared to (23) was reported previously in [11], where a compact difference scheme was proposed for equation (2). It was found here in numerical experiments that the application of a non-uniform grid along the r-axis has made it possible to increase the numerical solution accuracy by two-three times in comparison with the uniform grid case, see Tables V and VI.

TABLE VII. ERROR $\|\Delta U\|_2$ AND CONVERGENCE RATE νU_2 ON A SEQUENCE OF GRIDS,

$$u(x,y)=\left((x+1)^{5/2}-(x+1)\right)\left((y+1)^{5/2}-(y+1)\right),$$

$0 \le r \le 1$.

$N_\theta - 1$	$N_r - 1$	$\|\delta u\|_2$	ν_2
Uniform grid			
75	12	3.4103E−03	
100	16	2.3446E−03	1.30
150	24	1.3717E−03	1.32
200	32	9.3142E−04	1.35
250	40	4.8153E−03	1.37
Non-uniform grid along the r axis ($\lambda_{r,L}$ = 0.4, $\lambda_{r,R}$ = 1.0) and uniform grid along the θ axis			
75	12	3.0055E−03	
100	16	1.7395E−03	1.90
150	24	8.2044E−04	1.85
200	32	4.7586E−04	1.89
250	40	3.1154E−04	1.90

TABLE VIII. ERROR $\|\Delta U\|_\infty$ AND CONVERGENCE RATE νU_∞ ON A SEQUENCE OF GRIDS,

$$u(x,y)=\left((x+1)^{5/2}-(x+1)\right)\left((y+1)^{5/2}-(y+1)\right),$$

$0 \le r \le 1$.

$N_\theta - 1$	$N_r - 1$	$\|\delta u\|_\infty$	ν_∞
Uniform grid			
75	12	2.0122E−02	
100	16	1.5867E−02	0.83
150	24	1.0719E−02	0.97
200	32	4.8329E−02	1.02
250	40	4.2320E−02	0.90
Non-uniform grid along the r axis ($\lambda_{r,L}$ = 0.4, $\lambda_{r,R}$ = 1.0) and uniform grid along the θ axis			
75	12	9.8639E−03	
100	16	6.8212E−03	1.28
150	24	3.7591E−03	1.47
200	32	2.5292E−03	1.38
250	40	1.8740E−03	1.34

In the case of the test solution (25) (see Tables VII and VIII), the accuracy of the numerical solution obtained by the CLR method proves to be much higher than in the case of test (24). When a discrete analog of the L_2 norm is used the numerical solution errors in the case of using a non-uniform grid on the r-axis are less than in the case of a uniform grid with the same number of nodes by the factors from 1.1 to 2.6, and the difference in the accuracy increases with the increasing number of grid nodes. In addition, the convergence rate ν_2 differs in the case of test (25) much less from the second order than in the case of test (24) when the non-uniform grid is used on the polar axis.

IV. CONCLUSIONS

A new version of the method of collocations and least residuals has been presented for the numerical solution of the Poisson equation in polar coordinates. The method has been verified on two test problems having the exact analytic solutions. It is shown that if the radial coordinate origin does not belong to the computational region then the proposed method has the second –order accuracy.

If the singularity — the radial coordinate origin — enters the computational region, then the application of a non-uniform grid along the radial coordinate enables an increase in the numerical solution accuracy by factors from 1.7 to 5 in comparison with the uniform grid case.

One can also note that the CLR method is well parallelizable. One can partition the entire computational region along the boundaries of grid cells into several subregions containing approximately equal number of cells. In each subregion, the global problem computation can be performed in parallel.

REFERENCES

[1] C. Peskin, “Numerical analysis of blood flow in the heart,” J. Comput. Phys., vol. 25, pp. 220-252, 1977.

[2] F. Gibou, R. Fedkiw, L.-T. Cheng, and M. Kang, “A second-order-accurate symmetric discretization of the poisson equation on irregular domains”, J. Comput. Phys., vol. 176, pp. 205-227, 2002.

[3] F. Gibou and R. Fedkiw, “A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem”, J. Comput. Phys., vol. 202, pp. 577-601, 2005.

[4] H. Johansen and P. Colella, “A cartesian grid embedded boundary method for poisson’s equation on irregular domains”, J. Comput. Phys., vol. 147, pp. 60-85, 1998.

[5] A.V. Shapeev and V.P. Shapeev, “Difference schemes of increased order of accuracy for solving elliptical equations in domain with curvilinear boundary”, Computat. Math. and Math. Phys., vol. 40, No. 2, pp. 213-221, 2000.

[6] V.V. Belyaev and V.P. Shapeev, “The method of collocations and least squares on adaptive grids in a region with curved boundary”, Vychislitelnye tehnologii, vol. 5, No. 4, pp. 12-21, 2000.

