Journal of Nonlinear and Convex Analysis
Volume 26, Number 7, 2025, 2127–2144

Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

THE FINITE ELEMENT METHOD BASED ON ORTHOGONAL STEP FUNCTION GROUPS

JICHAO SUN, SHUYING CHEN, AND XIAOKUN YANG*

Abstract. In the conventional finite element method, a polynomial function, which uses the coordinate values of each node in the element as parameters, serves as an interpolation function to approximate the field function piecewise. When solving large and complex engineering problems in this manner, the operator equation is transformed into a system of linear algebraic equations, with the number of unknowns reaching hundreds of thousands or even millions, making it suitable for iterative methods. However, its convergence properties and convergence rates are the key to solving these problems. This study aimed to propose a finite element method of orthogonal step function groups to solve operator equations for improving convergence. This method took the linear combination of orthogonal step functions as the approximation function of partition elements. In this study, the Galerkin method was used to transform the differential operator equation into a system of linear algebraic equations. Several approximate diagonalization methods of the coefficient matrix of the system of linear algebraic equations were provided to achieve better convergence performance. Also, two examples of solving the Poisson equation was used to demonstrate the performance. The study also addressed the treatment of boundary conditions for problems with different boundary values and discussed the completeness of the orthogonal step function group, thus providing a basis for their application in the Galerkin method.

1. Introduction

The finite element method (FEM) is a numerical analysis technique developed alongside computer technology in the mid-20th century. With its rigorous logic and clear concepts, FEM is widely used for analyzing various continuous media problems across various fields of science and engineering. It can effectively handle and solve complex problems, such as irregular boundary shapes, complex boundary functions, and objects with variable physical properties. The FEM is solved using matrix operations, which makes it well-suited for computer programming. These advantages give it great vitality. The application of FEM has expanded from two-dimensional (2D) problems in elasticity mechanics to three-dimensional (3D), plate, and shell

2020 *Mathematics Subject Classification.* 78M10, 65N30.

Key words and phrases. Finite element method, Galerkin method, orthogonal approximate eigenfunction group, OSFG.

This work was supported by the National Natural Science Foundation of China under Grant No. 62002084, the Opening Project of Automotive New Technique of Anhui Province Engineering Technology Research Center under Grant No. QCKJ202204A, the Key Projects of the Provincial Education Department under Grant No.2024AH050101, Supported by Anhui Province Key Laboratory of Intelligent Car Wire-Controlled Chassis System of Anhui Polytechnic University under Grant No. QCKJJ202403 and No.HX-2023-12-120.

*Corresponding author.

problems. It has also grown from addressing static equilibrium problems to tackling stability problems, dynamic problems and wave propagation problems. FEM scope has further broadened from solid mechanics to fluid mechanics, heat conduction, electromagnetic issues, and other continuous media domains [22, 11, 18, 4, 9]. It can be predicted that with the development of modern science, computational mathematics and computer technology, finite element method, as a numerical analysis tool with solid theoretical foundation and widely applied effectiveness, will play a greater role in economic construction and the development of science and technology, and will be further developed and improved[16, 3, 19, 12, 20, 10].

The numerical methods for solving linear algebraic equations can be classified into two categories: direct methods and iterative methods. The disadvantage of the direct methods is that they do not always guarantee accurate results, this is mainly because the accumulation and propagation of errors in the arithmetic operations cannot be controlled in the process of multiple elimination unknowns and back substitutions. For high-order equations, the calculation accuracy is significantly reduced due to the continuous accumulation of round-off errors. Although higher-order equations with sparse coefficient matrices can be solved using direct methods, maintaining matrix sparsity in operation is challenging. Additionally, direct methods have certain shortcomings, such as complex calculation formulas, complex procedures, and the requirement for more storage units. These methods are commonly used to solve small to medium-sized linear equation systems. In contrast, the iterative methods preserve matrix sparsity and offer simple computations and easier programming. These methods do not require storing zero elements of the coefficient matrix, which allows them to occupy less storage space. Hence, iterative methods are highly effective for solving large linear algebraic equations, especially for large, sparse matrices. However, the success of these methods depends on their convergence and the speed of convergence [6, 1].

The finite element analysis of large-scale and complex engineering problems can have hundreds of thousands or even millions of degrees of freedom. Linear algebraic equations can be obtained after discretizing a complex continuum. Ensuring the convergence and stability of these numerical solutions is the main focus of finite element theory.

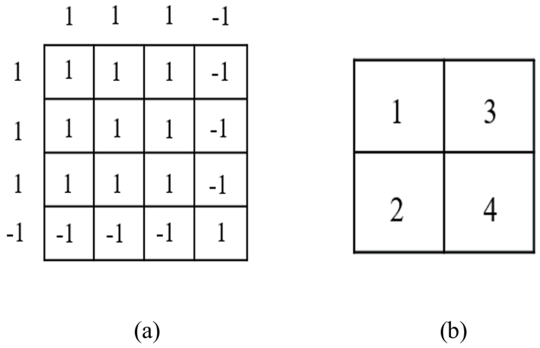
In conventional FEM, a polynomial function, using the coordinate values of each node in mesh elements as parameters, is used as an interpolation function, approximating the field function piecewise to complete the discretization process. The present study used the orthogonal step function group (OSFG)-based FEM to improve the convergence of linear algebraic equations. It took the linear combination of a set of orthogonal step functions as the approximation function of the partition element. It could improve the convergence, mitigate the shortcomings of the conventional FEM in solving singularity problems, and reduce the amount of numerical integration operations.

2. Orthogonal step function group

2.1. **Defining the OSFG.**

First, let us divide the interval [a, b] into n equal parts, assuming that a function exists whose value is 1 in one of these parts and 0 in the remaining ones. Such a function is called the unit pulse basis function in n equally divided intervals. For example, the unit pulse basis function, which is equal to 1 in the ith part and 0 in other parts, is denoted as p_i. The linear combination of the unit impulse basis functions $p_i (i = 1, 2, \ldots, n)$ has the form of $\alpha\,(x) = \sum_{i=1}^{n} \alpha_i p_i$ where α_i is a real coefficient. It is called a step function on the interval of n equal partitions. The unit impulse basis function p_i, $i = 1, 2, \ldots, n$ is considered the n-dimensional space base. All step functions in the interval of n equal partitions constitute the entire linear space as $\alpha\,(x) = \sum_{i=1}^{n} \alpha_i p_i$. If n individual of functions on the interval of n equal partitions are orthogonal, they are called "n-order orthogonal step function group"(hereinafter referred to as n-order OSFG). These functions correspond one-to-one with an n-order row orthogonal matrix. For example, the Walsh matrix corresponds to the Walsh OSFG, and the discrete cosine transform matrix corresponds to the discrete cosine OSFG. Similarly, 2D and 3D OSFG can be constructed. The interval [a, b] in 1D becomes a bounded and closed region in the multidimensional condition. At this time, the region Ω is divided into basic graphs. The two-dimensional basic graphs can be triangles or parallelograms. The 3D basic graphs can be cubes or other geometric shapes. A function that is equal to 1 on a basic graph and is equal to 0 on any other basic graphs is called unit pulse basis function, and a function that is constant on each basic graph is also a step function. Our research was initially limited to region that could only be divided into basic graphs. As long as the number of orthogonal step functions in the region Ω was equal to the number of basic graphs into which the region Ω was divided, the orthogonal step functions on the region Ω and the unit pulse basis functions on the basic graphs can linearly represent each other. 2D or 3D n orthogonal step functions on n basic graphs are also referred to as n-order orthogonal step function groups, or n-order OSFGs.

