*Journal of Mathematical Sciences and Applications, 2014, Vol. 2, No. 3, 39-42*
Available online at http://pubs.sciepub.com/jmsa/2/3/3

DOI:10.12691/jmsa-2-3-3

# New Approach for Numerical Solution of Poisson's Equation by Cubic Spline

**M. Yousefi[1,*], J. Rashidinia[1], M. Yousefi[2], N.S BahrololoumiMofrad[2], Mehdi Moudi[3]**

[1]School of Mathematics, Iran University of Science & Technology, Narmak, Tehran 16844-13114, Iran
[2]Energy and Mechanical Engineering Department, Abbaspour- power and water-College of Engineering, ShahidBeheshti University, P.O. Box.16765-1719, Tehran, Iran
[3]Institute of Applied Materials - Reliability of Components and Systems (IAM-ZBS) Engelbert-Arnold-Straße Karlsruhe, Germany
*Corresponding author: mah_yousefi@yahoo.com

*Received December 23, 2014; Revised December 29, 2014; Accepted December 31, 2014*

**Abstract** We consider the solution of various boundary value problems for Poisson's equation in the unit square using a nodal cubic spline collocation method and modifications of it which produce optimal fourth order approximations. Uniform partition of the square with cost $O(N^2 \log N)$ using a direct fast Fourier transform method. The numerical results exhibit super convergence phenomena.

***Keywords:*** *Nodal collocation, Poisson's equation, cubic spline, convergence analysis*

**Cite This Article:** M. Yousefi, J. Rashidinia, M. Yousefi, N.S BahrololoumiMofrad, and Mehdi Moudi, "New Approach for Numerical Solution of Poisson's Equation by Cubic Spline." *Journal of Mathematical Sciences and Applications*, vol. 2, no. 3 (2014): 39-42. doi: 10.12691/jmsa-2-3-3.

## 1. Introduction

The first spline collocation method proposed for the solution of second-order two-point boundary value problems was a nodal cubic spline collocation(NCSC) method [1]. In the basic nodal cubic spline collocation (NCSC) method, an approximate solutionis sought in the space of bicubic splines and is determined by satisfying the differential equation and the boundary conditions at the partition nodes of $\Omega$ and $\partial\Omega$, respectively.This method is well known to be suboptimal; in fact, it is no better than second order, whereas fourth order is expected since the approximate solution is piecewise bicubic [2]. In [3], two optimal order NCSC methods, a two step method (TSM) and a one step method (OSM), were presented for the solution of the Dirichlet BVP. These methods are based on optimal NCSC methods for solving second-order two-point BVPs.De Boor [4] proved that classical nodal cubic spline collocation for solving two-point boundary value problems. Archer [5] and, independly, Daniel and Swartz [6] developed a modified nodal cubic spline collocation (MNCSC) scheme which is fourth order accurate. Houstis,Vavalis, and Rice [7]. Derived a fourth order MNCSC scheme for solving elliptic boundary value problems on rectangles. In [5], for the Helmholtz equation, matrix decomposition algorithms (MDA) with fast Fourier transformswere formulated and implemented to solve the two step method(TSM) collocation equations for each the boundary value problem (BVP), but, for the one step method, it was possible to formulate an MDA only for the Dirichlet BVP.

In this paper, we consider Poisson's equation with a constant coefficient in the unit square subject to the Dirichlet boundary value condition. We develop a new algorithm for the solution of the linear systems arising in the NCSC method for the Dirichlet problem. Our scheme involves perturbations of both the left- and right-hand sides. Numerical results show that our scheme exhibits super convergence phenomena while that [7] does not.

A brief outline of this paper is as follows. We give preliminaries in section 2. In section 3, the matrix vector of our sheme, and a direct fast Fourier transforms algorithm for solving the scheme are presented Section 4.) Includes numerical results obtained using our scheme.

