International Journal of Pure and Applied Mathematics
Volume 7 No. 3 2003, 263-270

A NUMERICAL TECHNIQUE FOR THE 3-D POISSON EQUATION

Salwa K. Abd-El-Hafiz[1], Gamal A.F. Ismail[2], Berlant S. Matit[3 §]

[1]Department of Engineering Mathematics
Faculty of Engineering
Cairo University
Giza, EGYPT
e-mail: salwahafiz@link.net

[2,3]Department of Applied Mathematics
Faculty for Girls
Ain Shams University
Asma Fahmy St. Heliopolis
Cairo, EGYPT
[3]e-mail: berlantm@operamail.com

Abstract: In this paper,we solve the three-dimensional Poisson equation with Dirichlet boundary conditions. The Poisson equation is, first, discretized using the finite difference method. We choose an indexing strategy for the grid points, which is essential for the efficient solution of Poisson equation at all grid points. Then, an iterative method is used to solve the resulting linear system. Finally, we present an example that demonstrates the accuracy of the method.

AMS Subject Classification: 65L12
Key Words: Poisson equation, Dirichlet boundary conditions, discretization, finite difference, successive over relaxation

Received: March 3, 2003

§Correspondence author

1. Introduction

The solution of Pisson equation is required in many physical problems [6, 5]. For example, Poisson equation occurs during the solution of Euler equations for gas dynamics [11] and the solution of the incompressible Navier-Stockes equations [3]. In this paper, we present an accurate method for the solution of the three-dimensional (3-D)Poisson equation

$$\bigtriangledown^2 \psi = f, \tag{1}$$

where f specify a forcing function and ψ is an analytic function. In two-dimensional (2-D) space, there exists a direct analytical method for solving equation (1) in a rectangular domain [4, 12, 14]. In this paper, we explain how to solve this equation in a 3-D cuboidal domain. Because Dirichlet boundary conditions are used, the problem has a unique solution [3, 8]. To solve Poisson equation,we use finite-difference approximations to replace the derivatives [10, 11] and produce a system of linear algebraic equations of the form

$$AW = B, \tag{2}$$

where A is an $N \times N$ matrix, B is a known $N \times 1$ matrix, and W is an unknown $N \times 1$ matrix. N is the total number of mesh points in the domain. Iterative methods that are discussed in [2, 9] are ideal for solving the resulting sparse matrix. More specifically, we use the successive over relaxation method (SOR) for the solution of the resulting system. Finally, we introduce an example and compare its exact solution with the numerical results of our algorithm. The efficiency and accuracy of our algorithm facilitate its application on 3-D problems that cannot be solved analytically.

2. Discretization and Finite Difference Formulation

The Dirichlet boundary value problem for Poisson equation is given by

$$\bigtriangledown^2 \psi(x,y,z) = f(x,y,z), \quad (x,y,z) \in R\,, \tag{3}$$

$$\psi(x,y,z) = G(x,y,z), \qquad (x,y,z) \in \partial R\,, \tag{4}$$

where $R = \{(x,y,z)|a_1 < x < a_2,\ b_1 < y < b_2,\ c_1 < z < c_2\}$, ∂R is the boundary of R, and f, G are continuous known functions on R. ψ is an analytic function.

To solve Poisson equation, we subdivide the domain with equally spaced lines parallel to x-, y- and z-axes. For $\Delta x = h$, $\Delta y = d$, $\Delta z = s$, where $h = \frac{a_2-a_1}{N_x}$, $d = \frac{b_2-b_1}{N_y}$,$s = \frac{c_2-c_1}{N_z}$ and N_x, N_y, N_z are integers, we define

$$\begin{aligned} \psi(x_i, y_j, z_k) &= \psi_{i,j,k} , \\ f(x_i, y_j, z_k) &= f_{i,j,k} , \end{aligned} \tag{5}$$

where $x_i = a_1 + ih$ for $i = 0, 1, ...N_x$, $y_j = b_1 + jd$ for $j = 0, 1, ...N_y$, and $z_k = c_1 + ks$ for $k = 0, 1, ...N_z$.

