# Symmetrized two-scale finite element discretizations for partial differential equations with symmetric solutions

Pengyu Hou* Fang Liu† Aihui Zhou‡

June 1, 2022

**Abstract**

In this paper, a symmetrized two-scale finite element method is proposed for a class of partial differential equations with symmetric solutions. With this method, the finite element approximation on a fine tensor product grid is reduced to the finite element approximations on a much coarse grid and a univariant fine grid. It is shown by both theory and numerics including electronic structure calculations that the resulting approximation still maintains an asymptotically optimal accuracy. Consequently the symmetrized two-scale finite element method reduces computational cost significantly.

**Key words.** symmetric, two-scale, finite element, partial differential equation.
**AMS subject classifications.** 65N15, 65N25, 65N30, 65N50.

## 1 Introduction

To improve the efficiency of solving multi-dimensional elliptic source and eigenvalue problems on tensor-product domains, the two-scale finite element method was proposed in [7, 12, 13]. With this method, low frequency parts of the finite element solution to an elliptic problem are captured on a coarse grid and high frequency components are handled by some univariant fine and coarse grids.

We understand that many quantities in science and engineering have symmetries such as wavefunctions, Hamiltonians, and interatomic potentials in quantum physics. It is shown in literature that efficient numerical methods can be designed to reduce the computational cost of solving problems with symmetric properties (see, e.g., [2, 6, 8] and

*School of Mathematical Sciences, Beijing Normal University, Beijing 100875, China (hou.pengyu@mail.bnu.edu.cn).

†School of Statistics and Mathematics, Central University of Finance and Economics, Beijing 102206, China (fliu@cufe.edu.cn).

‡LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (azhou@lsec.cc.ac.cn)

references cited therein). For instance, a symmetry-based decomposition approach was proposed to solve eigenvalue problems with spatial symmetries in [6]. By this approach, the original differential eigenvalue problem is decomposed into some eigenvalue subproblems that can be carried out in parallel. In addition, each subproblem requires only a smaller number of eigenpairs compared with the original problem. As a result, the computational cost is reduced.

To reduce the computational cost of approximations of the elliptic source and eigenvalue problems with symmetric solutions, in this paper, we study some symmetrized two-scale finite element discretizations.

Let us give an informal description of the main ideas and results in this paper. Set $k$ be a positive integer. The symmetric group $\mathrm{Sym}(k)$ is the set of bijections of $\{1, 2, \cdots, k\}$ to itself. Specially, a permutation $\sigma \in \mathrm{Sym}(k)$ which interchanges two letters $i$ and $j$ and leaves all the other letters unchanged is called a transposition. Denote the transposition $\sigma$ which interchanges $i$ and $j$ by $(i, j)$ [10].

Set $\Omega = (0, 1)^3$ and $\mathbf{x} = (x_1, x_2, x_3)$. We say that $u$ is a symmetric function on $\bar{\Omega}$ [2] if

$$u(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)}) = u(x_1, x_2, x_3), \qquad \forall \mathbf{x} \in \bar{\Omega},\ \sigma \in \mathrm{Sym}(3).$$

Let $T^{h_1,h_2,h_3}(\Omega)$ be a uniform tensor product grid on $\bar{\Omega}$. $S^{h_1,h_2,h_3}(\Omega)$ is the associated tensor-product space of piecewise trilinear functions on $\bar{\Omega}$. Let $u_{h_1,h_2,h_3} \in S^{h_1,h_2,h_3}(\Omega)$ be a finite element approximation to $u$.

In computation, the grid points of $T^{h_1,h_2,h_3}(\Omega)$ are usually numbered along the $x_1$-direction, the $x_2$-direction, and the $x_3$-direction, consecutively. Let $N_i = 1/h_i (i = 1, 2, 3)$ be a positive integer. Then the values of $u_{h_1,h_2,h_3}$ on these grid points are stored in a vector denoted by $U^{h_1,h_2,h_3}$ with the size $N_{dof} := \prod_{i=1}^{3}(N_i + 1)$. Let $G : \mathbb{C}^{N_{dof}} \to S^{h_1,h_2,h_3}(\Omega)$ be a bijection satisfying

$$G(U^{h_1,h_2,h_3}) = \sum_{i=1}^{N_{dof}} U_i^{h_1,h_2,h_3} \phi_i, \qquad \forall U^{h_1,h_2,h_3} \in \mathbb{C}^{N_{dof}},$$

where $U_i^{h_1,h_2,h_3}$ is the $i$-th component of $U^{h_1,h_2,h_3}$ and $\phi_i$ is the Lagrangian basis function of $S^{h_1,h_2,h_3}(\Omega)$ with $i = 1, 2, \cdots, N_{dof}$. Obviously, we have $u_{h_1,h_2,h_3} = G(U^{h_1,h_2,h_3})$.

