International Journal of Computational and Applied Mathematics.
ISSN 1819-4966 Volume 11, Number 1 (2016), pp. 71-87

http://www.ripublication.com

A Numerical Technique-Recursive form of Bi-cubic B-spline Collocation Solution to Laplace equation

Y. Rajashekhar Reddy [1], Ch. Sridhar Reddy [2], M. V. Ramana Murthy[3]

[1]Assistant Professor, Department of mathematics
JNT University college of Engineering Jagitial,
Nachupally (kondagattu) Karimnagar-505501, Telangana State, India
E-mail: yrsreddy4@gmail. com
[2]Associate Professor, Department of Mechanical engineering
JNT University college of Engineering Jagitial, Nachupally,
Karimnagar-505501, Telangana State, India
E-mail: Sridhar. jntuhcej@gmail. com
[3] Professor, Dept of mathematics Osmania University
Hyderabad, Telagana State, India-500074
E-mail: mv. rm50@gmail. com

Abstract

Tensor product of third degree B-spline basis functions is used as basis functions in collocation method for approximate solution of second order partial differential equations. Recursive form of B-spline function is used as basis in this present method. This method is applied to find the approximate solution of heat transfer governing partial differential equations with Dirchlet’s Boundary conditions for different kinds of domains. The results show its efficiency and consistency. The easiness of the method reduces the complexity and time consuming when compared with other existed methods.

Key words: B-splines, collocation, Laplace equations, Dirchlet’s Boundary value problems

1. Introduction

The widely used mesh based methods are finite volume method and finite element method. These methods depend on discretization of domains. It is required that sometimes irregular complex geometry should be discretized and total geometry is represented as composed of these small elements. These small elements which are parts in problem domain may not form exact given geometry. This leads to geometrical error. In the process of developing the approximate solution for the governing differential equations, the exact geometry is to be maintained in order to evaluate integration over the specified domain. The other possible error which is arised in modifying the governing differential equation into weak form, subsequently approximate solution is developed for this weak form only. Addition to these errors, mesh generation is more time consuming and costly.

Owing to the difficulty of above mesh methods in mesh generation, different types of mesh free methods are developed. Some of the widely used mesh free methods are dicussed[1–11]. The focus of the present study is concentrated on the heat conduction problems.

From above brief review on meshless method application in solving heat conduction problems, we can see that previous researchers have focused mainly on using EFG, SPH and MLPG method. However, the EFG method needs a background mesh for the integrals in the weak form, hence it is not really mesh less method; the SPH and DAM and MLWS method are built on the collocation point schemes, for which the selection of the collocation point are important, and the numerical accuracy goes down near the boundary.

A wide range of problems have been investigated by Atluri and his co-authors using MLPG method. Almost all of the previous works limited to heat conduction problems of regular domain. However, many problems in engineering are in irregular domain, and FVM and FEM are difficult to describe accurately boundaries of the irregular domain unless the mesh is very fine, or special grid generation method is adopted which is usually time-consuming. Mesh less methods can overcome this difficulty because they do not need mesh. Mesh less methods distribute arbitrarily scattering points in the problem domain, so they will have more advantages in solving problems with irregular domain than FVM and FEM. So in the present paper, we apply bi-cubic B-spline collocation method to compute the solution to Laplace equation.

Using B-spline basis functions as basis in collocation method, many inter mediatory evaluations can be avoided such as strong form differential equation to weak form conversion, evaluation of integrals which are essential for most of the existed methods and at the same time some additional information that is also required to implement such methods. In this manuscript, a methodology is developed for the solution of Laplace equations using Bi-cubic B-spline Collocation method. Before going to detailed implementation and application of the present method, definition and properties of B-spline basis function and its associated terminology is presented.

2 B-Spline Surfaces

A B-spline surface is defined as

$$U(x,y)=\sum_{i=1}^{m}\sum_{j=1}^{n}B_{i,j}\ M_{i,p}(x)\ N_{j,q}(y) \tag{1}$$

where $B_{i,j}$ are the vertices of the polygon net called control points, $M_{i,p}(x)$ is the p[th] degree B-spline basis function which is defined at the knot ‘x’ over the knot vector space KVx in X-direction and $N_{j,q}(y)$ is the q[th] degree B-spline basis function which is defined at the knot ‘y’ over the knot vector space KVy in Y-direction. The B-spline surface is also defined by rectangular array of control points. This form permits local control of curve shape. The degrees of its basis functions are defined independent of the control points.

The surface equation 2. 1 can be expressed at random point (x_r, y_s), in a matrix form as,

$$[U(x_r, y_s)]_{1\times 1} = Q_{1\times mn} * B_{mn\times 1} \qquad (2)$$

where

$$\begin{aligned} Q_{1\times mn} = [&M_{1,p}(x_r)N_{1,q}(y_s) \;\; M_{1,p}(x_r)N_{2,q}(y_s) \;\ldots\ldots\; M_{1,p}(x_r)N_{n,q}(y_s) \\ &M_{2,p}(x_r)N_{2,q}(y_s) \;\; M_{2,p}(x_r)N_{2,q}(y_s) \;\ldots\ldots M_{2,p}(x_r)N_{n,q}(y_s) \\ &\quad\vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ &M_{m,p}(x_r)N_{1,q}(y_s) \;\; M_{m,p}(x_r)N_{2,q}(y_s) \;\ldots\ldots M_{m,p}(x_r)N_{n,q}(y_s)\;)] \end{aligned}$$

$$B = [B_{1,1} \;\; B_{1.2} \ldots B_{1,n} \; B_{2,1} \; B_{2,2} \ldots B_{2,n} \ldots B_{m,1} \, B_{m,2} \ldots B_{m,n}]^T_{mn\times 1}$$

Equation (2) is the linear equation in ‘m*n’ control points.