## 2. Cubic Spline Function Properties and Cubic Spline Interpolation

We consider the Dirichlet boundary value problemfor Poisson's equation

$$\Delta u = f \quad in \quad \Omega, \qquad u = 0 \quad on\,\partial\Omega \tag{2.1}$$

Where $\Delta \equiv D_x^2 + D_y^2$ is the Laplacian, and $\Omega = (0,1)\times(0,1)$, and $\partial\Omega$ is the boundary of $\Omega$.

Let $\rho_x = \{x_i\}_{i=0}^{N+1}$ with $x_i = ih, h = 1/(N+1)$ be a uniform partition of $[0,1]$ in the x-direction(For simplicity, in the remainder of the paper, a uniform partition

$\rho_y = \{y_i\}_{i=0}^{N+1}$ of $[0,1]$ in the y-direction is such that $y_j = x_j$.)

Let $S_3$ denote the space of cubic splines

$$S_3 = \{v : v \in C^2([0,1]), v|_I \in P_3, 1 \le i \le N+1\}$$

Where $I_i = [x_{i-1}, x_i]$ and $P_3$ is the set of polynomials of degree ≤ 3 and let

$$S^D = \{v \in S_3 : v(0) = v(1) = 0\}$$

To introduce basis functions for $S_3$, we extend the partition $\rho_x$ using $xi = ih,\ i = -3,-2,-1, N+2, N+3, N+4$. As a basis for $S_3$, we choose the functions $\{\beta_m\}_{m=-1}^{N+2}$, where

$$\beta_m = \begin{cases} h^{-3} g_1(x - x_{m-2}), & x \in I_{m-1} \\ g_2((x - x_{m-1})/h), & x \in I_m \\ g_2((x_{m+1} - x)/h), & x \in I_{m+1} \\ h^{-3} g_1(x_{m+2} - x), & x \in I_{m+2} \\ 0 & otherwise \end{cases} \quad (2.2)$$

And

$$g_1(x) = x^3, \quad g_2(x) = 1 + 3x + 3x^2 - 3x^3 \quad (2.3)$$

These basis functions are such that, for $m = 0, \ldots, N+1$

$$\begin{cases} \beta_{m-1}(x_m) = 1, \beta_m(x_m) = 4, \beta_{m+1}(x_m) = 1, \\ \beta''_{m-1}(x_m) = 6/h^2, \beta''_m(x_m) = -12/h^2, \beta''_{m+1}(x_m) = 6/h^2. \end{cases} \quad (2.4)$$

As a basis for $S^D$, we choose the cubic splines $\{\beta_m^D\}_{m=0}^{N+1}$ defined in terms of splines by

$$\begin{cases} \beta_0^D = \beta_0 - 4\beta_{-1}, \beta_1^D = \beta_1 - \beta_{-1}, \beta_m^D = \beta_m, \ m = 2, 3, \ldots, N-1 \\ \beta_N^D = \beta_N - \beta_{N+2} \quad \beta_{N+1}^D = \beta_{N+1} - \beta_{N+2} \end{cases} \quad (2.5)$$

Cf.[8,Section 2].

It follows from (2.3) that

$$\begin{cases} B_0^D(x_1) = 1, & B_1^D(x_1) = 4, & B_1^D(x_2) = 1, \\ B_N^D(x_{N-1}) = 1, & B_N^D(x_N) = 4, & B_{N+1}^D(x_N) = 1, \end{cases} \quad (2.6)$$

$$\begin{cases} [B_0^D]''(x_0) = -\dfrac{36}{h^2}, \quad [B_1^D]''(x_0) = 0, \\ [B_0^D]''(x_1) = \dfrac{6}{h^2}, \quad [B_1^D]''(x_1) = -\dfrac{12}{h^2}, \quad [B_1^D]''(x_2) = \dfrac{6}{h^2} \\ [B_N^D]''(x_{N-1}) = \dfrac{6}{h^2}, \quad [B_N^D]''(x_N) = -\dfrac{12}{h^2}, \quad [B_{N+1}^D]''(x_N) = \dfrac{6}{h^2} \\ [B_N^D]''(x_{N+1}) = \dfrac{6}{h^2}, [B_{N+1}^D]''(x_{N+1}) = -\dfrac{36}{h^2}. \end{cases} \quad (2.7)$$

