Journal of Electromagnetic Analysis and Applications, 2014, 6, 303-308
Published Online September 2014 in SciRes. http://www.scirp.org/journal/jemaa
http://dx.doi.org/10.4236/jemaa.2014.610030

The Exact Formulation of the Inverse of the Tridiagonal Matrix for Solving the 1D Poisson Equation with the Finite Difference Method

Serigne Bira Gueye

Département de Physique, Faculté des Sciences et Techniques, Université Cheikh Anta Diop, Dakar-Fann, Sénégal
Email: sbiragy@gmail.com

Received 21 June 2014; revised 15 July 2014; accepted 8 August 2014

Abstract

A new method for solving the 1D Poisson equation is presented using the finite difference method. This method is based on the exact formulation of the inverse of the tridiagonal matrix associated with the Laplacian. This is the first time that the inverse of this remarkable matrix is determined directly and exactly. Thus, solving 1D Poisson equation becomes very accurate and extremely fast. This method is a very important tool for physics and engineering where the Poisson equation appears very often in the description of certain phenomena.

Keywords

1D Poisson Equation, Finite Difference Method, Tridiagonal Matrix Inversion, Thomas Algorithm, Gaussian Elimination, Potential Problem

1. Introduction

The finite difference method is a very useful tool for discretizing and solving numerically a differential equation. It is effectively a classical method of approximation based on Taylor series expansions that has help during the last years theoretical results to gain in accuracy, stability and convergence.

In fact, this method is very useful for solving for example Poisson equation. This elliptic equation appears very often in mathematics, physics, chemistry, biology and engineering. In one dimension, the resolution leads to a tridiagonal matrix in the case of centered difference approximation. This matrix, which is diagonally domi-

How to cite this paper: Gueye, S.B. (2014) The Exact Formulation of the Inverse of the Tridiagonal Matrix for Solving the 1D Poisson Equation with the Finite Difference Method. *Journal of Electromagnetic Analysis and Applications*, **6**, 303-308. http://dx.doi.org/10.4236/jemaa.2014.610030

nant, can be inverted with methods such as Gauss elimination, Thomas Algorithm Method [1]. These technics are powerful and very efficient.

We proposed here, a new and direct method of inversion of this tridiagonal matrix independently of the right-hand side. For Dirichlet-Dirichlet boundary problems, this innovative method is faster than the Thomas Algorithm. It gives better accuracy and is far more economical in terms of memory occupation.

First, the finite difference method is presented for the 1D Poisson equation. Secondly, the properties of the matrix associated with the Laplacian and its inverse are discussed. Then, the inverse matrix is determined and its properties are analyzed. Thus, verification is done considering an interesting potential problem, and the sensibility of the method is quantified.

2. Finite Difference Method and 1D Poisson Equation

We consider a function $\Phi(x)$ which satisfies the Poisson equation $\Delta\Phi(x) = f(x)$, in the interval $]a,b[$, where f is a specified function. $\Phi(x)$ fulfills the Dirichlet-Dirichlet boundary conditions $\Phi(a) = \Phi_a$ and $\Phi(b) = \Phi_b$. We consider an one-dimensional mesh with $N+2$ discrete points (x_i). Each point (x_i) is defined by $x_i = a + i\cdot\Delta x$, where $\Delta x = \frac{(b-a)}{N+1} = h$ being the step size. We define $\Phi_i \approx \Phi(x_i)$, $f_i = f(x_i)$, $i = 0,1,\cdots,N+1$.

We have chosen the centered difference approximation $\left(O\left(\Delta x^2\right)\right)$, in this work, for the fact that it gives a tridiagonal, diagonally dominant, and symmetric matrix. Considering all the above mentioned criteria, one can rewrite the 1D Poisson equation in a set of algebraic equations:

$$\Phi_{i-1} - 2\Phi_i + \Phi_{i+1} = h^2 f_i, \quad i = 1,2,\cdots,N. \tag{1}$$

One gets a linear system of N equations, which can be written in a matrix form [2]

$$\underbrace{\begin{pmatrix} -2 & 1 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & -2 & 1 & \ddots & \cdots & 0 \\ 0 & 0 & 0 & 1 & -2 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{pmatrix}}_{:=\boldsymbol{A}} \times \underbrace{\begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \\ \Phi_5 \\ \vdots \\ \Phi_{N-1} \\ \Phi_N \end{pmatrix}}_{:=\boldsymbol{\Phi}} = \underbrace{\begin{pmatrix} h^2 f_1 - \Phi_a \\ h^2 f_2 \\ h^2 f_3 \\ h^2 f_4 \\ h^2 f_5 \\ \vdots \\ h^2 f_{N-1} \\ h^2 f_N - \Phi_b \end{pmatrix}}_{:=\boldsymbol{F}} \tag{2}$$

