Available online at www.sciencedirect.com
ScienceDirect

Journal of Computational and Applied Mathematics 201 (2007) 175 – 181

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

A formally fourth-order accurate compact scheme for 3D Poisson equation in cylindrical coordinates

Ming-Chih Lai*, Jui-Ming Tseng

Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan

Received 7 July 2005; received in revised form 7 February 2006

Abstract

In this paper, we extend our previous work (M.-C. Lai, A simple compact fourth-order Poisson solver on polar geometry, J. Comput. Phys. 182 (2002) 337–345) to 3D cases. More precisely, we present a spectral/finite difference scheme for Poisson equation in cylindrical coordinates. The scheme relies on the truncated Fourier series expansion, where the partial differential equations of Fourier coefficients are solved by a formally fourth-order accurate compact difference discretization. Here the formal fourth-order accuracy means that the scheme is exactly fourth-order accurate while the poles are excluded and is third-order accurate otherwise. Despite the degradation of one order of accuracy due to the presence of poles, the scheme handles the poles naturally; thus, no pole condition is needed. The resulting linear system is then solved by the Bi-CGSTAB method with the preconditioner arising from the second-order discretization which shows the scalability with the problem size.

Keywords: Poisson equation; Cylindrical coordinates; Symmetry constraint; Fast Fourier transform; Bi-CGSTAB method

1. Introduction

In many physical problems, one often needs to solve the Poisson equation on a 3D non-Cartesian domain, such as cylindrical domain. For example, the projection method [1] in the simulation of incompressible flow in a pipe requires solving the pressure Poisson equation. It is convenient to rewrite the equation in cylindrical coordinates. The first problem that must be dealt with is the coordinate singularities (or poles) caused by the transformation. It is important to note that the occurrence of those singularities is due to the representation of the governing equation in the coordinates and the solution itself is regular if the source function and the boundary conditions are smooth.

For the past few years, the first author and his collaborators have developed a class of FFT-based fast direct solvers (FDSs) for Poisson equation on 2D [7] and 3D [6] cylindrical and spherical domains. The methods have three major features, namely, the coordinate singularities can be treated easily, the resulting linear equations can be solved efficiently by existing available fast algorithms, and the different boundary conditions can be handled without substantial differences. Besides, the method is easy to implement. Despite those aforementioned advantages of our algorithm, the numerical schemes in 3D domain [6] are only second-order accurate.

* Corresponding author.
E-mail address: mclai@math.nctu.edu.tw (M.-C. Lai).

0377-0427/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2006.02.011

Recently, the first author has developed a simple compact fourth-order Poisson solver on 2D polar geometry [5]. In fact, the scheme in [5] is formally fourth-order meaning that it has fourth-order accuracy only for the problem excluding the polar origin but degrades to third-order accuracy when the origin is included. Despite the degradation of one order of accuracy due to the presence of the pole, the scheme handles the pole naturally; thus, no pole condition is needed. There are a few papers in the literature that discuss fourth-order finite difference schemes for the Poisson equation in 2D polar [3,9,4] and 3D cylindrical coordinates [3,2]. However, those papers need to derive some special equations at $r=0$ (that is, the pole condition). In this paper, we shall extend the methodology presented in [5] to the 3D cylindrical geometry.

2. Formally fourth-order compact scheme for Poisson equation in cylindrical coordinates

As mentioned before, the goal of this paper is to derive a formally fourth-order compact scheme for Poisson equation in cylindrical coordinates. In particular, we consider the domain to be a fully circular cylinder so that the solution is periodic in azimuthal (θ) direction, and we also restrict the Dirichlet boundary conditions on the top, bottom and the sidewall boundaries. Thus, the Poisson problem in a finite circular cylinder $\Omega=\{0<r\leqslant 1, 0\leqslant\theta<2\pi, 0\leqslant z\leqslant 1\}$ can be written in cylindrical coordinates as

$$\frac{\partial^2 u}{\partial r^2}+\frac{1}{r}\frac{\partial u}{\partial r}+\frac{1}{r^2}\frac{\partial^2 u}{\partial\theta^2}+\frac{\partial^2 u}{\partial z^2}=f(r,z,\theta), \tag{2.1}$$

$$u(r,1,\theta)=u_{\mathrm{T}}(r,\theta), \quad u(r,0,\theta)=u_{\mathrm{B}}(r,\theta), \quad u(1,z,\theta)=u_{\mathrm{S}}(z,\theta). \tag{2.2}$$