2.2. **Two-dimensional OSFG examples.** In Figure 1, (1, 1) and (1, -1) are two orthogonal vectors. The product of the orthogonal vector components in the horizontal and vertical directions forms a 2D vector. The number in the small block in Figure 1(a) is the product of the orthogonal vector components in the horizontal and vertical directions outside the corresponding large block in the row and column.



	1	1	1	-1
1	1	1	1	-1
1	1	1	1	-1
1	1	1	1	-1
-1	-1	-1	-1	1

(a)

1	3
2	4

(b)

Figure 1. (a)Formation of 2D vectors (b)The method of segmentation.

Figure 1(a) is divided into four small blocks according to Figure 1(b), forming a square-shape basis of four 2D orthogonal systems, as shown in Figure 2. The basis

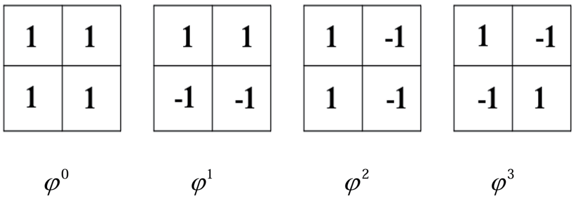


FIGURE 2. The square-shape base of a 2 D orthogonal system.

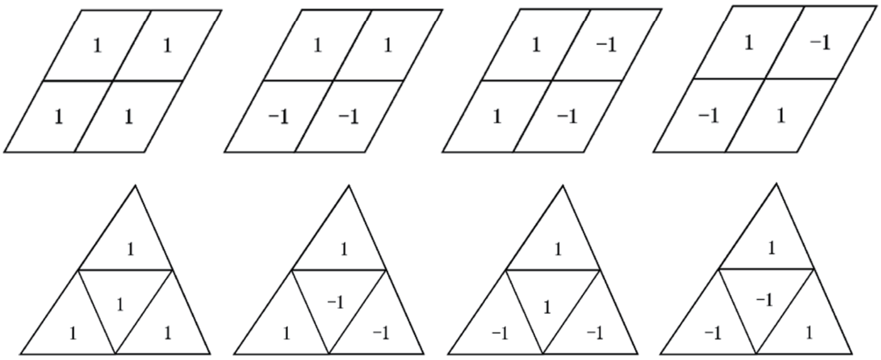


FIGURE 3. Basis of the 2D function group of parallelograms and triangles.

of the 2D function group of parallelograms and triangles is shown in Figure 3. Each graph of parallelograms (including squares) and triangles is called the basic graph.

The length, width, and height of a cube are each divided into two equal parts, resulting in the cube being divided into eight smaller, equal parts.

The basis of the orthogonal function group in 3D space can be derived by imitating the construction method of the orthogonal function group in 2D space.

3. Orthogonal approximate eigenfunction group

Let p_i be a sequence of unit pulse basis functions with equal width connected at the beginning and end. Their linear combination $\alpha\left(x\right) = \sum_{i=-\infty}^{+\infty} \alpha_i p_i$ forms a step function on an infinite interval, where α_i is called the strength of the pulse basis function. The interval occupied by the unit pulse basis function p_i is Δ_i, and the length is h. When using the Galerkin method to solve the differential operator equation, a difference operator L_D approximately replaces the differential operator. When the step function $\sum_{i=-\infty}^{+\infty} u_i p_i$ on the infinite interval passes through the system L_D, it becomes the step function$\sum_{i=-\infty}^{+\infty} r_i p_i$. If the difference operator is $L_D u_i = u_{i-1} - 2u_i + u_{i+1}$, let the input sequence u_i be$\ldots 0$, u_n, u_{n+1}, u_{n+2}, u_{n+3}, 0,... [14]. We can get the output sequence r_i as$\ldots 0$, u_n, r_n, r_{n+1}, r_{n+2}, r_{n+3}, u_{n+3}, 0,..., and a fourth-order matrix relationship exists.

$$\begin{bmatrix}-2 & 1 & 0 & 0\\ 1 & -2 & 1 & 0\\ 0 & 1 & -2 & 1\\ 0 & 0 & 1 & -2\end{bmatrix}\begin{bmatrix}u_n\\ u_{n+1}\\ u_{n+2}\\ u_{n+3}\end{bmatrix}=\begin{bmatrix}r_n\\ r_{n+1}\\ r_{n+2}\\ r_{n+3}\end{bmatrix}. \tag{3.1}$$

Assuming the coefficient matrix of equation (3.1) is A, the characteristic equation is $|\ \lambda\ E-A|=0$, the eigenvalues are solved as λ_k, $k=1,\ 2,\ 3,\ 4$, and the corresponding orthogonal characteristic functions are $\left(\varphi_n^k,\varphi_{n+1}^k,\varphi_{n+2}^k,\varphi_{n+3}^k\right)^{\mathrm{T}}$, where T represents transpose. The step function group $\varphi^k(x)=\sum_{i=n}^{n+3}\varphi_i{}^k p_i$, $k=1,\ 2,\ 3,\ 4$ is called a fourth-order orthogonal approximation characteristic function group of the difference operator L_D, and its pulse basis function intensity sequence is $\ldots 0,\varphi_n^k,\varphi_{n+1}^k,\varphi_{n+2}^k,\varphi_{n+3}^k,0\ldots.$ After applying the second-order difference operator L_D, we get $\ldots 0,\varphi_n^k,\lambda_k\varphi_n^k,\lambda_k\varphi_{n+1}^k,\lambda_k\varphi_{n+2}^k,\lambda_k\varphi_{n+3}^k,\varphi_{n+3}^k,0\ldots.$

Next, we take $\left(0,\varphi_n^k,\varphi_{n+1}^k,\varphi_{n+2}^k,\varphi_{n+3}^k,0\right)$ as the kth row of matrix U to form matrix U and take $\left(\varphi_n^k,\lambda_k\varphi_n^k,\lambda_k\varphi_{n+1}^k,\lambda_k\varphi_{n+2}^k,\lambda_k\varphi_{n+3}^k,\varphi_{n+3}^k\right)$ as the kth row of matrix R to form matrix R. These matrices take the following forms:

$$\begin{aligned}U&=\begin{bmatrix}0 & \varphi_n^1 & \varphi_{n+1}^1 & \varphi_{n+2}^1 & \varphi_{n+3}^1 & 0\\ 0 & \varphi_n^2 & \varphi_{n+1}^2 & \varphi_{n+2}^2 & \varphi_{n+3}^2 & 0\\ 0 & \varphi_n^3 & \varphi_{n+1}^3 & \varphi_{n+2}^3 & \varphi_{n+3}^3 & 0\\ 0 & \varphi_n^4 & \varphi_{n+1}^4 & \varphi_{n+2}^4 & \varphi_{n+3}^4 & 0\end{bmatrix}\\ &=\begin{bmatrix}0 & -1.3450 & 2.1763 & -2.1763 & 1.3450 & 0\\ 0 & 1.5747 & -0.9732 & -0.9732 & 1.5747 & 0\\ 0 & 0.8313 & 0.5137 & -0.5137 & -0.8313 & 0\\ 0 & 0.1420 & 0.2298 & 0.2298 & 0.1420 & 0\end{bmatrix}.\end{aligned} \tag{3.2}$$

