An efficient scheme for solution of two-dimensional Laplace's equation

Ariba Arshad[1], R.C. Mittal[2], Seema Sharma[3]

[1]Department of Mathematics and Statistics, Gurukula Kangri Vishwavidyalaya, Haridwar, Uttarakhand

[2]Department of Mathematics, Jaypee Institute of Technology, Noida, Uttar Pradesh

[3]Department of Mathematics and Statistics, Gurukula Kangri Vishwavidyalaya, Haridwar, Uttarakhand

* Corresponding author's email: [1] aribadav1992@gmail.com, [2] rcmmmfma@iitr.ac.in, [3] dikshitseema@yahoo.com

doi:https://doi.org/10.21467/proceedings.100.3

ABSTRACT

The main objective of this paper is to analyse the solution of two-dimensional Laplace's equation using an efficient scheme, which is based on collocation of modified bi-cubic B-spline functions. This scheme is applied to obtain the approximate solution of Laplace's equation with Dirichlet boundary conditions for two illustrative examples. For these examples, the results obtained by this scheme have been compared with exact solutions and solutions available in literature to check the accuracy and versatility of present scheme.

Keywords: Modified bi-cubic B-splines, Laplace's equation, Hockney method.

1 Introduction

Consider the well known two-dimensional Laplace's equation defined on a rectangular domain Ω=[a,b]×[c,d] given by

$$u_{xx} + u_{yy} = 0, \tag{1}$$

with Dirichlet boundary conditions,

$$\left.\begin{array}{lll} u(a,y) = f_0(y), & u(b,y) = f_1(y) & c \le y \le d \\ u(x,c) = f_2(x), & u(x,d) = f_3(x) & a \le x \le b \end{array}\right\} \tag{2}$$

Laplace's equation plays a very important role in science fields like fluid dynamics, heat transfer, gravitation, electromechanics, magnetism and many others [1]. Laplace's equation is the steady state heat equation in the study of heat conduction. It satisfies the velocity potential for the steady

Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-942709-6-6 (eBook)

flow of incompressible non-viscous fluid. Laplace's equation arises in static deflection of a membrane. Laplace's equation is used in determining structure of astronomical object from spectrum. Laplace's equation is associated with equilibrium or steady state problems such as steady state temperature distribution, steady state stress distribution, steady state potential distribution and steady state flows.

In the last few decades, a huge amount of work has been done to analyse the solutions of two-dimensional Laplace's equation using various numerical techniques. Khaled [2] developed sinc and adomian decomposition method to investigate Laplace's equation. Shabbir et al. [3] employed a Galerkin technique to solve two-dimensional Laplace's equation. Hamid et al. [4] analysed Laplace's equation by using bi-cubic B-spline interpolation method. Patil and Prasad [5, 6] obtained the numerical solution of Laplace's equation using grid less techniques, finite difference method, finite element method and Markov chain method. Buralieva et al. [7] presented wavelet-Galerkin method to solve two-dimensional Laplace's equation. A recursive form of bi-cubic B-spline collocation method for Laplace's equation has been employed by Reddy et al. [8]. Chopade and Rastogi [9] employed finite difference method and finite element method for solution of two-dimensional Laplace's equation.

In this paper, a scheme based on modified bi-cubic B-spline functions has been employed to obtain the numerical solution of two-dimensional Laplace's equation with Dirichlet boundary conditions. To demonstrate the efficiency and accuracy of the present scheme, convergence and comparison studies have been accomplished for two test problems.

The brief outline of the paper is as follows: In section 2, modified cubic B-spline functions and modified bi-cubic B-spline functions are explained. The description of scheme to solve two-dimensional Laplace's equation with Dirichlet boundary conditions is also given in section 2. Section 3 consists of two examples to verify the efficiency and accuracy of the proposed scheme. The conclusions are given in section 4 to summarize the findings of this work.

2 Description of Method

Spline is a tool which was originally used by draftsman to draw a smooth curve passing through the specified points in a plane. A B-spline is a spline function that has minimal support with respect to a given degree, smoothness and domain partition.

Divide the rectangular domain Ω=[a,b]$\times$[c,d] into a mesh of rectangular finite elements designated by set of mesh points $\{(x_i, y_j)$: i=0(1)n, j=0(1)m$\}$. The intervals (x_{i-1}, x_i), i=1(1)n taken along X-axis are of same length h_x and intervals (y_{j-1}, y_j), j=1(1)m taken along Y-axis are of same length h_y.