2.1. Fourier mode equations

Since the solution u is periodic in θ, we can approximate it by the truncated Fourier series as

$$u(r,z,\theta)=\sum_{n=-N/2}^{N/2-1}\hat{u}_n(r,z)\mathrm{e}^{\mathrm{i}n\theta}, \tag{2.3}$$

where $\hat{u}_n(r,z)$ is the complex Fourier coefficient given by

$$\hat{u}_n(r,z)=\frac{1}{N}\sum_{k=0}^{N-1}u(r,z,\theta_k)\mathrm{e}^{-\mathrm{i}n\theta_k}, \tag{2.4}$$

and $\theta_k=2k\pi/N$ with N the number of grid points along a circle. The above transformation between the physical space and Fourier space can be efficiently performed by the fast Fourier transform (FFT) with $\mathrm{O}(N\log_2 N)$ arithmetic operations.

Substituting the expansions of Eq. (2.3) into Eq. (2.1), and equating the Fourier coefficients, we derive $\hat{u}_n(r,z)$ satisfying the PDE

$$\frac{\partial^2\hat{u}_n}{\partial r^2}+\frac{1}{r}\frac{\partial\hat{u}_n}{\partial r}+\frac{\partial^2\hat{u}_n}{\partial z^2}-\frac{n^2}{r^2}\hat{u}_n=\hat{f}_n(r,z), \quad 0<r\leqslant 1, \quad 0\leqslant z\leqslant 1, \tag{2.5}$$

$$\hat{u}_n(r,0)=\hat{u}_{\mathrm{B}}^n(r), \quad \hat{u}_n(r,1)=\hat{u}_{\mathrm{T}}^n(r), \quad \hat{u}_n(1,z)=\hat{u}_{\mathrm{S}}^n(z). \tag{2.6}$$

Here, the nth Fourier coefficient of the right-hand side function $\hat{f}_n(r,z)$ and the boundary values $\hat{u}_{\mathrm{S}}^n(z)$, $\hat{u}_{\mathrm{T}}^n(r)$, $\hat{u}_{\mathrm{B}}^n(r)$ are defined in a similar fashion as Eq. (2.4). In the following subsection, we shall use the notations $U(r,z)=\hat{u}_n(r,z)$ and $F(r,z)=\hat{f}_n(r,z)$, respectively.

Using the truncated Fourier series expansion, the original 3D Poisson equation now becomes a set of 2D Fourier mode equations. In fact, we only need to solve half of Fourier modes, say $n=0,1,\ldots,N/2-1$ since u is a real valued function and we have $u_{-n}(r,z)=\overline{u_n(r,z)}$. Furthermore, since those Fourier mode equations are fully decoupled, they can be solved in parallel. After we solve those Fourier mode equations and obtain the values of $\hat{u}_n(r,z)$, the solution

$u(r, z, \theta)$ can be obtained via the inverse FFT as Eq. (2.3). In [6], we have developed a second-order finite difference scheme to solve the above Fourier mode equation. In the following subsection, we will derive a formally fourth-order accurate compact scheme for Eq. (2.5).

2.2. Formally fourth-order compact difference discretization

In order to derive a fourth-order finite difference approximation to Eq. (2.5), obviously, the first and second derivatives, U_r, U_{rr} and U_{zz}, must be approximated to fourth-order accurately. To proceed, let us write down some difference formulas for the first and second derivatives with the truncation errors $\mathrm{O}(\Delta r^4)$ and $\mathrm{O}(\Delta z^4)$ as follows:

$$U_r = \delta_r^1 U - \frac{\Delta r^2}{6} U_{rrr} + \mathrm{O}(\Delta r^4), \tag{2.7}$$

$$U_{rr} = \delta_r^2 U - \frac{\Delta r^2}{12} U_{rrrr} + \mathrm{O}(\Delta r^4), \tag{2.8}$$

$$U_{zz} = \delta_z^2 U - \frac{\Delta z^2}{12} U_{zzzz} + \mathrm{O}(\Delta z^4). \tag{2.9}$$

Here δ_r^1, δ_r^2 and δ_z^2 are the centered difference operators for the first and second derivatives, defined as

$$\delta_r^1 U_{ij} = \frac{U_{i+1,j} - U_{i-1,j}}{2\,\Delta r}, \quad \delta_r^2 U_{ij} = \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta r^2}, \tag{2.10}$$

$$\delta_z^2 U_{ij} = \frac{U_{i,j+1} - 2\,U_{i,j} + U_{i,j-1}}{\Delta z^2}, \tag{2.11}$$

where U_{ij} are the discrete values defined at the grid points (r_i, z_j). As in [6], we choose a shifted grid to avoid the polar singularity as

$$r_i = (i - 1/2)\,\Delta r, \quad z_j = j\,\Delta z, \tag{2.12}$$

for $1 \leqslant i \leqslant L+1$; $0 \leqslant j \leqslant M+1$, with $\Delta r = 2/(2L+1)$ and $\Delta z = 1/(M+1)$. Note that, unlike the traditional mesh [11], we do not put the grid points on the polar axis directly, thus; no pole conditions are needed.

