Tension spline method for the solution of elliptic equations

Homa Zadvana and Jalil Rashidinia

Department of Mathematics and statistics, Central Tehran Branch, Islamic Azad University, Tehran, Iran; School of Mathematics, Iran University of Science and Technology, Tehran, Iran

ABSTRACT

In this paper, two classes of methods are developed for the solution of two-dimensional elliptic partial differential equations. We have used tension spline function approximation in both x and y spatial directions and a new scheme of order O(h^4 + A^4) has been obtained. The convergence analysis of the methods has been carried out. Numerical examples are given to illustrate the applicability and accurate nature of our approach.

1. Introduction

Elliptic partial differential equations often arise in many fields of engineering and theoretical physics such as electric potential in electrostatics, electromagnetic scattering theory [1], micro and nano electronic devise physics, membrane vibration and radar scattering [2, 3].

We consider the linear second-order two-dimensional elliptic equation of the form

\[ A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial y^2} = Cu + f(x,y), \quad (x,y) \in \Omega, \]

subjected to Dirichlet boundary conditions

\[ u(x,y) = g(x,y), \quad (x,y) \in \partial\Omega, \]

where A, B and C are positive constants and \( \Omega = \{(x,y); 0 < x, y < 1\} \) with the boundary \( \partial\Omega \). We assume that the solution \( u(x,y) \) and forcing function \( f \) are sufficiently smooth and have required continuous partial derivatives. The well-known Poisson equation and modified Helmholtz equation are special cases of Equation (1).

Various numerical methods have been discussed and developed for solving elliptic partial differential equations. For the solution of two-dimensional Poisson equation, fourth-order compact difference schemes have discussed in [4, 5] and also a new family of high-order finite difference schemes is proposed in [6] by using implicit finite difference formulas. Sixth-order compact scheme is developed for the Poisson equation in [7] based on the fourth-order nine-point difference approximation, combined with Richardson extrapolation. Fourth-order finite difference methods for the solution of Helmholtz equation are presented in [8, 9]. Cubic spline collocation method was introduced for Poisson equation by Romenski [10]. Bialecki used orthogonal spline collocation method for the solution of Poisson equation [11]. The method of collocation at nodal points based on the tensor product of cubic splines for Helmholtz equation with Dirichlet boundary conditions was first analysed independently by Cavendish and Ito in their PhD theses and second-order convergence of the method was proved. Christara has developed quadratic spline collocation methods for elliptic problems in the unit square [12]. These methods which have third order of accuracy, collocate a perturbed differential equation. Houstis et al. analysed a new class of collocation methods using cubic spline for the general form of elliptic PDEs and derived fourth order of accuracy [13]. Quadratic spline collocation methods for two-dimensional Helmholtz problem are developed in [14], it is shown that the methods are fourth-order accurate at the nodes and collocation points. Rashidinia et al. proposed the cubic spline method for the singular elliptic boundary value problems in [15, 16]. Islam Khan et al. used tension spline method to solve second-order singularly perturbed boundary-value problems [17] also tension spline method for the solution of nonlinear partial differential equations have discussed in [18, 19]. Mohanty et al. have used a new finite difference scheme for two-dimensional elliptic problems [20], also they proposed...
cubic spline finite difference method of order \(O(k^2 + h^4)\) for two-dimensional quasi-linear elliptic equations [21]. Higher-order accurate finite volume schemes for two-dimensional Helmholtz equations are developed in [22]. For numerically solving inhomogeneous elliptic partial differential equations, a Chebyshev polynomial scheme combined with the method of fundamental solutions (MFS) and the equilibrated collocation Trefftz method have been presented in [23] and a particular solution using the standard polynomial basis function has been derived in [24]. A new wavelet multigrid method based on Daubechies filter coefficients is proposed in [25] for the numerical solution of elliptic type equations.

In this paper, we have presented the non-polynomial tension spline method for two-dimensional linear elliptic equation (1). We discussed the derivation of non-polynomial tension spline approximation and by using tension spline approximation in both spatial directions, two classes of nine-point scheme have been obtained. The truncation error of the scheme is carried out and convergence of our approach is proved.