$$\begin{aligned}R&=\begin{bmatrix}\varphi_n^1 & \lambda_1\varphi_n^1 & \lambda_1\varphi_{n+1}^1 & \lambda_1\varphi_{n+2}^1 & \lambda_1\varphi_{n+3}^1 & \varphi_{n+3}^1\\ \varphi_n^2 & \lambda_2\varphi_n^2 & \lambda_2\varphi_{n+1}^2 & \lambda_2\varphi_{n+2}^2 & \lambda_2\varphi_{n+3}^2 & \varphi_{n+3}^2\\ \varphi_n^3 & \lambda_3\varphi_n^3 & \lambda_3\varphi_{n+1}^3 & \lambda_3\varphi_{n+2}^3 & \lambda_3\varphi_{n+3}^3 & \varphi_{n+3}^3\\ \varphi_n^4 & \lambda_4\varphi_n^4 & \lambda_4\varphi_{n+1}^4 & \lambda_4\varphi_{n+2}^4 & \lambda_4\varphi_{n+3}^4 & \lambda_4\varphi_{n+3}^4\end{bmatrix}\\ &=\begin{bmatrix}-1.3450 & 0.3717 & -0.6015 & 0.6015 & -0.3717 & 1.3450\\ 1.5747 & -0.6015 & 0.3717 & 0.3717 & -0.6015 & 1.5747\\ 0.8313 & 0.6015 & -0.3717 & 0.3717 & 0.6015 & -0.8313\\ 0.1420 & -0.3717 & -0.6015 & -0.6015 & -0.3717 & 0.1420\end{bmatrix}.\end{aligned} \tag{3.3}$$

Using the orthogonality between $\varphi^k(x)=\sum_{i=n}^{n+3}\varphi_i{}^k p_i$, $k=1,\ 2,\ 3,\ 4$, it is known that $\boldsymbol{U}\cdot\boldsymbol{R}^T$ is a diagonal matrix.

Similarly, for a second-order difference operator system $L_D u_{\mathrm{i}}=u_{i-1}-2u_{\mathrm{i}}+u_{i+1}$, when the input sequence is... 0, u_n, u_{n+1}, $0,\ldots$, the input and output can also obtain the following second-order matrix relationship:

$$\begin{bmatrix}-2 & 1\\ 1 & -2\end{bmatrix}\begin{bmatrix}u_n\\ u_{n+1}\end{bmatrix}=\begin{bmatrix}r_n\\ r_{n+1}\end{bmatrix}. \tag{3.4}$$

By obtaining the eigenvectors of the coefficient matrix, the corresponding U, R can also be obtained.

$$U=\begin{bmatrix}0&1&1&0\\0&1&-1&0\end{bmatrix}. \tag{3.5}$$

$$R=\begin{bmatrix}1&-1&-1&1\\1&-3&3&-1\end{bmatrix}. \tag{3.6}$$

Here $\begin{bmatrix}0&1&1&0\\0&1&-1&0\end{bmatrix}$ is a matrix representation of the second-order orthogonal approximation eigenfunction group. It can also be simplified as $\begin{bmatrix}1&1\\1&-1\end{bmatrix}$.

When the input sequence is $\ldots 0,\ u_n,\ u_{n+1},\ldots,u_{n+K-1},\ 0,\ldots,$ the relationship between the input and the output can be obtained as a Kth-order square matrix, and the concept of Kth-order orthogonal approximate eigenfunction groups can also be introduced. Generally, for a difference operator L_D, if a step function φ satisfies $L_D\varphi \approx \lambda\varphi$, where λ is constant, then the step function φ is called an approximate eigenfunction of the difference operator L_D.

4. OSFG-based FEM

4.1. Related concepts.

The linear combination of a set of orthogonal step functions is taken as the approximation function of the subdivision element. Such OSFG-based FEM can be used for solving partial differential and integral equations. This study was limited to partial differential equations. In OSFG-based FEM, the coefficient matrix M of the linear algebraic equation transformed from the partial differential equation is a sparse matrix close to the diagonal block matrix composed of small matrix blocks. It is multiplied by the diagonal block matrix, in which each small matrix block is the inverse matrix of each small matrix block in the matrix M, to achieve a matrix close to the diagonal matrix. If the orthogonal step function group adopts the orthogonal approximate eigenfunction group, each small matrix block of the matrix M is a small diagonal matrix, and such FEM becomes orthogonal approximate eigenfunction group (OAEG) based FEM. The approximate diagonalization of the coefficient matrix of linear algebraic equations have improved the convergence of the iterative method. The following example illustrate the diagonalization method.

Example 4.1. Find the electrostatic potential between two infinite large parallel plates Φ: A board is located at $x = 0$, $\Phi = 0$ V; the other is at $x = 1$ m, $\Phi = 1$ V. The two parallel plates are filled with a medium with a dielectric constant ε (F/m). The charge density varies between them $\rho(x) = -(x+1)\varepsilon$ (C/m^3). This problem can be mathematically described by Poisson's equation and reduced to a second-order differential equation: $\frac{d^2\varphi}{dx^2} = x + 1$, $0 < x < 1$. Its boundary conditions are:

$$\Phi|_{x=0} = 0,\ \Phi|_{x=1} = 1. \tag{4.1}$$

Let $\tilde{\varPhi} = \psi\left(n\right) + \sum_{i=1}^{8} c_i\varphi_i\left(n\right)$, where $\varphi_i\left(n\right)$ is the basis function of the OSFG, $\psi\left(n\right)$ is used to satisfy boundary conditions, and c_i is an undetermined coefficient. We expand the domain of the definition of $\psi\left(n\right)$ and $\varphi_i\left(n\right)$ from the interval [0,1] to the interval $[-\frac{1}{8}, \frac{9}{8}]$ and divide the interval $[-\frac{1}{8}, \frac{9}{8}]$ into 10 equal small intervals. Within each equal small interval, $\psi\left(n\right)$ and $\varphi_i\left(n\right)$ are constants, and $\psi\left(n\right)$ is 1 within a small interval $[1, \frac{9}{8}]$, taking zero values in the remaining small intervals. Thus, interval [0,1] includes eight small intervals. Within each small interval, the value of $\varphi_i\left(n\right)$ is the component value of the row vector of the following matrix:

$$U_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The symbol $\otimes$ stands for Kronecker product; outside the interval [0,1], the value of $\varphi_{\mathrm{i}}\left(n\right)$ is zero. Besides, $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ is the OAEG used to approach the function value on each partition element of the four equal parts partitioned from the interval [0,1]. Therefore, the method is the OAEG-based FEM.

The difference equation is derived from the differential equation $\frac{d^2\varPhi}{dx^2} = x + 1$, and the second-order derivative $\frac{d^2\varPhi(x)}{dx^2}$ is replaced by the second-order difference quotient $\frac{1}{h^2}L_D\varPhi\left(n\right) = \frac{1}{h^2}[\varPhi\left(n-1\right) + \varPhi\left(n+1\right) - 2\varPhi\left(n\right)]$. Thus, we can obtain $\frac{1}{h^2}L_D\varPhi\left(n\right) = x\left(n\right) + 1$ and replace $\varPhi\left(n\right)$ with $\tilde{\varPhi}\left(n\right) = \psi\left(n\right) + \sum_{i=1}^{8} c_i\varphi_{\mathrm{i}}\left(n\right)$ to obtain

$$\frac{1}{h^2}L_D\left(\psi\left(n\right) + \sum_{i=1}^{8} c_i\varphi_i\left(n\right)\right) = x\left(n\right) + 1. \tag{4.2}$$

where $\psi\left(n\right)$, $\varphi_i\left(n\right)$, and $x(n)$ represent the intensity sequences of pulse basis functions of each step function. $x(n)$ is taken as the value of x at the midpoint of each small interval. For step functions $\alpha\left(x\right) = \sum_{i=n}^{n+K\text{-}1}\alpha_i p_i$ and $\beta\left(x\right) = \sum_{i=n}^{n+K\text{-}1}\beta_i p_i$, we define their inner product as: $\left\langle\alpha\left(x\right), \beta\left(x\right)\right\rangle = \sum_{i=n}^{n+K\text{-}1}\alpha_i\beta_i$. According to the Galerkin method, we take the dot product of both sides of formula (4.2) with $\varphi_j\left(n\right)$ to get

$$\begin{aligned} \frac{1}{h^2}\sum_{i=1}^{8}\mathrm{c}_i\left\langle L_D\varphi_i\left(n\right), \varphi_j\left(n\right)\right\rangle &+ \frac{1}{h^2}\left\langle L_D\psi\left(n\right), \varphi_j\left(n\right)\right\rangle \\ &= \left\langle x\left(n\right) + 1, \varphi_j\left(n\right)\right\rangle, j = 1, 2, \ldots, 8. \end{aligned} \tag{4.3}$$