For the given *‘mn’ control points* and chosen ‘p’ degree B-spline basis function in X-direction, ‘q’ degree B-spline basis function in Y-direction decides the knot vector in each direction. The relation among control points (m or n), degree of the basis functions and the number of knot vectors is

Number of knot vectors = number of control points + degree of the basis function+ 1

The equation (2) should be evaluated at each control point. Result of this the required surface points are obtained to generate the surface.

Suppose ‘$m\times n$’ control points are given, the equation (2) should be evaluated at ‘$m\times n$’ points with knot vectors defined based on the degree of basis[11-14].

- Some important Properties of B-spline surfaces

i) The maximum degree of the surface in each parametric direction is equal to the number of defining polygon vertices in that direction

ii) The continuity of the surface in each parametric direction is one less than the degree in each direction

3 B-Spline Collocation Method for 2D

Tensor product of B-spline basis function is used as basis function in collocation method to find the numerical solution for second order partial differential equations. Recursive form of B-spline function is employed as basis in normal collocation method. This method is developed based on the assumptions that the knot vectors are associated with the computational nodal points and constants in approximate solution are treated as control points.

Considering the second order partial differential equation of the temperature distribution $U(x, y)$ over the region $a \le x \le b$, $c \le y \le d$

$$\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}= f(x,y) \quad , \qquad (3)$$

with the boundary conditions

$$\left\langle \begin{array}{ll} U(x,c) = g_1(x), & a \le x \le b \\ U(x,d) = g_2(x), & a \le x \le b \\ U(a,y) = h_1(y), & c \le y \le d \\ \text{and} & \\ U(b,y) = h_2(y), & c \le y \le d \end{array} \right\rangle \qquad (3a)$$

where a,b,c,d are constants $,g_1(x),g_2(x)$ *are functions in x and*

$h_1(y),h_2(y)$ *are functions in y.*

Let $$U^h(x,y)=\sum_{i=-3}^{m-1}\sum_{j=-3}^{n-1} B_{i,j}\; M_{i,p}(x)\; N_{j,q}(y) \qquad (4)$$

where

$B_{i,j}$'*s are control points*

$M_{i,p}$ *is the pth degree* **B** – *spline basis function in the X – direction*

and

$N_{j,q}(y)$ *qth degree B – spline basis function in the Y – direction* be the approximate solution of heat distribution function $U(x,y)$ for which the governing partial differential equation (3). The function $U(x,y)$ is the temperature distribution over the considered computational domain.

Particularly, the approximate solution is assumed based on the third degree B-spline basis function that is employed in the collocation method. In order to have the B-spline property which is partition of unity, three additional knots should be taken both sides of knot vector space

Let the computational domain be $a \le x \le b, c \le y \le d$, the nodes in X – *direction* are $\{a = x_1, x_2, x_3, \ldots, x_{m-1}, x_m = b\}$ and the nodes in Y – *direction* are $\{c = y_1, y_2, y_3, \ldots, y_{n-1}, y_n = d\}$.

Assuming that the knot vectors in each direction are nodes in each direction respectively and control points are treated as constants or unknowns in equation (4)

The first and second order partial derivatives of approximate function with respect to x are derived by differentiating the component function $M_{i,p}(x)$ with respect to x. i. e.

$$\frac{\partial U^h(x,y)}{\partial x} = \sum_{i=-3}^{m-1}\sum_{j=-3}^{n-1} B_{i,j} \frac{\partial M_{i,p}(x)}{\partial x} N_{j,q}(y) \tag{5}$$

$$\frac{\partial^2 U^h(x,y)}{\partial x^2} = \sum_{i=-3}^{m-1}\sum_{j=-3}^{n-1} B_{i,j} \frac{\partial^2 M_{i,p}(x)}{\partial x^2} N_{j,q}(y) \tag{6}$$

Similarly, the first and second order partial derivatives of approximate solution with respect to y is given as

$$\frac{\partial U^h(x,y)}{\partial y} = \sum_{i=-3}^{m-1}\sum_{j=-3}^{n-1} B_{i,j} M_{i,p}(x) \frac{\partial N_{j,q}(y)}{\partial y} \tag{7}$$

$$\frac{\partial^2 U^h(x,y)}{\partial y^2} = \sum_{i=-3}^{m-1}\sum_{j=-3}^{n-1} B_{i,j} M_{i,p}(x) \frac{\partial^2 N_{j,q}(y)}{\partial y^2} \tag{8}$$

Similarly,

$$\frac{\partial^2 U^h(x,y)}{\partial y \partial x} = \sum_{i=-3}^{m-1}\sum_{j=-3}^{n-1} B_{i,j} \frac{\partial M_{i,p}(x)}{\partial x} \frac{\partial N_{j,q}(y)}{\partial y} \tag{9}$$

4 Collocation Method

Collocation method is a numerical technique. It is used to establish the relations among control points which are used to express the linear combination of the base functions. This method converts assumed approximate solution in the form of system of linear equations and become a powerful tool in developing various approximate methods because of its point based and discrete nature. Residue is the difference between the exact solution and the approximate solution. The residue is made to zero at some discrete nodal values in order to get the constraints among the control points. General working procedure of collocation method.

If B-spline functions are used as basises in approximate solution and the collocation procedure is followed to obtain the system of linear equations in control points of approximation solution then this method is known as B-spline collocation method.