At the mesh points (or grid points) inside R, we approximate equation (3) at each point (x_i, y_j, z_k) by

$$\begin{aligned} & \nabla^2\psi(x, y, z) \\ = & \frac{\psi(x_{i+1}, y_j, z_k) - 2\psi(x_i, y_j, z_k) + \psi(x_{i-1}, y_j, z_k)}{h^2} \\ + & \frac{\psi(x_i, y_{j+1}, z_k) - 2\psi(x_i, y_j, z_k) + \psi(x_i, y_{j-1}, z_k)}{d^2} \\ + & \frac{\psi(x_i, y_j, z_{k+1}) - 2\psi(x_i, y_j, z_k) + \psi(x_i, y_j, z_{k-1})}{s^2} \\ - & \frac{h^2}{12}\frac{\partial^4\psi(\xi_i, y_j, z_k)}{\partial x^4} - \frac{d^2}{12}\frac{\partial^4\psi(x_i, \eta_j, z_k)}{\partial x^4} - \frac{s^2}{12}\frac{\partial^4\psi(x_i, y_j, \theta_k)}{\partial z^4} \\ = & f(x_i, y_j, z_k) , \end{aligned} \tag{6}$$

with $\xi_i \in [x_{i-1}, x_{i+1}]$, $\eta_j \in [y_{j-1}, y_{j+1}]$ and $\theta_k \in [z_{k-1}, z_{k+1}]$.

By deleting the error terms involving the fourth derivatives of ψ, we get a new set of equations for approximating unknowns $\psi_B \approx \psi : \{\psi_B(x_i, y_j, z_k)|(x_i, y_j, z_k) \in R\}$. The values of ψ_B at boundary mesh points on ∂R are given by

$$\psi_B(x_i, y_j, z_k) = G(x_i, y_j, z_k), \quad (x_i, y_j, z_k) \in \partial R. \tag{7}$$

Hence, the resulting linear system is

$$\begin{aligned} \nabla^2\psi_B(x, y, z) &= \frac{\psi_B(x_{i+1}, y_j, z_k) - 2\psi_B(x_i, y_j, z_k) + \psi_B(x_{i-1}, y_j, z_k)}{h^2} \\ &+ \frac{\psi_B(x_i, y_{j+1}, z_k) - 2\psi_B(x_i, y_j, z_k) + \psi_B(x_i, y_{j-1}, z_k)}{d^2} \\ &+ \frac{\psi_B(x_i, y_j, z_{k+1}) - 2\psi_B(x_i, y_j, z_k) + \psi_B(x_i, y_j, z_{k-1})}{s^2} \\ &= f(x_i, y_j, z_k). \end{aligned} \tag{8}$$

If $\psi(x,y,z)$ is four times continuously differentiable over $R\bigcup\partial R$, it can be shown that for some c,

$$\max_{R\bigcup\partial R}|\psi(x_i,y_j,z_k)-\psi_B(x_i,y_j,z_k)|\le c\ \max\{h^2,d^2,s^2\}. \tag{9}$$

For interior mesh points, we can simplify equation (8) to

$$\begin{aligned}2\psi_B(x_i,y_j,z_k)[d^2s^2+h^2d^2+h^2s^2]-d^2s^2[\psi_B(x_{i+1},y_j,z_k)\\+\psi_B(x_{i-1},y_j,z_k)]-h^2s^2[\psi_B(x_i,y_{j+1},z_k)+\psi_B(x_i,y_{j-1},z_k)]\\-h^2d^2[\psi_B(x_i,y_j,z_{k+1})+\psi_B(x_i,y_j,z_{k-1})]=-h^2d^2s^2f(x_i,y_j,z_k).\end{aligned} \tag{10}$$

For simplicity , we will also drop the subscript B.

Thus, we have a system of $(N_x-1)\times(N_y-1)\times(N_z-1)$ linear equations and $(N_x+1)\times(N_y+1)\times(N_z+1)$ grid points. The boundary conditions (4) give the values of ψ at 2 $(1+N_xN_y+N_xN_z+N_zN_y)$ points. We then have a system of $(N_x-1)\times(N_y-1)\times(N_z-1)$ equations in the same number of unknowns.