Let the approximate solution $U_{nm}(x,y)$ of Laplace's equation (1) be [10]

$$U_{nm}(x,y)=\sum_{i=0}^{n}\sum_{j=0}^{m}w_{i,j}\phi_{i,j}(x,y). \tag{3}$$

where, $w_{i,j}$ are to be determined. The modified bi-cubic B-spline function $\phi_{i,j}(x,y)$ is given as

$$\phi_{i,j}(x,y)=\phi_i(x)\phi_j(y) \tag{4}$$

where, modified cubic B-spline basis functions $\phi_i(x)$ are given as

$$\left.\begin{aligned}
&\phi_o(x)=\psi_o(x)+2\psi_{-1}(x)\\
&\phi_1(x)=\psi_1(x)-\psi_{-1}(x)\\
&\phi_j(x)=\psi_j(x), for \quad j=2(1)(n-2)\\
&\phi_{n-1}(x)=\psi_{n-1}(x)-\psi_{n+1}(x)\\
&\phi_n(x)=\psi_n(x)+2\psi_{n+1}(x)
\end{aligned}\right\} \tag{5}$$

where, ψ_{-1},ψ_o,ψ_1,...,ψ_{n-1},ψ_n,ψ_{n+1} are cubic B-spline basis functions over the interval $[a,b]$ defined as follows

$$\psi_i(x)=\frac{1}{h_x^3}\left\{\begin{array}{cc}
(x-x_{i-2})^3, & x\in[x_{i-2},x_{i-1})\\
(x-x_{i-2})^3-4(x-x_{i-1})^3, & x\in[x_{i-1},x_i)\\
(x_{i+2}-x)^3-4(x_{i+1}-x)^3, & x\in[x_i,x_{i+1})\\
(x_{i+2}-x)^3, & x\in[x_{i+1},x_{i+2})\\
0, & otherwise
\end{array}\right\}. \tag{6}$$

At a particular knot x_i, there exist only three cubic B-splines ψ_{i-1}, ψ_i, ψ_{i+1} with positive values. The values of $\phi_i(x)$, $\phi_i'(x)$ and $\phi_i''(x)$ along x-direction at different knots have been tabulated, respectively, in Tables 1- 3. Similarly, along y-direction, the values $\phi_j(y)$, $\phi_j'(y)$ and $\phi_j''(y)$ can be obtained by replacing i with j, x with y and h_x with h_y. Each modified bi-cubic B-spline $\phi_{i,j}(x,y)$ covers sixteen elements of domain Ω and each finite element of domain Ω is covered by sixteen modified bi-cubic B-splines with their peaks being at the knot (x_i,y_j) of the domain.

Table 1: Values of modified cubic B-spline functions along x-direction

x	$\phi_o(x)$	$\phi_1(x)$	$\phi_2(x)$		$\cdots$		$\phi_{n-2}(x)$	$\phi_{n-1}(x)$	$\phi_n(x)$
x_o	6	0							
x_1	1	4	1						
x_2		1	4	1					
$\cdots$			$\cdots$	$\cdots$	$\cdots$				
x_{n-2}						1	4	1	
x_{n-1}							1	4	1
x_n								0	6

Table 2: Values of first derivatives of modified cubic B-spline functions along x-direction

x	$\phi'_o(x)$	$\phi'_1(x)$	$\phi'_2(x)$		$\cdots$		$\phi'_{n-2}(x)$	$\phi'_{n-1}(x)$	$\phi'_n(x)$
x_o	$-\frac{6}{h_x}$	$\frac{6}{h_x}$	0						
x_1	$-\frac{3}{h_x}$	0	$\frac{3}{h_x}$						
x_2		$-\frac{3}{h_x}$	0	$\frac{3}{h_x}$					
$\cdots$			$\cdots$	$\cdots$	$\cdots$				
x_{n-2}						$-\frac{3}{h_x}$	0	$\frac{3}{h_x}$	
x_{n-1}							$-\frac{3}{h_x}$	0	$\frac{3}{h_x}$
x_n							0	$-\frac{6}{h_x}$	$\frac{6}{h_x}$