Substituting the equations (4)-(9) in governing differential equation (4. 3). Then we have

$$\sum_{i=-3}^{m-1}\sum_{j=-3}^{n-1} B_{i,j} \frac{\partial^2 M_{i,p}(x)}{\partial x^2} N_{j,q}(y) + \sum_{i=-3}^{m-1}\sum_{j=-3}^{n-1} B_{i,j} M_{i,p}(x) \frac{\partial^2 N_{j,q}(y)}{\partial y^2} = f(x,y) \tag{10}$$

Let (x_r, y_s) be the any nodal point in the computational domain and expanding the equation (10) then we have

$$\sum_{i=-3}^{m-1} \frac{\partial^2 M_{i,p}(x_r)}{\partial x^2} \left[B_{i,-3} N_{-3,q}(y_s) + B_{i,-2} N_{-2,q}(y_s) + B_{i,-1} N_{-1,q}(y_s) + \cdots + B_{i,n-1} N_{n-1,q}(y_s) \right]$$

$$+ \sum_{i=-3}^{m-1} M_{i,p}(x_r) \left[B_{i,-3} \frac{\partial^2 N_{-3,q}(y_s)}{\partial y^2} + B_{i,-2} \frac{\partial^2 N_{-2,q}(y_s)}{\partial y^2} + B_{i,-1} \frac{\partial^2 N_{-1,q}(y_s)}{\partial y^2} + \cdots + B_{i,n-1} \frac{\partial^2 N_{n-1,q}(y_s)}{\partial y^2} \right] = f(x_r, y_s)$$

Expressing the above formulation in matrix form,

$$[A_{cd}] * [B] = f(x_r, y_s) \tag{11}$$

$$[A_{cd}] = \begin{bmatrix} A'_{-3} & A'_{-2} & A'_{-1} & \cdot & \cdot & \cdot & A'_{(m-1)} \end{bmatrix}_{1\times(m+2)*(n+2)}$$

$$A_{-3} = \begin{bmatrix} \frac{\partial^2 M_{-3,p}(x_r)}{\partial x^2} N_{-3,q}(y_s) + M_{-3,p}(x_r) \frac{\partial^2 N_{-3,q}(y_s)}{\partial y^2} \\ \frac{\partial^2 M_{-3,p}(x_r)}{\partial x^2} N_{-2,q}(y_s) + M_{-3,p}(x_r) \frac{\partial^2 N_{-2,q}(y_s)}{\partial y^2} \\ \frac{\partial^2 M_{-3,p}(x_r)}{\partial x^2} N_{-1,q}(y_s) + M_{-3,p}(x_r) \frac{\partial^2 N_{-1,q}(y_s)}{\partial y^2} \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ \frac{\partial^2 M_{-3,p}(x_r)}{\partial x^2} N_{n-1,q}(y_s) + M_{-3,p}(x_r) \frac{\partial^2 N_{n-1,q}(y_s)}{\partial y^2} \end{bmatrix}^T_{(n+2)}$$

$$A_{-2} = \begin{bmatrix} \frac{\partial^2 M_{-2,p}(x_r)}{\partial x^2} N_{-3,q}(y_s) + M_{-2,p}(x_r) \frac{\partial^2 N_{-3,q}(y_s)}{\partial y^2} \\ \frac{\partial^2 M_{-2,p}(x_r)}{\partial x^2} N_{-2,q}(y_s) + M_{-2,p}(x_r) \frac{\partial^2 N_{-2,q}(y_s)}{\partial y^2} \\ \frac{\partial^2 M_{-2,p}(x_r)}{\partial x^2} N_{-1,q}(y_s) + M_{-2,p}(x_r) \frac{\partial^2 N_{-1,q}(y_s)}{\partial y^2} \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ \frac{\partial^2 M_{-2,p}(x_r)}{\partial x^2} N_{n-1,q}(y_s) + M_{-2,p}(x_r) \frac{\partial^2 N_{n-1,q}(y_s)}{\partial y^2} \end{bmatrix}^T_{1\times(n+2)}$$

$$B=\begin{bmatrix} B_{-3} \\ B_{-2} \\ B_{-1} \\ . \\ . \\ . \\ B_{(m-1)} \end{bmatrix}_{(m+2)*(n+2)\times 1}$$

where $B_{-3}=\begin{bmatrix} B_{-3,-3} \\ B_{-3,-2} \\ B_{-3,-1} \\ . \\ . \\ . \\ B_{-3,(n-1)} \end{bmatrix}_{(n+2)\times 1}$ $B_{-2}=\begin{bmatrix} B_{-2,-3} \\ B_{-2,-2} \\ B_{-2,-1} \\ . \\ . \\ . \\ B_{-2,(n-1)} \end{bmatrix}_{(n+2)\times 1}$ $B_{-1}=\begin{bmatrix} B_{-1,-3} \\ B_{-1,-2} \\ B_{-1,-1} \\ . \\ . \\ . \\ B_{-1,(n-1)} \end{bmatrix}_{(n+2)\times 1}$ $B_{(m-1)}=$

$$\begin{bmatrix} B_{(m-1),-3} \\ B_{(m-1),-2} \\ B_{(m-1),-1} \\ . \\ . \\ . \\ B_{(m-1),(n-1)} \end{bmatrix}_{(n+2)\times 1}$$

$$\left[A_{cd}\right] * \left[B\right] = \left[f(x_r, y_s)\right] \tag{12}$$

$$A_{-1}=\begin{bmatrix} \frac{\partial^2 M_{-1,p}(x_r)}{\partial x^2} N_{-3,q}(y_s) + M_{-1,p}(x_r)\frac{\partial^2 N_{-3,q}(y_s)}{\partial y^2} \\ \frac{\partial^2 M_{-1,p}(x_r)}{\partial x^2} N_{-2,q}(y_s) + M_{-1,p}(x_r)\frac{\partial^2 N_{-2,q}(y_s)}{\partial y^2} \\ \frac{\partial^2 M_{-1,p}(x_r)}{\partial x^2} N_{-1,q}(y_s) + M_{-1,p}(x_r)\frac{\partial^2 N_{-1,q}(y_s)}{\partial y^2} \\ . \quad . \quad . \\ . \quad . \quad . \\ . \quad . \quad . \\ \frac{\partial^2 M_{-1,p}(x_r)}{\partial x^2} N_{n-1,q}(y_s) + M_{-1,p}(x_r)\frac{\partial^2 N_{n-1,q}(y_s)}{\partial y^2} \end{bmatrix}^T_{1\times(n+2)}$$