2. Non-polynomial spline functions

We consider the solution domain \(\Omega = [0, 1] \times [0, 1]\) and choose grid spacing \(h > 0\) and \(k > 0\), in the \(x\)-direction and \(y\)-direction, respectively. The mesh points are defined by \((x_i, y_j) = (ih, jk), \ i = 0, 1, \ldots, m + 1, j = 0, 1, \ldots, n + 1,\) where \(m\) and \(n\) are positive integers. In each segment \([x_i, x_{i+1}]\) let \(s_{1j}(x)\) be the tension spline function which interpolates \(u(x, y_j)\) and define

\[
s_{1j}(x) = a_{1j} + b_{1j}(x - x_i) + c_{1j} \sinh \lambda_1 (x - x_i) + d_{1j} \cosh \lambda_1 (x - x_i),
\]

where \(a_{1j}, b_{1j}, c_{1j}\) and \(d_{1j}\) are unknown coefficients and \(\lambda_1\) is free parameter. Also, we let \(s_{2j}(y)\) be the tension spline function interpolating \(u(x_i, y)\) in each segment \([y_j, y_{j+1}]\) and define

\[
s_{2j}(y) = a_{2j} + b_{2j}(y - y_j) + c_{2j} \sinh \lambda_2 (y - y_j) + d_{2j} \cosh \lambda_2 (y - y_j),
\]

where \(a_{2j}, b_{2j}, c_{2j}\) and \(d_{2j}\) are unknown coefficients and \(\lambda_2\) is an arbitrary parameter. To determine explicit expressions for all of coefficients we first denote

\[
s_{1j}(x_i) = u_{ij}, \quad s_{1j}(x_{i+1}) = u_{i+1,j}, \quad s_{2j}(y_j) = u_{ij},
\]

\[
s_{2j}(y_{j+1}) = u_{i,j+1}, \quad s_{2j}'(x_i) = M_{1ij}, \quad s_{2j}'(x_{i+1}) = M_{1i+1,j},
\]

\[
s_{2j}'(y_j) = M_{2ij}, \quad s_{2j}'(y_{j+1}) = M_{2i,j+1}.
\]

By using (4) and (5) and from algebraic manipulations we derive

\[
a_{1j} = u_{ij} - \frac{1}{\lambda_1} M_{1ij}, \quad b_{1j} = \frac{1}{h} (u_{i+1,j} - u_{ij}) + \frac{1}{\omega_1 \lambda_1} (M_{1ij} - M_{1i+1,j}),
\]

\[
c_{1j} = \frac{1}{\lambda_1} (\csc(\omega_1) M_{1i+1,j} - \coth(\omega_1) M_{1ij}),
\]

\[
d_{1j} = \frac{1}{\lambda_1} M_{1ij},
\]

where \(\omega_1 = \lambda_1 h\). And also

\[
a_{2j} = u_{ij} - \frac{1}{\lambda_2} M_{2ij}, \quad b_{2j} = \frac{1}{k} (u_{i,j+1} - u_{ij}) + \frac{1}{\omega_2 \lambda_2} (M_{2ij} - M_{2i+1,j}),
\]

\[
c_{2j} = \frac{1}{\lambda_2} (\csc(\omega_2) M_{2i+1,j} - \coth(\omega_2) M_{2ij}),
\]

\[
d_{2j} = \frac{1}{\lambda_2} M_{2ij},
\]

where \(\omega_2 = \lambda_2 k\).

