Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2021.012575

ARTICLE

A Meshless Collocation Method with Barycentric Lagrange Interpolation for Solving the Helmholtz Equation

Miaomiao Yang, Wentao Ma and Yongbin Ge[*]

Institute of Applied Mathematics and Mechanics, Ningxia University, Yinchuan, 750021, China

[*]Corresponding Author: Yongbin Ge. Email: gyb@nxu.edu.cn

Received: 05 July 2020 Accepted: 21 September 2020

ABSTRACT

In this paper, Chebyshev interpolation nodes and barycentric Lagrange interpolation basis function are used to deduce the scheme for solving the Helmholtz equation. First of all, the interpolation basis function is applied to treat the spatial variables and their partial derivatives, and the collocation method for solving the second order differential equations is established. Secondly, the differential matrix is used to simplify the given differential equations on a given test node. Finally, based on three kinds of test nodes, numerical experiments show that the present scheme can not only calculate the high wave numbers problems, but also calculate the variable wave numbers problems. In addition, the algorithm has the advantages of high calculation accuracy, good numerical stability and less time consuming.

KEYWORDS

Helmholtz equation; Chebyshev interpolation nodes; Barycentric Lagrange interpolation; meshless collocation method; high wave number; variable wave number

1 Introduction

In the present work, we consider the following two-dimensional Helmholtz equation:

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+k^2(x,y)\,u(x,y)=f(x,y),\quad (x,y)\in\Omega \tag{1.1}$$

where $\Omega=[a,b]\times[c,d]$ is the problem region, $u(x,y)$ is an unknown function of x and y. $k(x,y)$ is the wave number, which could be either a constant or a function of x and y. $f(x,y)$ is a source term. The Dirichlet boundary conditions are given as:

$$\begin{cases} u(a,y)=g_1(y),\ u(b,y)=g_2(y), & y\in[c,d] \\ u(x,c)=g_3(x),\ u(x,d)=g_4(x), & x\in[a,b] \end{cases} \tag{1.2}$$

Helmholtz equation is an elliptical partial differential equation, which represents the solution of the wave equation independent of time and it arises from time-harmonic wave propagation [1]. This equation can simulate a variety of physical phenomena, including vibration analysis, water wave propagation, electromagnetic scattering, acoustic scattering, and radar scattering, etc. [2–6]. There is a long history about the development of wave propagation [7]. When the wavenumber is large, the solution of Helmholtz equation is highly oscillatory, so there are some difficulties in obtaining the numerical solution. Meanwhile, because of the complexity of variable wave number Helmholtz equation, it also brings great difficulties to numerical calculation, which still need to be further studied. Therefore, the research on numerical solution method of Helmholtz equation has important theoretical value and practical significance. A variety of numerical methods have been developed to solve the Helmholtz equation, such as the finite element method, boundary element method, finite difference method, meshless method and so on [8–11]. Among them, the finite element method is the most widely used method, but the calculation accuracy of the finite element method will sharply reduce with the increase of the wave number of the equation. At the same time, the meshless method is favored by many researchers because of its many advantages, such as simple discretization, easy derivation of format, easy realization of calculation program, high calculation accuracy and good numerical stability.

A new boundary method with method of external sources for eigenproblems was presented by Reutskiy [12] to solve Helmholtz equation in simply and multiply connected domains. However, this method depends on the grid, so when it applied to eigenproblem in irregular region, meshless technique is needed. Chen et al. [13] put forward a first-order system least square method for solving high wave number problems, and then a nontrivial decomposition is used to solve the first-order system least square problem, which was a new method different from the standard finite element method. Dogan et al. [14] studied the dispersion error of the Helmholtz equation by using the local meshless boundary integral equation method and the radial basis integral equation method. Both methods used the second-order polynomial and the frequency dependent radial basis function to interpolate the potential field, which can solve the high wave number problems. But the cost is to solve a larger linear system, which is computationally intensive. Based on the meshless and boundless analysis of Helmholtz equation, Chen et al. [15] proposed a boundless Burton Miller method by using Burton Miller formula. This method can find the unique solution for all wave numbers, and can also deal with Helmholtz equation under the high wave numbers. However, the cost of calculation is expensive and the derivation process is complicated. Qu et al. [16] developed a local basic solution method which was different from the traditional basic solution method. According to the nodes, the calculation domain was divided into several overlapping subdomains. Finally, the sparse strip matrix was used to solve large-scale problems. Whereas, the numerical implementation is complicated. By using an asymptotic expansion that removes the singularities up to several leading orders, Britt et al. [17] put forward a method that can keep high order convergence even if there is singular solution for Helmholtz equation. Whereas, it did not give out the results in the case of large wave number. A circular boundary knot method was studied by Lin et al. [18] by placing the collocation nodes on the boundary surface in a circular form to solve the high wave number Helmholtz equation, the BKM matrix obtained has the block structure of cyclic matrix and can be solved effectively. Which needs to solve a lot of small matrices, the calculation storage capacity is large. Wang et al. [19] considered the Helmholtz equation in the outer domain, and proposed a finite difference scheme in polar or spherical coordinates, which can solve high wave number problems. Whereas, the "pollution effect" in the domain of the origin of coordinates is inevitable. In addition, the common disadvantage of the above literature is that there is few mention of the case of variable wave number problem.

In this paper, based on the Chebyshev nodes, the scheme for solving the Helmholtz equation is derived by using barycentric Lagrange interpolation basis function. This method is simple in theory, needs few interpolation nodes, and the final discrete matrix is easy to be dealt with. It can not only calculate the high wave number problems, but also calculate the variable wave number problems. It has the advantages including short calculation time, high accuracy and good stability. In this meshless method, the point collocation technique is used for forming the final system of linear algebraic equations. Thus, the present method is a variant of the well-known finite point method [20]. And another analogous method is the localized method of fundamental solutions [21]. The remainder of this paper is arranged as follows: in Section 2, the calculation formula of the two-dimensional barycentric Lagrange interpolation and some theoretical knowledge of the simplified matrix are introduced. In Section 3, the interpolation collocation method and its discrete matrix for calculating the two-dimensional Helmholtz Eq. (1.1) are derived. In Section 4, the numerical results for some test problems are given and compared with the results in the literature. Finally, conclusions are included in Section 5.

2 Barycentric Lagrange Interpolation

2.1 Barycentric Lagrange Interpolation Formula

Let $x_i, i = 1, 2, \ldots, m$, be a given interpolation node, and the corresponding function value is f_i, then Lagrange polynomial interpolation formula can be expressed as

$$p(x) = \sum_{i=1}^{m} L_i(x) f_i \tag{2.1}$$

where $L_i(x)$ is Lagrange interpolation basis function in the form of

$$L_i(x) = \frac{\prod_{i=1}^{m} (x - x_i)}{\prod_{i=1, i \neq j}^{m} (x_j - x_i)} \tag{2.2}$$

According to Eq. (2.1), it can be concluded that when new interpolation nodes are added, the interpolation basis function needs to be recalculated, which increases the calculation amount. However, if we start from the basis function and extract the common part, its expression will be changed and the calculation amount will be reduced consequently.

Set

$$l(x) = (x - x_1)(x - x_2) \ldots (x - x_m) \tag{2.3}$$

Defines weight of the barycentric Lagrange interpolation function as

$$\omega_i = \frac{1}{\prod_{i=1, i \neq j}^{m} (x_j - x_i)} \tag{2.4}$$

As usual, we can get the following algorithm flow diagram (see Fig. 1) for computing ω_i.

Then, the interpolation basis function (2.2) can be changed to

$$L_i(x) = l(x) \frac{\omega_i}{x - x_i}, \quad i = 1, 2, \ldots, m \tag{2.5}$$

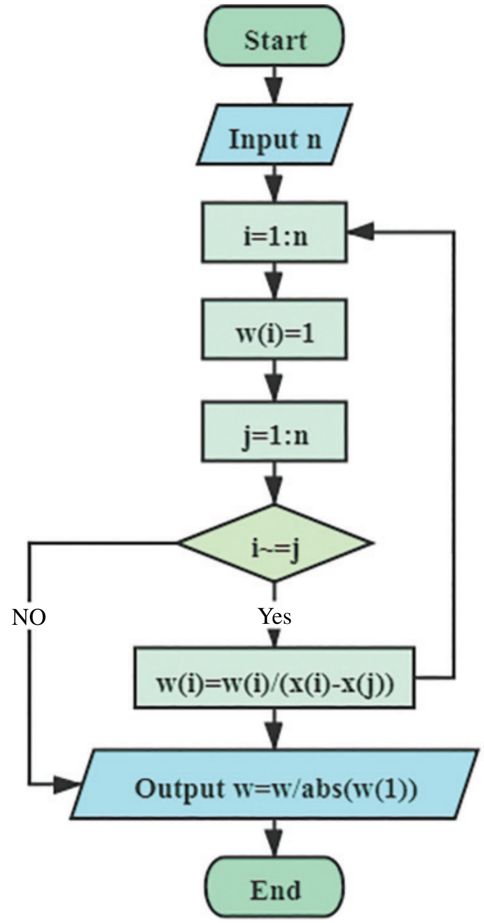


Figure 1: The algorithm flow diagram of ω_i

Substitute Eq. (2.5) into Eq. (2.1) to get

$$p(x) = l(x) \sum_{i=1}^{m} \frac{\omega_i}{x - x_i} f_i \tag{2.6}$$

The Eq. (2.6) is called the improved Lagrange interpolation formula [22]. The part of main improvement is to construct Eq. (2.3), thus rewriting the interpolation basis function Eq. (2.2) into the form of Eq. (2.5). The purpose is with number of Lagrange interpolation nodes increases, the computation amount decreases a lot, from $\mathrm{O}\left(m^2\right)$ to $\mathrm{O}(m)$.

By using Eq. (2.6) to interpolate the constant 1 with $(x_i, 1)$, $i = 1, 2, \ldots, m$, then we have

$$1 = \sum_{i=1}^{m} L_i(x) = l(x) \sum_{i=1}^{m} \frac{\omega_i}{x - x_i}$$

After simplification, it can be obtained

$$l(x) = \frac{1}{\sum_{i=1}^{m} \frac{\omega_i}{x - x_i}} \tag{2.7}$$

Substituting Eq. (2.7) into Eq. (2.6), we have

$$p(x) = \frac{\sum_{i=1}^{m} \frac{\omega_i}{x - x_i} f_i}{\sum_{i=1}^{m} \frac{\omega_i}{x - x_i}} \tag{2.8}$$

The Eq. (2.8) is called the barycentric Lagrange interpolation formula [22]. Compared with Eq. (2.6), Eq. (2.8) overcomes the Runge phenomenon effectively while maintaining the same computational cost $\mathrm{O}(m)$.