For any $\sigma \in \mathrm{Sym}(3)$, there holds $u_{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}} = G(U^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}})$. We define a vector transformation $\mathcal{T}_\sigma : \mathbb{C}^{N_{dof}} \to \mathbb{C}^{N_{dof}}$ satisfying

$$\mathcal{T}_\sigma(U^{h_1,h_2,h_3}) = U^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}}, \quad \forall U^{h_1,h_2,h_3} \in \mathbb{C}^{N_{dof}}.$$

Then $U^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}}$ can be obtained from $U^{h_1,h_2,h_3}$ by a symmetrization operation (Algorithm 3.1). Let $R_\sigma : S^{h_1,h_2,h_3}(\Omega) \to S^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}}(\Omega)$ be a bijection defined by

$$R_\sigma = G \circ \mathcal{T}_\sigma \circ G^{-1}. \tag{1.1}$$

We get

$$u_{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}} = R_\sigma(u_{h_1,h_2,h_3}), \quad \forall u_{h_1,h_2,h_3} \in S^{h_1,h_2,h_3}(\Omega). \tag{1.2}$$

Note that the two-scale finite element solution [7, 12] of a linear elliptic source problem with the Dirichlet boundary condition is a linear combination of the standard finite element solutions on four different scale grids. That is,

$$B^h_{H,H,H} u = u_{h,H,H} + u_{H,h,H} + u_{H,H,h} - 2u_{H,H,H}.$$

Hence, for a linear source problem with symmetric solution, we obtain from (1.2) that $u_{H,h,H} = R_{(1,2)}(u_{h,H,H})$ and $u_{H,H,h} = R_{(1,3)}(u_{h,H,H})$. Consequently,

$$B^h_{H,H,H}u = u_{h,H,H} + R_{(1,2)}(u_{h,H,H}) + R_{(1,3)}(u_{h,H,H}) - 2u_{H,H,H}.$$

It is seen from (1.1) that the amount of operations for getting $R_{(1,2)}(u_{h,H,H})$ or $R_{(1,3)}(u_{h,H,H})$ from $u_{h,H,H}$ is essentially that for calculating $\mathcal{T}_{(1,2)}(U^{h,H,H})$ or $\mathcal{T}_{(1,3)}(U^{h,H,H})$ from $U^{h,H,H}$ by the symmetrization algorithm (Algorithm 3.1), respectively. It is only $18N_{dof}$ with $N_{dof} := (n-1)(N-1)(N-1)$ if $n = 1/h$ and $N = 1/H$. While the amount of operations for computing $u_{H,h,H}$ or $u_{H,H,h}$ by the standard finite element method is far from $18N_{dof}$, due to creating stiffness matrix and solving linear algebraic systems. Hence the symmetrization algorithm is quite efficient. Similarly, the symmetrization algorithm can be also combined with the postprocessed two-scale finite element method [11] and the two-scale postprocessed finite element method [15] respectively to construct new and more efficient algorithms.

This paper is structured as follows. In section 2, some preliminaries are presented and the existing related results that will be used in this paper are called. In section 3, a vector transformation operator associated with symmetric functions is constructed. A symmetrization algorithm is then proposed to perform this vector transformation operator. In section 4 and section 5, combining the symmetrization algorithm with the two-scale finite element method, some symmetrized two-scale finite element discretizations are developed for the linear source and eigenvalue problems with symmetric solutions, respectively. In section 6, several numerical results, including the applications to electronic structure calculations, are presented to illustrate the efficiency of our approaches. Finally, some concluding remarks are given in section 7.

## 2 Preliminaries

Let $\Omega = (0,1)^d$ with $d \geq 2$. The standard notation for Sobolev spaces $W^{s,p}(\Omega)$ and their associated norms and seminorms will be used [4]. Set $H^s(\Omega) = W^{s,2}(\Omega)$ and $\|\cdot\|_{s,\Omega} = \|\cdot\|_{s,2,\Omega}$. Let $H^1_0(\Omega) = \{v \in H^1(\Omega) : v\mid_{\partial\Omega} = 0\}$, $H^{-1}(\Omega)$ be the dual of $H^1_0(\Omega)$, and $(\cdot,\cdot)$ be the standard $L^2(\Omega)$ inner product. We shall use $A \lesssim B$ to mean that $A \leq CB$ for some constant $C$ which does not depend on any grid parameters.