From the continuity of the first derivatives of spline functions \(s_{1j}(x)\) and \(s_{2j}(y)\) at \((x_i, y_j)\), we get the following consistency relations:

\[
\frac{1}{h^2}(u_{i-1,j} - 2u_{ij} + u_{i+1,j}) = \alpha_1 M_{1i-1,j} + 2\beta_1 M_{1ij} + \alpha_1 M_{1i+1,j},
\]

\[
\frac{1}{k^2}(u_{i,j-1} - 2u_{ij} + u_{i,j+1}) = \alpha_2 M_{2ij-1} + 2\beta_2 M_{2ij} + \alpha_2 M_{2ij+1},
\]

where

\[
\alpha_1 = \frac{1}{\omega_1^2} (1 - \omega_1 \csc(\omega_1)),
\]

\[
\beta_1 = \frac{1}{\omega_1^2} (\omega_1 \coth(\omega_1) - 1),
\]

\[
\alpha_2 = \frac{1}{\omega_2^2} (1 - \omega_2 \csc(\omega_2)),
\]

\[
\beta_2 = \frac{1}{\omega_2^2} (\omega_2 \coth(\omega_2) - 1).
\]

When \(\lambda_1 \to 0\) that \(\omega_1 \to 0\), then \((\alpha_1, \beta_1) \to (1/6, 1/3)\) and also when \(\lambda_2 \to 0\), then \((\alpha_2, \beta_2) \to (1/6, 1/3)\) so consistency relations of tension splines defined in (6) and (7) reduce to the following cubic spline relations, respectively:

\[
(u_{i-1,j} - 2u_{ij} + u_{i+1,j}) = \frac{h^2}{6} (M_{1i-1,j} + 4M_{1ij} + M_{1i+1,j}),
\]

\[
(u_{i,j-1} - 2u_{ij} + u_{i,j+1}) = \frac{k^2}{6} (M_{2ij-1} + 4M_{2ij} + M_{2ij+1}).
\]
3. Spline numerical methods

At the grid point \((x_i, y_j)\), Equation (1) may be discretized as

\[
Au_{xx}(x_i, y_j) + Bu_{yy}(x_i, y_j) = Cu_{x}(x_i, y_j) + f(x_i, y_j). \tag{10}
\]

We next develop an approximation for Equation (1) in which we use the non-polynomial tension spline functions approximations of second derivatives in \(x\)-direction and also in \(y\)-direction:

\[
u_{xx}(x_i, y_j) \equiv M_{11j}, \tag{11}
\]

\[
u_{yy}(x_i, y_j) \equiv M_{22j}. \tag{12}
\]

From (10) and by neglecting the truncation error, we get

\[
AM_{11j} + BM_{22j} = Cu_{x} + f_{xj}, \tag{13}
\]

\[
AM_{11j+1} + BM_{22j+1} = Cu_{x} + f_{xj+1}, \tag{14}
\]

\[
AM_{11j+1} + BM_{22j+1} = Cu_{x} + f_{xj+1}. \tag{15}
\]

Similar to (6), for the \((j - 1)\)-th and \((j + 1)\)-th levels in \(y\)-direction we have

\[
(u_{i-1j} - 2u_{ij} + u_{i+1j}) = h^2(\alpha_1 M_{11j-1} + 2\beta_1 M_{11j-1} + \alpha_1 M_{11j-1}), \tag{16}
\]

\[
(u_{i-1j+1} - 2u_{ij+1} + u_{i+1j+1}) = h^2(\alpha_1 M_{11j+1} + 2\beta_1 M_{11j+1} + \alpha_1 M_{11j+1}). \tag{17}
\]

Similar to (7), for the \((i - 1)\)-th and \((i + 1)\)-th levels in \(x\)-direction we have

\[
(u_{i-1j-1} - 2u_{i-1j} + u_{i+1j-1}) = k^2(\alpha_2 M_{22j-1} + 2\beta_2 M_{22j-1} + \alpha_2 M_{22j-1}), \tag{18}
\]

\[
(u_{i+1j-1} - 2u_{ij+1} + u_{i+1j+1}) = k^2(\alpha_2 M_{22j+1} + 2\beta_2 M_{22j+1} + \alpha_2 M_{22j+1}). \tag{19}
\]