2.2 Bivariate Barycenter Lagrange Interpolation Formula

Let the $u(x, y)$ be a function about the variable x and y, and define the interval as $(x, y) \in [a, b] \times [c, d]$. When the interval $[a, b]$ and $[c, d]$ is discretized into m and n nodes, respectively, then there are $m \times n$ tensor product interpolation nodes in the whole domain $\left(x_i, y_j\right)$, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$. Then the barycentric Lagrange interpolation formula [23] of the $u(x, y)$ is

$$u(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} L_i(x) M_j(y) u_{ij} \tag{2.9}$$

where $u_{ij} = u(x_i, y_j)$. $L_i(x)$ and $M_j(y)$ are the interpolation basis functions in the x and y direction, respectively.

$$L_i(x) = \frac{\frac{\omega_i}{x - x_i}}{\sum_{k=1}^{m} \frac{\omega_k}{x - x_k}}, \quad M_j(y) = \frac{\frac{v_j}{y - y_j}}{\sum_{k=1}^{n} \frac{v_k}{y - y_k}}, \quad i = 1, 2, \ldots, m, \ j = 1, 2, \ldots, n \tag{2.10}$$

where the weight is

$$\omega_i = \frac{1}{\prod_{i \neq k}^{m} (x - x_k)}, \quad v_j = \frac{1}{\prod_{j \neq k}^{n} (y - y_k)} \tag{2.11}$$

2.3 Interpolation Knot

We can see from the expression of the barycentric Lagrange interpolation weight function Eqs. (2.4) or (2.11) that the selection of the interpolation weight is only related to the distribution of the nodes. Therefore we consider the three kinds of nodes as interpolation nodes and test nodes. They are random nodes, uniform nodes and Chebyshev nodes. Because random nodes and uniforms nodes can be directly implemented in programming language, this part only gives the Chebyshev nodes formula:

Chebyshev nodes of the first kind:

$$x_i = \cos\frac{(2i+1)\pi}{2m+2}, \quad i = 1, 2, \ldots, m \tag{2.12}$$

Chebyshev nodes of the second kind:

$$x_i = \cos\frac{i\pi}{m}, \quad i = 1, 2, \ldots, m \tag{2.13}$$

What needs to be explained is that for two types of Chebyshev node formulas, the defined interval is $[-1, 1]$. As for the general interval $[a, b]$, the coordinate transformation formula $\bar{x} = \frac{x(b-a)}{2} + \frac{(b+a)}{2}$ of the interval can be used.

2.4 Differential Matrix of Barycenter Lagrange Interpolation

The differential matrix was first found in the study of Chebyshev quasi-spectral method [24]. The barycentric Lagrange interpolated differential matrix can directly obtain the derivative function of the unknown function in the Helmholtz equation at the computing node. Therefore, the differential matrix is a very important part in the solution of the barycentric interpolation Lagrange collocation method.

When m nodes are inserted on the interval $[a, b]$, there is $a = x_1 < x_2 < \cdots < x_m = b$, to set the function value of the unknown function $u(x)$ at the node x_i is $u_i = u(x_i)$, $i = 1, 2, \ldots m$, then the barycentric Lagrange interpolation formula of the function $u(x)$ is

$$u(x) = \sum_{i=1}^{m} L_i(x)\, u_i \tag{2.14}$$

In which, $L_i(x) = \frac{\frac{\omega_i}{x-x_i}}{\sum_{i=1}^{m}\frac{\omega_i}{x-x_i}}$ is the barycentric interpolation basis function, and $\omega_i = \frac{1}{\prod_{i\neq j}^{m} x_i - x_j}$ is the weight of gravity interpolation.

Using Eq. (2.14), the k derivative of the function $u(x)$ at the node $x_j\,(j = 1, 2, \ldots, m)$ can be expressed as

$$u^{(k)}\left(x_j\right) = \frac{d^k u\left(x_j\right)}{dx^k} = \sum_{i=1}^{n} L_i^{(k)}\left(x_j\right) u_i = \sum_{i=1}^{n} D_{ij}^{(k)} u_i, \quad k = 1, 2, \ldots$$

In the form of a matrix as

$$u^{(k)} = D^{(k)} u \tag{2.15}$$

where $u^{(k)} = \left[u_1^{(k)}, u_2^{(k)}, \ldots, u_m^{(k)}\right]^T$ and $u = [u_1, u_2, \ldots, u_m]^T$ are the vector. The matrix $D^{(k)}$ is as the k order differential matrix of the function $u(x)$, and $D_{ij}^{(k)} = L_i^{(k)}\left(x_j\right)$. In addition, we can calculate $D^{(k)}$ (see [25]) with

$$D_{ij}^{(k)} = L_i^{(k)}\left(x_j\right) = \begin{cases} \dfrac{k}{x_i - x_j}\left(\dfrac{\omega_j}{\omega_i} D_{jj}^{(k-1)} - D_{ij}^{(k-1)}\right), & i \neq j \\ -\displaystyle\sum_{i=0, i\neq j}^{m} D_{ij}^{(k)}, & i = j \end{cases} \tag{2.16}$$

When the solution of some partial differential equation is obtained by numerical method, it is also necessary to use the differential matrix to calculate the derivative values of the unknown function on calculating nodes.

2.5 Kronecker Product of a Matrix

Kronecker product is an operation form between two matrices of any size and a special form of tensor product. The Kronecker product of any matrix $A = \left(a_{ij}\right)_{m\times n}$ and matrix $B = \left(b_{ij}\right)_{p\times q}$ is $A \otimes B$, recorded as following

$$A \otimes B = \left(a_{ij}B\right)_{m\cdot p\times n\cdot q} \tag{2.17}$$

where $a_{ij}B = \begin{bmatrix} a_{ij}b_{11} & a_{ij}b_{12} & \cdots & a_{ij}b_{1q} \\ a_{ij}b_{21} & a_{ij}b_{22} & \cdots & a_{ij}b_{2q} \\ \vdots & \vdots & \cdots & \vdots \\ a_{ij}b_{p1} & a_{ij}b_{p2} & \cdots & a_{ij}b_{pq} \end{bmatrix}$

Using the Kronecker product of the barycentric Lagrange interpolation differential matrix, the discrete equation of the Helmholtz equation can be expressed as a simple matrix form.

2.6 Relationship between Partial Derivative of Differential Equation and Differential Matrix

The Kronecker product with the barycentric Lagrange interpolated differential matrix can also be used. The unknown function in the Helmholtz equation has the following corresponding relation between the partial derivative of the variable and the Kronecker product of the corresponding differential matrix.

The number of nodes for the unknown function $u(x,y)$ in x and y directions of the two-dimensional Helmholtz equation are m and n then by using Eq. (2.17), the expressions of the differential matrices corresponding to the partial derivative of the differential equation about the variables x and y can be briefly expressed as

$$\frac{\partial^2 u(x,y)}{\partial x^2} = D^{(2)} \otimes I_n, \quad \frac{\partial^2 u(x,y)}{\partial y^2} = I_m \otimes C^{(2)}, \quad u(x,y) = I_m \otimes I_n \tag{2.18}$$

where I_m and I_n represent the unit matrices of m order and n order, respectively, and the $D^{(2)}$ and $C^{(2)}$ are barycentric interpolates of the second-order matrices at the node x_1, $x_2, \ldots, x_m$ and y_1, $y_2, \ldots, y_n$, respectively.

Using the above marks, the discrete forms of partial differential equations and boundary conditions can be written directly into matrix form, which makes programming much easier.

3 Discrete Equations of the Two-Dimensional Helmholtz Equation

By substituting the barycenter Lagrange interpolation Eq. (2.9) into Eq. (1.1), we can obtain

$$\sum_{j=1}^{n}\sum_{i=1}^{m}\left[L_i''(x)\,M_j(y)+L_i(x)\,M''{}_j(y)+k^2(x,y)\,L_i(x)\,M_j(y)\right]u_{ij}=f(x,y) \tag{3.1}$$

If Eq. (3.1) holds at the nodes of (x_I, y_J), $I=1,2,\ldots,m; J=1,2,\ldots,n$, then Eq. (3.1) is replaced by the following

$$\sum_{j=1}^{n}\sum_{i=1}^{m}\left[L_i''(x_I)\,M_j(y_J)+L_i(x_I)\,M_J''(y_J)+k^2(x_I,y_J)\,L_i(x_I)\,M_j(y_J)\right]u_{ij}=f(x_I,y_J) \tag{3.2}$$

Set $k^2(x_I,y_J)=k^2{}_{ij}, f(x_I,y_J)=f_{ij}, L_i(x_I)=L_{iI}, M_j(y_J)=L_{jJ}$. From the properties of the Lagrange interpolation basis, $L_i(x_I)=\delta_{iI}, M_j(x_J)=\delta_{jJ}, L_i^{(r)}(x_I)=D_{iI}^{(r)}, M_j^{(r)}(y_J)=C_{jJ}^{(r)}, r=1,2,$ and using Eqs. (2.15), (2.16) and (2.18), then Eq. (3.2) can also be expressed as

$$\left[\left(D^{(2)}\otimes I_n\right)+\left(I_m\otimes C^{(2)}\right)+\begin{pmatrix}k_{1n}^2 & \cdots & 0\\ \vdots & \vdots & \vdots\\ 0 & \cdots & k_{mn}^2\end{pmatrix}(I_m\otimes I_n)\right]\begin{pmatrix}u_{1n}\\ \vdots\\ u_{mn}\end{pmatrix}=\begin{pmatrix}f_{1n}\\ \vdots\\ f_{mn}\end{pmatrix} \tag{3.3}$$

Let

$$\begin{aligned}
&\mathrm{k}^2=diag\left(k_{11}^2,\ldots,k_{1n}^2,k_{21}^2,\ldots,k_{2n}^2,\ldots,k_{m1}^2,\ldots,k_{mn}^2\right),\\
&U=[u_{11},\ldots u_{1n},u_{21},\ldots,u_{2n},\ldots,u_{m1},\ldots,u_{mn}]^T,\\
&F=[f_{11},\ldots f_{1n},f_{21},\ldots,f_{2n},\ldots,f_{m1},\ldots,f_{mn}]^T
\end{aligned}$$

Then Eq. (3.3) can be written into

$$\left[\left(D^{(2)}\otimes I_n\right)+\left(I_m\otimes C^{(2)}\right)+\mathrm{k}^2\,(I_m\otimes I_n)\right]U=F \tag{3.4}$$

That is

$$LU=F \tag{3.5}$$

In which $L=\left(D^{(2)}\otimes I_n\right)+\left(I_m\otimes C^{(2)}\right)+\mathrm{k}^2\,(I_m\otimes I_n)$.

The above two-dimensional Helmholtz equation is discretized by the barycentric Lagrange interpolation method, and the final scheme is Eq. (3.5). In addition, the boundary conditions need to be discretized, and the following discrete formulas for boundary conditions are elaborated.

By employing Eqs. (2.17) and (2.18), we obtain the matrix form of the boundary condition in Eq. (1.2)

$$\begin{cases}\left(e_m^1\otimes I_n\right)U=g_1,\left(e_m^n\otimes I_n\right)U=g_2\\ \left(I_m\otimes e_n^1\right)U=g_3,\left(I_m\otimes e_n^m\right)U=g_4\end{cases} \tag{3.6}$$

where e_m^1 and e_m^m represent the first and m lines of the first order differential matrix of the m order unit matrix, e_n^1 and e_n^n represent the first and n lines of the first order differential matrix of the n order unit matrix, and $g_1 = [g_1(y_1), g_1(y_2), \ldots, g_1(y_n)]^T, g_2 = [g_2(y_1), g_2(y_2), \ldots, g_2(y_n)]^T$, $g_3 = [g_3(x_1), g_3(x_2), \ldots, g_3(x_m)]^T, g_4 = [g_4(x_1), g_4(x_2), \ldots, g_4(x_m)]^T$.

Combining Eqs. (3.5) and (3.6), we can obtain the value of the function $u(x, y)$ on each node.

4 Numerical Examples

In this section, we solve some test problems to demonstrate the effectiveness and accuracy of the barycentric Lagrange interpolation method. Six numerical examples of Helmholtz equation including high wavenumber and variable wavenumber problems are given. In which, the maximum absolute error (E_∞) and relative error (E_r) are defined as follows:

$$E_\infty = \max_{1 \le i \le Nt, 1 \le j \le Mt} |u_{ij} - ue_{ij}|,\ E_r = \sqrt{\frac{\sum_{i=1}^{Nt}\sum_{j=1}^{Mt}\left(u_{ij} - ue_{ij}\right)^2}{\sum_{i=1}^{Nt}\sum_{j=1}^{Mt} {ue_{ij}}^2}} \tag{4.1}$$

where u_{ij} and ue_{ij} represent numerical and exact solutions, respectively. Nt and Mt are the numbers of test nodes in the x and y directions, respectively.