Then, we find the inner products $\left\langle L_D\varphi_i\left(n\right), \varphi_j\left(n\right)\right\rangle$, $\left\langle L_D\psi\left(n\right), \varphi_j\left(n\right)\right\rangle$, and $\left\langle x\left(n\right) + 1, \varphi_j\left(n\right)\right\rangle$ and substitute them into equation (4.2) to obtain

$$\begin{bmatrix} -2 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -6 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -2 & 0 & 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & -6 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -2 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & 0 & -6 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -2 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & -6 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{bmatrix} = \frac{1}{64} \begin{bmatrix} 2.25 \\ -0.125 \\ 2.75 \\ -0.125 \\ 3.25 \\ -0.125 \\ 3.75 \\ -0.125 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}. \tag{4.4}$$

Using the matrix notation, equation (4.4) is expressed as $M_1 \bullet c = \frac{1}{64}I\text{-}H$, where $c = [c_1,c_2,c_3,c_4,c_5,c_6,c_7,c_8]^{\mathrm{T}}$, M_1 is the coefficient matrix of the equation, and its elements are $\langle L_D\varphi_i(n), \varphi_j(n)\rangle = m_{ij}^1$, because $[\varphi_{\mathrm{i}}(1)\ \varphi_{\mathrm{i}}(2)...\varphi_{\mathrm{i}}(8)]$ is the row vector of matrix U_1, the difference operator L_D acts on each row of matrix U_1 to obtain matrix R_1. Let ${R_1}^{\mathrm{T}}$ be the transposed matrix of R_1; then $M_1 = U_1 \cdot {R_1}^{\mathrm{T}}$. Assume that $M_1 = X + D + S$, where X is the lower triangular matrix, D is the diagonal matrix, and S is the upper triangular matrix. Taking $A_1 = -D^{-1}(X + S)$, we can transform equation (4.4) into $c{=}A_1c + f_1$ and derive the spectral radius $\rho(A_1) = 0.8851$. Therefore, the equations are convergent using the iterative method [21, 2], yielding = [0.0480, -0.0215, 0.1723, -0.0411, 0.3824, -0.0645, 0.6942, -0.0919]$^{\mathrm{T}}$. The calculated values of $\tilde{\Phi}(n)$ are listed in Table 1.

TABLE 1. Calculated values of $\tilde{\Phi}(n)$

Interval	$[-\frac{1}{8}\ \ 0]$	$[0, \frac{1}{8}]$	$[\frac{1}{8}, \frac{1}{4}]$	$[\frac{1}{4}, \frac{3}{8}]$	$[\frac{3}{8}, \frac{1}{2}]$	$[\frac{1}{2}, \frac{5}{8}]$	$[\frac{5}{8}, \frac{3}{4}]$	$[\frac{3}{4}, \frac{7}{8}]$	$[\frac{7}{8}, 1]$	$[1, \frac{9}{8}]$
$\tilde{\Phi}(n)$	0	0.0265	0.0696	0.1312	0.2133	0.3179	0.4469	0.6023	0.7860	1

4.2. **Improvement in the OSFG-based FEM convergence.**

Method 4.2. Equation (4.4) $M_1 \bullet c = \frac{1}{64}I\text{-}H$ can be solved by the iterative method, and its convergence performance can be improved. It can be achieved using the matrix row transformation method: $\begin{bmatrix} -16 & 2 & -9 & -3 \\ -2 & 4 & -3 & -1 \\ -9 & 3 & -16 & -2 \\ -3 & 1 & -2 & -4 \end{bmatrix} \times \begin{bmatrix} -2 & 0 & 1 & 1 \\ 0 & -6 & -1 & -1 \\ 1 & -1 & -2 & 0 \\ 1 & -1 & 0 & -6 \end{bmatrix} =$ $\begin{bmatrix} 20 & & & \\ & -20 & & \\ & & 20 & \\ & & & 20 \end{bmatrix}$. Let $\tilde{N} = \begin{pmatrix} -16 & 2 & -9 & -3 \\ -2 & 4 & -3 & -1 \\ -9 & 3 & -16 & -2 \\ -3 & 1 & -2 & -4 \end{pmatrix}$ and $\begin{pmatrix} \tilde{N} & 0 \\ 0 & \tilde{N} \end{pmatrix} = K$, $M_2 =$

$$\begin{pmatrix} \tilde{N} & 0 \\ 0 & \tilde{N} \end{pmatrix} M_1 = \begin{pmatrix} 20 & 0 & 0 & 0 & -6 & -6 & 0 & 0 \\ 0 & -20 & 0 & 0 & -2 & -2 & 0 & 0 \\ 0 & 0 & 20 & 0 & -14 & -14 & 0 & 0 \\ 0 & 0 & 0 & 20 & 2 & 2 & 0 & 0 \\ 0 & 0 & -14 & 14 & 20 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & -20 & 0 & 0 \\ 0 & 0 & -6 & 6 & 0 & 0 & 20 & 0 \\ 0 & 0 & -2 & 2 & 0 & 0 & 0 & 20 \end{pmatrix}.$$

$M_2 \bullet c = K\left(\frac{1}{64}I\text{-}H\right)$is obtained by multiplying the two ends of equation (4.4) with matrix K; it is written in a form that is easy to iterate as $c{=}A_2c{+}f_2$. Thus, we can calculate the corresponding spectral radius $\rho\,(A_2) = 0.8$. The method combines two small pieces of the M_1 matrix into one large piece and then diagonalizes it. This method can be used for the general OSFG-based FEM and repeated until the spectral radius satisfies the requirements.

Method 4.3. We divide the [a, b] interval into P equal parts, each being a partition element. The OSFG-based FEM approach implies taking a linear combination of the members of N-order OSFG as the approximation function of each partition element, where $P \times N$ orthogonal step functions are called a P-arrangement of N-order OSFG. As the whole-domain functions on interval [a, b], they take the values of orthogonal functions in their respective partition units and zero outside their respective partition units.

First, we take the second-order OAEG $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}$ to form four independent orthogonal functions with more zero elements, thus obtaining the following fourth-order OSFG: $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

This fourth-order OSFG is the extended group of the second-order OAEG, which is treated as the approximating function in the FEM partition unit. The second-order arrangement of the extended group of the second-order OAEG is

$$U_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.5}$$

We use the difference operator $L_D\varPhi\,(n) = \varPhi\,(n-1) + \varPhi\,(n+1) - 2\varPhi\,(n)$, which acts on each row of U_3, to obtain matrix R_3. Then, we derive matrix ${R_3}^{\mathrm{T}}$ as the transpose of R_3. Finally, we calculate $M_3 = U_3 \cdot {R_3}^{\mathrm{T}}$, which is the coefficient matrix of a system of linear algebraic equations transformed by an operator equation when the row vectors of U_3 are the basis functions, as follows:

$$M_3 = U_3 \cdot {R_3}^{T} = \begin{bmatrix} -2 & 1 & 1 & 0 & 0 & & & 0 \\ 1 & -2 & 0 & 1 & 0 & & & \\ 1 & 0 & -6 & -1 & 0 & & & \\ 0 & 1 & -1 & -2 & 1 & & & 0 \\ 0 & & 0 & 1 & -2 & 1 & 1 & 0 \\ 0 & & & 0 & 1 & -2 & 0 & 1 \\ 0 & & & 0 & 1 & 0 & -6 & -1 \\ 0 & & & 0 & 0 & 1 & -1 & -2 \end{bmatrix}. \tag{4.6}$$

The spectral radius corresponding to matrix M_3 is $\rho\,(A_3) = = 0.9207$. The spectral radius corresponding to the matrix M_5 is still large, and hence the further diagonalization of matrix M_3 is required. This can be achieved by the following two methods.