Let $\mathbb{N}_0$ be the set of all nonnegative integers. For $w \in W^{s,p}(\Omega)$, $\mathbf{x} = (x_1, x_2, \cdots, x_d) \in \Omega$, and $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_d) \in \mathbb{N}^d_0$, set

$$(D^\alpha w)(\mathbf{x}) = (\frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_d} w}{\partial x_d^{\alpha_d}})(\mathbf{x}) \qquad \text{and} \qquad |\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_d.$$

Furthermore, set $\mathbf{0} = (0, \cdots, 0) \in \mathbb{R}^d$, $\mathbf{e} = (1, \cdots, 1) \in \mathbb{R}^d$, and $\mathbb{Z}_d = 1, 2, \cdots, d$. For each $i \in \mathbb{Z}_d$, let $\mathbf{e}_i = (0, \cdots, 0, 1, 0, \cdots, 0) \in \mathbb{R}^d$, which is the vector whose $i$-th component is one and other components are zero. Write $\hat{\mathbf{e}}_i = \mathbf{e} - \mathbf{e}_i$.

We shall use the following Sobolev space

$$W^{G,4}(\Omega) = \{w \in H^3(\Omega) : D^\alpha w \in L^2(\Omega), \mathbf{0} \leq \alpha \leq 3\mathbf{e}, |\alpha| = 4\}$$

with its natural norm $\|\cdot\|_{W^{G,4}(\Omega)}$ [16].

## 2.1 A source problem

Consider the following elliptic source problem:

$$\begin{cases} Lu = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{2.1}$$

Here $f \in H^{-1}(\Omega)$ and $L$ is defined by

$$Lu = -\sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}\big(a_{ij}\frac{\partial u}{\partial x_j}\big) + \sum_{i=1}^{d} b_i \frac{\partial u}{\partial x_i} + Vu,$$

where $a_{ij} \in W^{1,\infty}(\Omega)$, $b_i$, $V \in L^{\infty}(\Omega)$, and $\sum_{i,j=1}^{d} a_{ij}x_i x_j \geq c\sum_{k=1}^{d} x_k^2$ on $\bar{\Omega}$ for some constant $c > 0$.

A weak form of (2.1) is: find $u \in H_0^1(\Omega)$ such that

$$a(u,v) = (f,v) \quad \forall v \in H_0^1(\Omega), \tag{2.2}$$

where

$$a(u,v) = \int_{\Omega} \Big( \sum_{i,j=1}^{d} a_{ij} \frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_j} + \sum_{i=1}^{d} b_i \frac{\partial u}{\partial x_i} v + Vuv \Big) \quad \text{and} \quad (f,v) = \int_{\Omega} fv. \tag{2.3}$$

For each $f \in H^{-1}(\Omega)$, assume that there is a unique solution $u \in H_0^1(\Omega)$ of (2.2).

Set $N \in \mathbb{N}_0/\{0\}$ and $h = 1/N \in [0,1]$. Let $T^h[0,1]$ be the uniform grid with grid size $h$. Let $h_j = 1/N_j$ for $N_j \in \mathbb{N}_0/\{0\}$, $j \in \mathbb{Z}_d$. Set $\mathbf{h} = (h_1, \ldots, h_d)$ and $h = \max\limits_{j \in \mathbb{Z}_d} h_j$. Define a tensor-product grid on $\bar{\Omega} = [0,1]^d$ by

$$T^{\mathbf{h}}(\Omega) := T^{h_1}[0,1] \times \cdots \times T^{h_d}[0,1].$$

Let $S^h[0,1] \subset C[0,1]$ be the piecewise linear finite element space on $T^h[0,1]$. The tensor product spaces of piecewise $d$-linear functions on $\bar{\Omega}$ are

$$S^{\mathbf{h}}(\Omega) := S^{h_1}[0,1] \otimes \cdots \otimes S^{h_d}[0,1] \qquad \text{and} \qquad S_0^{\mathbf{h}}(\Omega) := S^{\mathbf{h}}(\Omega) \cap H_0^1(\Omega).$$

The standard Galerkin finite element method for solving (2.2) is: find $u_{\mathbf{h}} \in S_0^{\mathbf{h}}(\Omega)$ such that

$$a(u_{\mathbf{h}}, v) = (f,v) \quad \forall\ v \in S_0^{\mathbf{h}}(\Omega). \tag{2.4}$$

Define the Galerkin projector $P_{\mathbf{h}} : H_0^1(\Omega) \mapsto S_0^{\mathbf{h}}(\Omega)$ by

$$a(w - P_{\mathbf{h}}w, v) = 0 \quad \forall v \in S_0^{\mathbf{h}}(\Omega), \quad \forall w \in H_0^1(\Omega). \tag{2.5}$$

Obviously $P_{\mathbf{h}}w = w_{\mathbf{h}}$, $\forall w \in H_0^1(\Omega)$.