Example 1 [26]:

Consider the following non-homogeneous Helmholtz equation

$$u_{xx} + u_{yy} + k^2 u = \left(k^2 - \pi^2 - k^2\pi^2\right)\sin(\pi x)\sin(k\pi y), \quad (x, y) \in [0, 1] \times [0, 1]$$

with the boundary conditions:

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

The exact solution is

$$u(x, y) = \sin(\pi x)\sin(k\pi y)$$

Tab. 1 gives the calculation results of E_∞ for the four kinds of interpolation nodes with $Nt = Mt = N$, $k = 5$. In which, N is the number of interpolation nodes in the domain, Nt and Mt are the numbers of test nodes, and the test nodes type are the second kind of Chebyshev nodes (that is Chebyshev II) rather than the first kind of Chebyshev nodes (that is Chebyshev I). The results show that the random interpolation nodes are bad for our proposed method, the uniform interpolation nodes are not good and the two kinds of Chebyshev interpolation nodes are very good. What's more, the numerical results show that the accuracy of the Chebyshev node is highest, the stability is best, and the calculation error is smallest. Therefore, we use the two kinds of Chebyshev nodes rather than the random or uniform nodes as the interpolation nodes to calculate the Tabs. 2 and 3 for Example 1. In addition, the numerical results of the Chebyshev II is a little better than that of the Chebyshev I.

Table 1: The E_∞ with four kinds of interpolation nodes when $Nt = Mt = N, k = 5$ for Example 1

N	Random	Uniform	Chebyshev I	Chebyshev II
16	3.254(+0)	2.591(−2)	3.246(−5)	9.017(−6)
20	4.562(+0)	1.903(−4)	4.982(−8)	1.118(−8)
24	1.011(+8)	2.442(−2)	3.602(−11)	7.128(−12)

Table 2: The E_∞ with two kinds of interpolation nodes and three kinds of test nodes when $Nt = Mt = 100, k = 5$ for Example 1

N	Chebyshev I			Chebyshev II		
	Random	Uniform	Chebyshev II	Random	Uniform	Chebyshev II
16	3.245(−5)	3.246(−5)	3.246(−5)	1.846(−5)	1.833(−5)	1.844(−5)
20	4.982(−8)	4.982(−8)	4.982(−8)	2.646(−8)	2.596(−8)	2.645(−8)
24	3.602(−11)	3.602(−11)	3.602(−11)	1.898(−11)	1.895(−11)	1.896(−11)

Table 3: The E_∞ with two kinds of interpolation nodes and three kinds of test nodes when $N = 24, k = 5$ for Example 1

(Nt, Mt)	Chebyshev I			Chebyshev II		
	Random	Uniform	Chebyshev II	Random	Uniform	Chebyshev II
(24,24)	3.602(−11)	3.602(−11)	3.602(−11)	1.746(−11)	1.828(−11)	7.128(−12)
(50,50)	3.601(−11)	3.602(−11)	3.602(−11)	1.695(−11)	1.743(−11)	1.663(−11)
(90,60)	3.602(−11)	3.602(−11)	3.602(−11)	1.875(−11)	1.828(−11)	1.783(−11)
(100,150)	3.598(−11)	3.602(−11)	3.602(−11)	1.890(−11)	1.887(−11)	1.888(−11)
(150,100)	3.600(−11)	3.602(−11)	3.602(−11)	1.848(−11)	1.897(−11)	1.890(−11)
(500,500)	3.602(−11)	3.602(−11)	3.602(−11)	1.902(−11)	1.900(−11)	1.902(−11)

Then the E_∞ with two kinds of Chebyshev interpolation nodes and three kinds of test nodes when $Nt = Mt = 100, k = 5$ and $N = 24, k = 5$ is given in Tabs. 2 and 3, respectively. What should we know is that the two kinds of interpolation nodes here are Chebyshev I and the Chebyshev II. Three kinds of test nodes are random test nodes, uniform test nodes and Chebyshev II test nodes all the time. According to the calculation results, firstly, we can know that the results of Chebyshev II is better than Chebyshev I and there is almost no difference between three kinds of test nodes. In addition, the error, error order and the convergence order is almost constant as Nt and Mt change, which means that the number of test nodes has little influence on our numerical results. So we can take the value of Nt and Mt with any reasonable number in the left numerical examples. What's more, the numerical results of the Chebyshev II is a little bit better than the Chebyshev I and the convergence order with the two kinds of interpolation nodes is almost identical, therefore, we use the Chebyshev II interpolation nods and three kinds of test nodes to calculate the left numerical examples in our work.

Finally, the calculation results of E_∞ with the 4th-order method in [26] and with the present scheme are compared when $Nt = 130, Mt = 140, k = 30$ in Tab. 4. The results show that the error order and the convergence order of the scheme in this paper reaches $\mathrm{O}\left(10^{-9}\right)$ and 12th-order, respectively when the number of collocation points is $N = 128$, while that in the literature just reaches $\mathrm{O}\left(10^{-4}\right)$ and 4th-order, respectively. It shows that the calculation accuracy of the present scheme is much better than that in the literature.

Table 4: The E_∞ when $Nt = 130, Mt = 140, k = 30$ for Example 1

N	4th-order [26]	Present scheme		
		Random	Uniform	Chebyshev II
64	1.28(−2)	1.097(−5)	1.074(−5)	1.097(−5)
128	7.10(−4)	1.445(−9)	4.660(−9)	4.521(−9)

The node distribution for the four kinds of interpolation nodes with $Nt = Mt = N, k = 5$ is shown in Fig. 2. We can observe that the random nodes are distributed randomly within the interval, uniform nodes are distributed equably in the interval and the two kinds of Chebyshev nodes are concentrated at the boundaries while the middle is sparse relatively.

Then the exact solution with Chebyshev test nodes is plotted in Fig. 3a-III, numerical solutions with random test nodes are plotted in Fig. 3b-I, uniform test nodes in Fig. 3b-II and Chebyshev test nodes in Fig. 3b-III; errors with random test nodes in Fig. 3c-I, uniform test nodes in Fig. 3c-II and Chebyshev test nodes in Fig. 3c-III with $Nt = 130$, $Mt = 140$, $k = 30$, $N = 12$. Among them, because the exact solutions with three kinds of test nodes are almost the same, we only give the figure with Chebyshev test nodes. In addition, it can be observed that all numerical solutions with the three kinds of test nodes agree well with the exact solution. What's more, because of the addition of test nodes on the boundaries to calculate the numerical solution with proposed method, these nodes' values on the boundaries are no longer given by the exact solution and resulting in some error's appearance on the boundaries. It can also be observed from Fig. 3c that the errors are equivalent with that in the interior region.

Finally, the convergence order with $Nt = Mt = 100, k = 5$ and $Nt = 130, Mt = 140, k = 30$ is shown in Fig. 4. It can be seen that with the wave number of k increases, the order of convergence decreases. Meanwhile, the convergence order of the two kinds of interpolation nodes Chebyshev I and Chebyshev II is almost identical. And then the convergence order of the proposed method is much higher than that in the literature.

Example 2 [27]:

Consider the following non-homogeneous Helmholtz equation

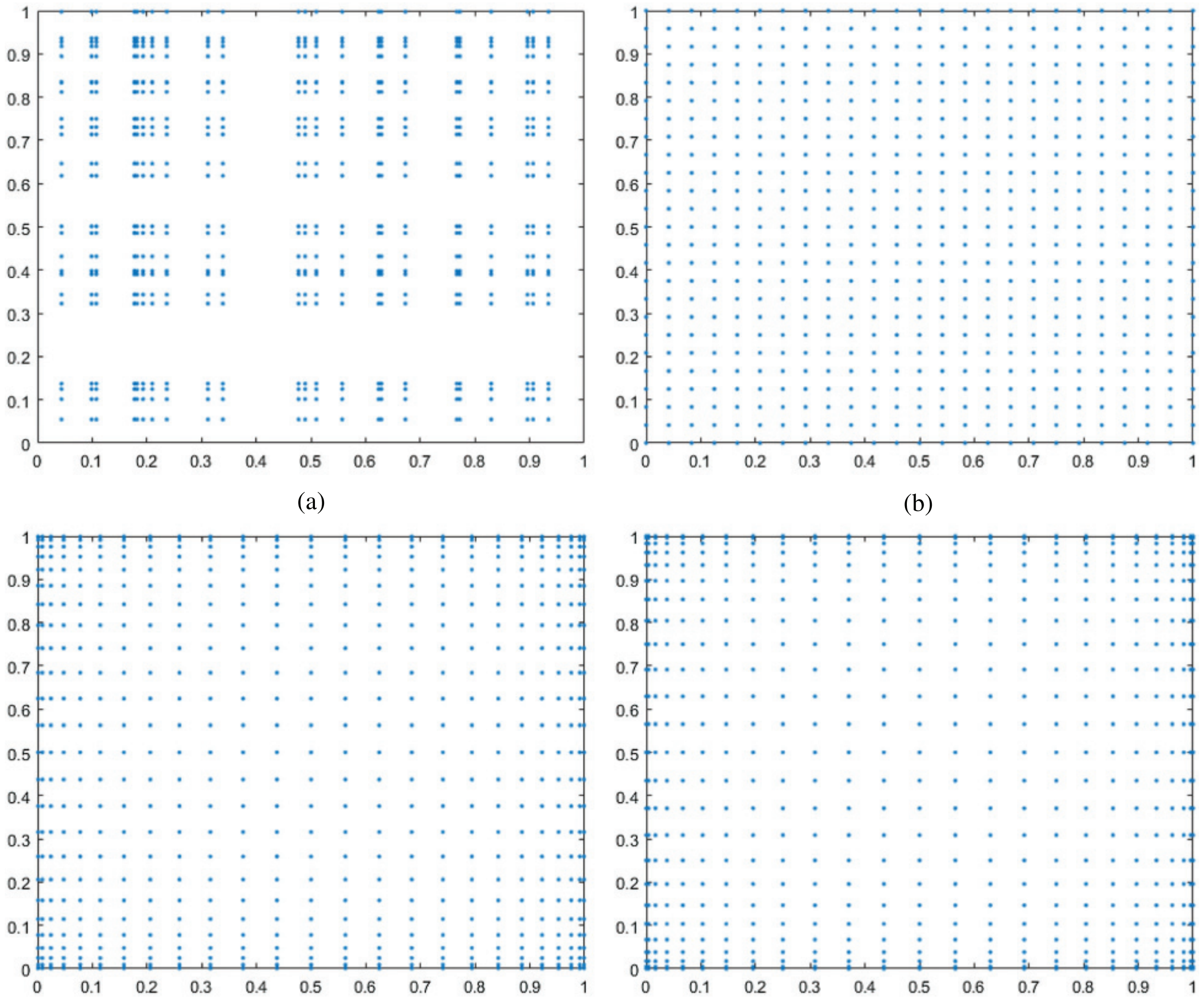
$$\begin{aligned}
&u_{xx} + u_{yy} + k^2 u = f(x,y), \quad (x,y) \in [0,\pi] \times [0,\pi] \\
&f(x,y) = 2\alpha^2 \{\sinh(\alpha x)\sinh[\alpha(\pi - x)] - \cosh(\alpha x)\cosh[\alpha(\pi - x)]\} \\
&+2\beta^2 \{\sinh(\beta y)\sinh[\beta(\pi - y)] - \cosh(\beta y)\cosh[\beta(\pi - y)]\} \\
&+k^2 \{\sinh(\alpha x)\sinh[\alpha(\pi - x)] + \sinh(\beta y)\sinh[\beta(\pi - y)]\}
\end{aligned}$$

with the boundary conditions:

$$u\,(0,y) = u\,(\pi,y) = \sinh\,(\beta y)\,\sinh\,[\beta\,(\pi - y)]$$
$$u\,(x,0) = u\,(x,\pi) = \sinh\,(\alpha x)\,\sinh\,[\alpha\,(\pi - x)]$$

The exact solution is

$$u\,(x,y) = \sinh\,(\alpha x)\,\sinh\,[\alpha\,(\pi - x)] + \sinh\,(\beta y)\,\sinh\,[\beta\,(\pi - y)]$$