Table 3: Values of second derivatives of modified cubic B-spline functions along x-direction

x	$\phi''_o(x)$	$\phi''_1(x)$	$\phi''_2(x)$		$\cdots$		$\phi''_{n-2}(x)$	$\phi''_{n-1}(x)$	$\phi''_n(x)$
x_o	0	0	0						
x_1	$\frac{6}{h_x^2}$	$-\frac{12}{h_x^2}$	$\frac{6}{h_x^2}$						
x_2		$\frac{6}{h_x^2}$	$-\frac{12}{h_x^2}$	$\frac{6}{h_x^2}$					
$\cdots$			$\cdots$	$\cdots$	$\cdots$				
x_{n-2}						$\frac{6}{h_x^2}$	$-\frac{12}{h_x^2}$	$\frac{6}{h_x^2}$	
x_{n-1}							$\frac{6}{h_x^2}$	$-\frac{12}{h_x^2}$	$\frac{6}{h_x^2}$
x_n							0	0	0

The values of $w_{i,j}$ on the boundary of the domain Ω can be obtained by using approximate solution (3) in the boundary conditions (2).

The boundary condition $u(x,c) = f_2(x)$ gives

$$6\begin{bmatrix} 6 & 0 & & & & & & & \\ 1 & 4 & 1 & & & & & & \\ & 1 & 4 & 1 & & & & & \\ & & & \cdots & \cdots & \cdots & & & \\ & & & \cdots & \cdots & \cdots & \cdots & & \\ & & & & & 1 & 4 & 1 & \\ & & & & & & 1 & 4 & 1 \\ & & & & & & & 0 & 6 \end{bmatrix}_{(n+1)\times(n+1)} \begin{bmatrix} w_{0,0} \\ w_{1,0} \\ w_{2,0} \\ \cdots \\ \cdots \\ w_{n-2,0} \\ w_{n-1,0} \\ w_{n,0} \end{bmatrix}_{(n+1)\times 1} = \begin{bmatrix} f_2(x_0) \\ f_2(x_1) \\ f_2(x_2) \\ \cdots \\ \cdots \\ f_2(x_{n-2}) \\ f_2(x_{n-1}) \\ f_2(x_n) \end{bmatrix}_{(n+1)\times 1} . \quad (7)$$

The boundary condition $u(x,d) = f_3(x)$ yields

$$6\begin{bmatrix} 6 & 0 & & & & & & \\ 1 & 4 & 1 & & & & & \\ & 1 & 4 & 1 & & & & \\ & & \cdots & \cdots & \cdots & & & \\ & & & \cdots & \cdots & \cdots & & \\ & & & & 1 & 4 & 1 & \\ & & & & & 1 & 4 & 1 \\ & & & & & & 0 & 6 \end{bmatrix}_{(n+1)\times(n+1)} \begin{bmatrix} w_{0,m} \\ w_{1,m} \\ w_{2,m} \\ \cdots \\ \cdots \\ w_{n-2,m} \\ w_{n-1,m} \\ w_{n,m} \end{bmatrix}_{(n+1)\times 1} = \begin{bmatrix} f_3(x_0) \\ f_3(x_1) \\ f_3(x_2) \\ \cdots \\ \cdots \\ f_3(x_{n-2}) \\ f_3(x_{n-1}) \\ f_3(x_n) \end{bmatrix}_{(n+1)\times 1}. \quad (8)$$

The boundary condition $u(a,y) = f_0(x)$ leads to

$$6\begin{bmatrix} 4 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & \cdots & \cdots & \cdots & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 4 \end{bmatrix}_{(m-1)\times(m-1)} \begin{bmatrix} w_{0,1} \\ w_{0,2} \\ w_{0,3} \\ \cdots \\ \cdots \\ w_{0,m-3} \\ w_{0,m-2} \\ w_{0,m-1} \end{bmatrix}_{(m-1)\times 1} = \begin{bmatrix} f_0(y_1) - 6w_{0,0} \\ f_0(y_2) \\ f_0(y_3) \\ \cdots \\ \cdots \\ f_0(y_{m-3}) \\ f_0(y_{m-2}) \\ f_0(y_{m-1}) - 6w_{0,m} \end{bmatrix}_{(m-1)\times 1}. \quad (9)$$