In order to have fourth-order approximations for U_r, U_{rr} and U_{zz}, we need to approximate the higher order partial derivatives U_{rrr}, U_{rrrr} and U_{zzzz} in Eqs. (2.7)–(2.9) to be second-order accurate. In addition, those approximations should involve at most the neighboring nine-point stencil to meet the compactness requirement. To accomplish this, we differentiate Eq. (2.5) with respect to r and z and obtain the higher order partial derivatives of U as

$$U_{rrr} = F_r - \frac{U_{rr}}{r} + \frac{1+n^2}{r^2} U_r - \frac{2n^2}{r^3} U - U_{zzr}, \tag{2.13}$$

$$U_{rrrr} = F_{rr} - \frac{F_r}{r} + \frac{3+n^2}{r^2} U_{rr} - \frac{3+5n^2}{r^3} U_r + \frac{8n^2}{r^4} U + \frac{U_{zzr}}{r} - U_{zzrr}, \tag{2.14}$$

$$U_{zzzz} = F_{zz} - U_{rrzz} - \frac{U_{rzz}}{r} + \frac{n^2}{r^2} U_{zz}. \tag{2.15}$$

Now the partial derivatives U_{rrr}, U_{rrrr} and U_{zzzz} are written in terms of lower order partial derivatives which are no higher than second-order in r and z. Using the standard centered difference approximations to those lower order partial derivatives in Eqs. (2.13)–(2.15) and substituting those approximations into Eqs. (2.7)–(2.9) and Eq. (2.5), we obtain the following difference scheme

$$\begin{aligned}
&\delta_r^2 U_{i,j} - \frac{\Delta r^2}{12}\left[\delta_r^2 F_{i,j} - \frac{1}{r_i}\delta_r^1 F_{i,j} + \frac{3+n^2}{r_i^2}\delta_r^2 U_{i,j} - \frac{3+5n^2}{r_i^3}\delta_r^1 U_{i,j}\right.\\
&\quad\left. + \frac{8n^2}{r_i^4}U_{i,j} + \frac{1}{r_i}\delta_r^1\delta_z^2 U_{i,j} - \delta_r^2\delta_z^2 U_{i,j}\right] + \frac{1}{r_i}\delta_r^1 U_{i,j}\\
&\quad - \frac{\Delta r^2}{6r_i}\left[\delta_r^1 F_{i,j} - \frac{1}{r_i}\delta_r^2 U_{i,j} + \frac{1+n^2}{r_i^2}\delta_r^1 U_{i,j} - \frac{2n^2}{r_i^3}U_{i,j} - \delta_r^1\delta_z^2 U_{i,j}\right] - \frac{n^2}{r_i^2}U_{i,j}\\
&\quad + \delta_z^2 U_{i,j} - \frac{\Delta z^2}{12}\left[\delta_z^2 F_{i,j} - \delta_z^2\delta_r^2 U_{i,j} - \frac{1}{r_i}\delta_z^2\delta_r^1 U_{i,j} + \frac{n^2}{r_i^2}\delta_z^2 U_{i,j}\right] = F_{i,j},
\end{aligned} \tag{2.16}$$

for $1 \leqslant i \leqslant L$, $1 \leqslant j \leqslant M$. Note that, the scheme involves centered difference approximations to first or second-order partial derivatives in r and z so only a nine-point compact stencil is used. The detailed nine-point stencil coefficients are summarized in the Appendix so we omit here. One can also see that if the order terms of Δr^2 and Δz^2 are neglected in Eq. (2.16), then the scheme recovers to the usual second-order accurate scheme as in [6].