It also follows from (2.5),(2.6),(2.4), and (2.3) that, for i=1,…,N,

$$B_m^D(x_i) - \frac{h^2}{6}[B_m^D]''(x_i) = \begin{cases} 6, m = i \\ 0, m \ne i \end{cases} \quad m = 0, \ldots, N+1 \quad (2.8)$$

And the matrix-vector form of

$$\phi_{i,j} = \sum_{m=1}^{N} c_{i,m} \sum_{n=1}^{N} c_{j,n} \psi_{m,n}, \quad i, j = 1, \ldots, N \quad (2.9)$$

Is

$$\phi = (C_1 \otimes C_2)\psi \quad (2.10)$$

Where $C_1 = (c_{i,m}^{(1)})_{i \in I, m \in M}, C_1 = (c_{j,n}^{(2)})_{j \in J, n \in N}$. And

$$\phi = [\phi_{1,1}, \ldots, \phi_{1,J'}, \ldots, \phi_{I',1}, \ldots, \phi_{I',J'}]^T,$$

$$\psi = [\psi_{1,1}, \ldots, \psi_{1,N'}, \ldots, \psi_{M',1}, \ldots, \psi_{M',N'}]^T$$

# 3. The Approximate Solution of Poisson's Equation

In this section, we consider equation (2.1). we introduce

$$S^D = S_3 \cap \{v : v(0) = v(1) = 0\}$$

Our NCSC scheme for solving (2.1) is formulated as follows: seek $u_h^D \in S^D \otimes S^D$ such that

$$\begin{aligned} &\Delta u_h(x_i, y_j) - \frac{h^2}{36} D_x^2 D_y^2 u_h(x_i, y_j) \\ &= f(x_i, y_j) - \frac{h^2}{144}\Delta f(x_i, y_j) \\ &i, j = 0, \ldots, N+1 \\ &u = 0 \quad on \ \partial\Omega \end{aligned} \quad (3.1)$$

(3.1) is equivalent to

$$2D_x^2 D_y^2 u_h(x_i, y_j) = \Delta f(x_i, y_j), \quad i, j = 0, N+1 \quad (3.2)$$

$$\begin{aligned} &D_x^2 u_h(x_i, y_j) - \frac{h^2}{36} D_x^2 D_y^2 u_h(x_i, y_j) \\ &= f(x_i, y_j) - \frac{h^2}{144}\Delta f(x_i, y_j) \quad i = 0, N+1, j = 1, \ldots, N \end{aligned} \quad (3.3)$$

$$\begin{aligned} &D_y^2 u_h(x_i, y_j) - \frac{h^2}{36} D_x^2 D_y^2 u_h(x_i, y_j) \\ &= f(x_i, y_j) - \frac{h^2}{144}\Delta f(x_i, y_j) \\ &i = 1, \ldots N, j = 0, N+1 \end{aligned} \quad (3.4)$$

$$\begin{aligned} &\Delta u_h(x_i, y_j) - \frac{h^2}{36} D_x^2 D_y^2 u_h(x_i, y_j) \\ &= f(x_i, y_j) - \frac{h^2}{144}\Delta f(x_i, y_j) \quad i, j = 1, \ldots, N \end{aligned} \quad (3.5)$$

The scheme (3.1)is motivated by the fourth order finite difference method for (2.1).