## 2.2 An eigenvalue problem

Assume that $a(\cdot,\cdot)$ is the same as (2.3) with $b_i = 0,\ i = 1,2,\cdots,d$ and $(a_{ij})$ is symmetric. Consider the following eigenvalue problem

$$a(u,v) = \lambda(u,v) \quad \forall v \in H_0^1(\Omega), \tag{2.6}$$

where $\lambda$ is the eigenvalue of the symmetric bilinear form $a(\cdot,\cdot)$ relative to the $L^2(\Omega)$ inner product $(\cdot,\cdot)$ and $u$ is the corresponding eigenvector. The eigenvalues of (2.6) are real numbers satisfying $\lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots$ [1]. Assume that the corresponding eigenvectors $u_1, u_2, u_3, \ldots$ satisfy $(u_i, u_j) = \delta_{ij}$ for $i, j = 1, 2, \cdots$ with $\delta_{ij}$ the Dirac delta.

Consider the standard Galerkin finite element method for solving (2.6): find a pair $(\lambda_{\mathbf{h}}, u_{\mathbf{h}}) \in \mathbb{R} \times S_0^{\mathbf{h}}(\Omega)$ satisfying

$$a(u_{\mathbf{h}}, v) = \lambda_{\mathbf{h}}(u_{\mathbf{h}}, v) \quad \forall v \in S_0^{\mathbf{h}}(\Omega), \tag{2.7}$$

which has finite eigenvalues $\lambda_{1,\mathbf{h}} < \lambda_{2,\mathbf{h}} \leq \cdots \leq \lambda_{n_{\mathbf{h}},\mathbf{h}}$, where $n_{\mathbf{h}} = \dim\ S_0^{\mathbf{h}}(\Omega)$. Assume that the corresponding eigenvectors $u_{1,\mathbf{h}}, u_{2,\mathbf{h}}, \ldots, u_{n_{\mathbf{h}},\mathbf{h}}$ satisfy $(u_{i,\mathbf{h}},\ u_{j,\mathbf{h}}) = \delta_{ij}$. The Rayleigh principle [1] implies that $\lambda_i \leq \lambda_{i,\mathbf{h}}$ for $i = 1, 2, \ldots, n_{\mathbf{h}}$.

Suppose that $h = \max\limits_{j\in\mathbb{Z}_d} h_j \ll 1, \lambda$ is simple, and $(\lambda_{\mathbf{h}}, u_{\mathbf{h}})$ is an approximation to $(\lambda, u)$ which satisfies (2.7) and Lemma 3.2 in [11] with $\|u_{\mathbf{h}}\|_{0,\Omega} = 1$. There holds [1]

$$\lambda_{\mathbf{h}} - \lambda + \|u - u_{\mathbf{h}}\|_{0,\Omega} + h\|u - u_{\mathbf{h}}\|_{1,\Omega} \lesssim h^2. \tag{2.8}$$

**Lemma 2.1.** *[7] There holds*

$$\lambda_{\mathbf{h}} - \lambda = \lambda(u, u - P_{\mathbf{h}}u) + \mathcal{O}(h^4). \tag{2.9}$$

# 3 A vector transformation operator and its implementation

In this section, we will introduce a vector transformation operator associated with symmetric functions. A so-called symmetrization algorithm will be proposed to implement this vector transformation. We will combine the symmetrization algorithm with the two-scale finite element method to design new and efficient algorithms in Section 4.

We call $u$ a symmetric function on $\bar{\Omega}$ if

$$u(x_{\sigma(1)}, x_{\sigma(2)}, \cdots, x_{\sigma(d)}) = u(x_1, x_2, \cdots, x_d), \qquad \forall \mathbf{x} \in \bar{\Omega},\ \sigma \in \mathrm{Sym}(d). \tag{3.1}$$

Thus we have

$$\begin{aligned} u(i_{\sigma(1)}h_{\sigma(1)}, i_{\sigma(2)}h_{\sigma(2)}, \cdots, i_{\sigma(d)}h_{\sigma(d)}) &= u(i_1h_1, i_2h_2, \cdots, i_dh_d), \\ i_k &= 0, 1, \cdots, N_k; k = 1, 2, \cdots, d. \end{aligned} \tag{3.2}$$