In order to close the linear system, the numerical boundary values $U_{0,j}$, $U_{L+1,j}$ and $U_{i,0}$, $U_{i,M+1}$ should be supplied. The numerical boundary value $U_{0,j}$ can be given by $U_{0,j} = (-1)^n U_{1,j}$ due to the symmetry constraint of Fourier coefficients $(\hat{u}_n(-\Delta r/2, z_j) = (-1)^n \hat{u}_n(\Delta r/2, z_j))$ [5]. The other numerical boundary condition can be easily obtained by the given Dirichlet boundary values $U_{L+1,j} = \hat{u}_{\mathrm{S}}^n(z_j)$, $U_{i,0} = \hat{u}_{\mathrm{B}}^n(r_i)$ and $U_{i,M+1} = \hat{u}_{\mathrm{T}}^n(r_i)$. Throughout this paper, we denote the linear system of Eq. (2.16) as $\tilde{A_n}\tilde{U_n} = \tilde{F_n}$, where the modes to be solved are $n = 0, 1, \ldots, N/2 - 1$, as mentioned before.

3. Numerical results

In this section, we perform some numerical tests on the accuracy and efficiency of our scheme. Since the matrix of the resulting linear system in Eq. (2.16) is non-symmetric, we use the BiConjugate gradient stabilized method (Bi-CGSTAB) [12] to solve the linear systems. The stopping criterion of the convergence is based on the relative residual where the tolerance ranges from 10^{-9}–10^{-13} depending on the different Fourier modes. The reason for choosing different tolerances with different modes is because the Fourier coefficients decay rapidly as the frequency gets higher, so it is natural to choose larger tolerance when the frequency is lower and the smaller tolerance for the higher frequency modes. Note that, the relative residual here is defined as $\|r^{(k)}\|_\infty / \|\tilde{F_n}\|_\infty$, where $r^{(k)}$ is the residual after the kth iteration.

Table 1 shows the maximum relative errors for three different solutions of Poisson equation in cylindrical coordinates. In all our tests, we use L mesh points in the radial (r) direction. The mesh size $M = L$ and $N = 2L$ are used in the axial (z) and azimuthal (θ) directions, respectively. We define the relative maximum error E_L with mesh resolution L as $E_L = \|u_{\mathrm{ext}} - U\|_\infty / \|u_{\mathrm{ext}}\|_\infty$, where u_{ext} is the exact solution. The rate of convergence is then computed by the formula $\log_2(E_{L/2}/E_L)$.

If we somehow need to approximate the values at the poles ($r = 0$) based on the computed values on our grid points, we can simply use the interpolation formula given in the work of Lele [8]. That is, we employ the midpoint interpolation formula in the radial direction to obtain the values at the poles. The number in the parentheses of second column in Table 1 shows the relative maximum errors at $r = 0$. One can immediately observe that the errors at poles are comparably accurate with the global errors in the first and third examples. The second example has zero value at the poles and the interpolation almost provides the exact values as the maximum errors show smaller numbers than the machine double precision.

From Table 1, we can see that the errors of the solutions show third-order convergence for all examples in the case of solid cylinder ($0 < r \leqslant 1$). The loss of one order of accuracy seems to come from the discretization near the polar origin. This can be seen from the following truncation error analysis. In Eq. (2.5), the $U_r\,(=\partial\hat{u}_n/\partial r)$ term is divided by r. So the second-order approximation of U_{rrr} in Eq. (2.7) is divided by an $\mathrm{O}(\Delta r)$ term near the origin, which makes

Table 1
The relative maximum errors for different solutions to Poisson equation in cylindrical coordinates

L	$0 < r \leqslant 1$		$0.5 < r \leqslant 1$	
	$\lVert E_L \rVert_\infty$	Rate	$\lVert E_L \rVert_\infty$	Rate
$u(r,z,\theta) = \mathrm{e}^{r\cos\theta + r\sin\theta + z}$				
8	7.8137E − 05 (2.3219E − 06)		1.5465E − 07	
16	9.8506E − 06 (3.8096E − 07)	2.99	1.0920E − 080	3.82
32	1.2566E − 06 (3.4869E − 08)	2.97	7.3409E − 10	3.90
64	1.5941E − 07 (4.1319E − 08)	2.98	4.7580E − 11	3.95
128	2.0202E − 08 (5.3541E − 08)	2.98	3.0286E − 12	3.98
$u(r,z,\theta) = r^3(\cos\theta + \sin\theta)z(1-z)$				
8	9.1438E − 04 (1.7916E − 18)		8.8994E − 07	
16	1.0755E − 04 (7.3172E − 19)	3.09	6.4128E − 08	3.80
32	1.3008E − 05 (6.5169E − 19)	3.05	4.3173E − 09	3.89
64	1.5966E − 06 (2.1539E − 19)	3.03	2.8035E − 10	3.95
128	1.9762E − 07 (8.3404E − 19)	3.01	1.7854E − 11	3.95
$u(r,z,\theta) = \cos(\pi(r^2\cos^2\theta + r\sin\theta))\sin(\pi z^2)$				
8	7.4000E − 03 (5.8546E − 04)		7.5000E − 03	
16	3.3150E − 04 (4.4018E − 05)	4.48	1.7101E − 05	8.78
32	4.0782E − 05 (3.1563E − 06)	3.02	1.2221E − 06	3.81
64	5.0424E − 06 (2.1540E − 07)	3.02	8.1011E − 08	3.92
128	6.2440E − 07 (9.0675E − 09)	3.01	5.2187E − 09	3.96