(1) Let $N = \begin{bmatrix} -2 & 1 & 1 & 0 \\ 1 & -2 & 0 & 1 \\ 1 & 0 & -6 & -1 \\ 0 & 1 & -1 & -2 \end{bmatrix}$, and the inverse matrix of N is N^{-1}. Let $\begin{pmatrix} N^{-1} & 0 \\ 0 & N^{-1} \end{pmatrix} M_3 = M_4$, and the spectral radius corresponding to M_4 can be obtained as $\rho\left(A_4\right) = 0.8$.

(2) Alternatively, we perform row transformation on

$$\begin{bmatrix} -2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -2 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -6 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & -2 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and get

$$\begin{bmatrix} 10 & 0 & 0 & 0 & -8 & -5 & -1 & -2 \\ 0 & 2 & 0 & 0 & -1 & -2 & 0 & -1 \\ 0 & 0 & -10 & 0 & 1 & 0 & 2 & -1 \\ 0 & 0 & 0 & 10 & -2 & -5 & 1 & -8 \end{bmatrix}.$$

If $\tilde{N} = \begin{bmatrix} -8 & -5 & -1 & -2 \\ -1 & -2 & 0 & -1 \\ 1 & 0 & 2 & -1 \\ -2 & -5 & 1 & -8 \end{bmatrix}$, we get $M_5 = \begin{pmatrix} \tilde{N} & 0 \\ 0 & \tilde{N} \end{pmatrix} M_3$

This yields:

$$M_5 = \begin{bmatrix} 10 & 0 & 0 & 0 & -2 & & & 0 \\ 0 & 2 & 0 & 0 & -1 & & & \\ 0 & 0 & -10 & & -1 & & & \\ 0 & 0 & 0 & 10 & -8 & & & 0 \\ 0 & & 0 & -8 & 10 & 0 & 0 & 0 \\ & & & -1 & 0 & 2 & 0 & 0 \\ & & & 1 & 0 & 0 & -10 & \\ 0 & & 0 & -2 & 0 & 0 & 0 & 10 \end{bmatrix}. \tag{4.7}$$

The spectral radius corresponding to M_5 is $\rho\left(A_5\right) = 0.8$.

The equation $Mc{=}B$ is rewritten as $c{=}Ac{+}f$, the initialization vector $c^{(0)}$ is given arbitrarily, and we can get a vector sequence $\left(c^{(0)}c^{(1)}c^{(2)}\ldots c^{(k)}\ldots\right)$ using the iterative formula $c^{(k)} = Ac^{(k-1)} + f$ $(k = 1, 2, \ldots)$, $\|A\|$ is the norm of matrix A, and $\frac{\left\|c^{(k)}-c*\right\|}{\left\|\mathrm{c}^{(0)}-\mathrm{c}*\right\|} \leq \left\|A^k\right\|$ is obtained by $c^{(k)} - c* = A^k\left(c^{(0)} - c*\right)$.

To make $\frac{\left\|c^{(k)}-c*\right\|}{\left\|c^{(0)}-c*\right\|} \leq \mu$, as long as $\left\|A^k\right\| \leq \mu$, we can get $k \geq \frac{-\log\mu}{-\frac{1}{k}\log\left\|A^k\right\|}$.

It was proven in previous studies [21, 2] that $\lim\limits_{k\to\infty}\left\|A^k\right\|^{\frac{1}{k}} = \rho\left(A\right)$. For large k values, we can use the following approximation of k to achieve the solution accuracy μ :

$$k \approx \frac{-\log\mu}{-\log\rho\left(A\right)}. \tag{4.8}$$

As shown in equation (4.8), the smaller the value of $\rho\left(A\right)$, the smaller the required k is. For example, using M_5 as the coefficient matrix of the system of linear equations in the aforementioned example, when the number of iterations is k_5, the precision drops to 0.0001 times the initial error, that is $\mu = 0.0001$, and we get $k_5 = 41.2754$. If one ignores the number of addition operations and only considers the number of multiplication and division operations, the matrix M_5 needs to multiply and divide eight times per iteration. For matrix M_5, when $\left\|c^{(k)} - c*\right\|$ drops to 0.0001 times of the initial distance $\left\|c^{(0)} - c^{(*)}\right\|$, a total of 336 (=42×8) multiplication or division operations are required.

It can be seen that $c^{(k)}$ is convergent to $c*$, and the descending speed of distance $\left\|c^{(k)} - c*\right\|$ between them is related to two factors. Smaller $\rho(A)$ values and more zero elements of matrix A correspond to faster convergence.

In actual calculations, the following methods can be used due to the unknown $\left\|c^{(0)} - c^{(*)}\right\|$: it is known from the discipline of computational methods, $\left\|c^{(k)} - c*\right\| \leq \frac{\|A\|^k}{1-\|A\|}\left\|c^{(1)} - c^{(0)}\right\|$ we get $\frac{\left\|c^{(k)}-c*\right\|}{\left\|c^{(1)}-c^{(0)}\right\|} \leq \frac{\|A\|^k}{1-\|A\|}$ when $\|A\| < 1$, $c^{(\mathrm{k})}$ is convergent to $c*$. The distance $\left\|c^{(k)} - c*\right\|$ between them decreases to ε multiple of the initial distance $\left\|c^{(1)} - c^{(0)}\right\|$ and is less than $\frac{\|A\|^k}{1-\|A\|}$. (Let $c = (c_1, c_2, c_3, \ldots c_n)$, the 2-norm of vector C is $\|c\|_2 = \sqrt{\sum_{i=1}^{n} c_i^2}$ and $\lambda_{\max}\left(A^T A\right)$ is the largest eigenvalue of $A^T A$.) It can be obtained by taking the vector norm $\|\bullet\|$ as $\|\bullet\|_2$ and the matrix norm $\|A\|$ as $\|A\|_2 = \sqrt{\lambda_{\max}\left(A^T A\right)}$ to get

$$\varepsilon = \frac{\left\|c^{(k)}-c^{(*)}\right\|_2}{\left\|c^{(1)}-c^{(0)}\right\|_2} \leq \frac{\|A\|_2^k}{1-\|A\|_2}. \tag{4.9}$$

We can use equation (4.9) to obtain an estimation formula for the convergence number k after reaching a certain accuracy ε. The benefit of using an extended group of orthogonal approximate eigenfunction group is that the matrix M_5 has more zero elements when the matrix M_3 becomes the matrix M_5.

4.3. **Two-dimensional example.**

The FEM of the OSFG can also be used in two and three dimensions, as discussed in the following example.