The boundary condition $u(b,y) = f_1(x)$ gives

$$6\begin{bmatrix} 4 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & \cdots & \cdots & \cdots & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 4 \end{bmatrix}_{(m-1)\times(m-1)} \begin{bmatrix} w_{n,1} \\ w_{n,2} \\ w_{n,3} \\ \cdots \\ \cdots \\ w_{n,m-3} \\ w_{n,m-2} \\ w_{n,m-1} \end{bmatrix}_{(m-1)\times 1} = \begin{bmatrix} f_1(y_1) - 6w_{n,0} \\ f_1(y_2) \\ f_1(y_3) \\ \cdots \\ \cdots \\ f_1(y_{m-3}) \\ f_1(y_{m-2}) \\ f_1(y_{m-1}) - 6w_{n,m} \end{bmatrix}_{(m-1)\times 1}. \quad (10)$$

The tridiagonal system of equations (7-10) can be solved by Thomas algorithm to obtain the values of $w_{i,j}$ on the boundary of domain Ω. Now, satisfying the equation (1) on the internal mesh points of domain Ω, a system of (n-1)×(m-1) equations is obtained, which is as follows:

$$\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j}\phi_i''(x_k)\phi_j(y_{k'}) + \sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j}\phi_i(x_k)\phi_j''(y_{k'}) = 0, \qquad k = 1(1)(n-1), k' = 1(1)(m-1). \tag{11}$$

The system of equations (11) can be expressed in matrix form as follows:

$$(\phi'' \otimes \phi + \phi \otimes \phi'') \begin{bmatrix} \overrightarrow{\delta_1} \\ \overrightarrow{\delta_2} \\ \cdots \\ \cdots \\ \overrightarrow{\delta}_{m-2} \\ \overrightarrow{\delta}_{m-1} \end{bmatrix}_{(m-1)\times 1} = \begin{bmatrix} \overrightarrow{\gamma_1} \\ \overrightarrow{\gamma_2} \\ \cdots \\ \cdots \\ \overrightarrow{\gamma}_{m-2} \\ \overrightarrow{\gamma}_{m-1} \end{bmatrix}_{(m-1)\times 1} \tag{12}$$

where, $\otimes$ is the Kronecker product and

$$\phi''\otimes\phi = \frac{1}{h_x^2}\begin{bmatrix} -12A & 6A & & & & & & \\ 6A & -12A & 6A & & & & & \\ & & & \cdots & \cdots & \cdots & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & 6A & -12A & 6A \\ & & & & & & 6A & -12A \end{bmatrix}_{(m-1)\times(m-1)},$$

$$A = \begin{bmatrix} 4 & 1 & & & & & & \\ 1 & 4 & 1 & & & & & \\ & & & \cdots & \cdots & \cdots & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & 1 & 4 & 1 \\ & & & & & & 1 & 4 \end{bmatrix}_{(n-1)\times(n-1)},$$

$$\phi\otimes\phi'' = \frac{1}{h_y^2}\begin{bmatrix} 4B & B & & & & & & \\ B & 4B & B & & & & & \\ & & & \cdots & \cdots & \cdots & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & B & 4B & B \\ & & & & & & B & 4B \end{bmatrix}_{(m-1)\times(m-1)},$$

$$B = \begin{bmatrix} -12 & 6 & & & & & & & \\ 6 & -12 & 6 & & & & & & \\ & 6 & -12 & 6 & & & & & \\ & & & \cdots & \cdots & \cdots & & & \\ & & & & \cdots & \cdots & \cdots & & \\ & & & & & & 6 & -12 & 6 \\ & & & & & & & 6 & -12 \end{bmatrix}_{(n-1)\times(n-1)} ,$$

$$\overrightarrow{\delta_j} = \begin{bmatrix} w_{1,j} \\ w_{2,j} \\ \cdots \\ \cdots \\ w_{n-2,j} \\ w_{n-1,j} \end{bmatrix}_{(n-1)\times 1} \qquad for \quad j = 1, 2, ..., (m-1) \quad ,$$

$$\overrightarrow{\gamma_1} = \begin{bmatrix} -12(w_{0,0} + w_{1,0} + w_{2,0} + w_{0,1} + w_{0,2}) \\ -12(w_{1,0} + w_{2,0} + w_{3,0}) \\ \cdots \\ \cdots \\ -12(w_{n-3,0} + w_{n-2,0} + w_{n-1,0}) \\ -12(w_{n-2,0} + w_{n-1,0} + w_{n,0} + w_{n,1} + w_{n,2}) \end{bmatrix}_{(n-1)\times 1} ,$$