Thus, solving the 1D Poisson equation means to invert the negative definite, and regular $N \times N$-matrix $\boldsymbol{A} = (a_{ij})$. Its inverse, that we noted $\boldsymbol{B} = (b_{ij})$, is also symmetric. Both matrices have the following properties:

$$a_{ij} = \begin{cases} -2, & i = j \\ 1, & |i-j| = 1 \\ 0, & |i-j| > 1 \end{cases} \tag{3}$$

and

$$\begin{cases} -2b_{i1} + b_{i2} = \delta_i^1 \\ b_{ij-1} - 2b_{ij} + b_{ij+1} = \delta_i^j, \quad 1 < j < N, \\ b_{iN-1} - 2b_{iN} = \delta_i^N \end{cases} \tag{4}$$

where δ_i^j is the Kronecker's delta.

3. The Inverse of Matrix A

From (4), we derive successively the following interesting relations:

$$b_{ij+1} = b_{ij} + b_{i1} \quad \text{and} \quad b_{ij} = j b_{i1} + (j-1) \tag{5}$$

with (5), one sees that the matrix $\boldsymbol{B}$ is entirely determined if the term b_{11} is known. This term can be determined by observing the behavior of $\boldsymbol{B}$ for different N values: It holds

$$b_{11} = \frac{-N}{N+1} \tag{6}$$

From (5) and (6), we get

$$\begin{cases} b_{1j} = -\dfrac{N-(j-1)}{N+1} \\ b_{i1} = -\dfrac{N-(i-1)}{N+1} \end{cases} \tag{7}$$

Now, the matrix $\boldsymbol{B}$ is completely and exactly determined. $\boldsymbol{B} = (b_{ij});\ i, j = 1, 2, \cdots, N$ with

$$b_{ij} = \begin{cases} -j\dfrac{[N-(i-1)]}{N+1}, & i \ge j \\ -i\dfrac{[N-(j-1)]}{N+1}, & i < j \end{cases}; \tag{8}$$

$\boldsymbol{B} =$

$$\frac{-1}{N+1}\begin{pmatrix} N & (N-1) & (N-2) & \cdots & N-(j-1) & \cdots & 3 & 2 & 1 \\ (N-1) & 2(N-1) & 2(N-2) & \cdots & 2[N-(j-1)] & \cdots & 6 & 4 & 2 \\ (N-2) & 2(N-2) & 3(N-2) & \cdots & 3[N-(j-1)] & \cdots & 9 & 6 & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [N-(i-1)] & 2[N-(i-1)] & 3[N-(i-1)] & \cdots & i[N-(i-1)] & \cdots & 3i & 2i & i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 3 & 6 & 9 & \cdots & 3j & \cdots & 3(N-2) & 2(N-2) & (N-2) \\ 2 & 4 & 6 & \cdots & 2j & \cdots & 2(N-2) & 2(N-1) & (N-1) \\ 1 & 2 & 3 & \cdots & j & \cdots & (N-2) & (N-1) & N \end{pmatrix},$$

The solution of the 1D Poisson equation is obtained with a simple, extremely fast matrix multiplication: $\boldsymbol{\Phi} = \boldsymbol{BF}$. Thus, the numerical resolution of the 1D Poisson equation which is an interesting topic in physics and engineering is made easy and very accurate.

Analysis

A first analysis of the matrix $(\boldsymbol{B})$ let us believe that, this new method possesses an algorithm complexity of $(O(N^2))$, which is situated between the Gauss eliminations $(O(N^3))$ and the one of Thomas's $(O(N))$ [1].