By using Equations (6), (7) and (13)–(19) and after manipulating them we obtain

\[
p_1(u_{i-1j} - u_{i-1j+1} + u_{i+1j} - u_{i+1j+1}) + p_2(u_{i-1j} + u_{i+1j}) + p_3(u_{ij} - u_{ij+1}) + p_4u_{ij} = r_1(f_{ij} - f_{ij+1}) + r_2(f_{ij} + f_{ij+1}) + r_3(f_{ij} - f_{ij+1}) + r_4f_{ij}, \tag{20}
\]

where

\[
p_1 = Ak^2\alpha_2 + Bh^2\alpha_1 - Cr_1,
\]

\[
p_2 = 2Ak^2\beta_2 - 2Bh^2\alpha_1 - Cr_2,
\]

\[
p_3 = 2Bh^2\beta_1 - 2Ak^2\alpha_2 - Cr_3,
\]

\[
p_4 = -4Ak^2\beta_2 - 4Bh^2\beta_1 - Cr_4,
\]

\[
r_1 = h^2k^2\alpha_1\alpha_2,
\]

\[
r_2 = 2h^2k^2\alpha_1\beta_2,
\]

\[
r_3 = 2h^2k^2\alpha_2\beta_1,
\]

\[
r_4 = 4h^2k^2\beta_1\beta_2. \tag{21}
\]

4. Truncation error

The partial differential equation (10) may be discretized as

\[
f_{ij} = (AD_x^2 + BD_y^2) - Cu_{xij}, \tag{23}
\]

where \(D_x^2 = (\partial^2 / \partial x^2)\) and \(D_y^2 = (\partial^2 / \partial y^2)\). Also we have

\[
f_{i+1, j} = (AD_x^2 + BD_y^2) - Cu_{x(i+1)j},
\]

\[
f_{ij+1} = (AD_x^2 + BD_y^2) - Cu_{xij+1}.
\]

Now, by using the above notations and expanding both sides of (20) in Taylor series in terms of \(u(x_i, y_j)\) and its derivatives, we obtain the truncation error of the method as follows:

\[
T_{ij} = \left[2(\alpha_2 + \beta_2)(1 - 2\alpha_1 - 2\beta_1)A\bar{h}^2k^2D_x^2 + 2(\alpha_1 + \beta_1)(1 - 2\alpha_2 - 2\beta_2)B\bar{h}^2k^2D_y^2 + \alpha_2(1 - 2\alpha_1 - 2\beta_1)A\bar{h}^4k^4D_x^4 + \alpha_1(1 - 2\alpha_2 - 2\beta_2)B\bar{h}^4k^4D_y^4 + 2\left(\frac{1}{12} - \alpha_1\right)(\alpha_2 + \beta_2)A\bar{h}^6k^6D_x^6 + \left(\frac{1}{12} - \alpha_2\right)(\alpha_1 + \beta_1)B\bar{h}^6k^6D_y^6 + \alpha_2\left(\frac{1}{12} - \alpha_1\right)A\bar{h}^8k^8D_x^8 + \alpha_1\left(\frac{1}{12} - \alpha_2\right)B\bar{h}^8k^8D_y^8 + 2\left(\frac{1}{30} - \alpha_1\right)(\alpha_2 + \beta_2)A\bar{h}^{10}k^{10}D_x^{10} + 2\left(\frac{1}{30} - \alpha_2\right)(\alpha_1 + \beta_1)B\bar{h}^{10}k^{10}D_y^{10} \right] u_{ij}. \tag{24}
\]

From (24) we conclude that
5. Convergence analysis

The scheme (20) may be written in the following matrix form:

\[ PU = F, \]  