$$\overrightarrow{\gamma_j} = \begin{bmatrix} -12(w_{0,j-1} + w_{0,j} + w_{0,j+1}) \\ 0 \\ \cdots \\ \cdots \\ 0 \\ -12(w_{n,j-1} + w_{n,j} + w_{n,j+1}) \end{bmatrix}_{(n-1)\times 1} \qquad for \quad \text{j=2,3,...,(m-2),}$$

$$\overrightarrow{\gamma}_{m-1} = \begin{bmatrix} -12(w_{0,m}+w_{1,m}+w_{2,m}+w_{0,m-1}+w_{0,m-2}) \\ \\ -12(w_{1,m}+w_{2,m}+w_{3,m}) \\ \cdots \\ \cdots \\ -12(w_{n-3,m}+w_{n-2,m}+w_{n-1,m}) \\ \\ \\ -12(w_{n-2,m}+w_{n-1,m}+w_{n,m}+w_{n,m-1}+w_{n,m-2}) \end{bmatrix}_{(n-1)\times 1}.$$

The system of equations (12) has been solved using Hockney method [11]. This scheme has computational cost O(p.log(p)), where, p=(n-1)(m-1), the total number of internal mesh points. Once the values of $w_{i,j}$ on internal mesh points are obtained, the approximate solution $U_{nm}(x,y)$ can be achieved at any point of the domain Ω, substituting $w_{i,j}$ in relation (3).

3 Numerical experiments and discussion

In this section, two numerical examples have been considered to verify the accuracy and efficiency of present scheme. All numerical computations have been performed here by using MATLAB. To verify the accuracy of the present scheme following two error norms have been calculated

$$\text{Absolute error at mesh point } (x_i,y_j) = |u(x_i,y_j) - U_{nm}(x_i,y_j)|$$
$$\text{Absolute error norm } (L_\infty) = \max_{i,j}|u(x_i,y_j) - U_{nm}(x_i,y_j)|$$

where, $u(x_i,y_j)$ is the exact solution and $U_{nm}(x_i,y_j)$ is corresponding approximate solution at mesh point (x_i,y_j) obtained by present scheme.

Example 1 : Consider the two-dimensional Laplace's equation over the domain $[0,1]\times[0,1]$,

$$u_{xx} + u_{yy} = 0, \tag{13}$$

with boundary conditions,

$$\left.\begin{array}{ll} u(0,y)=5, & u(1,y)=5 \\ u(x,0)=5, & u(x,1)=5 \end{array}\right\} \tag{14}$$

The exact solution of this problem is

$$\text{u(x,y)=5}$$

To test the convergence, the absolute errors in approximate solution $U_{nm}(x,y)$ have been calculated for different number of meshes 10×10, 15×15 and 20×20 and the absolute errors in solution

$U_{nm}(x, y)$ for 10×10 number of meshes have been presented in Table 4. It is observed that the approximate solutions are correct upto 15 decimal places. L_∞ error norm and CPU time to find approximate solution $U_{nm}(x, y)$ for different number of meshes are reported in the Table 5. The approximate solutions of Laplace's equation in example 1 have been depicted in figure 1 for 10×10 number of meshes.

Table 4: Absolute error in $U_{nm}(x, y)$ for example 1 for 10×10 number of meshes

Mesh point	Absolute error
(0.1,0.1)	8.8818e-16
(0.5,0.1)	8.8818e-16
(0.9,0.1)	1.7763e-15
(0.3,0.3)	8.8818e-16
(0.7,0.3)	1.7763e-15
(0.9,0.3)	8.8818e-16
(0.1,0.5)	8.8818e-16
(0.5,0.5)	1.7763e-15
(0.9,0.5)	1.7763e-15
(0.1,0.7)	8.8818e-16
(0.7,0.7)	1.7763e-15

Table 5: L_∞ and CPU time for example 1 for different number of meshes

Number of meshes	L_∞	CPU time (sec.)
10×10	1.4210e-14	0.03
15×15	4.8850e-14	0.04
20×20	1.4743e-13	0.06