A deeper analysis of the matrix $(\boldsymbol{B})$ shows that the complexity brought by the Thomas method is largely improved in this study. In addition, one can see a close link between its row vectors and column vectors.

The matrix $(\boldsymbol{B})$ is also persymmetric:

$$b_{ij} = b_{N-j+1, N-i+1}$$

All the information about it, can be found in the upper triangle (in gray color, see **Figure 1**).

Further, we can even find very interesting relations in this matrix which can help refining the final solution.

That is what we effectively did, and one can see a direct solution for Φ_N at the point x_N, which can be expressed by

$$\Phi_N = -h^3 \sum_{i=1}^{N} i \cdot f_i \tag{9}$$

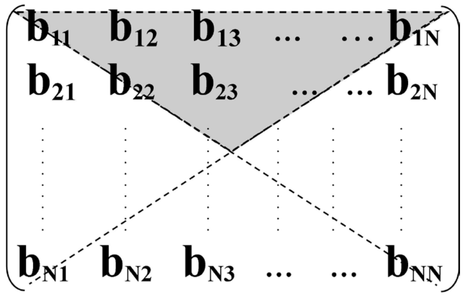

Figure 1. Matrix symmetries.

Also a direct solution for Φ_{N-1} at the point x_{N-1} is:

$$\Phi_{N-1} = -h^3\left[\left[\sum_{i=1}^{N-1} 2 \cdot i \cdot f_i\right] + (N-1) \cdot f_N\right] \tag{10}$$

Generally, a very important recurrence relation can be obtained, which gives all solutions:

$$\Phi_{N-k} = -h^3\left[(k+1)\cdot\left[\sum_{i=1}^{N-k} i \cdot f_i\right] + (N-k)\cdot\left[\sum_{i=N-k+1}^{N}(N-(i-1))\cdot f_i\right]\right], \quad k = 0, 1, \cdots, N-1 \tag{11}$$

which is equivalent to:

$$\Phi_k = -h^3\left[(N-k+1)\cdot\left[\sum_{i=1}^{k} i \cdot f_i\right] + k\cdot\left[\sum_{i=k+1}^{N}(N-(i-1))\cdot f_i\right]\right], \quad k = 1, 2, \cdots, N \tag{12}$$

This very innovative Equation (12) gives directly and accurately all the solution that we are looking for. It proves that our method is direct, faster than the one of Thomas's in this context and gives as well better accuracy. Furthermore, it is far more economical in terms of memory occupation. This is due to the fact that the matrix $(\boldsymbol{B})$ does not necessitate to be generated. A programmer does not need to declare nor to define the matrix $(\boldsymbol{B})$ in his code.

In conclusion to this, we can say that the matrix $(\boldsymbol{B})$ is the key of this efficient new method. This matrix $(\boldsymbol{B})$, which is the inverse of matrix $(\boldsymbol{A})$, is determined explicitly, directly, and independently of the right-hand side of the Poisson equation.

N.B.: One can prove using mathematical induction that $\det(\boldsymbol{A}) = (-1)^N (N+1)$. It holds for the (i, j) co-factor of $\boldsymbol{A}$: $CofA_{ij} = (-1)^{N+1} j \cdot [N-(i-1)]$, $i \geq j$. We call the matrix $\boldsymbol{B}$ **Bira's Matrix**.

4. Verification with a Potential Problem

We consider a scalar potential $\Phi(x)$, defined in [0, 1], which satisfies $\Delta\Phi(x) = \frac{\partial^2\Phi(x)}{\partial x^2} = f(x) = -\cos^2(\pi(x-1/2))$. $\Phi(x)$ fulfills the following boundary conditions: $\Phi(0) = \Phi(1) = 0$. The exact solution is

$$\Phi_{\text{exact}}(x) = -\frac{x^2}{4} + \left[\frac{\cos\left(\pi\left(x-\frac{1}{2}\right)\right)}{2\pi}\right]^2 + \frac{x}{4} \tag{13}$$