(27)

where \( U \) is the solution vector in the form \( U = [u_1, u_2, \ldots, u_n, \ldots, u_{m,1}, u_{m,2}, \ldots, u_{m,n}]^T \) and \( F \) is the right-hand side vector. \( P \) is a square block tridiagonal matrix of order \( n \times n \) and can be expressed as \( P = L + D + R \), where \( D \) is block diagonal matrix which each block of \( D \) is a triangular matrix in the form \( tri[p_2, p_3, p_4, p_5] \). \( L \) and \( R \) are strictly block lower triangular and strictly block upper triangular matrices, respectively. The blocks of \( R \) and \( L \) are equals to each other and are in the form \( tri[p_1, p_3, p_1] \), where the triangular matrix \( T = tri[a, b, a] \) is defined by

\[ T = (t_{ij}) = \begin{cases} 
  b, & i = j, \\
  a, & |i - j| = 1, \\
  0, & \text{otherwise}.
\end{cases} \]

From relations (21) and (22) we have

\[ |p_1| = |Cr_1 - Ak^2a_2 - Bh^2a_1| \leq Cr_1, \]
\[ |p_2| \leq 2Ak^2\beta_2, \]
\[ |p_3| \leq 2Bh^2\beta_1. \]

Then using (25) and (26) we can obtain

\[ i) \ |p_1| + |p_2| + |p_3| < |p_4|, \]
\[ ii) \ 2|p_1| + |p_2| + 2|p_3| < 2|p_4|, \]
\[ iii) \ 2|p_1| + 2|p_2| + |p_3| < 2|p_4|, \]
\[ iv) \ 4|p_1| + 2|p_2| + 2|p_3| < 4|p_4|. \]

(28)

From (28) we conclude that \( P \) is a diagonally dominant matrix and the system (27) has unique solution. For solving the system (27) by \( P = L + D + R \), the block Jacobi method is as follows:

\[ U^{(n+1)} = M_J U^{(n)} + D^{-1} f, \]

where \( M_J \), the Jacobi iteration matrix is defined by \( M_J = -D^{-1}(L + R) \). Also the block Gauss–Seidel method is

\[ U^{(n+1)} = M_{GS} U^{(n)} + (L + D)^{-1} f, \]

where \( M_{GS} = -(L + D)^{-1} R \).

For the nine-point scheme (20) the Jacobi iteration formula is

\[ u_{ij}^{(n+1)} = -\frac{1}{p_4} \left[ p_1(u_{i-1,j-1}^{(n)} + u_{i-1,j+1}^{(n)} + u_{i+1,j-1}^{(n)} + u_{i+1,j+1}^{(n)}) + p_2(u_{i-1,j}^{(n)} + u_{i+1,j}^{(n)}) + p_3(u_{i,j-1}^{(n)} + u_{i,j+1}^{(n)}) \right] + \frac{1}{p_4} \left[ r_1(f_{i-1,j-1} + f_{i-1,j+1} + f_{i+1,j-1} + f_{i+1,j+1}) + r_2(f_{i-1,j} + f_{i+1,j}) + r_3(f_{i,j-1} + f_{i,j+1}) + r_4f_{ij} \right]. \]

(29)

The error equation of the iteration is

\[ \varepsilon_{ij}^{(n+1)} = -\frac{1}{p_4} \left[ p_1(\varepsilon_{i-1,j-1}^{(n)} + \varepsilon_{i-1,j+1}^{(n)} + \varepsilon_{i+1,j-1}^{(n)} + \varepsilon_{i+1,j+1}^{(n)}) + p_2(\varepsilon_{i-1,j}^{(n)} + \varepsilon_{i+1,j}^{(n)}) + p_3(\varepsilon_{i,j-1}^{(n)} + \varepsilon_{i,j+1}^{(n)}) \right], \]

(30)

where the values of the error are equal to zero on the boundary of the unit square.

Equation (30) can be solved by the method of separation of variables, we let

\[ \varepsilon_{ij}^{(n+1)} = \xi \varepsilon_{ij}^{(n)}, \]

(32)

where \( \xi \) is the propagation factor. By substituting (31) and (32) in (30), we get

\[ W_{j-1} - \gamma W_j + W_{j+1} = 0 \]

(33)

and

\[ h_1 V_{j-1} + h_2 V_j + h_3 V_{j+1} = 0. \]

(34)

where \( \gamma \) is an arbitrary parameter to be determine and

\[ h_1 = p_1 \gamma + p_2, \quad h_2 = p_3 \gamma + p_4 \xi, \]

(35)

and boundary conditions are \( V_0 = V_{m+1} = 0 \) and \( V_0 = W_{n+1} = 0 \).