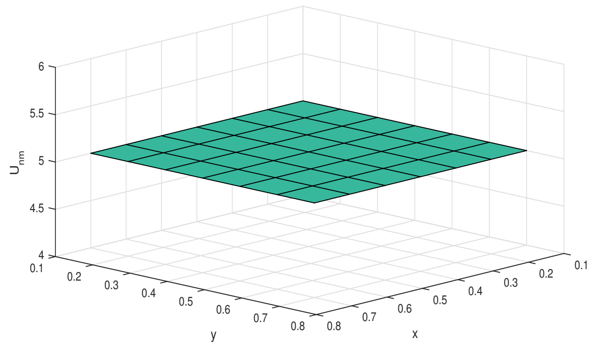

Figure 1: Approximate solution $U_{nm}(x, y)$ of example 1 for 10×10 number of meshes

Example 2 : Consider the two-dimensional Laplace's equation over the domain $[0, 1]\times[0, 1]$ as taken by Hamid et al. [4]

$$u_{xx} + u_{yy} = 0,$$

with boundary conditions

u(x,0)=x(1-x),
u(x,1)=0,
u(0,y)=0,
u(1,y)=0,

The exact solution given by Hamid et al. [4] is

$$u(x, y) = \sum_{i=1}^{\infty} -\frac{4sin(i\pi x)((-1)^i - 1)sinh(i\pi(1-y))}{sinh(i\pi)i^3\pi^3} \tag{15}$$

To choose appropriate number of meshes, the absolute errors in approximate solution $U_{nm}(x, y)$ at different mesh points for 5×5, 10×10, 15×15 and 20×20 number of meshes have been computed and absolute errors in $U_{nm}(x, y)$ for 20×20 number of meshes are given in Table 6. It is observed that obtained results are correct upto 4D places. L_∞ error norm and CPU time for calculation of $U_{nm}(x, y)$ for different number of meshes have been presented in Table 7. It is observed that the error decreases as the size of mesh gets finer and finer. Further, it demonstrates that the present scheme is quite economical with respect to computational time. The approximate solution for 20×20 number of meshes have been depicted in figure 2. Table 8 presents the comparison of approximate solutions of Laplace's equation with exact results and those given by Hamid et al. [4] and Reddy et al. [12] for 5×5 number of meshes. It is observed that the solutions obtained by present scheme using modified bi-cubic B-spline functions are better than those obtained by other two methods.

Table 6: Absolute error in $U_{nm}(x, y)$ for example 2 for 20×20 number of meshes

Mesh point	Absolute error
(0.1,0.5)	1.5451e-05
(0.1,0.7)	1.1090e-06
(0.1,0.9)	6.9824e-07
(0.3,0.7)	4.7856e-06
(0.3,0.9)	1.4955e-06
(0.5,0.5)	6.4274e-05
(0.5,0.9)	1.5908e-06
(0.7,0.7)	4.7856e-06
(0.7,0.9)	1.4955e-06
(0.9,0.3)	7.9266e-05
(0.9,0.5)	1.5451e-05
(0.9,0.9)	6.9824e-07

Table 7: L_∞ and CPU time for example 2 for different mesh sizes

Number of meshes	L_∞	CPU time (sec.)
5×5	3.5119e-02	0.10
10×10	7.2649e-03	0.15
15×15	1.7932e-03	0.20
20×20	9.5347e-04	0.23

Table 8: Comparison of approximate solutions for example 2 for 5×5 number of meshes

Mesh point	Exact	Present scheme	Hamid et al.[4]	Reddy et al. [12]
(0.25,0.25)	0.0832	0.0581	0.0788	0.0779
(0.25,0.5)	0.0364	0.0375	0.0335	0.0323
(0.25,0.75)	0.0137	0.0138	0.0123	0.0115
(0.5,0.25)	0.1159	0.0826	0.1121	0.1094
(0.5,0.5)	0.0513	0.0531	0.0473	0.0441
(0.5,0.75)	0.0194	0.0195	0.0174	0.0157
(0.75,0.25)	0.0832	0.0581	0.0788	0.0685
(0.75,0.5)	0.0364	0.0375	0.0335	0.0276
(0.75,0.75)	0.0137	0.0138	0.0123	0.0098