3. Indexing Strategy and Definition of the Linear System of Equations

We denote the interior mesh point (x_i,y_j,z_k) by P_Γ, where Γ is defend as:

$$\begin{aligned}\Gamma=i+(N_y-1-j)(N_x-1)+(N_z-1-k)(N_x-1)(N_y-1)\\ \text{for}\ \ i=1,...,N_x-1,\ j=1,...,N_y-1\ \text{ and }\ k=1,...,N_z-1.\end{aligned} \tag{11}$$

For example, consider $N_x=4$, $N_y=5$ and $N_z=3$. In this case, we have $(4-1)\times(5-1)\times(5-1)=24$ interior mesh points, which are given as follows:

$$P_1=(x_1,y_4,z_2),\quad P_2=(x_2,y_4,z_2),\quad P_3=(x_3,y_4,z_2),\quad P_4=(x_1,y_3,z_2)$$

$$P_5=(x_2,y_3,z_2),...,P_{22}=(x_1,y_1,z_1),\quad P_{23}=(x_2,y_1,z_1)$$

and $P_{24}=(x_3,y_1,z_1)$.

Expressing equation (8) in terms of the relabeled grid points $\psi_i=\psi(P_i)$ implies the following equations:

For P_1:

$$\begin{aligned}&2\psi_1[d^2s^2+h^2d^2+h^2s^2]-d^2s^2\psi_2-h^2s^2\psi_4-d^2h^2\psi_{13}\\&=d^2s^2\psi_{0,4,2}+h^2s^2\psi_{1,5,2}+d^2h^2\psi_{1,4,3}-d^2s^2h^2f_1.\end{aligned}$$

For P_2:

$$2\psi_2[d^2s^2 + h^2d^2 + h^2s^2] - d^2s^2\psi_3 - d^2s^2\psi_1 - s^2h^2\psi_5 - d^2h^2\psi_{14}$$
$$= h^2s^2\psi_{2,5,2} + h^2d^2\psi_{2,4,3} - d^2s^2h^2f_2.$$

For P_3:

$$2\psi_3[d^2s^2 + h^2d^2 + h^2s^2] - d^2s^2\psi_2 - h^2s^2\psi_6 - d^2h^2\psi_{15}$$
$$= d^2s^2\psi_{4,4,2} + h^2s^2\psi_{3,5,2} + d^2h^2\psi_{3,4,3} - d^2s^2h^2f_3.$$

Similar equations are written for the rest of the grid points, where the right hand sides of the equations are obtained from the boundary conditions. Thus, Poisson equation in 3-D gives us a system of equations that is very sparse (i.e., most of coefficients in each equation are zero). So, it is better to use the successive over relaxation (SOR) iteration method, which converges fairly quickly. It has a run time of $O(N^{\frac{3}{2}})$, where N is the number of grid points. This run time is comparable with the most efficient direct matrix inversion algorithms.

Now, a simple algorithm, which defines the resulting linear system of equations, can be written. The algorithm evaluates the elements of the matrix A and the vector B used in the linear system of equations $AW = B$, where $W = [\psi_1, \psi_2, ..., \psi_{(N_x-1)\times(N_y-1)\times(N_z-1)}]$. The evaluation is performed by relabling the interior mesh points used in equation (10) according to the indexing strategy of equation (11). Points on the boundary ∂R update the vector B. Points in the domain R update the coefficients matrix, A.

4. Numerical Results

Example. In order to test the algorithm, consider the following Poisson equation.

$$\bigtriangledown^2 \psi(x,y,z) = -3\Pi^2 \sin(\Pi x)\sin(\Pi y)\sin(\Pi z),$$
$$0 < x < 1, \quad 0 < y < 1, \quad 0 < z < 1,$$

with $\psi = 0$ on the boundary. The exact solution [13] is

$$\psi(x,y,z) = \sin(\Pi x)\sin(\Pi y)\sin(\Pi z).$$

To approximate the exact solution, we use our technique with $N = N_x = N_y = N_z = 2, 4, 8$ and 10. The approximate solution in the case of $N = 10$ at

$z = 0.5$ is shown in Figure 1. The error in our solution has a minimum absolute value of 8.27×10^{-39}, which is obtained near the boundary. The maximum absolute value of the error is 2.94×10^{-3}, which is reached near the middle of the domain R.