The numbers in the parentheses are the relative maximum errors at the poles $r = 0$ which the approximated values at $r = 0$ are calculated based on the interpolation of computed values in the radial direction. Since the solution is zero at the poles for the second example, we show the maximum errors at $r = 0$ instead.

the approximation of U_{rrr}/r first-order accurate. This has the consequence that the overall truncation error of the U_r/r term in the vicinity of the origin is $\mathrm{O}(\Delta r^3)$ and thus so is Eq. (2.5). However, this loss of accuracy does not appear when solving the problem on a cylinder that excludes the polar singularity such as the case of $0 < a \leqslant r \leqslant 1$. This can be explained as follows.

As in the solid cylinder case, we need to solve Eq. (2.5) with the Dirichlet boundary condition at $r = 1$ and an additional boundary condition must be imposed at $r = a(a \geqslant \Delta r)$. Instead of setting a grid as in Eq. (2.12), we choose a regular grid in the radial direction as

$$r_i = a + \mathrm{i}\Delta r, \quad i = 0, 1, \ldots, L, L+1, \tag{2.17}$$

with the mesh width $\Delta r = (1-a)/(L+1)$. Now the second-order approximation of U_{rrr} in Eq. (2.7) is divided by an $\mathrm{O}(a + \Delta r)$ term instead of an $\mathrm{O}(\Delta r)$ term, so the truncation error of the U_{rrr}/r term is still $\mathrm{O}(\Delta r^2)$ which makes the discretization error of U_r/r term is $\mathrm{O}(\Delta r^4)$. Therefore, the overall truncation error of Eq. (2.5) is $\mathrm{O}(\Delta r^4)$. One can see in Table 1 that the fourth-order convergence indeed can be achieved for all test examples for the case of $0.5 \leqslant r \leqslant 1$.

It is also worth mentioning that the error norm in the third example decreases from $L = 8$ to $L = 16$ by a factor of 4.48 or even 8.78 which is much better than expected. We attribute this behavior to the under-resolution of Fourier expansions in the case of $L = 8$. That is, there are not enough Fourier modes to represent the solution accurately. Once we have enough Fourier modes ($L = 16$–128), the dominant discretization error comes from the radial and axial directions so the rate of convergence is exactly the same as we have expected.

In order to speed up the convergence of Bi-CGSTAB iteration, we have applied different preconditioners which include Block Jacobi (BJ) [10], Incomplete LU factorization (ILU) and the FDS arising from the second-order discretization for Eq. (2.5) [6]. Here, we solve Eq. (2.16) with the Fourier mode number $n = 1$. The tolerance for the relative residual is chosen as 10^{-9}.

Table 2 shows the number of iterations and the CPU time in seconds needed to be convergent for different preconditioners. The column of “Bi-CGSTAB” is the one without any preconditioner which, as expected, has the largest number of iterations. The preconditioners BJ and ILU both need double iterations when the grid points are doubled. One can see that, the FDS preconditioner turns out to be the most efficient one since it has the least number of iterations

Table 2
The performance comparison of Bi-CGSTAB with different preconditioners (BJ—Block Jacobi, ILU—Incomplete LU factorization, FDS—Fast direct solver) for the cylindrical case

L	Bi-CGSTAB	BJ	ILU	FDS
8	25(0.016)	16(0.031)	6(0.016)	7(0.015)
16	52(0.078)	34(0.109)	9(0.031)	7(0.062)
32	97(0.453)	65(0.844)	19(0.265)	7(0.281)
64	182(3.141)	134(6.875)	38(2.187)	7(1.703)
128	366(22.18)	246(32.95)	73(10.562)	6(5.313)

The column of “Bi-CGSTAB” means no preconditioner used. The first number represents the number of iterations while the number in parentheses represents the CPU time in seconds.

and the iterations are kept to be a constant when we double the grid points. In addition, the FDS outperforms other preconditioners significantly in terms of CPU time as the grid size gets larger.