Example 4.4. In a long, straight rectangular metal slot, the side wall and bottom surface potentials are zero, and the head cover potential is 100 (relative). As shown in Figure 4, inside the electrolytic cell, the potential φ meets the condition of $\Delta\varphi = 0$ (Δ is the Laplace operator).

$$\varphi = \begin{cases} \varphi_1 = 100 & \text{head cover potential} \\ \varphi_2 = 0 & \text{the potential of the side wall and bottom surface.} \end{cases} \tag{4.10}$$

We use the squares in Figure 2 as the OSFG, put $\varphi^0, \varphi^1, \varphi^2, \varphi^3$ to the position of each block 1,2,3,4 in Figure 4, and distinguish their locations by the subscript of $\varphi^0, \varphi^1, \varphi^2, \varphi^3$, the value on the outside of each block is zero.

Thus, we can get a group of 16 functions orthogonal to each other: $\varphi_1^0, \varphi_1^1, \varphi_1^2, \varphi_1^3$; $\varphi_2^0, \varphi_2^1, \varphi_2^2, \varphi_2^3$; $\varphi_3^0, \varphi_3^1, \varphi_3^2, \varphi_3^3$; $\varphi_4^0, \varphi_4^1, \varphi_4^2, \varphi_4^3$, where each superscript represents the

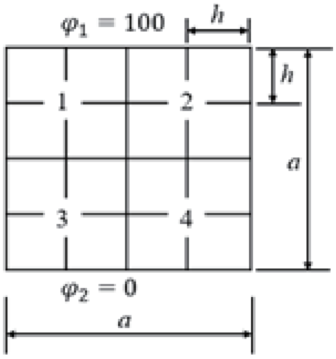
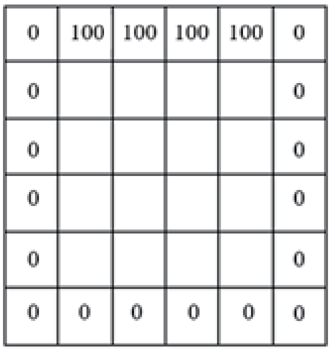


FIGURE 4. Long, straight rectangular metal slot.



0	100	100	100	100	0
0					0
0					0
0					0
0					0
0	0	0	0	0	0

FIGURE 5. Function graph of ψ.

type of function. Let ψ be the function in Figure 5. It takes zero values in the blank space, satisfies the boundary conditions, and is orthogonal to φ_j^i, where $i = 0, 1, 2, 3$, $j = 1, 2, 3, 4$. Let the solution that satisfies both the equation and the boundary conditions be $\varphi \approx \sum_{j=1}^{4} \sum_{i=0}^{3} c_j^i \varphi_j^i + \psi$. Next, we substitute it into equation $\Delta\varphi = 0$ to generate the remainder:

$$\varepsilon = \Delta\left(\sum_{j=1}^{4}\sum_{i=0}^{3} \mathrm{c}_j^i \varphi_j^i + \psi\right) = \sum_{j=1}^{4}\sum_{i=0}^{3} c_j^i \Delta\varphi_j^i + \Delta\psi. \tag{4.11}$$

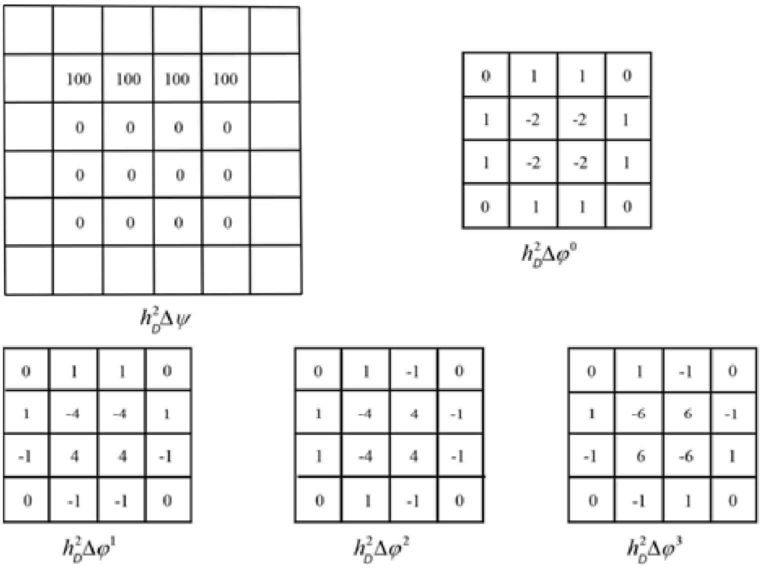
Using the Galerkin method $\langle \varepsilon, \varphi_n^m \rangle = 0$, $m = 0, 1, 2, 3$, $n = 1, 2, 3, 4$, we can get the equation for c_j^i :

$$\sum_{j=1}^{4}\sum_{i=0}^{3} \left\langle \Delta\varphi_j^i, \varphi_n^m \right\rangle c_j^i + \langle \Delta\psi, \varphi_n^m \rangle = 0. \tag{4.12}$$

So, let us solve $\Delta\varphi_j^i$ and $\Delta\psi$. Due to $\frac{\partial^2}{\partial z^2}\varphi = 0$, $\therefore \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ the difference quotient is used instead of differential. The midpoint of each small box is the sampling point, and the value of the sampling point is the value of the entire small box. The distance between the midpoints is the step size h. If a sampling point is marked as (r, s), the coordinates of the four adjacent sampling points are $(r + 1, s)$,$(r - 1, s)$,$(r, s + 1)$, and $(r, s - 1)$, where r, s are integers.

(4.13)
$$\Delta\varphi \approx \frac{1}{h^2}\left[\varphi\left(r+1,s\right)+\varphi\left(r-1,s\right)+\varphi\left(r,s+1\right)+\varphi\left(r,s-1\right)-4\varphi\left(r,s\right)\right] = \Delta_D\varphi.$$

From equation (4.13), we get $h^2\Delta_D\psi$, $h^2\Delta_D\varphi^0$, $h^2\Delta_D\varphi^1$, $h^2\Delta_D\varphi^2$, and $h^2\Delta_D\varphi^3$, as shown in Figure 6. It is clear that φ^0, φ^1, φ^2 and φ^3 form the orthogonal approximation characteristic function group of the operator $h^2\Delta_D$ (φ^0, φ^1, φ^2 and φ^3 all satisfy approximate equations $h^2\Delta_D\varphi \approx \lambda\varphi$). If the 1D difference operator is L_D, $L_D u_{\mathrm{i}} = u_{i-1} - 2u_i + u_{i+1}$. The orthogonal approximate characteristic functions of the operator L_D are $\varphi^1(x)$ and $\varphi^2(x)$, which satisfy $L_D\varphi^1(x) \approx \lambda_1\varphi^1(x)$. $L_D\varphi^2(x) \approx \lambda_2\varphi^2(x)$, wherein λ_1 and λ_2 are eigenvalues. Then, the tensor product $\varphi(x, y)$ of $\varphi^1(x)$ and $\varphi^2(y)$ is the orthogonal approximate characteristic function of the 2D difference operator $h^2\Delta_D$. The corresponding eigenvalue is $\lambda_1 + \lambda_2$.



	100	100	100	100	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

$h_D^2\Delta\psi$

0	1	1	0
1	-2	-2	1
1	-2	-2	1
0	1	1	0

$h_D^2\Delta\varphi^0$

0	1	1	0
1	-4	-4	1
-1	4	4	-1
0	-1	-1	0

$h_D^2\Delta\varphi^1$

0	1	-1	0
1	-4	4	-1
1	-4	4	-1
0	1	-1	0

$h_D^2\Delta\varphi^2$

0	1	-1	0
1	-6	6	-1
-1	6	-6	1
0	-1	1	0

$h_D^2\Delta\varphi^3$

FIGURE 6. Difference substitution after $h^2\Delta$ operator action.