$$A_{m-1} = \begin{bmatrix} \frac{\partial^2 M_{(m-1),p}(x_r)}{\partial x^2} N_{-3,q}(y_s) + M_{(m-1),p}(x_r) \frac{\partial^2 N_{-3,q}(y_s)}{\partial y^2} \\ \frac{\partial^2 M_{(m-1),p}(x_r)}{\partial x^2} N_{-2,q}(y_s) + M_{(m-1),p}(x_r) \frac{\partial^2 N_{-2,q}(y_s)}{\partial y^2} \\ \frac{\partial^2 M_{(m-1),p}(x_r)}{\partial x^2} N_{-1,q}(y_s) + M_{(m-1),p}(x_r) \frac{\partial^2 N_{-1,q}(y_s)}{\partial y^2} \\ \vdots \\ \frac{\partial^2 M_{(m-1),p}(x_r)}{\partial x^2} N_{n-1,q}(y_s) + M_{(m-1),p}(x_r) \frac{\partial^2 N_{n-1,q}(y_s)}{\partial y^2} \end{bmatrix}^T_{1\times(n+2)}$$

Equation (12) is the matrix form of equation (10) at single computational domain node point (x_r, y_s). It is the equation in $(m+2)\times(n+2)$ control points which is the result of assumption that the approximate solution (4) satisfies the governing partial differential equation (3) at the computational domain point (x_r, y_s). The computational domain consists (m)*(n) node points. The equation (12) should be evaluated at each these computational domain node and assumed that these node points are collocation points also. Then the equation (12) becomes system of (m)*(n) linear equations in (m+3)*(n+3) control points which are unknowns.

The Matrix form of above system of (m)*(n) linear equations is given below as

$$\left[A_{cd}\right]_{m*n\times(m+2)*(n+2)} * \left[B\right]_{(m+2)*(n+2)\times 1} = \left[f(x_r, y_s)\right]_{m*n\times 1} \quad (13)$$

The Matrix A_{cd} is not square matrix and yet Boundary conditions are not applied to approximate solution.

Applying Boundary conditions to approximate solution (eq. 4)

The total number of nodal points on boundary is $(2m+2n-4)$ and the assumption is that the approximate solution satisfies the boundary conditions then applying equation (4) for given boundary conditions (3a), we have

Along the boundary points, we have

$$U(x,y) = U^h(x,y) = \sum_{i=-3}^{m-1} \sum_{j=-3}^{n-1} B_{i,j}\, M_{i,p}(x)\, N_{j,q}(y)$$

Applying all the boundary conditions to approximate solution, we get

i) Taking the boundary along y=c, we have

$$y = c, U^h(x, y) = g_1(x)$$

$$\sum_{i=-3}^{m-1}\sum_{j=-3}^{n-1} B_{i,j}\ M_{i,p}(x)\ N_{j,q}(c) = g_1(x)$$

$$\sum_{i=-3}^{m-1} M_{i,p}(x)\begin{bmatrix} B_{i,-3}N_{-3,q}(c) + B_{i,-2}N_{-2,q}(c) + B_{i,-1}N_{-1,q}(c) + \ .\ .\ . \\ + B_{i,n-1}N_{n-1,q}(c) \end{bmatrix} = g_1(x)$$

Expressing the above equation into the matrix form, we have

$$A_b * B = g_1(x) \tag{14}$$

where

$$A_b = \begin{bmatrix} M_{-3,p}(x)N_{-3,q}(c) \\ M_{-3,p}(x)N_{-2,q}(c) \\ M_{-3,p}(x)N_{-1,q}(c) \\ . \quad . \quad . \\ . \quad . \quad . \\ . \quad . \quad . \\ M_{-3,p}(x)N_{(n-1),q}(c) \\ M_{-2,p}(x)N_{-3,q}(c) \\ M_{-2,p}(x)N_{-2,q}(c) \\ M_{-2,p}(x)N_{-1,q}(c) \\ .... \\ M_{-2,p}(x)N_{(n-1),q}(c) \\ M_{-1,p}(x)N_{-3,q}(c) \\ M_{-1,p}(x)N_{-2,q}(c) \\ M_{-1,p}(x)N_{-1,q}(c) \\ .... \\ M_{-1,p}(x)N_{(n-1),q}(c) \\ M_{(m-1),p}(x)N_{-3,q}(c) \\ M_{(m-1),p}(x)N_{-2,q}(c) \\ .... \\ M_{(m-1),p}(x)N_{(n-1),q}(c) \end{bmatrix}^T_{1\times(m+2)*(n+2)}$$

The matrix form (14) is extended to all nodal points along the line y=c then the system of m-linear equations are obtained in (m+3)*(n+3) control points and its matrix form is given as

$$(A_b)_{m\times(m+2)*(n+2)} * B = [g_1(x)]_{m\times 1} \tag{15}$$

where ‘x’ takes the m-nodes along the line y=c;
Similarly along the remaining boundary lines, we have
Along the above line i. e y=d

$$(A_a)_{m\times(m+2)*(n+2)} * B = [g_2(x)]_{m\times 1}. \tag{16}$$

where x takes above boundary line nodes.
Along the left boundary nodal point

$$(A_l)_{n\times(m+2)*(n+2)} * B = [h_1(y)]_{n\times 1} \tag{17}$$

Along the right boundary nodal points, we have the linear system of n-equations, so we have

$$(A_r)_{n\times(m+2)*(n+2)} * B = [h2(y)]_{n\times 1} \tag{18}$$

Assembling all the systems of linear equations which are generated over the computational domain boundary points i. e combining all the equations (15) (16), (17) and (18), then we have

$$\begin{bmatrix} A_b \\ A_a \\ A_l \\ A_r \end{bmatrix}_{si} * \left[B\right]_{(m+3)*(n+3)\times 1} = \begin{bmatrix} g_1(x) \\ g_2(x) \\ h_1(y) \\ h_2(y) \end{bmatrix}_{(2m+2n-4)\times 1} \quad (19)$$

where $si = (2m + 2n - 4) \times (m + 2) * (n + 2)$

Assembling all the matrices in order to form the global matrix, they are (13) and (19)