For $i = 0, N+1, 1 \le j \le N$ and $1 \le i \le N, 0 \le j \le N+1$, and

$$D_y^l D_x^2 u_h^D(x_i, y_j) = D_y^l f(x_i, y_j), \qquad i, j = 0, N+1 \quad (3.6)$$

Where l in (3.6)is eather 1 or 2.
Since

$$u_h^D(x,y) = \sum_{m=0}^{N+1}\sum_{n=0}^{N+1} u_{m,n}\beta_m^D(x)\beta_n^D(y) \qquad (3.7)$$

And involve $(N+2)^2$ equations in the unknown coefficients $\{u_{m,n}\}_{m,n=0}^{N+1}$

It follows from (2.4) and (2.5) that, for $0 \le m \le N+1$

$$2\sum_{m=0}^{N+1}\sum_{n=0}^{N+1} u_{m,n}[B_m^D]''(x_i)[B_n^D]''(y_j) = \Delta f(x_i, x_j) \quad (3.8)$$

Substituting (3.7) into (3.2), we obtain

$$\sum_{m=0}^{N+1}\sum_{n=0}^{N+1} u_{m,n}\left( \begin{array}{l} [B_m^D]''(x_i)B_n^D(y_j) \\ + \left[ B_m^D(x_i) - \frac{h^2}{36}[B_n^D]''(x_j) \right]\left[ B_n^D \right]''(y_j) \end{array} \right) \quad (3.9)$$

$$= f(x_i, y_j) - \frac{h^2}{144}\Delta f(x_i, y_j) \quad i, j = 1, ..., N$$

Substituting (3.7) into, we obtain (3.3)

$$\sum_{m=1}^{N}\sum_{n=1}^{N} u_{m,n}\left( \begin{array}{l} [B_m^D]''(x_i)B_n^D(y_j) \\ + \left[ B_m^D(x_i) - \frac{h^2}{36}[B_n^D]''(x_j) \right]\left[ B_n^D \right]''(y_j) \end{array} \right) \quad (3.10)$$

$$= \omega_{i,j} \qquad i, j = 1, ..., N$$

Where

$$\omega_{i,j} = f(x_i, y_j) - \frac{h^2}{144}\Delta f(x_i, y_j)$$
$$- \sum_{m=0,N+1}\sum_{n=0}^{N+1} u_{m,n}\left( \begin{array}{l} [B_m^D]''(x_i)B_n^D(y_j) \\ + \left[ B_m^D(x_i) - \frac{h^2}{36}[B_n^D]''(x_j) \right]\left[ B_n^D \right]''(y_j) \end{array} \right)$$
$$- \sum_{m=1}^{N}\sum_{n=0,N+1} u_{m,n}\left( \begin{array}{l} [B_m^D]''(x_i)B_n^D(y_j) \\ + \left[ B_m^D(x_i) - \frac{h^2}{36}[B_n^D]''(x_j) \right]\left[ B_n^D \right]''(y_j) \end{array} \right)$$

Each of the methods gives rise to a linear systemof the form

$$A_1 \otimes B_2 + (B_1 \otimes A_2)u = f$$

Where the matrices $A_1$, $B_1$ are $M \times M$ with $M = N, N+1, or \ \ N+2$, where the matrices $A_2$, $B_2$ are $(N+2)\times(N+2)$, and where $\otimes$ denotes the matrix tensor product (cf.,[8]), thus we can (3.10) as

$$\left[ A \otimes B + \left( B - \frac{h^2}{36}A \right) \otimes A \right] u = v \qquad (3.11)$$

Where $u = [u_{1,1}, ..., u_{N,N}]^T$, $\lambda = [\lambda_{1,1}, ..., \lambda_{N,N}]^T$ and for $\{B_m^D\}_{m=0}^{N+1}$, we introduce $N \times N$ matrices $A$ and $B$ defined by

$$A = (a_{i,m})_{i,m=1}^N, a_{i,m} = \left[ B_m^D \right]_{j,n=1}^N,$$
$$B = (b_{j,n})_{j,n=1}^N b_{j,n} = B_n^D(y_j) \qquad (3.12)$$