In fact, the 2D difference operator $h^2\Delta_D\varphi$
$= \left[\varphi\left(r+1,s\right)+\varphi\left(r-1,s\right)+\varphi\left(r,s+1\right)+\varphi\left(r,s-1\right)-4\varphi\left(r,s\right)\right]$
$= \left[\varphi\left(r+1,s\right)-2\varphi\left(r,s\right)+\varphi\left(r-1,s\right)\right]+\left[\varphi\left(r,s+1\right)-2\varphi\left(r,s\right)+\varphi\left(r,s-1\right)\right]$. If the $\varphi\left(x{,}y\right)$ is tensor product of $\varphi^1(x)$ and $\varphi^2(y)$, then $h^2\Delta_D\varphi\left(x{,}y\right)$ is approximately transformed into $\lambda_1\varphi\left(x{,}y\right) + \lambda_2\varphi\left(x{,}y\right) = (\lambda_1 + \lambda_2)\varphi\left(x{,}y\right)$.

Using the difference operator to obtain the approximate values of the inner product $\left\langle h^2\Delta\varphi_j^i,\varphi_n^m\right\rangle$ and inner product $\left\langle h^2\Delta\psi,\varphi_n^m\right\rangle$ and substituting them into equation (4.12), the following equation can be obtained:

$$Mc+B=0. \tag{4.14}$$

where $B=[200,200,0,0,200,200,0,0,0,0,0,0,0,0,0,0]^{\mathrm{T}}$. $c=[c_1^0,\ c_1^1,\ c_1^2,\ c_1^3,\ c_2^0,\ c_2^1,\ c_2^2,\ c_2^3,\ c_3^0,\ c_3^1,\ c_3^2,\ c_3^3,\ c_4^0,\ c_4^1,\ c_4^2,\ c_4^3]^{\mathrm{T}}$. $M=\begin{bmatrix} M_{11} & M_{12}\\ M_{21} & M_{22}\end{bmatrix}$, thereinto

$$M_{11}=\begin{pmatrix} -8 & 0 & 0 & 0 & 2 & 0 & 2 & 0\\ 0 & -16 & 0 & 0 & 0 & 2 & 0 & 2\\ 0 & 0 & -16 & 0 & -2 & 0 & -2 & 0\\ 0 & 0 & 0 & -24 & 0 & -2 & 0 & -2\\ 2 & 0 & -2 & 0 & -8 & 0 & 0 & 0\\ 0 & 2 & 0 & -2 & 0 & -16 & 0 & 0\\ 2 & 0 & -2 & 0 & 0 & 0 & -16 & 0\\ 0 & 2 & 0 & -2 & 0 & 0 & 0 & -24 \end{pmatrix}$$

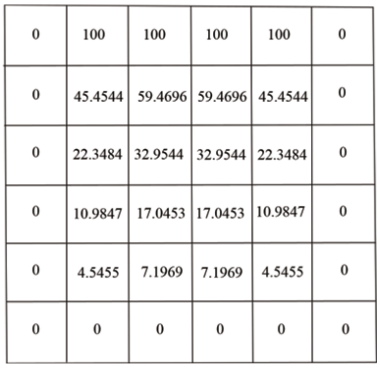
$$M_{12}=\begin{pmatrix} 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0\\ -2 & -2 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0\\ 0 & 0 & -2 & -2 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0\\ 0 & 0 & 0 & 0 & -2 & -2 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2\\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & -2 \end{pmatrix}$$

$$M_{21}=\begin{pmatrix} 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0\\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0\\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0\\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2\\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{pmatrix}$$

$$M_{22}=\begin{pmatrix} -8 & 0 & 0 & 0 & 2 & 0 & 2 & 0\\ 0 & -16 & 0 & 0 & 0 & 2 & 0 & 2\\ 0 & 0 & -16 & 0 & -2 & 0 & -2 & 0\\ 0 & 0 & 0 & -24 & 0 & -2 & 0 & -2\\ 2 & 0 & -2 & 0 & -8 & 0 & 0 & 0\\ 0 & 2 & 0 & -2 & 0 & -16 & 0 & 0\\ 2 & 0 & -2 & 0 & 0 & 0 & -16 & 0\\ 0 & 2 & 0 & -2 & 0 & 0 & 0 & -24 \end{pmatrix}$$

Using the iterative method, formula (4.14) takes the form of $c=Ac+f$, yielding the spectral radius of matrix A, $\rho(A)=0.6667<1$. The iterative method converges, and the solution is as follows: $c=[c_1^0,\ c_1^1,\ c_1^2,\ c_1^3,\ c_2^0,\ c_2^1,\ c_2^2,\ c_2^3,\ c_3^0,\ c_3^1,\ c_3^2,$

c_3^3, c_4^0, c_4^1, c_4^2, $c_4^3]^{\mathrm{T}}$ = [40.0567, 12.4053, -6.1553, -0.8523, 40.0567, 12.4053, 6.1553, 0.8523, 9.9431, 4.0719, -2.1780, -0.8523, 9.9431, 4.0719, 2.1780,0.8523]$^{\mathrm{T}}$. Then, we calculate $\varphi \approx \sum_{j=1}^{4}\sum_{i=0}^{3} c_j^i \varphi_j^i + \psi$ and get the distribution of internal potential inside the electrolytic cell shown in Figure 7:



0	100	100	100	100	0
0	45.4544	59.4696	59.4696	45.4544	0
0	22.3484	32.9544	32.9544	22.3484	0
0	10.9847	17.0453	17.0453	10.9847	0
0	4.5455	7.1969	7.1969	4.5455	0
0	0	0	0	0	0

FIGURE 7. Graphical function $\varphi \approx \sum_{j=1}^{4}\sum_{i=0}^{3} c_j^i \varphi_j^i + \psi$

5. Completeness of the basis function for the OSFG-based FEM

When using the weighted residual method to solve operator equations, the basis function family and the weight function family should form a complete system. Now, let us demonstrate the completeness of the OSFG-based finite element basis function.

First, it is limited to the region Ω that can only be divided into basic graphs; a Lebesgue square-integrable function is defined on region Ω. The real variable function theory shows that a continuous function always exists for a Lebesgue square-integrable function defined on a bounded closed region Ω, and the distance between them can be arbitrarily small, As long as the diameter of the support domain of the impulse basis function is sufficiently small, the distance between the continuous function and the linear combination of the impulse basis functions on the bounded closed region Ω is arbitrarily small according to the uniform continuity of the continuous function on the bounded closed region. The OSFG-based finite element basis functions and the impulse basis functions can linearly represent each other. Therefore, the linear combination of the OSFG-based finite element basis function is dense everywhere in the Lebesgue square-integrable function space. This proves the completeness of the basis function of the FEM of the orthogonal step functions group [13, 15, 17, 7, 5, 8].

We consider the Lebesgue square-integrable function on a general bounded closed region Ω'. A bounded closed region Ω' can always be covered by a graph concatenated by basic graphs. Among these basic shapes, only the following two types are considered: the entire graph falls inside the region Ω' or the graph falls on the boundary of the region Ω'. They are called internal and boundary basic graphs, respectively. The region where the internal and boundary basic graphs are combined is denoted as Ω, and the region Ω covers the bounded closed region Ω'. We define a function on a region Ω as the function that is the original Lebesgue square-integrable function on the region Ω' and a function whose value is defined as zero on the difference set between Ω and Ω'. This defines a new Lebesgue square-integrable function on Ω, which can be approximated by a linear combinations of OSFG-based finite element basis functions on Ω. The linear combinations also approximate the original Lebesgue square-integrable function on Ω'. Such is proved the completeness of the OSFG-based finite element basis function.