Next we illustrate how to get the values of $u(\mathbf{x})$ on the grid points of $T^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}(\Omega)$ from those on the grid points of $T^{h_1,h_2,\cdots,h_d}(\Omega)$ in computation. The values of $u(\mathbf{x})$ on the grid points of $T^{h_1,h_2,\cdots,h_d}(\Omega)$ are stored in a vector $U^{h_1,h_2,\cdots,h_d}$ with the size $N_{dof} :=$

$\prod_{i=1}^{d}(N_i+1)$. Usually the grid points of $T^{h_1,h_2,\cdots,h_d}(\Omega)$ are numbered along the $x_1$-direction, the $x_2$-direction, $\cdots$, and the $x_d$-direction, consecutively. That is, the $\hat{I}$-th component of $U^{h_1,h_2,\cdots,h_d}$ satisfies

$$\begin{aligned} U_{\hat{I}}^{h_1,h_2,\cdots,h_d} &= u(i_1h_1, i_2h_2, \cdots, i_dh_d),\\ &\qquad i_k = 0,1,\cdots,N_k; k=1,2,\cdots,d, \end{aligned} \tag{3.3}$$

where the subscript

$$\hat{I} := I(i_1,i_2,\cdots,i_d) = \sum_{l=2}^{d}\left(i_l\prod_{j=1}^{l-1}(N_j+1)\right)+i_1+1. \tag{3.4}$$

The values of $u(\mathbf{x})$ on the grid points of $T^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}(\Omega)$ are contained in the vector $U^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}$. We obtain from (3.2)-(3.4) that

$$\begin{aligned} U_{I(i_{\sigma(1)},i_{\sigma(2)},\cdots,i_{\sigma(d)})}^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}} &= U_{I(i_1,i_2,\cdots,i_d)}^{h_1,h_2,\cdots,h_d},\\ &\qquad i_k=0,1,\cdots,N_k; k=1,2,\cdots,d, \end{aligned} \tag{3.5}$$

with

$$I(i_{\sigma(1)},i_{\sigma(2)},\cdots,i_{\sigma(d)}) = \sum_{l=2}^{d}\left(i_{\sigma(l)}\prod_{j=1}^{l-1}(N_{\sigma(j)}+1)\right)+i_{\sigma(1)}+1.$$

Hence for any $\sigma\in \mathrm{Sym}(d)$, one can get $U^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}$ from $U^{h_1,h_2,\cdots,h_d}$. We can define a vector transformation operator $\mathcal{T}_\sigma : \mathbb{C}^{N_{dof}} \to \mathbb{C}^{N_{dof}}$ satisfying

$$\mathcal{T}_\sigma(U^{h_1,h_2,\cdots,h_d}) = U^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}, \qquad \forall U^{h_1,h_2,\cdots,h_d}\in\mathbb{C}^{N_{dof}}. \tag{3.6}$$

This vector transformation operator can be implemented by Algorithm 3.1, which is called the symmetrization algorithm.

**Algorithm 3.1. Input:** *$h_1$, $h_2$, $\cdots$, $h_d$, $U^{h_1,h_2,\cdots,h_d}$, $\sigma$.*
**Output:** *$U^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}$.*

```
$N_1 = 1/h_1$; $N_2 = 1/h_2$; $\cdots$; $N_d = 1/h_d$.
for $i_1 = 0 : N_1$ do
    for $i_2 = 0 : N_2$ do
        $\cdots$
        for $i_d = 0 : N_d$ do
            $U_{I(i_{\sigma(1)},i_{\sigma(2)},\cdots,i_{\sigma(d)})}^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}} = U_{I(i_1,i_2,\cdots,i_d)}^{h_1,h_2,\cdots,h_d}$.
return $U^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}$.
```

Calculating $I(i_1,i_2,\cdots,i_d)$ in (3.4) requires $d^2$ operations. Hence the amount of operations for Algorithm 3.1 is $2d^2N_{dof}$. Next we illustrate the idea of Algorithm 3.1 further for $d=2,3$, respectively.