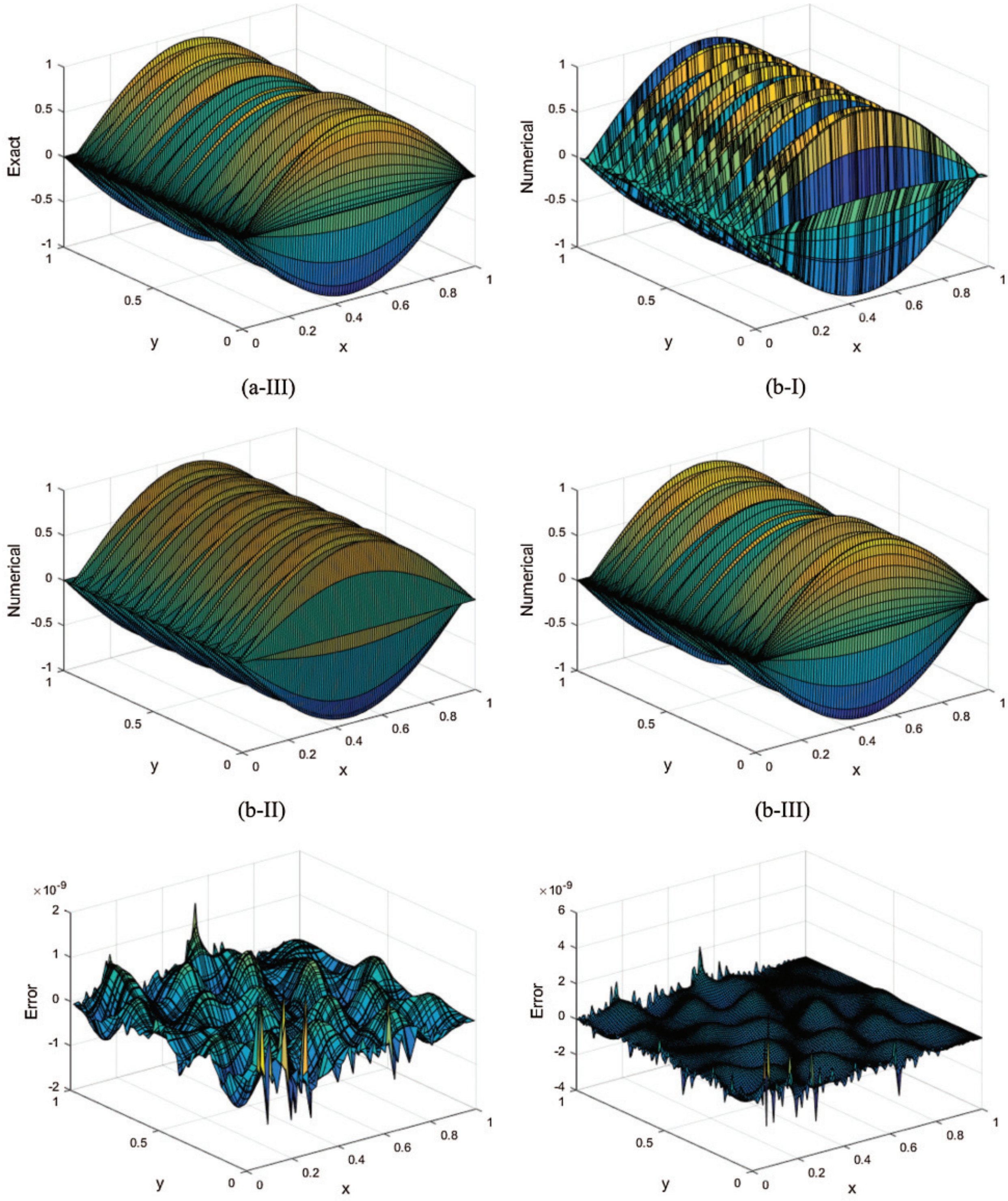
(a)

(b)

(c)

(d)

Figure 2: Random (a), uniform (b), Chebyshev I (c) and Chebyshev II (d) interpolation nodes when $Nt = Mt = N, k = 5$ for Example 1



(a-III)

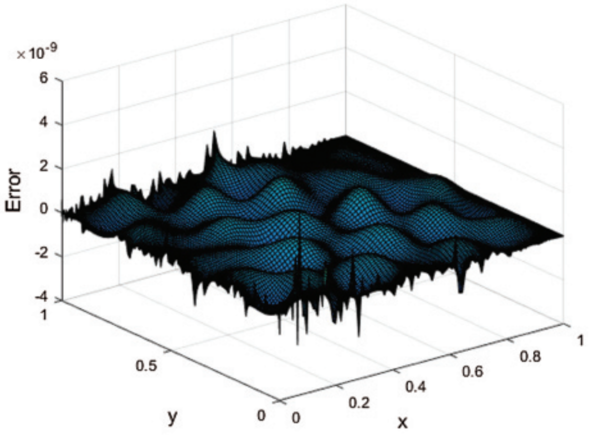
(b-I)

(b-II)

(b-III)

(c-I)

(c-II)



(c-III)

Figure 3: Exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) when $Nt = 130, Mt = 140, k = 30, N = 128$ for Example 1

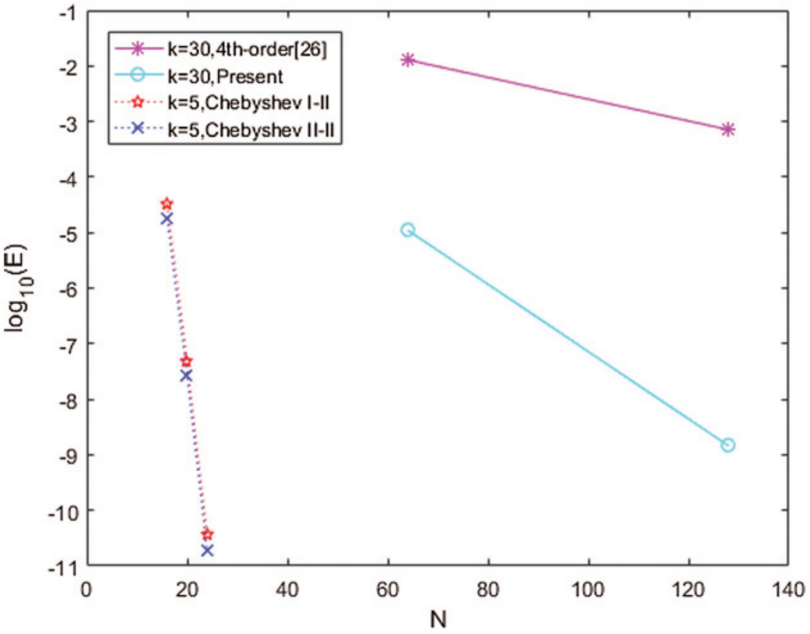


Figure 4: The convergence order with $Nt = Mt = 100, k = 5$ and $Nt = 130, Mt = 140, k = 30$ for Example 1

Tab. 5 gives the E_∞ computed by the present scheme and compares with EB-6 method in [27] when $\alpha = 0.5, \beta = 0.7, Nt = Mt = 120, k = 6.4$. The results show that the error order of the proposed scheme reaches $O\left(10^{-11}\right)$ when the number of collocation points is $N = 16$, while that in the literature, the error order only reaches $O\left(10^{-7}\right)$ when $N = 16$ and $O\left(10^{-10}\right)$ when $N = 64$. Therefore, the number of calculation points needed in the present scheme is much less than that in the literature and we can save more computational amount. In addition, we notice the error order of the proposed scheme decreases gradually and the error increase gradually with the increase of the number of nodes N, but the overall calculation result is better than that in the literature. What's more, convergence order of the proposed method is the 21th-order, while that in the literature is the 6th-order, so the precision of present method is also better than the method in the literature. Compared with the finite difference method, the barycentric Lagrange interpolation collocation method does not mean that the more nodes there are, the better the calculation results will be. Sometimes, we can obtain the better results with comparably a few nodes.

Table 5: The E_∞ when $\alpha = 0.5, \beta = 0.7, Nt = Mt = 120, k = 6.4$ for Example 2

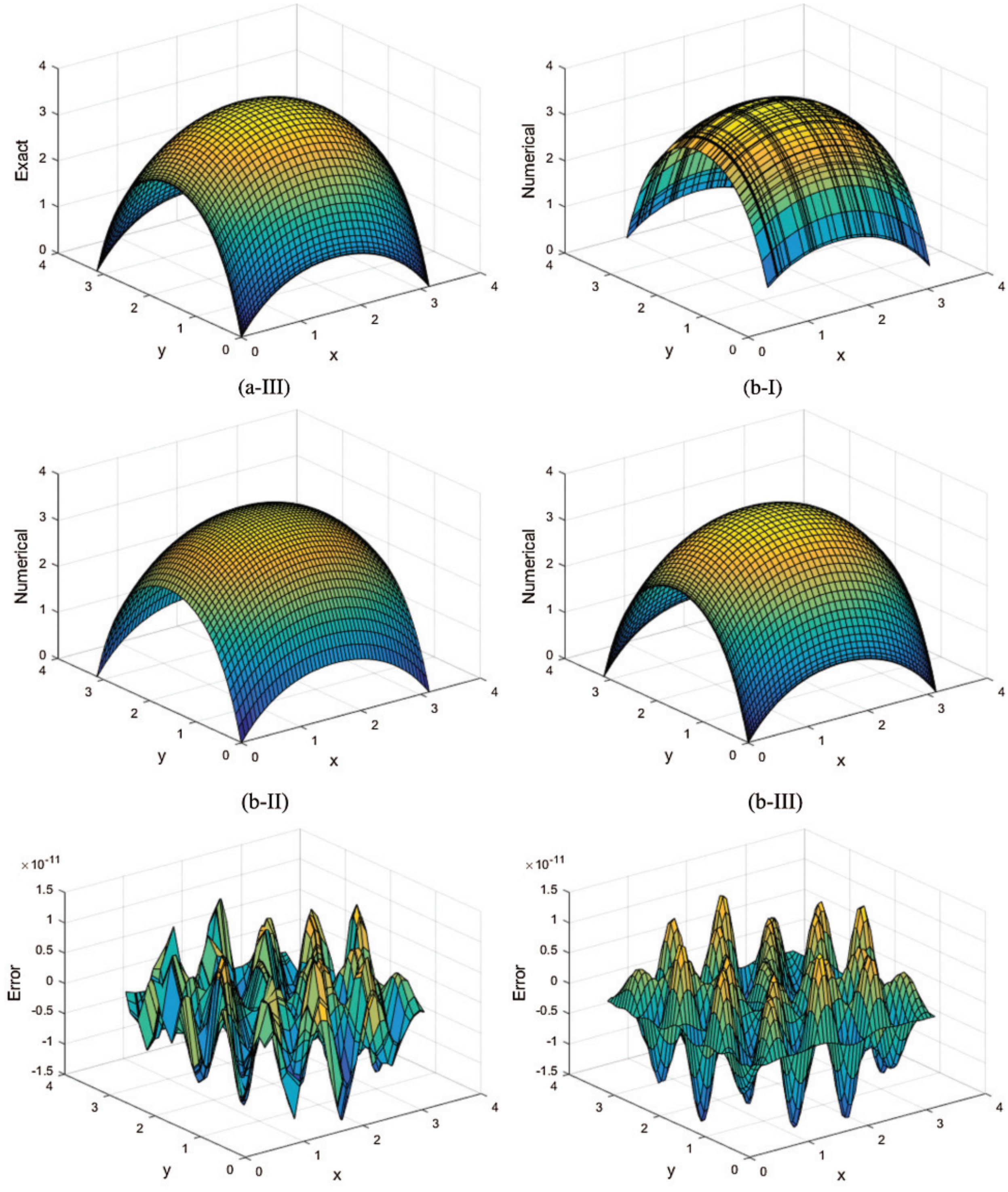
N	EB-6 [27]	Present scheme		
		Random	Uniform	Chebyshev
8	8.44(−5)	2.887(−6)	2.888(−6)	2.879(−6)
16	8.58(−7)	1.153(−11)	1.566(−11)	1.154(−11)
32	1.31(−9)	9.147(−11)	9.136(−11)	9.149(−11)
64	2.02(−10)	1.568(−9)	1.569(−9)	1.569(−9)

The E_∞, E_r and CPU time with $\alpha = 0.6, \beta = 0.8, Nt = Mt = 50, k = 10$ are given in Tab. 6. It shows that the proposed scheme can approximate the exact solution with a few nodes. It is necessary to note that for the determination of CPU time in our work, we select the longest time taken by the three test nodes method. Even so, we still find that for the two-dimensional Helmholtz equation, the CPU time with the present scheme is very small when the number of collocation points is less, which shows that this method is less time-consuming and can save calculation time effectively.