The solution of difference equation (33) subject to homogeneous boundary conditions is

\[ W_j = w_1 \sin(k\pi q), \quad 1 \leq j \leq n, 1 \leq q \leq n \]

(36)

with \( \gamma = 2 \cos(k\pi q) \) and \( w_1 \) an arbitrary constant.
From (28-(iv)) and (39), we conclude that
and for any values of $h$, the Jacobi method converges for any initial guess by
and convergence orders are tabulated in Tables 1–3.

Also, the solution of (34) with homogeneous boundary conditions is

$$V_i = v_1 \sin(h \pi p i), \quad 1 \leq i \leq m, \quad 1 \leq p \leq m, \quad (37)$$
$$l_2 = -2l_1 \cos(h \pi p), \quad (38)$$

and $v_1$ is an arbitrary constant. By eliminating $h_1$ and $l_2$ from (35) and (38), we get

$$\xi = \frac{1}{p^4} \left( -4p_1 \cos(k \pi q) \cos(h \pi p) - 2p_2 \cos(h \pi p) \right.$$
$$\left. - 2p_3 \cos(k \pi q) \right). \quad (39)$$

From (28-(iv)) and (39), we conclude that $|\xi| < 1$. Therefore, the Jacobi method converges for any initial guess and for any values of $h$ and $k$. Also the spectral radius $\rho$ of Jacobi and Gauss–Seidel iteration matrices are related by

$$\rho(M_{GS}) = \rho(M_1)^2.$$

Thus the Gauss–Seidel method converges too.

### 6. Numerical results

In this section, we consider the following examples with the known exact solutions. We applied our methods to solve these examples, with various values of $h$ and $k$. The computed solutions at grid points are compared with the exact solutions, the maximum absolute errors and convergence orders are tabulated in Tables 1–3.

The orders of convergence are established from the following relation:

$$\log \left( \frac{\|u - u_1\|}{\|u - u_2\|} \right) \approx \log \left( \frac{h_1}{h_2} \right),$$

(40)

where $u$ is the exact solution and $u_1$ and $u_2$ are computed solutions in two consequence steps $(h_1, k_1)$ and $(h_2, k_2).

**Example 6.1:** Consider the equation

$$\frac{\partial^2 u}{\partial x^2} + 2 \frac{\partial^2 u}{\partial y^2} = \pi^2 u - 3\pi \sin(\pi y)(2\sin(\pi x)$$
$$+ 2\pi x \cos(\pi x) + \pi x), \quad 0 < x, y < 1,$$

subjected to Dirichlet boundary condition $u = 0$ in boundaries of unit square. The exact solution of this problem is

$$u(x, y) = x \sin(\pi y)(1 + \cos(\pi x)).$$

We applied our methods $T_1$ and $T_2$ to solve this problem. The computed solutions are compared with exact solutions at grid points and the maximum absolute errors of solutions are tabulated in Table 1. The graphs of the exact and computed solutions are given in Figure 1.

**Example 6.2:** Consider the equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 2u + 6e^{x+y}(2xy^2 + 2x^2y - 4xy),$$

$$0 < x, y < 1,$$

(41)

subjected to Dirichlet boundary condition $u = 0$ in boundaries of unit square. The exact solution is

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

We applied method $T_2$ to solve this problem. The maximum absolute errors and orders of convergence are tabulated in Table 2 and are compared with the results in reference [13].