## 3.1 Two dimensional case

Let $\Omega = (0,1)^2$. We see that there is only one element $\sigma = (1,2)$ in Sym(2) besides the identical operator. Assume that $u$ is a symmetric function on $\bar{\Omega}$. Namely,

$$u(x_2, x_1) = u(x_{\sigma(1)}, x_{\sigma(2)}) = u(x_1, x_2), \quad \forall \mathbf{x} \in \bar{\Omega}. \tag{3.7}$$

Consequently for the grid points of $T^{h_2,h_1}(\Omega) = T^{h_{\sigma(1)},h_{\sigma(2)}}(\Omega)$ and $T^{h_1,h_2}(\Omega)$, we have

$$\begin{aligned} u(i_2h_2, i_1h_1) = u(i_{\sigma(1)}h_{\sigma(1)}, i_{\sigma(2)}h_{\sigma(2)}) = u(i_1h_1, i_2h_2),& \\ i_k = 0, 1, \cdots, N_k; k = 1, 2.& \end{aligned} \tag{3.8}$$

In implementation, the values of $u(x_1, x_2)$ on the grid points of a uniform grid $T^{h_1,h_2}(\Omega)$ are stored in a vector $U^{h_1,h_2}$ with the size $N_{dof} := (N_1+1) \times (N_2+1)$. Usually the grid points of a uniform grid $T^{h_1,h_2}(\Omega)$ are numbered along the $x_1$-direction and the $x_2$-direction, consecutively. That is, the $\hat{I}$-th component of $U^{h_1,h_2}$ satisfies

$$U^{h_1,h_2}_{\hat{I}} = u(i_1h_1, i_2h_2), \qquad i_k = 0, 1, \cdots, N_k; k = 1, 2, \tag{3.9}$$

where the subscript

$$\hat{I} := I(i_1, i_2) = i_2(N_1 + 1) + i_1 + 1. \tag{3.10}$$

The values of $u(x_2, x_1)$ on the grid points of $T^{h_2,h_1}(\Omega)$ are stored in the vector $U^{h_2,h_1}$. By (3.8) – (3.10), we have

$$U^{h_2,h_1}_{I(i_2,i_1)} = U^{h_1,h_2}_{I(i_1,i_2)}, \qquad i_k = 0, 1, \cdots, N_k; k = 1, 2, \tag{3.11}$$

with

$$I(i_2, i_1) = i_1(N_2 + 1) + i_2 + 1.$$

That is, one can get $U^{h_2,h_1}$ from $U^{h_1,h_2}$. Let $d = 2$ in (3.6). It follows that $\mathcal{T}_{(1,2)}(U^{h_1,h_2}) = U^{h_2,h_1}$.

For example, if $h_1 = \frac{1}{2}$ and $h_2 = 1$, it is shown in Figure 1 that the values of $u(x_1, x_2)$ on the grid points of the uniform grids $T^{\frac{1}{2},1}(\Omega)$ and $T^{1,\frac{1}{2}}(\Omega)$ are stored in the vectors $U^{\frac{1}{2},1}$ and $U^{1,\frac{1}{2}}$, respectively. By Algorithm 3.1, $U^{1,\frac{1}{2}} = \mathcal{T}_{(1,2)}(U^{\frac{1}{2},1})$ can be obtained from $U^{\frac{1}{2},1}$. Similarly, if $h_1 = \frac{1}{4}$ and $h_2 = \frac{1}{2}$, The process of obtaining the vector $U^{\frac{1}{2},\frac{1}{4}}$ from $U^{\frac{1}{4},\frac{1}{2}}$ is illustrated in Figure 2, in which $\hat{U} := U^{\frac{1}{4},\frac{1}{2}}$ and $\tilde{U} := U^{\frac{1}{2},\frac{1}{4}}$ for simplicity.

The process of obtaining the vectors $U^{h_{\sigma(1)},h_{\sigma(2)}} = \mathcal{T}_{\sigma}(U^{h_1,h_2})$ from $U^{h_1,h_2}$ can be implemented by Algorithm 3.1 with $d = 2$, which demands $8N_{dof}$ operations.

## 3.2 Three dimensional case

Let $\Omega = (0,1)^3$. Assume that $u$ is a symmetric function on $\bar{\Omega}$. That is,

$$u(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)}) = u(x_1, x_2, x_3), \quad \forall \mathbf{x} \in \bar{\Omega}, \ \sigma \in \text{Sym}(3).$$

Thus for the grid points of $T^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}}(\Omega)$ and $T^{h_1,h_2,h_3}(\Omega)$, we have

$$\begin{aligned} u(i_{\sigma(1)}h_{\sigma(1)}, i_{\sigma(2)}h_{\sigma(2)}, i_{\sigma(3)}h_{\sigma(3)}) = u(i_1h_1, i_2h_2, i_3h_3),& \\ i_k = 0, 1, \cdots, N_k; k = 1, 2, 3.& \end{aligned} \tag{3.12}$$