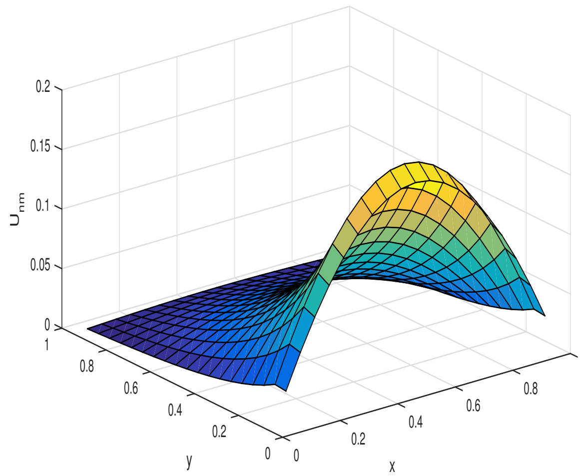

Figure 2: Approximate solution $U_{nm}(x, y)$ of example 2 for 20×20 number of meshes

4 Conclusion

In this paper, the two-dimensional Laplace's equation with Dirichlet boundary conditions has been solved by using modified bi-cubic B-spline functions. To check the efficiency and accuracy of present scheme, two examples of two-dimensional Laplace's equation have been considered. The absolute errors and absolute error norms (L_∞) are presented for both examples. To choose appropriate number of meshes, absolute errors have been calculated for different number of meshes. Further, approximate solutions and absolute errors for present scheme have been compared with two methods namely bi-cubic B-spline interpolation method and bi-cubic B-spline collocation method. It is observed that present scheme gives better results in comparison to other methods. The approximate solutions using 10×10 and 20×20 number of meshes are presented for example 1 and example 2, respectively. The calculations have been carried out using MATLAB.

Reference

[1] Smith, G.D.: *Numerical Solutions of Partial Differential Equations: Finite Difference Methods*, Oxford University Press, New York, (1985).

[2] Al-Khaled, K.: Numerical solutions of the Laplace's equation. Appl. Math. Comput. **170** 1271–1283 (2005).

[3] Shabbir, M., Rafiq, M., Ahmed, M.O., Pervaiz, A., Siddique, R.: Finite element solution for two-dimensional Laplace's equation with dirichlet boundary conditions. Pak. J. Engg. Appl. Sci. **10** 97-102 (2012).

[4] Hamid, N.N.H., Majid, A.A., Ismail, A.I.M.: B-spline interpolation method for two-dimensional Laplace's equations. (2012) DOI:10.1063/1.4801243.

[5] Patil, P.V., Prasad, J.S.V.R.K.: A numerical grid and grid less (mesh less) techniques for the solution of two-dimensional Laplace's equation. Adv. Appl. Sci. Res. **5 (1)** 150-155 (2014).

[6] Patil, P.V., Prasad, J.S.V.R.K.: Numerical solution for two-dimensional Laplace's equation with dirichlet boundary conditions. Int. Org. Sci. Res. J. Math. **6 (4)** 66-75 (2013).

[7] Buralieva, J.V., Hadzieva, E., Saneva, K.H.V.: On a numerical solution of the Laplace's equation. Adv. Math. Sci. J. **4 (2)** 209-220 (2015).

[8] Reddy, Y.R., Reddy, C.S., Murthy M.V.R.: A Numerical technique-recursive form of bi-cubic B-spline collocation solution to Laplace's equation. Int. J. Comput. Appl. Math. **11 (1)** 71-87 (2016).

[9] Chopade, P.P., Rastogi, P.S.: Numerical method algorithms for solution of two-dimensional Laplace's equation in electrostatics. Amer. J. Comput. Appl. Math. **8 (4)** 65-69 (2018).

[10] Mittal, R.C., Tripathi, A.: Numerical solutions of two-dimensional Burgers'equations using modified bi-cubic B-spline finite elements. Eng. Comput. **32 (5)** 1275-1306 (2015).

[11] Hockney, R.W.: A fast direct solution of Poisson's equation using fourier analysis. J. Assoc. Comput. Mach. **12** 95-113 (1965).

[12] Reddy, Y.R., Reddy, C.S., Murthy M.V.R.: Numerical solution to Poisson's equation by recursive form of bi-cubic B-spline collocation solution. Int. J. Appl. Sci. Math. **3 (4)** 144-152 (2016).