Table 6: The E_∞, E_r and CPU time when $\alpha = 0.6, \beta = 0.8, Nt = Mt = 50, k = 10$ for Example 2

N	E_∞			E_r			CPU time
	Random	Uniform	Chebyshev	Random	Uniform	Chebyshev	
8	1.859(−5)	1.813(−5)	1.845(−5)	1.802(−6)	1.519(−6)	2.215(−6)	0.270
12	1.844(−9)	1.842(−9)	1.818(−9)	2.084(−10)	2.072(−10)	2.320(−10)	0.280
16	1.419(−11)	1.448(−11)	1.437(−11)	2.197(−12)	1.929(−12)	2.051(−12)	0.439
20	1.847(−8)	1.847(−8)	1.857(−8)	2.681(−9)	2.847(−9)	3.042(−9)	0.946

Fig. 5. gives the exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) with $\alpha = 0.6, \beta = 0.8, Nt = Mt = 50, k = 10, N = 16$. It shows that the numerical solution with three kinds of test nodes have the similar shape and are in good agreement with the exact solution as well. What's more, Figs. 3c-I, 3c-II and 3c-III show that the error order reaches $\mathrm{O}\left(10^{-11}\right)$.



(a-III)

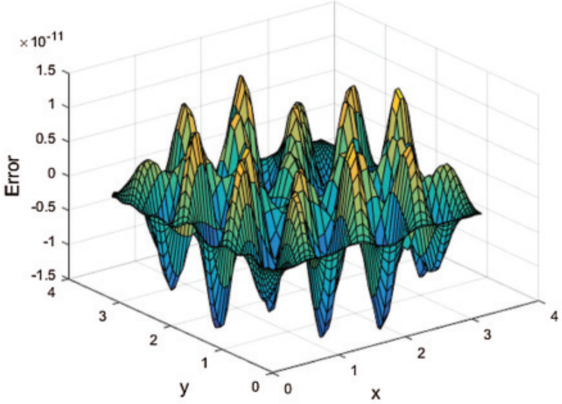
(b-I)

(b-II)

(b-III)

(c-I)

(c-II)



(c-III)

Figure 5: Exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) when $\alpha = 0.6, \beta = 0.8, Nt = Mt = 50, k = 10, N = 16$ for Example 2

Example 3 [28]:

Consider the following non-homogeneous high wavenumber Helmholtz equation

$$u_{xx} + u_{yy} + k^2 u = -a^2 \sin(ax) \sin(ky), \quad (x, y) \in [0, 1] \times [0, 1]$$

with the boundary conditions:

$$u(0, y) = 0, u(1, y) = \sin(a) \sin(ky)$$
$$u(x, 0) = 0, u(x, 1) = \sin(ax) \sin(k)$$

The exact solution is

$$u(x, y) = \sin(ax) \sin(ky)$$

The calculation results with the 6th-order method in [28] and the present scheme are compared under different wave numbers with $k = 24$ and $k = 48$ when $a = 1, Nt = 130, Mt = 150$ in Tab. 7. The results show that the calculation error and error order of the present scheme are much better than that in the literature. Therefore, it is more suitable for solving high wavenumber problems as well. Then the E_∞, E_r and CPU time with $a = 0.5, Nt = Mt = 60, k = 12$ are given in Tab. 8. It shows the advantages of the present scheme, such as less number of collocation points, short calculation time, high precision and small error.

Fig. 6. gives the exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) with $a = 1, Nt = 130, Mt = 150, k = 48, N = 128$, respectively. It shows that the numerical solutions agree very well with the exact solution, and the error of three kinds of test nodes has little difference.

Example 4 [29]:

Consider the following non-homogeneous variable wavenumber Helmholtz equation

$$u_{xx} + u_{yy} + k(x, y)^2 u = \left[k(x, y)^2 - 13\right] \sin(2x + 3y), \quad (x, y) \in [0, 1] \times [0, 1]$$

with the boundary conditions:

$$u(0,y)=\sin(3y),u(1,y)=\sin(2+3y)$$
$$u(x,0)=\sin(2x),u(x,1)=\sin(2x+3)$$

The exact solution is

$$u(x,y)=\sin(2x+3y)$$

Table 7: The E_∞ with different wave numbers when $a=1, Nt=130, Mt=150$ for Example 3

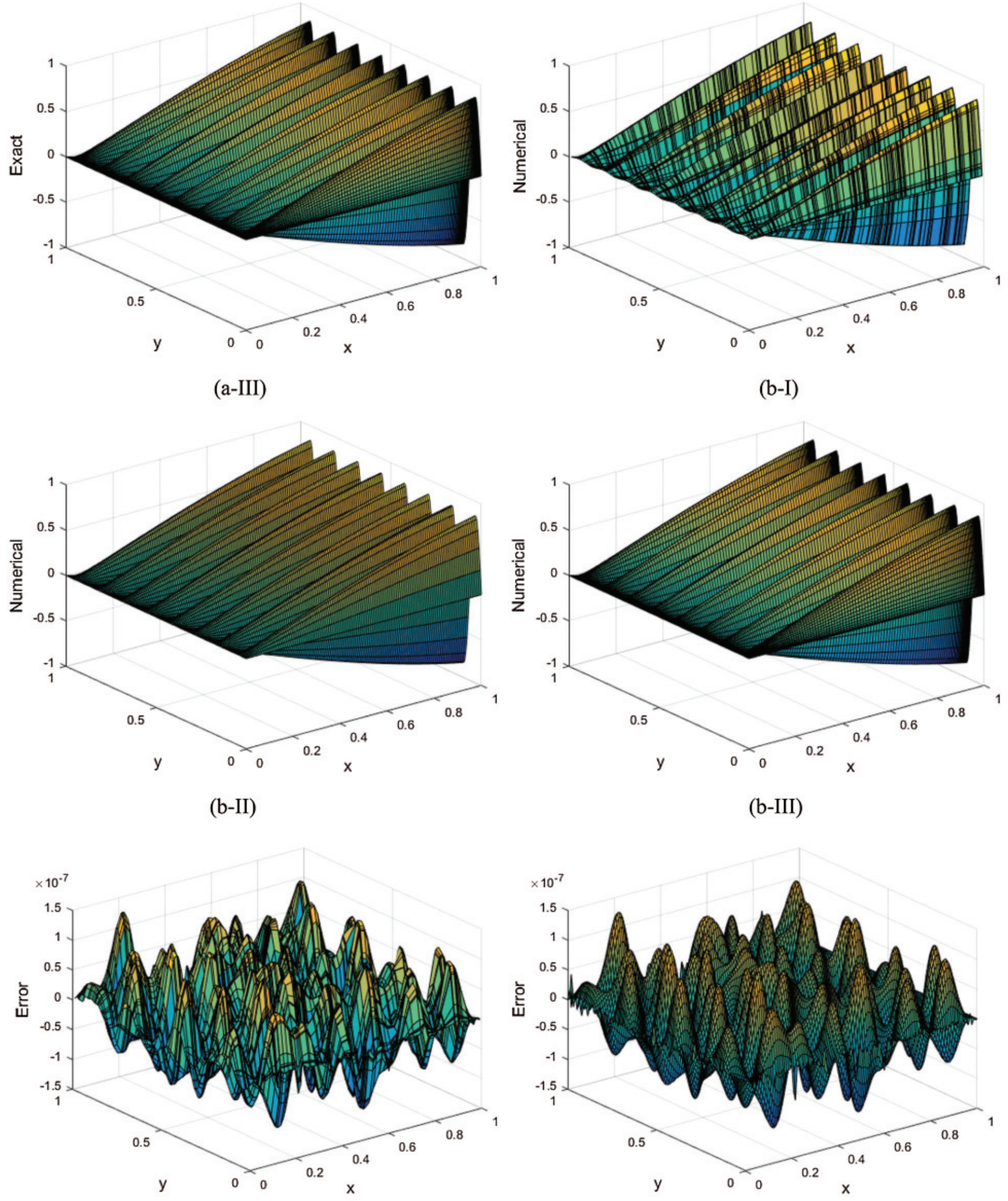
N		6th-order [28]	Present scheme		
			Random	Uniform	Chebyshev
$k=24$	64	4.60(−3)	1.935(−9)	2.752(−9)	4.374(−9)
	128	5.99(−5)	6.654(−8)	1.147(−7)	1.120(−7)
$k=48$	64	1.43	1.249(−8)	1.287(−8)	1.286(−8)
	128	1.83(−2)	1.406(−7)	1.392(−7)	1.398(−7)

Table 8: The E_∞, E_r and CPU time when $a=0.5, Nt=Mt=60, k=12$ for Example 3

N	E_∞			E_r			CPU time
	Random	Uniform	Chebyshev	Random	Uniform	Chebyshev	
12	2.659(−4)	2.711(−4)	2.597(−4)	4.930(−4)	4.654(−4)	4.064(−5)	0.941
16	4.241(−7)	4.269(−7)	4.252(−7)	7.224(−7)	7.327(−7)	6.486(−7)	0.617
24	1.882(−11)	1.847(−11)	2.102(−11)	1.368(−11)	1.250(−11)	1.433(−11)	1.688
32	2.259(−11)	3.657(−11)	3.658(−11)	1.537(−11)	1.897(−11)	2.240(−11)	1.392

Tab. 9 shows the E_∞ when $Nt = 150, Mt = 130$ with variable wavenumber $k(x,y) = \sqrt{\cos(x)+e^y}$. Compared with the MHADI method in [29], the error order of the present scheme is $\mathrm{O}\left(10^{-12}\right)$ with $N=16$, while that in the literature is only $\mathrm{O}\left(10^{-2}\right)$. In addition, the accuracy in the literature is the 4th-order, while that in this paper is up to the 9th-order. So the present scheme has higher calculation accuracy compared with that in the literature. We notice that although the error order of the proposed scheme decreases with the increase of the number of nodes, the overall calculation results are still better than that in the literature. Then the E_∞, E_r at different collocation points and CPU time with $Nt=Mt=50$, $k(x,y)=\sqrt{\sin(x)+e^y}$ are given in Tab. 10. The numerical results show that compared with the finite difference, the barycentric Lagrange interpolation method is more suitable for solving the problem of variable wavenumber. At the same time, we can obtain the better results with relatively a few nodes.

Fig. 7. gives the exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) with $Nt=Mt=50, N=16, k(x,y)=\sqrt{\sin(x)+e^y}$, respectively. It shows that although k is the variable wavenumber, the numerical solutions are also in good agreement with the exact solution, and the error is steady as a whole.