$\left[A_{cd}\right]_{m*n\times(m+2)*(n+2)} * \left[B\right]_{(m+3)*(n+3)\times 1} = \left[f(x_r, y_s)\right]_{m*n\times 1}$ and

$$\begin{bmatrix} A_b \\ A_a \\ A_l \\ A_r \end{bmatrix}_{si} * \left[B\right]_{(m+3)*(n+3)\times 1} = \begin{bmatrix} g_1(x) \\ g_2(x) \\ h_1(y) \\ h_2(y) \end{bmatrix}_{(2m+2n-4)\times 1}$$

where $si = (2m + 2n - 4) \times (m + 2) * (n + 2)$

i. e. The number of equations generated in the computational domain are m*n and the number of equations obtained by using boundary conditions are 2m+2n but corner nodes are used two times therefore deducting these repetitions then we have 2m+2n-4 boundary conditions are only included in constructing the global matrix. The repeated evaluations at corner nodes are neglected

$$\begin{bmatrix} A_{cd} \\ A_b \\ A_a \\ A_l \\ A_r \end{bmatrix}_{(m*n+2m+2n-4)\times(m+2)*(n+2)} * \left[B\right]_{(m+2)*(n+2)\times 1} = \begin{bmatrix} f(x_r, y_s) \\ g_1(x) \\ g_2(x) \\ h_1(y) \\ h_2(y) \end{bmatrix}_{(m*n+2m+2n-4)\times 1}$$

This can be written in the simplified matrix form, we have

$$\left[A\right] * \left[B\right] = \left[C\right] \quad (20)$$

where

$$A = \begin{bmatrix} A_{cd} \\ A_b \\ A_a \\ A_l \\ A_r \end{bmatrix}_{(m*n+2m+2n-4)\times(m+3)*(n+3)}$$

$$C = \begin{bmatrix} f(x_r, y_s) \\ g_1(x) \\ g_2(x) \\ h_1(y) \\ h_2(y) \end{bmatrix}_{(m*n+2m+2n-4)\times 1} \qquad \bar{B} = \begin{bmatrix} B_{-3,-3} & B_{-3,-2} & B_{-3,-1} & \cdot & \cdot & \cdot & B_{-3,n-1} \\ B_{-2,-3} & B_{-2,-2} & B_{-2,-1} & \cdot & \cdot & \cdot & B_{-2,n-1} \\ B_{-1,-3} & B_{-1,-2} & B_{-1,-1} & \cdot & \cdot & \cdot & B_{-3,-3} \\ & \cdot & & \cdot & & \cdot & \\ & \cdot & & \cdot & & \cdot & \\ & \cdot & & \cdot & & \cdot & \\ B_{m-1,-3} & B_{m-1,-2} & B_{m-1,-1} & \cdot & \cdot & \cdot & B_{m-1,n-1} \end{bmatrix}_{(m+3)\times(n+3)}$$

Solve the equation(20) for unknown constants (control points). Replacing these values with unknowns in equation (4) then the approximation solution becomes known approximate solution to the equation (3).
The whole assembly of matrices and its solution is obtained by coding in Matlab as based on the implementation procedure given below.

5 Numerical Example

In this section, a two dimensional numerical experiments are considered as in the part of testing of applicability of bi-cubic B-spline collocation method to various kinds of boundary value problems.
Laplace equation with the non–zero boundary conditions is considered under the numerical experiment and compared the obtained solution with the well established numerical technique Finite Element Method by taking as 80 elements (i. e. 81×81 partitions). This problem is solved by the present method considering the 3 sets of control points 5 X 5 (25 control points), 5 X 9 and 5 X13 where the first digit represents the number of control points in X-direction and the second digit represents the number of control points in Y-direction. The results are presented in table 1.

Governing differential equation for the temperature distribution $U(x, y)$ *is*

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \qquad\qquad 0 \le x, y \le 1$$

with the boundary Conditions

$$U(0, y) = 25^\circ C\ ; U(x,0) = 100^\circ C; \ U(1, y) = 50^\circ C; \ U(x,1) = 75^\circ C;$$

Given computational domain [0 1]×[0, 1]and boundary conditions are shown in below Figure 1

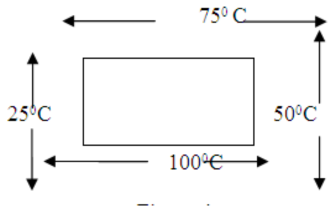

Figure 1

Case (i) : Partition of the domain for 5×5

Dividing each side 5 uniform parts of the interval [0, 1] gives the total 25 collocation points for the square computational domain. A third degree B-spline basis function is employed in collocation method to obtain the numerical solution for the numerical example. Knot vectors associates the nodal points in each direction. So, the knot vectors in X-direction are KVx = {0 0. 25 0. 5 0. 75 1} and the knot vectors in Y-direction KVy = {0 0. 25 0. 5 0. 75 1}. Three additional knot vectors are added both side of the knot vector space for both directions in order to maintain the partition of unity property of B-spline basis function. These additional knots are considered only to find the weights of knots at inside the knots in computational domain. These knots are not treated as collocation points because these knots are outside of the domain. The equation (4) is approximate solution with the assumption that the cubic degree B-spline is used as the basis function in collocation method. The computational domain for 5×5 partition has 25 nodes which are treated as knots to find B-spline base function.

The nodes in the computational domain are taken as the collocation points and the assumption that the approximate solution satisfies governing differential equation at these collocation points. This gives the system of 25-linear equations in control points (unknowns). 16 nodes are boundary points. The boundary conditions are given for the given governing differential equation. The approximate solution is the solution to the governing differential equation as the assumption in collocation method. Therfore, the equation (4) should satisfy the boundary conditions also.