[7] S. Popinet, “Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries”, J. Comput. Phys., vol. 190, pp. 572-600, 2003.

[8] H. Chen, C. Min, and F. Gibou, “A supra-convergent finite difference scheme for the Poisson

and heat equations on irregular domains and non-graded adaptive Cartesian grids", J. Sci. Computing, vol. 31, Nos. ½, pp. 19-60, 2007.

[9] P.N. Swartztrauber and R.A. Sweet, "The direct solution of the discrete Poisson equation on a disc", SIAM J. Numer. Anal., vol. 10, pp. 900-907, 1973.

[10] M.-C. Lai, W.-W. Lin, and W. Wang, "A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries", IMA J. Numer. Anal., vol. 22, No. 4, pp. 537-548, 2002.

[11] M.-C. Lai, "A simple compact fourth-order Poisson solver on polar geometry", J. Comput. Phys., vol. 182, pp. 337-345, 2002.

[12] E.V. Vorozhtsov, "New analytic solutions of the problem of gas flow in a casing with rotating disc", Lect. Notes Comput. Sci., vol. 5743, pp. 350-372, 2009.

[13] R. Verzicco and P. Orlandi, "A finite-difference scheme for three-dimensional incompressible flows in cylindrical coordinates", J. Comput. Phys., vol. 123, No. 2, pp. 402-414, 1996.

[14] J.F. Thompson, Z.U.A. Warsi, and C.W. Mastin, Numerical Grid Generation: Foundations and Applications. New York: North-Holland, 1985.

[15] P. Knupp and S. Steinberg, Fundamentals of Grid Generation. Boca Raton: CRC Press, 1994.

[16] L. Borges and P. Daripa, "A fast parallel algorithm for the Poisson equation on a disk", J. Comput. Phys., vol. 169, pp. 151-192, 2001.

[17] R.K. Ray and J.C. Kalita, "A transformation-free HOC scheme for incompressible viscous flows on nonuniform polar grids", Int. J. Numer. Methods in Fluids, vol. 62, pp. 683-708, 2010.

[18] A.G. Sleptsov, "Collocation-grid solution of elliptic boundary-value problems", Modelirovanie v mekhanike, Vol. 5(22), No. 2, pp. 101-126, 1991.

[19] L.G. Semin, A.G. Sleptsov, and V.P. Shapeev, "Method of collocations–least squares for Stokes equations", Vychislitelnye tehnologii, vol. 1, No. 2, pp. 90-98, 1996.

[20] V.I. Isaev and V.P. Shapeev, "High-accuracy versions of the collocations and least squares method for the numerical solution of the Navier–Stokes equations", Computat. Math. and Math. Phys., vol. 50, No. 10, pp. 1670-1681, 2010.

[21] V.I. Isaev and V.P. Shapeev, "High-order accurate collocations and least squares method for solving the Navier - Stokes equations", Dokl. Math., vol. 85, pp. 71-74, 2012.

[22] V.P. Shapeev and E.V. Vorozhtsov, "CAS application to the construction of the collocations and least residuals method for the solution of 3D Navier–Stokes equations", Lect. Notes Comput. Sci., vol. 8136, pp. 381-392, 2013.

[23] V.P. Shapeev, E.V. Vorozhtsov, V.I. Isaev, and S.V. Idimeshev, "The method of collocations and least residuals for three-dimensional Navier–Stokes equations", Vychisl. Metody Programm., vol. 14, No. 3, pp. 306-322, 2013.

[24] V.P. Shapeev and E.V. Vorozhtsov, "Application of computer algebra systems to the construction of the collocations and least residuals method for solving the 3D Navier–Stokes equations", Modeling and Analysis of Information Systems, vol. 21, No. 5, pp. 131-147, 2014.

[25] V.P. Shapeev and E.V. Vorozhtsov, "CAS application to the construction of the collocations and least residuals method for the solution of the Burgers and Korteweg–de Vries–Burgers equations, Lect. Notes Comput. Sci., vol. 8660, pp. 432-446, 2014.

[26] V.I. Isaev, V.P. Shapeev, and S.A. Eremin, "Investigation of the properties of the method of collocations and least squares for solving boundary-value problems for Poisson equation and Navier–Stokes equations", Vychislitelnye tehnologii, vol. 12, No. 3, pp. 53-70, 2007. 1994.

[27] J.H. Ferziger and M. Perić, Computational Methods for Fluid Dynamics, 3rd ed. Berlin: Springer-Verlag, 2002.

[28] H.A. Schwarz, "Über einen Grenzübergang durch alternierendes Verfahren", Vierteljahrsschrift der naturforschenden Gesellschaft in Zürich, vol. 15, pp. 272-286, 1870.

[29] G.H. Golub and C.F. Van Loan, Matrix Computations, 3rd ed. Baltimore: Johns Hopkins University Press, 1996.