With the finite difference method, we take $N = 100$, $\Delta x = h = \frac{1}{N+1}$, $x_i = i \cdot \Delta x$, $\Phi_i \approx \Phi(x_i)$, and $f_i = f(x_i) = -\cos^2\left(\pi\left(x_i - \frac{1}{2}\right)\right)$. The solution is

$$\begin{pmatrix}\Phi_1\\\Phi_2\\\Phi_3\\\Phi_4\\\Phi_5\\\vdots\\\Phi_{99}\\\Phi_{100}\end{pmatrix}=\frac{-1}{101}\begin{pmatrix}100&99&98&\cdots&\cdots&3&2&1\\99&198&196&\cdots&\cdots&6&4&2\\\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots&\vdots\\\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots&\vdots\\3&6&9&\cdots&\cdots&294&196&98\\2&4&6&\cdots&\cdots&196&198&99\\1&2&3&\cdots&\cdots&98&99&100\end{pmatrix}\times\begin{pmatrix}h^2f_1\\h^2f_2\\h^2f_3\\h^2f_4\\h^2f_5\\\vdots\\h^2f_{99}\\h^2f_{100}\end{pmatrix}, \tag{14}$$

Discussions

We define the variable $\varepsilon_i(100)$, which is the relative error at point x_i for $(N=100)$. $\Phi_{i\text{FDM}}$ represents Φ_i.

Generally, we have

$$\varepsilon_i(N)=\left|\frac{\Phi_{i\text{FDM}}-\Phi_{i\text{exact}}}{\Phi_{i\text{exact}}}\right| \tag{15}$$

We can also define the average value of the relative error for a given N: $\bar{\varepsilon}(N)$. For $N=100$, it is: $\bar{\varepsilon}(100)\approx 6.160\times10^{-5}$.

We obtain the following results, presented in **Table 1**.

The table shows that the solution is very accurate. Notwithstanding that we have been interested in determining the sensibility of the proposed method. Effectively, we have plotted $\bar{\varepsilon}(N)$ for different N values. We obtain a hyperbola, which can be predicted as proportional to $h^2=1/(N+1)^2=\Delta x^2$.

This curve is fitted with a function which can be defined as

$$\text{Trunc}(N)=\alpha\cdot h^2=\frac{\alpha}{(N+1)^2}, \tag{16}$$

where $\alpha\approx 0.62598$. We obtain two curves represented in **Figure 2**.

Table 1. Results and relative error.

i	x_i	$\Phi_{i\text{FDM}}$	$\Phi_{i\text{exact}}$	$\varepsilon_i(100)$
1	$9.90099009900990E-003$	$2.47532050462650E-003$	$2.47531263320003E-003$	$3.1799726459536950E-0006$
2	$1.98019801980198E-002$	$4.95054591793065E-003$	$4.95051434685110E-003$	$6.3773332093484231E-0006$
3	$2.97029702970297E-002$	$7.42539189910138E-003$	$7.42532094619246E-003$	$9.5555342911444034E-0006$
4	$3.96039603960396E-002$	$9.89938595762298E-003$	$9.89926009306004E-003$	$1.2714542476634171E-0005$
5	$4.95049504950495E-002$	$1.23718692811413E-002$	$1.23716731875297E-002$	$1.5850209475913139E-0005$
6	$5.94059405940594E-002$	$1.48419992841306E-002$	$1.48417179157622E-002$	$1.8957938019221409E-0005$
7	$6.93069306930693E-002$	$1.73087528675008E-002$	$1.73083715085596E-002$	$2.2033207515521614E-0005$
8	$7.92079207920792E-002$	$1.97709303765360E-002$	$1.97704346980271E-002$	$2.5071705121898528E-0005$
9	$8.91089108910891E-002$	$2.22271602418517E-002$	$2.22265363570344E-002$	$2.8069367498980424E-0005$
10	$9.90099009900990E-002$	$2.46759042854172E-002$	$2.46751388035270E-002$	$3.1022394496623757E-0005$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
94	$9.30693069306931E-001$	$1.73087528675008E-002$	$1.73083715085596E-002$	$2.2033207516323428E-0005$
95	$9.40594059405941E-001$	$1.48419992841306E-002$	$1.48417179157622E-002$	$1.8957938020390243E-0005$
96	$9.50495049504951E-001$	$1.23718692811413E-002$	$1.23716731875297E-002$	$1.5850209474510952E-0005$
97	$9.60396039603961E-001$	$9.89938595762299E-003$	$9.89926009306005E-003$	$1.2714542475933209E-0005$
98	$9.70297029702970E-001$	$7.42539189910138E-003$	$7.42532094619247E-003$	$9.5555342903267151E-0006$
99	$9.80198019801980E-001$	$4.95054591793065E-003$	$4.95051434685111E-003$	$6.3773332088227983E-0006$
100	$9.90099009900990E-001$	$2.47532050462650E-003$	$2.47531263320003E-003$	$3.1799726454280849E-0006$