(a-III)

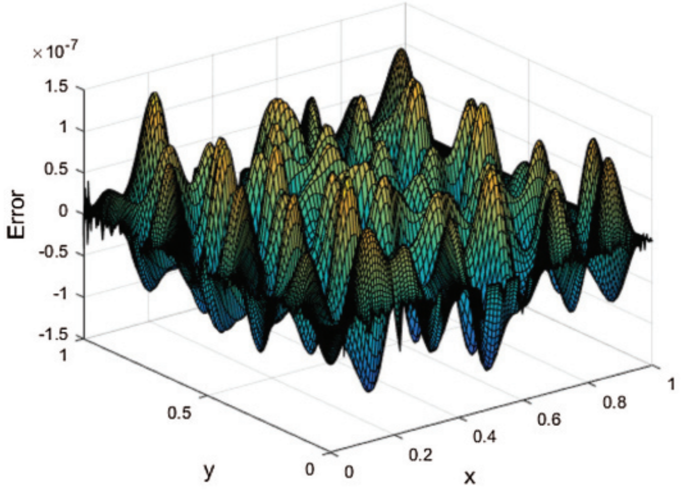
(b-I)

(b-II)

(b-III)

(c-I)

(c-II)



(c-III)

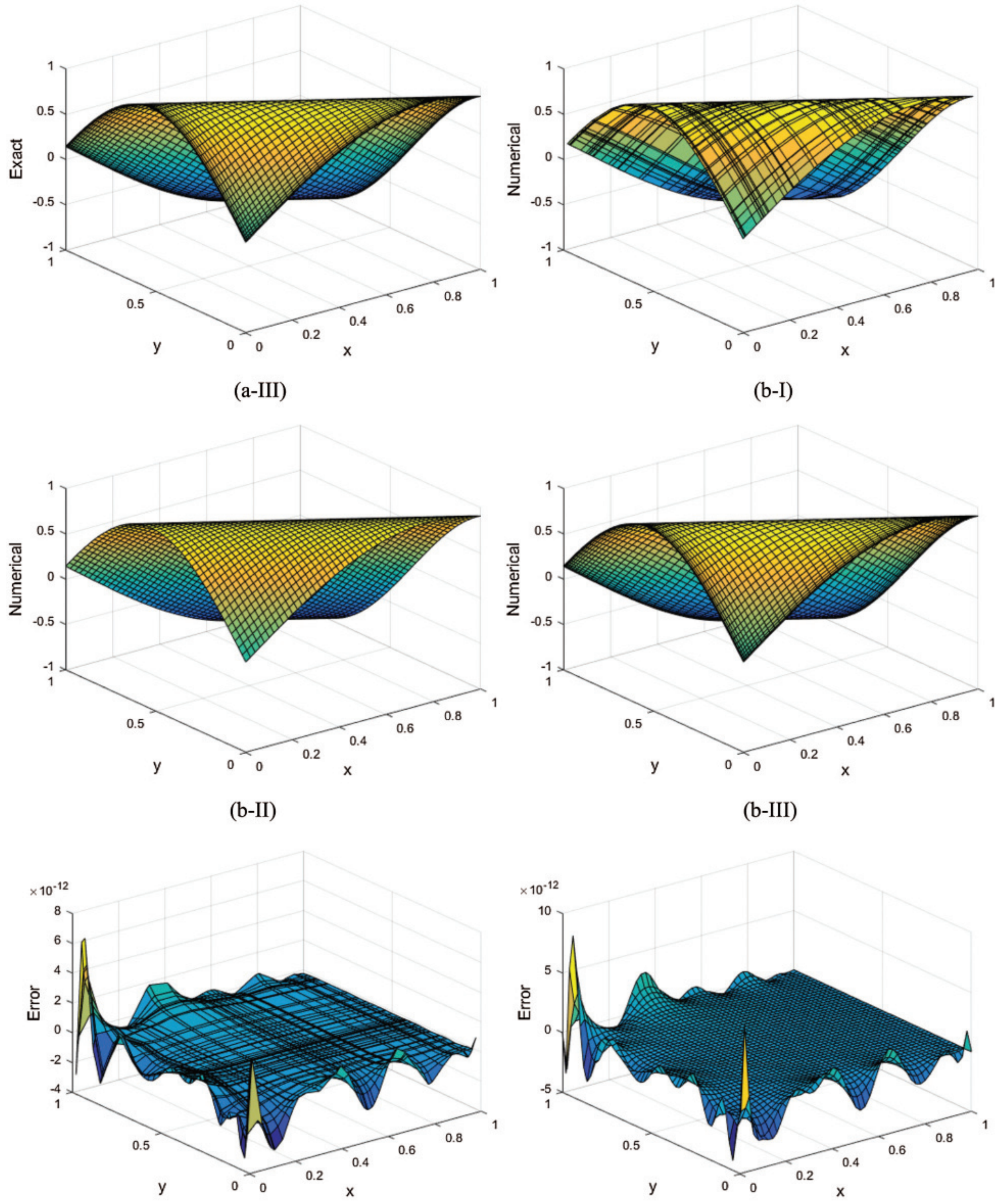
Figure 6: Exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) when $a = 1, Nt = 130, Mt = 150, k = 48, N = 128$ for Example 3

Table 9: The E_∞ when $Nt = 150, Mt = 130, k(x, y) = \sqrt{\cos(x) + e^y}$ for Example 4

N	MHADI [29]	Present scheme		
		Random	Uniform	Chebyshev
16	1.70(−2)	7.064(−12)	8.911(−12)	9.255(−12)
32	1.30(−3)	1.155(−10)	1.990(−10)	1.968(−10)
64	7.10(−5)	3.513(−9)	4.694(−9)	6.344(−9)
128	5.10(−6)	9.268(−8)	1.300(−7)	1.467(−7)

Table 10: The E_∞, E_r and CPU time when $Nt = Mt = 50, k(x, y) = \sqrt{\sin(x) + e^y}$ for Example 4

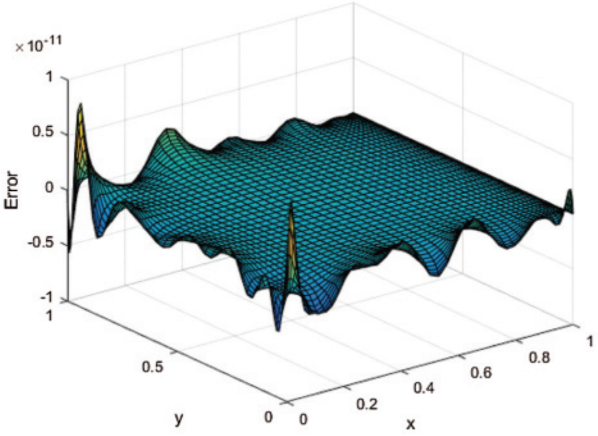
N	E_∞			E_r			CPU time
	Random	Uniform	Chebyshev	Random	Uniform	Chebyshev	
8	8.942(−7)	8.948(−7)	8.930(−7)	5.651(−7)	5.478(−8)	4.514(−7)	0.319
12	1.584(−11)	1.594(−11)	1.603(−11)	1.139(−11)	9.730(−12)	8.066(−12)	0.304
16	4.761(−12)	8.564(−12)	8.488(−12)	5.709(−13)	7.908(−13)	1.405(−12)	0.427
24	2.562(−11)	8.154(−11)	8.213(−11)	3.927(−12)	5.838(−12)	9.301(−12)	0.639



(a-III) (b-I)

(b-II) (b-III)

(c-I) (c-II)



(c-III)

Figure 7: Exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) when $Nt = Mt = 50, k(x, y) = \sqrt{\sin(x) + e^y}, N = 16$ for Example 4

Table 11: The E_∞, E_r and CPU time when $Nt = Mt = 50, k = 5$ for Example 5

N	E_∞			E_r			CPU time
	Random	Uniform	Chebyshev	Random	Uniform	Chebyshev	
20	5.305(−6)	5.755(−6)	5.704(−6)	3.381(−6)	3.382(−6)	3.404(−6)	0.732
24	4.838(−9)	5.502(−9)	5.078(−9)	1.209(−9)	2.156(−9)	2.148(−9)	1.004
40	1.686(−9)	1.969(−9)	1.431(−9)	5.450(−10)	4.917(−10)	5.305(−10)	5.726

Table 12: The E_∞, E_r and CPU time when $Nt = Mt = 50, k = 10$ for Example 5

N	E_∞			E_r			CPU time
	Random	Uniform	Chebyshev	Random	Uniform	Chebyshev	
24	5.317(−2)	6.593(−2)	6.160(−2)	2.441(−2)	2.356(−2)	2.423(−2)	0.951
32	5.982(-6)	5.982(−6)	6.141(−6)	2.614(−6)	2.614(−6)	2.549(−6)	2.329
48	1.112(−8)	1.313(−8)	1.056(−8)	5.243(−9)	5.010(−9)	5.036(−9)	10.638

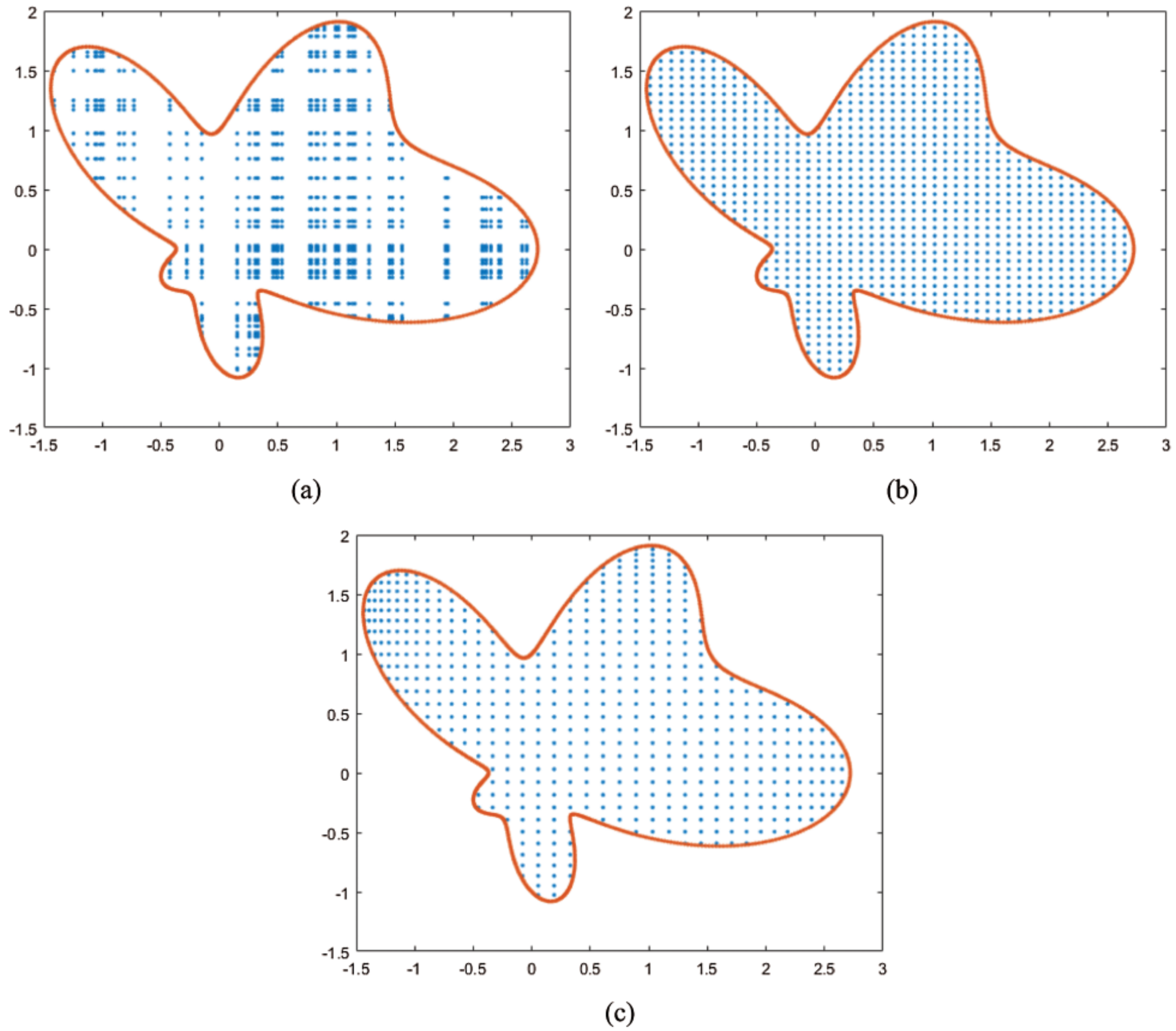
Example 5 [30]:

Consider the following homogeneous Helmholtz equation

$$u_{xx} + u_{yy} + k^2 u = 0, \quad (x, y) \in \Omega$$

with the Dirichlet boundary condition. The computational domain is an ameba-shape domain (see Fig. 7) with the boundary defined by the parametric equation:

$$\rho(\theta) = \exp(\sin(\theta)) \sin^2(2\theta) + \exp(\cos(\theta)) \cos^2(2\theta),$$
$$x = \rho(\theta) \cos(\theta), \ y = \rho(\theta) \sin(\theta), 0 \le \theta \le 2\pi$$

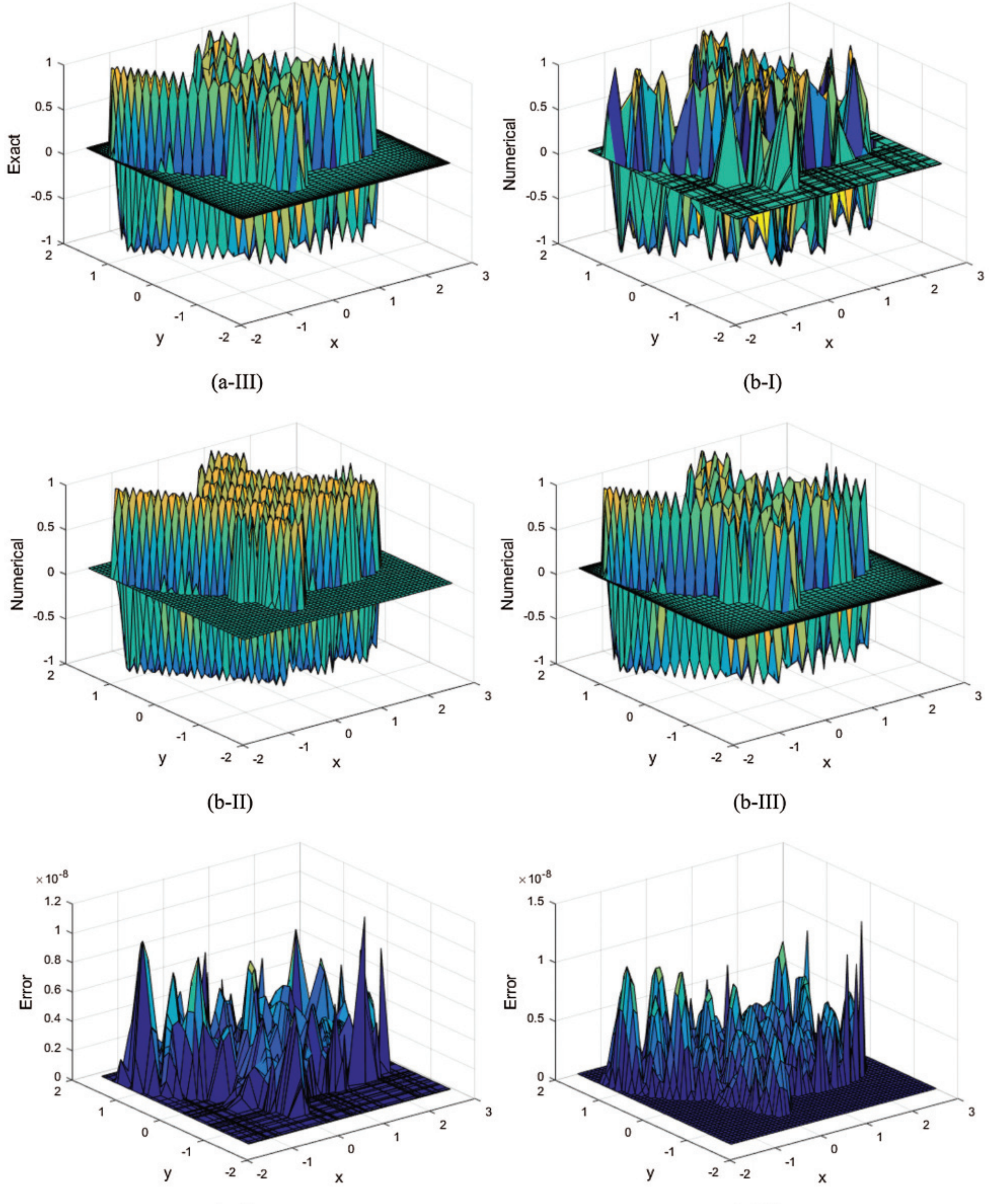


(a)

(b)

(c)

Figure 8: The ameba-shape domain with different test nodes: random (a), uniform (b) and Chebyshev (c) when $Nt = Mt = 50, k = 10, N = 48$ for Example 5



(a-III)

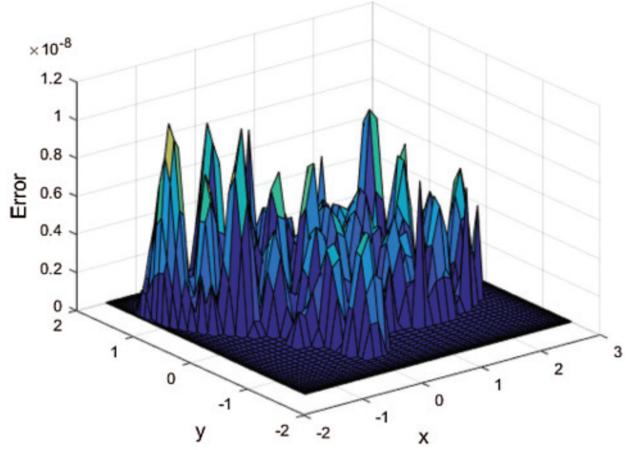
(b-I)

(b-II)

(b-III)

(c-I)

(c-II)



(c-III)

Figure 9: Exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) when $Nt = Mt = 50, k = 10, N = 48$ for Example 5

Table 13: The E_∞ when $Nt = Mt = 50$ for Example 6

N	Liu [30]	Present scheme		
		Random	Uniform	Chebyshev
20	1.7(−2)	1.275(−11)	2.991(−11)	4.136(−11)

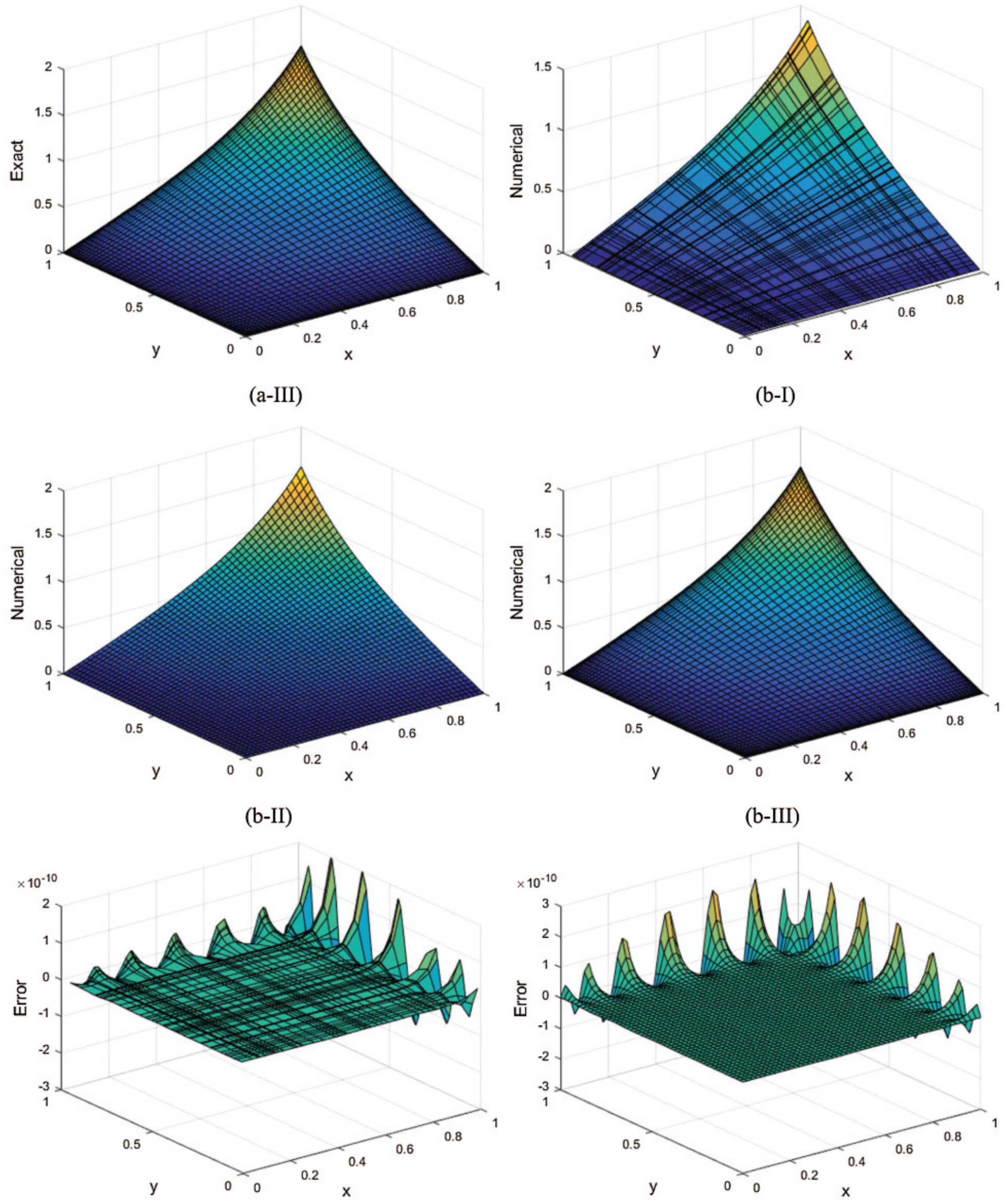
Table 14: The E_∞, E_r and CPU time when $Nt = Mt = 50$ for Example 6

N	E_∞			E_r			CPU time
	Random	Uniform	Chebyshev	Random	Uniform	Chebyshev	
8	1.183(−5)	1.573(−5)	1.552(−5)	3.867(−6)	6.715(−6)	7.788(−6)	0.203
12	3.981(−8)	5.827(−8)	5.834(−8)	1.182(−8)	2.188(−8)	2.674(−8)	0.244
16	1.746(−10)	2.228(−10)	2.215(−10)	6.391(−11)	7.581(−11)	9.435(−11)	0.340
24	2.021(−11)	4.141(−11)	4.207(−11)	7.508(−12)	1.067(−11)	1.218(−11)	0.482

The exact solution is

$$u(x,y) = \sin\left[\frac{k}{\sqrt{2}}(x+y)\right]$$

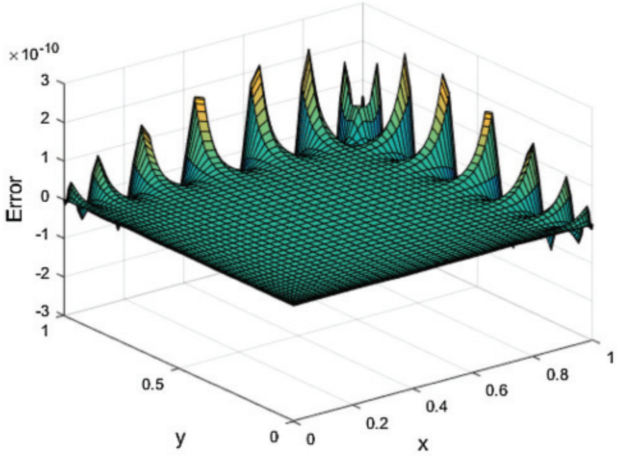
Tab. 11 and Tab. 12 give the E_∞, E_r and CPU time with an ameba-shape domain when $Nt = Mt = 50, k = 5$ and $Nt = Mt = 50, k = 10$, respectively. What should be noted is that the [30] only computed a square domain, so we do not make comparisons with it and just give the results of our proposed method. The results show that even in an arbitrary region, the accuracy and convergence order of the present method are still very high. It is obvious that the present method is an effective method with high precision.