**Example 6.3:** Consider the equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = c^2 \pi^2 u - 3c^2 \pi^2 \sin(c \pi x) \sin(c \pi y),$$

$$0 < x, y < 1,$$

(42)

subjected to Dirichlet boundary condition $u = 0$ in

| Table 1. The maximum absolute errors and orders of convergence of Example 6.1. |
| (h,k) | Method ($T_1$) c = 4 | Conv. order | Method ($T_2$) c = 4 | Conv. order | Method ($T_2$) c = 16 | Conv. order |
|-------|-----------------------|------------|-----------------------|------------|-----------------------|------------|
| (1/16, 1/16) | 2.4162 E−3 | 6.6144 E−6 | 9.6554 E−6 | 1.3588 E−6 | 6.0505 E−6 | 2.3373 E−6 |
| (1/16, 1/32) | 2.0264 E−3 | 5.6684 E−6 | 1.9399 E−7 | 4.0000 E−8 | 4.0000 E−8 | 2.3373 E−6 |
| (1/32, 1/16) | 6.0505 E−4 | 1.9399 E−7 | 4.0000 E−8 | 4.0000 E−8 | 4.0000 E−8 | 2.3373 E−6 |
| (1/64, 1/64) | 1.4752 E−4 | 2.0417 E−7 | 2.4167 E−8 | 4.0404 E−9 | 4.0404 E−9 | 2.3373 E−6 |
| (1/128, 1/128) | 2.3373 E−5 | 2.6667 E−8 | 9.8365 E−10 | 4.6744 E−11 | 4.6744 E−11 | 2.3373 E−6 |

| Table 2. The maximum absolute errors in Example 6.2. |
| (h,k) | Our method ($T_2$) | Conv. order | Method [13] | Conv. order |
|-------|----------------------|------------|---------------|------------|
| (1/16, 1/16) | 2.315 E−5 | 5.72 E−3 | 5.72 E−3 | 5.72 E−3 |
| (1/16, 1/32) | 4.758 E−6 | 8.29 E−4 | 8.29 E−4 | 8.29 E−4 |
| (1/32, 1/16) | 3.714 E−7 | 5.89 E−5 | 5.89 E−5 | 5.89 E−5 |
| (1/21, 1/21) | 1.525 E−7 | 4.21 E−5 | 4.21 E−5 | 4.21 E−5 |

| Table 3. The maximum absolute errors and orders of convergence for Example 6.3. |
| (h,k) | Method ($T_1$) c = 4 | Conv. order | Method ($T_2$) c = 4 | Conv. order | Method ($T_2$) c = 16 | Conv. order |
|-------|-----------------------|------------|-----------------------|------------|-----------------------|------------|
| (0.02, 0.02) | 1.0436 E−1 | 9.5330 E−3 | 1.8696 E−1 | 4.3000 E−3 |
| (0.1, 0.1) | 2.7798 E−2 | 5.6792 E−4 | 5.6792 E−4 | 5.6792 E−4 | 5.6792 E−4 | 4.0600 E−3 |
| (0.025, 0.025) | 1.9548 E−3 | 2.4167 E−6 | 3.8782 E−5 | 3.8782 E−5 | 3.8782 E−5 | 3.8800 E−3 |
| (0.02, 0.02) | 1.2470 E−3 | 6.5835 E−7 | 1.5800 E−5 | 4.0200 E−3 |
boundaries of unit square. The exact solution is
\[ u(x, y) = \sin(c\pi x) \sin(c\pi y). \]

We have solved this problem with methods $T_1$ and $T_2$ for two values of $c$. The maximum absolute errors and orders of convergence of methods are tabulated in Table 3.

7. Conclusion

We applied non-polynomial tension spline approximation in both space directions for the solution of second-order elliptic partial differential equations. Using the parameters adopted in cubic tension spline, we have obtained a scheme of order $O(h^2 + k^2)$. Also by appropriate choices of the parameters we have increased the order of accuracy to new scheme of order $O(h^4 + k^4)$. The results of the numerical examples show that our methods are accurate and easy applicable.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Jalil Rashidinia http://orcid.org/0000-0002-9177-900X