6. Differential operator equations on bounded closed regions

Considering that the differential operator equation on the bounded closed region Ω' : $L\varphi = f$ (inside the region Ω') and meeting the boundary conditions (on the boundary Γ), we expand the bounded closed region Ω' into a union Ω of basic graphs using the method described in Section 5 and use the linear combination of the members of orthogonal step functions groups on each internal basic graph as the approximate value of the desired function inside the region. The selection of OSFGs should be conducive to faster convergence when using the iterative method. Then, the difference operator L_D is used to replace the differential operator L; the Galerkin method is used to find the solution of the PDE. The following describes the processing of boundary points using a square as an example of the basic graph:

The first boundary value problem: At this point, the boundary condition is $\varphi = g$ on the boundary Γ, and the value of the center point on each basic graph of the boundary is the value of the boundary point closest to the center point that falls on the basic graph.

The second and third boundary value problems: At this point, the boundary condition on the boundary Γ is as follows:

$$\frac{\partial \varphi}{\partial n} + \mathrm{a}\varphi = g. \tag{6.1}$$

The boundary curve Γ passes through a square with a center point P (as shown in Fig. 8), where S is the closest point on the boundary curve Γ to point P, with its outer normal vector being $\boldsymbol{n}$.

$$\frac{\partial\varphi}{\partial n}_{|P} = \left\{\frac{\partial\varphi}{\partial x}\cos(n,x) + \frac{\partial\varphi}{\partial y}\cos(n,y)\right\}_P \approx \frac{\varphi(Q)-\varphi(P)}{h}\cos(n,x) + \frac{\varphi(R)-\varphi(P)}{h}\cos(n,y). \tag{6.2}$$

Therefore, at point P, the following approximate equation can be obtained:

$$\frac{\varphi(Q)-\varphi(P)}{h}\cos(n,x) + \frac{\varphi(R)-\varphi(P)}{h}\cos(n,y) + a\varphi\left(P\right) = g\left(S\right). \tag{6.3}$$

7. Conclusions

The FEM of the OSFG involves taking a linear combination of orthogonal step functions as the approximation function for the partition elements, thereby approximating the field function in segments. The differential operator equations should be differenced first, and then the Galerkin method should be used to transform the

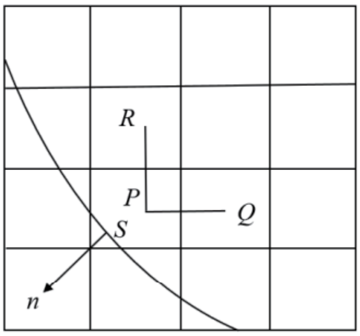


FIGURE 8. Processing of the boundary condition of square basic figure.

differential operator equations into a system of linear algebraic equations. The coefficient matrix of the system of linear algebraic equations is a sparse matrix. It is an approximate diagonal block matrix composed of many small block matrices, which can be approximately diagonalized by diagonalizing each small block or each consolidation block to reduce the spectral radius of the corresponding iteration matrix. Therefore, the number of iterations required to achieve a certain accuracy for the solution of the equation system is reduced, and consolidation block diagonalization can be reused until the spectral radius is satisfactory. This study introduced the concept and the construction method of OAEG of a given operator. If an orthogonal step function system adopted an orthogonal approximate characteristic function system and used it as the basis function in the Galerkin method, the differential operator equation was transformed into a linear algebraic equation system, in the coefficient matrix of the linear algebraic equation system, each small matrix itself was a small diagonal matrix. The OSFG adopted an extended group of orthogonal approximate characteristic function groups to reduce the number of nonzero elements in the corresponding iteration matrix. The OSFG was composed of jumping constant-valued functions and was equivalent to a set of pulse basis functions. Therefore, it simplified integration operations and helped to approximate singular points of the function being solved.

REFERENCES

[1] Å. Björck. *Iterative Methods. In: Numerical Methods in Matrix Computations(Part of the Texts in Applied Mathematics book series)* , Springer, Cham, 59 **(2015)**, 613-781.

[2] R. Burden and J. D. Faires. *Numerical Analysis:Tenth edition*, Stamford:Cengage Learning, 2018.

[3] P. G. Ciarlet and J. T. Oden. *The finite element method for elliptic problems*, Journal of Applied Mechanics, 45 **(1978)**, 968-969.

[4] R. Cook. *Concepts and applications of finite element analysis, fourth edition*, USA:John Wiley & Sons, 2001.

[5] G. B. Folland. *Real Analysis:Modern Techniques and Their Applications*, GUSA:Wiley-Blackwell, 1999.

[6] R. W. Hamming. *Introduction to applied numerical analysis*, New York: Dover Publications, 2012.

[7] C. Heil. *Introduction to real analysis*, Germany:Springer, 2019.

[8] L. Ralph. Jeffery. *The theory of functions of a real variable (Second Edition)*, Toronto:University of Toronto Press, 1951.

[9] J. M. Jin. *The finite element method in electromagnetics*, USA:Wiley-IEEE Press, 2014.

[10] N. Kachakhidze, J. Peradze and Z. Tsiklauri. *A galerkin-newton algorithm for solution of a kirchhoff-type static equation*, International Journal of Computational Methods, 19 **(2022)**, 2150057.

[11] Larson. *The finite element method: Theory, implementation, and applications*, Germany: Springer Berlin Heidelberg, 2013.

[12] Y. Liu and K. Ding. *A new coupling technique for the combination of Wavelet-Galerkin method with finite element method in solids and structures*, International Journal for Numerical Methods in Engineering, 112 **(2017)**, 1295-1322.

[13] B. Makarov and A. Podkorytov. *Real analysis: Measures, integrals and applications*, Germany:Springer, 2013.

[14] D. G. Manolakis and V. K. Ingle. *Applied digital signal processing: Theory and practice*, London:Cambridge University Press, 2018.

[15] I. P. Natanson. *Theory of functions of a real variable : (Teoria functsiy veshchestvennoy peremennoy)*, New York:Frederick Ungar Publishing CO, 1960.

[16] Nguyen. *Finite element methods: Parallel-sparse*, New York:Springer US, 2010.

[17] E. M. Stein and R. Shakarchi. *Functional analysis: Introduction to further topics in analysis*, Princeton:Princeton University Press, 2011.

[18] B. Szabó. *Introduction to finite element analysis*, WileyWiley, 2011.

[19] Thomee. *Galerkin finite element methods for parabolic problems*, Germany:Springer Berlin Heidelberg, 2010.

[20] C. Yang, Q. Li, H. Ma, J. Wang and S. Jiang. *Numerical calculation of the outer thickness of double-solid rotor asynchronous permanent magnetic coupling by finite element method*, International Journal of Materials & Product Technology, 39 **(2010)**, 339-346.

[21] D.M.Young. *Iterative solution of large linear systems*, Academy press, New york and London, 1971.

[22] O. C. Zienkiewicz. Origins, *milestones and directions of the finite element method-A personal view*, Archives of Computational Methods in Engineering 2 **(1995)**, 1–48.

Manuscript received February 12, 2024
revised September 12, 2024

J. Sun
School of computer and information, Anhui Polytechnic University, Wuhu 241000, China; Anhui Province Key Laboratory of Intelligent Car Wire-Controlled Chassis System, Anhui Polytechnic University
E-mail address: sjc@ahpu.edu.cn

S. Chen
Department of Computer Science and Technology Shandong Technology and Business University Yan Tai, China
E-mail address: flyuphighchen@126.com

X.Yang
School of Software, Jiangxi Agricultural University, Nanchang 330045, China
E-mail address: xkyang007@126.com