(a-III) (b-I)

(b-II) (b-III)

(c-I) (c-III)



(c-III)

Figure 10: Exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) when $Nt = Mt = 50, N = 16$ for Example 6

Then the Fig. 8. gives the computational domain with different test nodes: random (a), uniform (b) and Chebyshev (c) when $Nt = Mt = 50, k = 10, N = 48$. Similar to Fig. 1, we can observe the distribution of the three test nodes. Fig. 9. gives the exact (a) and numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) with $Nt = Mt = 50, k = 10, N = 48$. It shows that the even though under the general domain, the numerical solutions agree very well with the exact solution, and the errors of three kinds of test nodes have little difference.

Example 6 [30]:

Consider the following non-homogeneous modified Helmholtz equation

$$u_{xx} + u_{yy} - u = f(x,y), \quad (x,y) \in \Omega = [0,1] \times [0,1]$$
$$f(x,y) = \left(x^2 + y^2\right) \frac{2\sin(xy)}{\cos^3(xy)} - \tan(xy)$$

with the boundary conditions:

$$u(0,y) = 0, u(1,y) = \tan(y)$$
$$u(x,0) = 0, u(x,1) = \tan(x)$$

The exact solution is

$$u(x,y) = \tan(xy)$$

Lastly, we consider the non-homogeneous modified Helmholtz equation for Example 6. Tab. 13 gives calculation results with the present scheme and compared with [30]. Then the E_∞, E_r and CPU time with different collocation points under $Nt = Mt = 50$ is given in Tab. 14. We find that the error and convergence order of our proposed method are still much better than that in the literature. Fig. 10 gives the exact (a), numerical (b) solutions and error (c) with different test nodes: random (I), uniform (II) and Chebyshev (III) when $Nt = Mt = 50, N = 16$ for Example 6. It is interesting that the proposed scheme is also suitable for non-homogeneous modified Helmholtz equation, and the numerical solution and the exact solution are in good agreement as well, at the same time the error is stable as a whole.

5 Conclusions

Based on the Chebyshev interpolation nodes, we have developed a meshless collocation method for solving the two-dimensional Helmholtz equations with the Dirichlet boundary condition, by using the barycentric Lagrange interpolation basis function. Firstly, this meshless collocation method applies the interpolation basis function to treat the spatial variables and their partial derivatives, and establishes the collocation method for solving the differential equations. Secondly, at a given node, the multivariate basis function is substituted into the equation and its boundaries and the discrete algebraic equations are obtained. Then, the differential matrix is used to simplify it. To investigate the accuracy and efficiency of the proposed method, we conduct some numerical experiments by using three kinds of test nodes. Based on our research, we can draw some conclusions as follows:

(1) As for the choice of interpolation nodes, the numerical results show that Chebyshev II is better than Chebyshev I for the present scheme. As for test nodes, there is almost no difference for the numerical results with three kinds of test nodes. In addition, with the change of the number of test nodes Nt and Mt, the error is almost unchanged. That is say that the number of test nodes has little influence for the numerical results.

(2) By using comparably fewer nodes, the present scheme can obtain high-precision numerical results and keep good stability compared with those methods in the literature. It shows the advantages of the present scheme, such as smaller number of collocation points needed, short calculation time, high precision, small error and high efficiency. What's more, the numerical solutions agree well with the exact solutions.

(3) Compared with the methods in the literature, the present method can not only calculate the high wavenumber problems, but also calculate the variable wavenumber problems, and the numerical results are better than those in the literature. For modified Helmholtz equation, the results computed by the present scheme are far better than those in the literature as well.

(4) The present method can be extended to solve the 3D Helmholtz equations. We will report it in the near future.

Funding Statement: The authors would like to thank very mush to the anonymous reviewers whose constructive comments are helpful for this paper's revision. This work is partially supported by National Natural Science Foundation of China (11772165, 11961054, 11902170), Key Research and Development Program of Ningxia (2018BEE03007), National Natural Science Foundation of Ningxia (2018AAC02003, 2020AAC03059), and Major Innovation Projects for Building First-class Universities in China's Western Region (Grant No. ZKZD2017009).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Assari, P., Fatemeh, A., Cuomo, M. S. (2019). A numerical scheme for solving a class of logarithmic integral equations arisen from two-dimensional Helmholtz equations using local thin plate splines. *Applied Mathematics and Computation, 356,* 157–172. DOI 10.1016/j.amc.2019.03.042.

2. Wang, K. Y., Wang, Q. S. (2013). Study on vibration of membranes with Taylor polynomial method and error analysis for Helmholtz equation. *Advanced Materials Research, 763,* 234–237. DOI 10.4028/www.scientific.net/AMR.763.234.
3. Jo, C. H., Shin, C., Suh, J. H. (1996). An optimal 9-point, finite-difference, frequency-space, 2-D scalar wave extrapolato. *Geophysics, 61(2),* 529–537. DOI 10.1190/1.1443979.
4. Du, K. (2011). A composite preconditioner for the electromagnetic scattering from a large cavity. *Journal of Computational Physics, 230(22),* 8089–8108. DOI 10.1016/j.jcp.2011.07.011.
5. Wang, S., Hoop, M. V. D., Xia, J. L. (2010). Acoustic inverse scattering via Helmholtz operator factorization and optimization. *Journal of Computational Physics, 229(22),* 8445–8462. DOI 10.1016/j.jcp.2010.07.027.
6. Liu, G., Jayathilake, P. G., Khoo, B. C., Han, F., Liu, D. K. (2012). Conformal mapping for the Helmholtz equation: Acoustic wave scattering by a two dimensional inclusion with irregular shape in an ideal fluid. *Journal of the Acoustical Society of America, 131(2),* 1055–1065. DOI 10.1121/1.3675947.
7. Linton, C. M. (2010). Lattice sums for the Helmholtz equation. *SIAM Review, 52(4),* 630–674. DOI 10.1137/09075130X.
8. Swager, M. R., Gray, L. J., Fata, S. N. (2010). Galerkin boundary integral analysis for the 3D Helmholtz equation. *Computer Modeling in Engineering & Sciences, 58(3),* 297–234.
9. Tsai, C. C., Young, D. L. (2013). Using the method of fundamental solutions for obtaining exponentially convergent Helmholtz eigensolutions. *Computer Modeling in Engineering & Sciences, 94(1),* 175–205.
10. Wu, T. T. (2017). A dispersion minimizing compact finite difference scheme for the 2D Helmholtz equation. *Journal of Computational and Applied Mathematics, 311,* 497–512. DOI 10.1016/j.cam.2016.08.018.
11. Liu, G. R., Gu, Y. T. (2005). *An introduction to meshfree method and their programming*, Berlin, New York: Springer.
12. Reutskiy, S. Y. (2006). The method of external sources (MES) for eigenvalue problems with Helmholtz equation. *Computer Modeling in Engineering & Sciences, 12(1),* 27–39.
13. Chen, H. X., Qiu, W. F. (2017). A first order system least squares method for the Helmholtz equation. *Journal of Computational and Applied Mathematics, 309,* 145–162. DOI 10.1016/j.cam.2016.06.019.
14. Dogan, H., Popov, V., Ooi, E. H. (2015). Dispersion analysis of the meshless local boundary integral equation and radial basis integral equation methods for the Helmholtz equation. *Engineering Analysis with Boundary Elements, 50,* 360–371. DOI 10.1016/j.enganabound.2014.09.009.
15. Chen, L. C., Li, X. L. (2020). A Burton–Miller boundary element-free method for Helmholtz problems. *Applied Mathematical Modelling, 83,* 386–399. DOI 10.1016/j.apm.2020.03.009.
16. Qu, W. Z., Fan, C. M., Gu, Y. (2019). Localized method of fundamental solutions for interior Helmholtz problems with high wave number. *Engineering Analysis with Boundary Elements, 107,* 25–32. DOI 10.1016/j.enganabound.2019.06.018.
17. Britt, S., Petropavlovsky, S., Tsynkov, S., Turkel, E. (2015). Computation of singular solutions to the Helmholtz equation with high order accuracy. *Applied Numerical Mathematics, 93,* 215–241. DOI 10.1016/j.apnum.2014.10.006.
18. Lin, J., Chen, W., Chen, C. S., Jiang, X. R. (2020). Fast bundary knot method for solving axisymmetric Helmholtz problem with high wave number. *Computer Modeling in Engineering & Sciences, 94(6),* 485–505.
19. Wang, K., Wong, Y. S., Huang, J. Z. (2017). Solving Helmholtz equation at high wave numbers in exterior domains. *Applied Mathematics and Computation, 298,* 221–235. DOI 10.1016/j.amc.2016.11.015.
20. Li, X. L., Dong, H. Y. (2020). Error analysis of the meshless finite point method. *Applied Mathematics and Computation, 382,* 125326. DOI 10.1016/j.amc.2020.125326.
21. Qu, W. Z., Fan, C. M., Li, X. L. (2020). Analysis of an augmented moving least squares approximation and the associated localized method of fundamental solutions. *Computers and Mathematics with Applications, 80(1),* 13–30. DOI 10.1016/j.camwa.2020.02.015.
22. Berrut, J. P., Trefethen, L. N. (2004). Baryccentric lagrange interpolation. *SIAM Review, 46(3),* 501–517. DOI 10.1137/S0036144502417715.

23. Ma, W. T., Zhang, B. W., Ma, H. L. (2016). A meshless collocation approach with barycentric rational interpolation for two-dimensional hyperbolic telegraph equation. *Applied Mathematics and Computation, 279,* 236–248. DOI 10.1016/j.amc.2016.01.022.
24. Kumar, S. (2014). A new analytical modeling for fractional telegraph equation via Laplace transform. *Applied Mathematical Modelling, 38(13),* 3154–3163. DOI 10.1016/j.apm.2013.11.035.
25. Berrut, J. P., Baltensperger, R., Mittelmann, H. D. (2005). Recent developments in barycentric rational interpolation. *International Series of Numerical Mathematics, 151,* 27–51.
26. Fu, Y. P. (2008). Compact fourth-order finite difference schemes for Helmholtz equation with high wave numbers. *Journal of Computational Mathematics, 26,* 98–111.
27. Singer, I., Turkel, E. (2006). Sixth-order accurate finite difference schemes for the Helmholtz equation. *Journal of Computational Acoustics, 214(03),* 339–351. DOI 10.1142/S0218396X06003050.
28. Wu, T. T., Xu, R. M. (2018). An optimal compact sixth-order finite difference scheme for the Helmholtz equation. *Computers and Mathematics with Applications, 75(7),* 2520–2537. DOI 10.1016/j.camwa.2017.12.023.
29. Zhuang, Y., Sun, X. H. (2000). A high order ADI method for separable generalized Helmholtz equations. *Advances in Engineering Software, 31(8–9),* 585–591. DOI 10.1016/S0965-9978(00)00026-0.
30. Liu, C. S., Kuo, C. L. (2016). A multiple-scale Pascal polynomial triangle solving elliptic equations and inverse Cauchy problems. *Engineering Analysis with Boundary Elements, 62,* 35–43. DOI 10.1016/j.enganabound.2015.09.003.