Based on the implantation procedure which is given below for this numerical example is implemented in Matlab

- Implementation procedure of the method is given below

1. Assumption of approximate solution as the Cartesian product of B-spline basis function in each direction (eq. 4)
2. Substuiting the approximation solution (eq. 4) in governing differential equation (eq. 3)
3. System of linear equations is developed (eq. 13)
4. Imposing the boundary conditions (eq. 15, eq. 16, eq. 17 & eq. 18)
5. Assembling all the equations (eq. 13, eq. 15, eq. 16, eq. 17 &eq. 18)
6. Solve (eq. 20) for control points [B]
7. Substitute these control points (constants in eq. 4) in eq. 4

Table 1: presents the solution of B-spline collocation and Finite element solution

nodes	B-spline collocation solution			Finite element method
	5×5	5×9	5×13	81×81
(. 25,. 25)	62. 9287	61. 6276	61. 4712	62. 489
(. 25,. 5)	49. 7762	49. 7038	49. 6782	50. 131
(. 25,. 75)	53. 6773	52. 8259	52. 5859	53. 391
(. 5,. 25)	76. 9615	76. 1062	75. 8851	74. 859
(. 5,. 5)	64. 1916	63. 5460	63. 4533	62. 496
(. 5,. 75)	65. 4507	64. 9777	64. 8280	63. 732

(. 75,. 25)	73. 7777	72. 6618	72. 5283	71. 594
(. 75,. 5)	65. 0175	64. 7437	64. 6865	61. 263
(. 75,. 75)	65. 0000	65. 2114	65. 1246	62. 502

Table 2: Comparison of absolute relative errors

	Nodes	(.25,.25)	(.25,.5)	(.25,.75)	(.5,.25)	(.5,.5)	(.5,.75)	(.75,.25)	(.75,.5)	(.75,.75)
Absolute relative errors	5×5	0.0070	0.0071	0.0054	0.0281	0.0271	0.0270	0.0305	0.0613	0.0400
	5×9	0.0138	0.0085	0.0106	0.0167	0.0168	0.0195	0.0149	0.0568	0.0433
	5×13	0.0163	0.0090	0.0151	0.0137	0.0153	0.0172	0.0130	0.0559	0.0420

Some of the points of domain are evaluated by the present method and are shown in the above Table 1 and values at these nodes calculated by the Finite Element solution are included for the purpose of comparison and to calculate relative error. The present method performance over the domain schematically shown in Figure 2 by using the contours for this 5×5 partitions

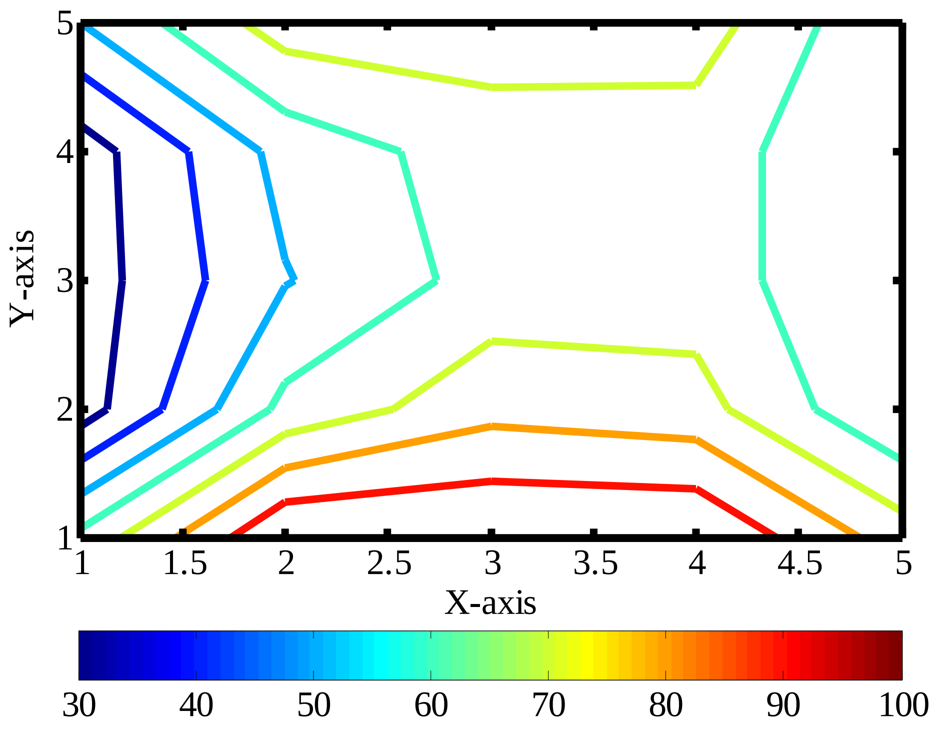

Figure 2: Bi-cubic B-spline collocation solutions for the 5×5 partition

The contours in the figure 2 shows the temperature distribution of the computational domain which is having the boundary temperature as shown in figure 1. The figure 2

reflects the temperature clearly, i. e. the contour which is close to the X-axis varies between the temperature 90^{0C} to 100^{0} C along the boundary provided temperature 100^{0} C. It is also observed from the figure 2 that the temperature is changing from100^{0}C to 75^{0}C vertically whereas horizontally varying the temperature from 25^{0}C to 50^{0}C.
To improve the smoothness of the contours of temperature distribution, the number of partitions of computational domain is increased from 5×5 to 5×9 which is discussed in the case (ii).