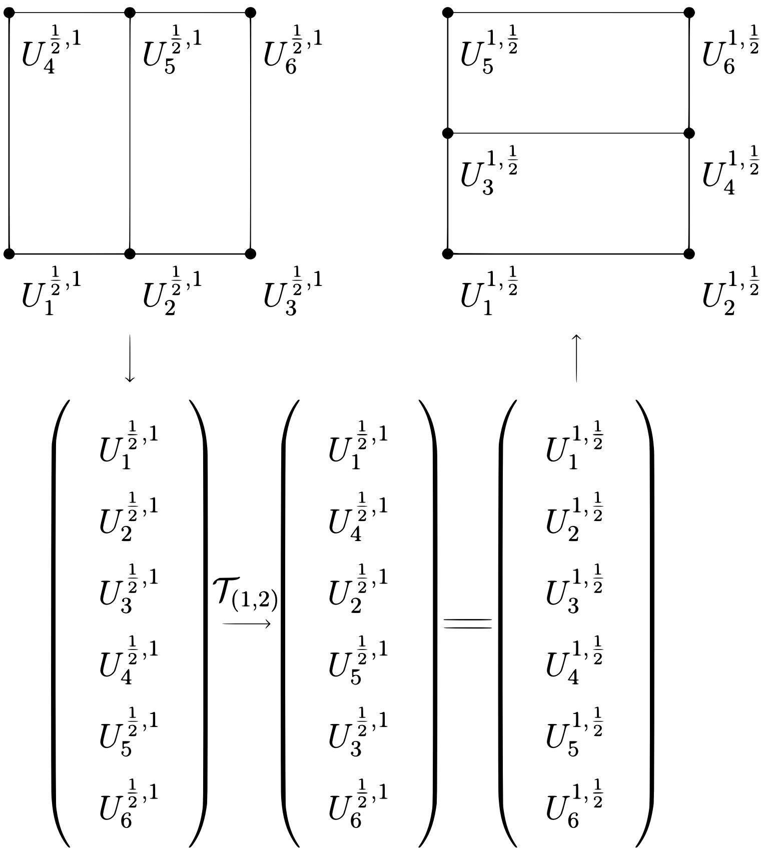
$U_4^{\frac{1}{2},1}$ $U_5^{\frac{1}{2},1}$ $U_6^{\frac{1}{2},1}$
$U_1^{\frac{1}{2},1}$ $U_2^{\frac{1}{2},1}$ $U_3^{\frac{1}{2},1}$
$U_5^{1,\frac{1}{2}}$ $U_6^{1,\frac{1}{2}}$
$U_3^{1,\frac{1}{2}}$ $U_4^{1,\frac{1}{2}}$
$U_1^{1,\frac{1}{2}}$ $U_2^{1,\frac{1}{2}}$
$\begin{pmatrix} U_1^{\frac{1}{2},1} \\ U_2^{\frac{1}{2},1} \\ U_3^{\frac{1}{2},1} \\ U_4^{\frac{1}{2},1} \\ U_5^{\frac{1}{2},1} \\ U_6^{\frac{1}{2},1} \end{pmatrix} \xrightarrow{\mathcal{T}_{(1,2)}} \begin{pmatrix} U_1^{\frac{1}{2},1} \\ U_4^{\frac{1}{2},1} \\ U_2^{\frac{1}{2},1} \\ U_5^{\frac{1}{2},1} \\ U_3^{\frac{1}{2},1} \\ U_6^{\frac{1}{2},1} \end{pmatrix} = \begin{pmatrix} U_1^{1,\frac{1}{2}} \\ U_2^{1,\frac{1}{2}} \\ U_3^{1,\frac{1}{2}} \\ U_4^{1,\frac{1}{2}} \\ U_5^{1,\frac{1}{2}} \\ U_6^{1,\frac{1}{2}} \end{pmatrix}$

Figure 1: The process of obtaining the vector $U^{1,\frac{1}{2}}$ from $U^{\frac{1}{2},1}$. For example, $U_2^{1,\frac{1}{2}} = U_{I(1,0)}^{1,\frac{1}{2}} = u(1,0) = u(0,1) = U_{I(0,1)}^{\frac{1}{2},1} = U_4^{\frac{1}{2},1}$ by (3.9), (3.10), and (3.11).