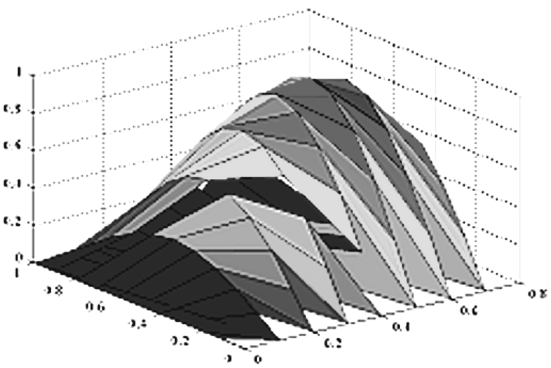

Figure 1: Approximate solution at z =0.5

We examine the error norm e at each value of N using the following equation

$$e = \left(\frac{\sum_{ijk} e_{ijk}^2}{m}\right)^{1/2},$$

where, m is the total number of grid points and $e_{ijk} = \psi_{ijk}$ (exact solution) - ψ_{ijk} (approximate solution). The results of computing the error norm at the different values of N are presented in Figure 2. As expected, the convergence is faster as $N \to \infty$.

References

[1] S. Abarbanel, A. Kumar, Compact Higher-Order Schemes for the Euler Equation, *Journal for Scientific Cmputing*, **3** (1988), 275-288.

[2] L.M. Adams, R.J. Leveque, D.M. Young, Analysis of SOR iteration for the 9-point Laplacian, *SIAM Journal of Numerical Analysis*, **25** (1988), 1156-1180.

[3] A. Brandt, Multi-Level adaptive solution to boundary value problems, *Math. of Computing*, **25** (1977), 333-390.

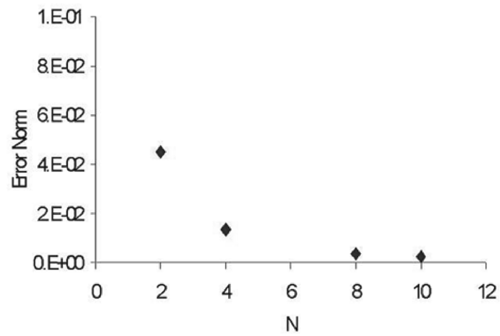

Figure 2: Convergence plot for the example

[4] R.L. Burden, J.D. Faires, *Numerical Analysis*, Prindle, Weber and Schmidt, Boston (1993).

[5] E.H. Chao, R.C. Davidson, S.F. Paul, Direct numerical solution of poisson equation in cylindrical coordinates, *Journal of Vacuum Scienceand Technology*, **17** (1997), 2050-2065.

[6] A.J. Garcia-Loureiro, T.F. Pena, J.M. Lopez-Gomzalez, L. Prat, Parallel domain decomposition applied to 3-D Poission equation for gradual HBT, *International Journal for Numerical Methods in Engineering*, **4** (2000), 639-652.

[7] C.J. Ho, F.H. Lin, Numerical simulation of three-dimensional incompressible flow by a new formulation, *International Journal of Numerical Methods in Fluids*, **17** (1996), 261-270.

[8] H. Jolansen, P. Collella, A cartesian grid embedded boundary method for Poisson equation on irregular domains, *Journal of Computing Physics*, **147** (1998), 60-85.

[9] Z. Li, A fast iterative algorithm for elliptic interface problems, *SIAM Journal of Numerical Analysis*, **35** (1998), 230-254.

[10] R.E. Lynch, J.R. Rice, High accuracy finite difference approximation to solutions of elliptic partial differential equations, In: *Proceedings of the National Academy of Science*, USA, **75** (1978), 2541-2544.

[11] W.F. Spotz, *High-Order Compact Finite Difference Schemes for Computational Mechanics*, PhD Thesis, University of Texas at Austin, TX (1995).

[12] T. Simchony, R. Chellappa, M. Shao, Direct analyticl methods for solving Poisson equations in computer vision problems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12** (1990), 435-446.

[13] K. Vipin, G. Ananth, G. Anshul, K. Gerorge, *Introduction to Parallel Computing Algorithm Design and Analysis*, Benjamin, Cummings, Redwood City (1994).

[14] F.A. William, *Numerical Methods for Partial Differential Equations*, Academic Press, New York (1977).