It follows from (2.4)-(2.7) that

$$A = 6h^{-2}P, \qquad B = P + 6I \qquad (3.13)$$

Where $I$ is the identity matrix and the $N \times N$ matrix $P$ is given by

$$P = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \qquad (3.14)$$

And (3.11) simplifies to

$$\frac{36}{h^2}[36P \otimes I + (36I + P) \otimes P]u = v \qquad (3.15)$$

We have

$$K^{-1}PK = \Lambda \qquad (3.16)$$

Where the $N \times N$ matrices $\Lambda$ and $Q$ are given by

$$\Lambda = diag(\lambda_i)_{i=1}^N, \qquad \lambda_i = -4\sin^2\frac{i\pi}{2(N+1)} \qquad (3.17)$$

$$K = (k_{i,j})_{i,j=1}^N, \qquad k_{i,j} = \left( \frac{2}{N+1} \right)^{1/2} \sin\frac{ij\pi}{N+1} \qquad (3.18)$$

Cf., [1, section 2].
Using (3.18), we see (3.15) is equivalent to

$$36h^{-2}(K \otimes I)[36P \otimes I + (36I + P) \otimes P]$$
$$(K \otimes I)(K^{-1} \otimes I)u = (K \otimes I)v \qquad (3.19)$$

If $u' = (K^{-1} \otimes I)u$, $v' = (K \otimes I)v$ and using (3.16) and (3.19) we obtain

$$36h^{-2}[36\Lambda \otimes I + (36I + \Lambda) \otimes T]u' = v' \qquad (3.20)$$

The system (3.20) reduces to the N independent systems

$$36h^{-2}[36\lambda_i \otimes I + (36I + \lambda_i) \otimes T]u_i' = v_i' \qquad (3.21)$$

We thus have the following algorithm for solving (3.15) :

Step 1. Compute $v' = (K \otimes I)v$
Step 2. Solve the N systems in (3.21)
Step 3. Compute $u = (K \otimes I)u'$

It follows from (3.18) that the cost of steps 1 and 3 is $O(N^2 \log N)$ each (cf.,[8]).In step 2, the systems are tridiagonal, so this step is performed at a cost $O(N^2)$. Consequently, the total cost of the algorithm is $O(N^2 \log N)$.

## 4. Numerical Experiment

In our numerical study, we used the following testproblem. Thecomputations were carried out in double precision. We determined the nodal and global errors using the formulas. test problem from [7] were considered.

We determined the nodal and global errors using the formulas

$$\|u\|_h = \max_{0\le i,j\le N+1} |u(x_i, y_j)|, \|u\|_{C(\bar\Omega)} \approx \max_{0\le i,j\le 1001} |u(x_i, y_j)|$$

Where

$$t_i = \frac{i}{1001}, \quad i = 1,\ldots,1001.$$

Convergence rates were determines using the formula

$$rate = \frac{\log(e_{N/2} / e_N)}{\log[(N+1)/(N/2+1)]}$$

Where $e_N$ is the error corresponding to partition $\rho_x \times \rho_y$.

*Problem D–1: Poisson's equation with exact solution*

$$u(x, y) = 3e^{xy}(x^2 - x)(y^2 - y)$$

**Table 1. Nodal errors and convergence rates for** $u, u_x, u_y$

| | $\| u-u_h \|_h$ | | $\| (u-u_h)_x \|_h$ | | $\| (u-u_h)_y \|_h$ | | $\| (u-u_h)_{xy} \|_h$ | |
|---|---|---|---|---|---|---|---|---|
| *N* | *Error* | *Rate* | *Error* | *Rate* | *Error* | *Rate* | *Error* | *Rate* |
| 8 | | | | | | | | |
| 16 | 6.014−06 | | 8.234−05 | | 7.008−05 | | 1.897−03 | |
| 32 | 4.999−07 | 4.087 | 7.001−06 | 4.002 | 6.015−06 | 4.027 | 1.569−04 | 3.987 |
| 64 | 3.326−09 | 4.036 | 4.879−07 | 4.051 | 5.298−07 | 4.002 | 1.231−05 | 4.001 |
| 128 | 1.007−10 | 4.001 | 3.451−08 | 4.034 | 3.876−08 | 4.001 | 1.204−06 | 3.861 |
| 256 | 0.246−11 | 3.986 | 2.002−09 | 4.000 | 2.068−09 | 3.997 | 1.001−07 | 4.007 |