Case (ii): Partition of the computational domain for 5×9
Number of collocation points is increased in case (ii) to improve the accuracy of the numerical solution. More divisions are taken then in case (i). Total 45collcations points are obtained by doing the 5×9 partitions. Table 1 present's present method solution at the same points as calculated earlier in table1. FEM solution and absolute relative error is also given at these points in Table2. Present solution at all the collocation points is illustrated by contour graph in Figure 2

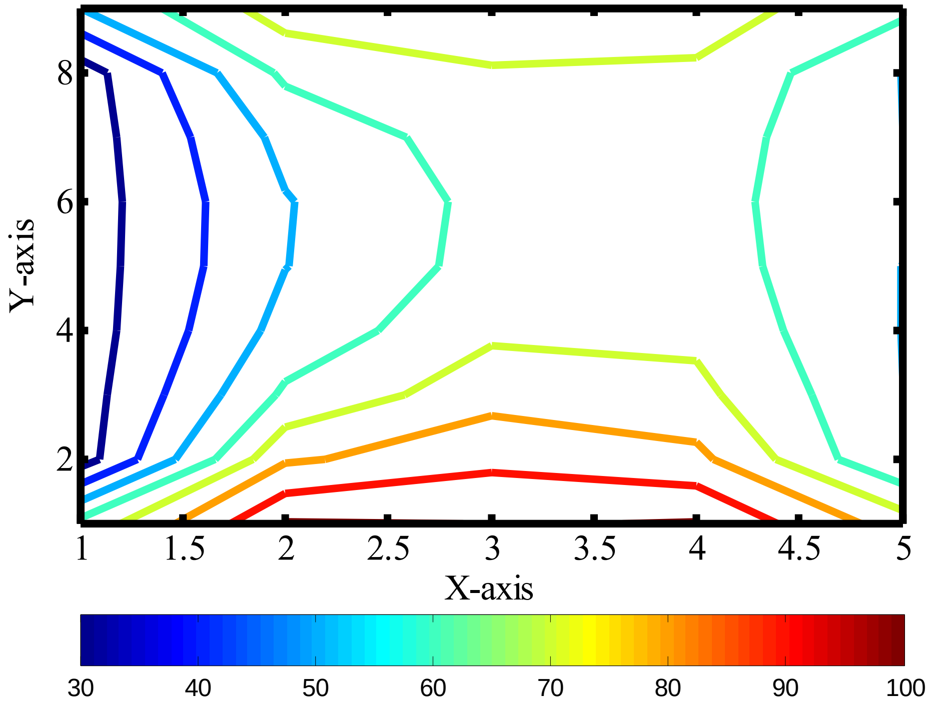

Figure 3: Bi-cubic B-spline collocation solution for the 5×9 partition

The smoothness of the temperature distribution contours for the partition 5×9 is improved when compared with the contours which are generated by the present solution for the domain 5×5. This is shown in the figure 3. This can be observed throughout domain. Further, the domain is made into more number of partitions in order to get the more smooth contours and test the convergence of the present B-spline collocation method which is studied in case(iii).

Case (iii)

Number of collocation points is increased in case (iii) to test the convergency of the numerical solution. More divisions are taken then in case (ii). Total 65collcations points are obtained by doing the 5×13 partitions. Table 1 presents present method solution at the same points as calculated earlier. FEM solution and absolute relative error is also given at these points in Table2. Present solution at all the collocation points is illustrated by contour graph in Figure 4

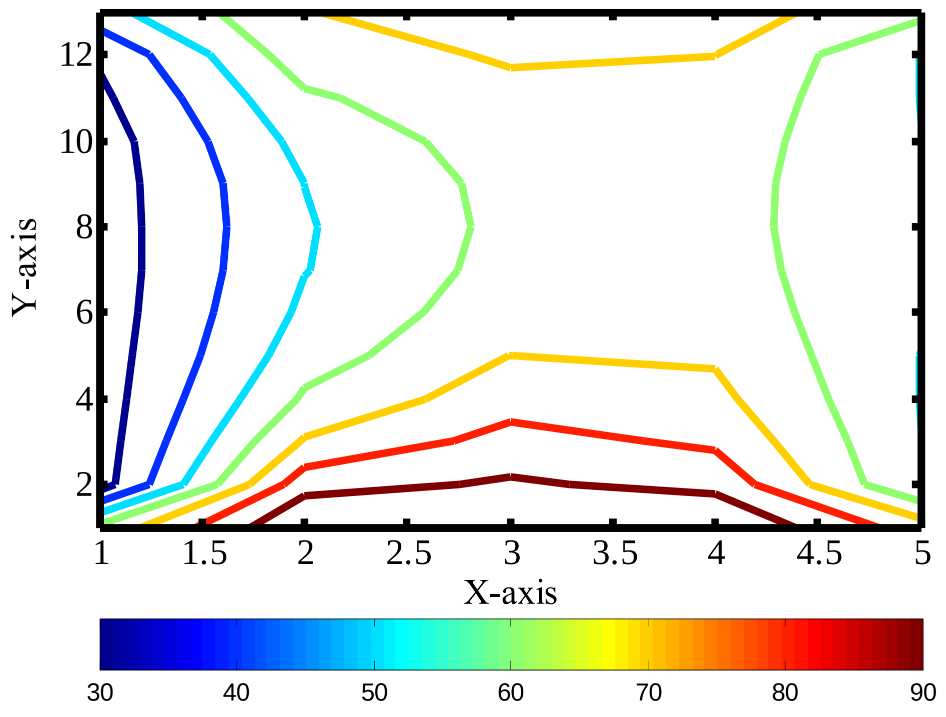

Figure 4: Bi-cubic B-spline collocation solution for the 5×13 partition

6 Results and Discussion

Laplace 2-D heat conduction problem with the boundary conditions is illustrated to demonstrate the present method. Tested the method by changing the number of collocation points. Intially, computational domain is made into 5×5 partitions and estimated the temperature at various nodes in the computational domain. These estimated values are shown in Table1. The temperature at mid-point (. 5,. 5) is 64.

1916. When increased the number of partitions of computational domain 5×5 to 5×9 and then to5×13, it is observed that temperature at the mid (. 5,. 5) is decreased from 64. 1916 to 63. 5460 and then to 63. 4533.