4. Conclusion

In this paper, we present a formally fourth-order compact difference scheme for 3D Poisson equation in cylindrical coordinates. The solver relies on the truncated Fourier series expansion, where the partial differential equations of Fourier coefficients are solved by a formal fourth-order compact difference discretization without pole conditions. The resulting linear system is then solved by the Bi-CGSTAB method with different preconditioners. The numerical results confirm the formal accuracy of our scheme. Meanwhile, the preconditioner arising from the second-order fast direct solver shows the scalability of Bi-CGSTAB with the problem size.

Acknowledgements

The authors are supported in part by the National Science council of Taiwan under research grant NSC-93-2115-M-009-008.

Appendix

In this appendix, we summarize the nine-point stencil coefficients of our scheme as follows. The scheme in Eq. (2.16) can be written in the form

$$\sum_p \sum_q A_{i+p,j+q} U_{i+p,j+q} = \sum_p \sum_q B_{i+p,j+q} F_{i+p,j+q}, \quad p, q = -1, 0, 1,$$

where

$$A_{i-1,j+1} = A_{i-1,j-1} = a_0 + b_0 - \alpha_i - \beta_i,$$

$$A_{i+1,j+1} = A_{i+1,j-1} = a_0 + b_0 + \alpha_i + \beta_i,$$

$$A_{i+0,j-1} = A_{i+0,j+1} = -2a_0 + 10b_0 - n^2 c_i,$$

$$A_{i-1,j+0} = 10a_0 - 2b_0 - (1+n^2)c_i - (1+3n^2)d_i + 2\alpha_i - 10\beta_i,$$

$$A_{i+1,j+0} = 10a_0 - 2b_0 - (1+n^2)c_i + (1+3n^2)d_i - 2\alpha_i + 10\beta_i,$$

$$A_{i+0,j+0} = -20a_0 - 20b_0 + 2(1-4n^2)c_i - n_i^2\gamma_i;$$

$$B_{i-1,j+1} = B_{i+1,j+1} = B_{i-1,j-1} = B_{i+1,j-1} = 0,$$

$$B_{i+0,j+1} = B_{i+0,j-1} = 1/12,$$

$$B_{i-1,j+0} = 1/12 - \lambda_i,$$

$$B_{i+1,j+0} = 1/12 + \lambda_i,$$

$$B_{i+0,j+0} = 2/3;$$

and

$$a_0 = 1/(12\Delta r^2), \quad b_0 = 1/(12\Delta z^2), \quad c_i = 1/(12r_i^2), \quad d_i = \Delta r/(24r_i^3),$$

$$\alpha_i = \Delta r/(24r_i\Delta z^2), \quad \beta_i = 1/(24r_i\Delta r), \quad \gamma_i = \Delta r^2/(3r_i^4), \quad \lambda_i = \Delta r/(24r_i).$$

References

[1] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comp. 22 (1968) 745–762.

[2] S.R.K. Iyengar, A. Goyal, A note on multigrid for the three-dimensional Poisson equation in cylindrical coordinates, J. Comput. Appl. Math. 33 (1990) 163–169.

[3] S.R.K. Iyengar, R. Manohar, High order difference methods for heat equation in polar cylindrical coordinates, J. Comput. Phys. 77 (1988) 425.

[4] M.K. Jain, R.K. Jain, M. Krishna, A fourth-order difference scheme for quasilinear Poisson equation in polar co-ordinates, Commun. Numerical Methods Eng. 10 (1994) 791–797.

[5] M.-C. Lai, A simple compact fourth-order Poisson solver on polar geometry, J. Comput. Phys. 182 (2002) 337–345.

[6] M.-C. Lai, W.-W. Lin, W. Wang, A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries, IMA J. Numer. Anal. 22 (2002) 537–548.

[7] M.-C. Lai, W.-C. Wang, Fast direct solvers for Poisson equation on 2D polar and spherical geometries, Numer. Meth. Partial Diff. Eq. 18 (2002) 56–68.

[8] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.

[9] R.C. Mittal, S. Gahlaut, High-order finite-difference schemes to solve Poisson's equation in polar coordinates, IMA J. Numer. Anal. 11 (1991) 261.

[10] B. Richard, et al., Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, PA, 1994.

[11] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Page 286–287, Wadsworth & Brooks/Cole, Belmant, CA, 1989.

[12] H.A. Van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 13 (1992) 631–644.