References

[1] Barkeshli K. Advanced electromagnetics and scattering theory. New York: Springer; 2015.
[2] Datta S. Quantum transport: atom to transistor. London: Cambridge University Press; 2005.
[3] Borden B. Radar imaging of airborne targets. New York: Taylor & Francis; 1999.
[4] Tian Z. An high order compact finite difference scheme for the second dimensional Poisson equation. J Northwest Univ. 1996;26(2):109–114.
[5] Zhang J. Multigrid method and fourth order compact scheme for 2d Poisson equation with unequal meshsize discretization. J Comput Phys. 2002;179:170–179.
[6] Zapata RIMU. High-order implicit finite difference schemes for the two-dimensional Poisson equation. Appl Math Comput. 2017;309:222–244.
[7] Wang Y, Zhang J. Sixth order compact scheme combined with multigrid method and extrapolation technique for 2d Poisson equation. J Comput Phys. 2009;228:137–146.
[8] Singer I, Turkel E. High-order finite difference methods for the Helmholtz equation. Comput Method Appl Math. 1998;163:343–358.
[9] Fu Y. Compact fourth order finite difference schemes for Helmholtz equation with high wave numbers. J Comput Math. 2008;26:98–111.
[10] Romenski V. The method of spline collocation for the Poisson equation. J Comput Acoust. 1979;81:81–86.
[11] Bialecki B. Superconvergence of orthogonal spline collocation in the solution of Poisson's equation. Numer Methods Partial Diff Eqn. 1999;15:285–303.
[12] Christara CC. Quadratic spline collocation methods for elliptic partial differential equations. BIT. 1994;34:33–61.
[13] Houstis E, Vavalis E, Rice J. Convergence of $O(h^4)$ cubic spline collocation methods for elliptic partial differential equations. SIAM J Numer Anal. 1988;25(1):54–74.
[14] Fairweather G, Karageorghis A, Maack J. Compact optimal quadratic spline collocation methods for the Helmholtz equation. J Comput Phys. 2011;230:2880–2895.
[15] Rashidinia J, Mohammadi R, Ghasemi M. Cubic spline solution of singularly perturbed boundary value problems with significant first derivatives. Appl Math Comput. 2007;190:1762–1766.
[16] Rashidinia J, Mohammadi R, Jalilian R. Spline solution of nonlinear singular boundary value problems. Int J Comput Math. 2008;85:39–52. doi:10.1080/00207160701293048
[17] Khan I, Aziz T. Tension spline method for second-order singularly perturbed boundary-value problems. Int J Comput Math. 2005;82(12):1547–1553. doi:10.1080/00207160410001684280
[18] Aghamohamadi M, Rashidinia J, Ezzati R. Tension spline method for solution of non-linear Fisher equation. Appl Math Comput. 2014;249:399–407.

[19] Rashidinia J, Mohammadi R. Tension spline solution of nonlinear sine-Gordon equation. Numer Algor. 2011;56:129–142. doi:10.1007/s11075-010-9377-x

[20] Mohanty R, Dey S. A new finite difference discretization of order four for \((\partial u/\partial n)\) for two dimensional quasi-linear elliptic boundary value problems. Int J Comput Math. 2001;76:505–516.

[21] Mohanty R, Jain MK, Dhall D. High accuracy cubic spline approximation for two dimensional quasi-linear elliptic boundary value problems. Int J Comput Math. 2013;37:155–171.

[22] Yaw K, Kossi E. Higher-order accurate finite volume discretization of Helmholtz equations with pollution effects reductions. Int J Innov Edu Res. 2018;6(2):130–148.

[23] Ghimire BK, Tian HY, Lamichhane A. Numerical solutions of elliptic partial differential equations using Chebyshev polynomials. Comput Math Appl. 2016;72:1042–1054.

[24] Dangal T, Chen C, Lin J. Polynomial particular solutions for solving elliptic partial differential equations. Comput Math Appl. 2017;73:60–70.

[25] Shiralashetti S, Kantli M, Deshi AB. A new wavelet multi-grid method for the numerical solution of elliptic type differential equations. Alex Eng J. 2018;57(1):203–209. doi:10.1016/j.aej.2016.12.007