We can see that the average absolute relative error is constantly decreasing as the number of collocation points are increased which is graphically shown in Figure 5. Consequence of these results, we can say that the present method is convergent. Figures 2, Figures 3 and Figure 4 presents the performance of present numerical method throughout the computational domain. Also it is observed from the Figures that smoothness is improved as the numbers of collocation points are increased.

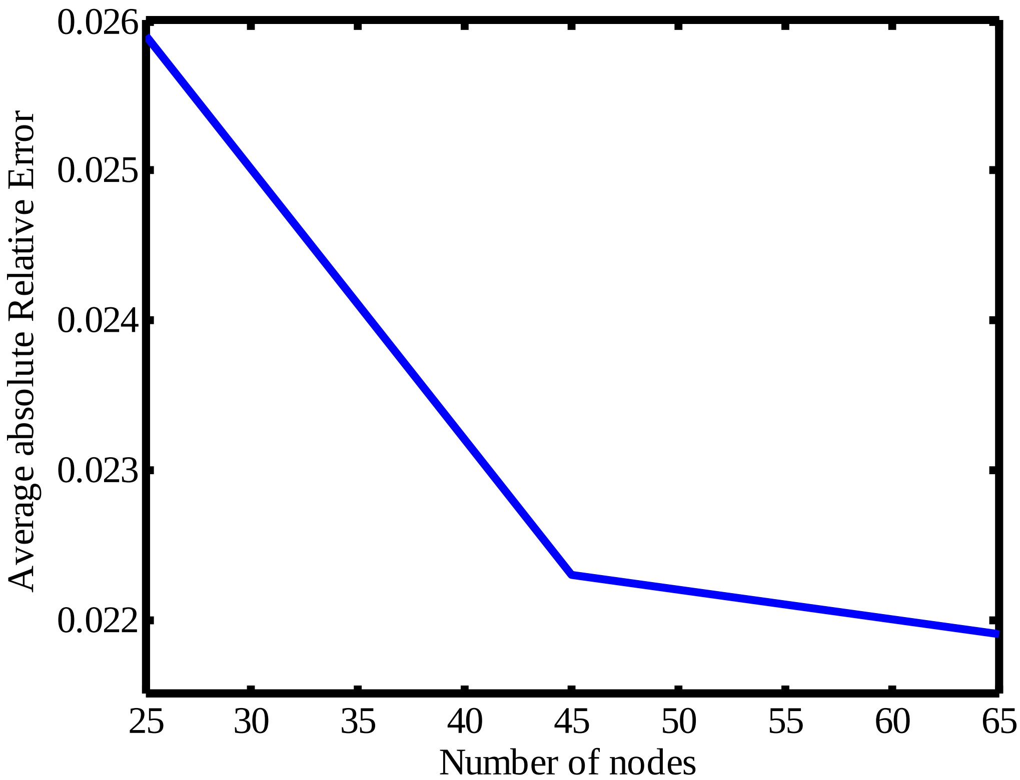

Figure 5 Compares the Average Absolute Relative Error and Number of nodes

7 Conclusions

Recursive form of Bi-cubic B-spline collocation method is developed and applied for the two dimensional Poisson's equation with temperature as the field variable. Dirchlet's forms of boundary conditions are considered for the approximate solution. The results obtained by using the bi-cubic B-spline collocation method are good agreement with the finite element solution.

References

[1] LiuG. R., M. B. Liu, Smoothed Particle Hydrodynamics: A Meshfree Particle Method, World Scientific Publishing Co. Pte. Ltd., Singapore, 2003

[2] Lucy L. B., A numerical approach to testing of the fission hypothesis, Astron. J. 8 (1977) 1013–1024.

[3] Monaghan J. J., Smoothed particle hydrodynamics, Annu. Rev. Astron. Astrophys 30 (1992) 543–574

[4] Belytschko. T, Y. Y. Lu, L. Gu, Element-free Galerkin methods, Int. J. Numer. Methods Eng. 37 (1994) 229–256.

[5] Onate. E, S. Idelsohn, O. Zienkiewicz, R. L. Taylor, A finite point method in computational mechanics application to convective transport and fluid flow, Int. J. Numer. Methods Eng. 39 (1995) 3839–3866.

[6] Atluri S. N., T. Zhu, A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics, Comput. Mech. 24 (1998) 348–372.

[7] Atluri S. N., T. Zhu, A new meshless local Petrov–Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation, Comput. Model. Simulat. Eng. 3 (1998) 196–197.

[8] Atluri S. N., S. P. Shen, The Meshless Local Petrov–Galerkin (MLPG) Method, Tech Science Press, Encino USA, 2002.

[9] Sadat. H, S. Couturier, Performance and accuracy of a meshless method for laminar natural convection, Numer. Heat Transfer, Part B 37 (2000) 455–467.

[10] Schere C. E, E. Kaszkuewicz and N. Mangiav acchi, A multilevel Schwarz shooting for the solution of the Poisson equation two dimensional incompressible flow simulations, Appl. Math. and Comput., 153(2004) 803-831.

[11] Carl De Boor On Calculating with B-plines. JOURNAL OF APPROXIMATION THEORY6, SO-62 (1972)

[12] Curry H. B and Schoenberg I. J., On spline distributions and their limits: the Polya distributions, Abstr. Bull. Amer. Math. Sot. 53 (1947), 1114

[13] Curry H. B. and I. J. Schoenberg, On Polya frequency functions IV: The fundamental spline functions and their limits, J. Anal. Math. 17 (1966), 71-107.

[14] CURRY. H. B AND I. J. SCHOENBERG On Polya frequency functions IV: The fundamental spline functions and their limits, J. Anal. Math. 17 (1966), 71-107.

[15] David F. Rogers and J. Alan Adams, “Mathematical Elements for Computer Graphics”, 2nd ed., Tata McGraw-Hill Edition, New Delh**.**