**Table 2. Global errors and convergence rates for** $u, u_x, u_y$ **and** $u_{xy}$

| | $\| u-u_h \|_{C(\bar\Omega)}$ | | $\| (u-u_h)_x \|_{C(\bar\Omega)}$ | | $\| (u-u_h)_y \|_{C(\bar\Omega)}$ | | $\| (u-u_h)_{xy} \|_{C(\bar\Omega)}$ | |
|---|---|---|---|---|---|---|---|---|
| *N* | *Error* | *Rate* | *Error* | *Rate* | *Error* | *Rate* | *Error* | *Rate* |
| 8 | | | | | | | | |
| 16 | 7.799−06 | | 0.987−04 | | 0.999−04 | | 1.985−03 | |
| 32 | 6.013−07 | 3.769 | 1.082−05 | 3.994 | 0.854−05 | 3.945 | 2.001−04 | 3.938 |
| 64 | 5.239−08 | 3.989 | 1.120−06 | 4.002 | 0.832−06 | 4.008 | 2.087−05 | 3.874 |
| 128 | 4.001−09 | 4.001 | 1.148−07 | 4.081 | 0.901−07 | 3.756 | 2.732−06 | 3.646 |
| 256 | 2.017−10 | 4.003 | 1.278−08 | 3.896 | 0.965−08 | 4.001 | 2.865−07 | 4.000 |

## 5. Conclusion

We see from the results in Table 1 and Table 2 that Scheme produces fourth order accuracy for u in both the discrete and the continuous maximum norms. We also observe super convergence phenomena since the derivative approximation at the partition nodes are of order four.

## References


[1] B. Bialecki, G. Fairweather, and A. Karageorghis, *Matrix decomposition algorithms for modified spline collocation for Helmholtz problems*, SIAM J. Sci. Comput., 24 (2003), pp. 1733-1753.

[2] B. Bialecki, G. Fairweather, and A. Karageorghis, *Optimal superconvergent one step nodal cubic spline collocation methods*, SIAM J. Sci. Comput., 27 (2005), pp. 575-598.

[3] E. N. Houstis, E. A. Vavalis, and J. R. Rice, *Convergence of* O(h[4]) *cubic spline collocation methods for elliptic partial differential equations*, SIAM J. Numer. Anal., 25 (1988), pp. 54-74.

[4] C. de Boor, *The Method of Projections as Applied to the Numerical Solution of Two Point Boundary Value Problems Using Cubic Splines*, Ph.D. thesis, University of Michigan, Ann Arbor, MI, 1966.

[5] D. Archer, *An* O(h[4]) *cubic spline collocation method for quasilinear parabolic equations*, SIAM J. Number. Anal., 14 (1977), pp. 620-637.

[6] J. W. Daniel and B. K. Swartz, *Extrapolated collocation for two-point boundary-value problems using cubic splines*, J. Inst. Math Appl., 16 (1975), pp. 161-174.

[7] E. N. Houstis, E. A. Vavalis, and J. R. Rice, *Convergence of* O(h[4]) *cubic spline collocation methods for elliptic partial differential equations*, SIAM J. Numer. Anal., 25 (1988), pp. 54-74.

[8] B. Bialecki and G. Fairweather, *Matrix decomposition methods for separable elliptic boundary value problems in two dimensions*, J.Comput. Appl. Math., 46 (1993), pp. 369-386.