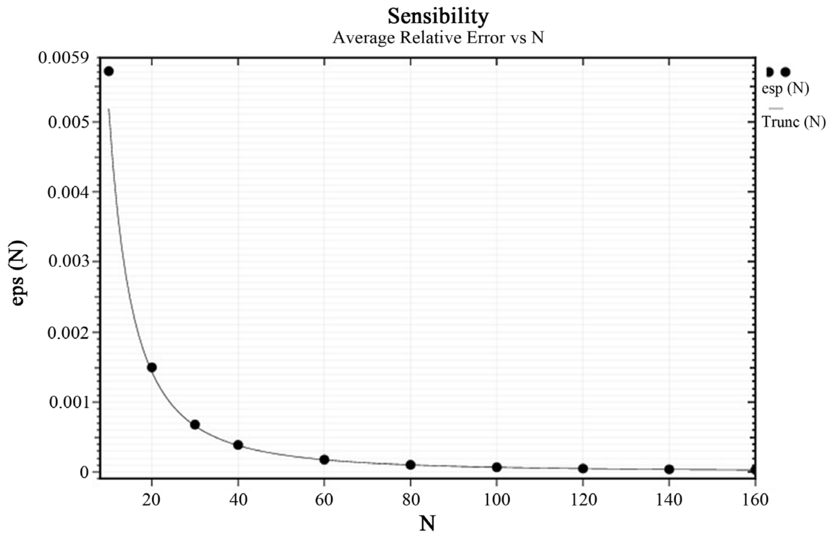

Figure 2. Sensibility.

We realize that the average relative error $\overline{\varepsilon}(N)$ behaves like a truncation error that we express in the following manner $\left|\dfrac{h^2\Phi_{\text{exact}}^{(4)}(c)}{12}\right|$. $\Phi_{\text{exact}}^{(4)}(c)$ is the fourth order derivative of the Φ_{exact} function in a point (here C) which belongs to the interval $[a,b]$.

For our given function Φ_{exact} and also the results from the fitting, we have the following relations [3]:

$$\overline{\varepsilon}(N) \approx \alpha \cdot h^2 = \frac{\alpha}{(N+1)^2} < \frac{h^2 4\pi^2}{12}, \tag{17}$$

This proves that the method is very accurate, naturally stable, robust, quick and precise.

5. Conclusions

This paper has provided a new improved method for solving the 1D Poisson equation with the finite difference method. Accurate results have been obtained with a sensibility found to be as the function of $1/(N+1)^2$. In fact, the inverse of the tridiagonal matrix, which is associated with this differential equation, is determined directly, exactly, and independently to the right-hand side. Thus, a new formulation of the solution is given with an algorithmic complexity of $O(N)$. With this innovative method, the 1D Poisson equation, with Dirichlet-Dirichlet boundary condition is solved, with only one programming loop. This new approach provides also gain in accuracy and economy in memory allocation.

A future work can consider Neumann or mixed boundary conditions.

Acknowledgements

I would like to thank my colleagues Dr. Cheikh Mbow and Dr. Kharouna Talla for benefit discussions and remarks that contribute to improving the quality of this paper.

References

[1] Conte, S.D. and de Boor, C. (1981) Elementary Numerical Analysis: An Algorithmic Approach. 3rd Edition, McGraw-Hill, New York, 153-157.

[2] Leveque, R.J.E. (2007) Finite Difference Method for Ordinary and Partial Differential Equations, Steady State and Time Dependent Problems. *SIAM*, 15-16. http://dx.doi.org/10.1137/1.9780898717839

[3] Mathews, J.H. and Kurtis, K.F. (2004) Numerical Methods Using Matlab. 4th Edition, Prentice Hall, Upper Saddle River, 323-325, 339-342.