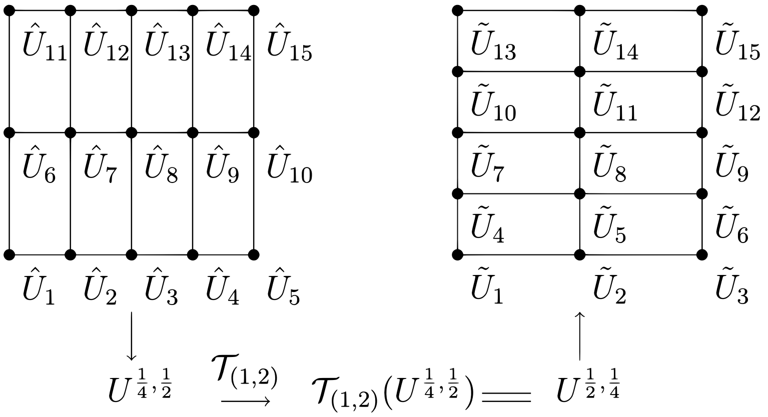
$\hat{U}_{11}$ $\hat{U}_{12}$ $\hat{U}_{13}$ $\hat{U}_{14}$ $\hat{U}_{15}$
$\hat{U}_6$ $\hat{U}_7$ $\hat{U}_8$ $\hat{U}_9$ $\hat{U}_{10}$
$\hat{U}_1$ $\hat{U}_2$ $\hat{U}_3$ $\hat{U}_4$ $\hat{U}_5$
$\tilde{U}_{13}$ $\tilde{U}_{14}$ $\tilde{U}_{15}$
$\tilde{U}_{10}$ $\tilde{U}_{11}$ $\tilde{U}_{12}$
$\tilde{U}_7$ $\tilde{U}_8$ $\tilde{U}_9$
$\tilde{U}_4$ $\tilde{U}_5$ $\tilde{U}_6$
$\tilde{U}_1$ $\tilde{U}_2$ $\tilde{U}_3$
$U^{\frac{1}{4},\frac{1}{2}} \xrightarrow{\mathcal{T}_{(1,2)}} \mathcal{T}_{(1,2)}(U^{\frac{1}{4},\frac{1}{2}}) = U^{\frac{1}{2},\frac{1}{4}}$

Figure 2: The process of obtaining the vector $\tilde{U} := U^{\frac{1}{2},\frac{1}{4}}$ from $\hat{U} := U^{\frac{1}{4},\frac{1}{2}}$. For example, $U_{11}^{\frac{1}{2},\frac{1}{4}} = U_{I(1,3)}^{\frac{1}{2},\frac{1}{4}} = u(\frac{1}{2},\frac{3}{4}) = u(\frac{3}{4},\frac{1}{2}) = U_{I(3,1)}^{\frac{1}{4},\frac{1}{2}} = U_9^{\frac{1}{4},\frac{1}{2}}$ by (3.9), (3.10), and (3.11).

In implementation, similarly, the values of $u(x_1, x_2, x_3)$ on the grid points of $T^{h_1,h_2,h_3}(\Omega)$ are stored in a vector denoted by $U^{h_1,h_2,h_3}$ with the size $N_{dof} := \prod_{i=1}^{3}(N_i + 1)$. The grid points of $T^{h_1,h_2,h_3}(\Omega)$ are usually numbered along the $x_1$-direction, the $x_2$-direction, and the $x_3$-direction, consecutively. Thus the $\hat{I}$-th component of $U^{h_1,h_2,h_3}$ satisfies

$$U_{\hat{I}}^{h_1,h_2,h_3} = u(i_1h_1, i_2h_2, i_3h_3),$$
$$i_k = 0, 1, \cdots, N_k; k = 1, 2, 3, \tag{3.13}$$

where the subscript

$$\hat{I} := I(i_1, i_2, i_3) = i_3(N_1 + 1)(N_2 + 1) + i_2(N_1 + 1) + i_1 + 1. \tag{3.14}$$

The values of $u(\mathbf{x})$ on the grid points of $T^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}}(\Omega)$ are contained in the vector $U^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}}$. By (3.12), (3.13), and (3.14), we obtain

$$U_{I(i_{\sigma(1)},i_{\sigma(2)},i_{\sigma(3)})}^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}} = U_{I(i_1,i_2,i_3)}^{h_1,h_2,h_3},$$
$$i_k = 0, 1, \cdots, N_k; k = 1, 2, 3, \tag{3.15}$$

with

$$I(i_{\sigma(1)}, i_{\sigma(2)}, i_{\sigma(3)}) = i_{\sigma(3)}(N_{\sigma(1)} + 1)(N_{\sigma(2)} + 1) + i_{\sigma(2)}(N_{\sigma(1)} + 1) + i_{\sigma(1)} + 1.$$

That is, the vector $U^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}}$ can be obtained from $U^{h_1,h_2,h_3}$. Let $d = 3$ in (3.6). It follows that $\mathcal{T}_\sigma(U^{h_1,h_2,h_3}) = U^{h_{\sigma(1)},h_{\sigma(2)},h_{\sigma(3)}}$.

For example, if $h_1 = \frac{1}{4}$, $h_2 = \frac{1}{2}$, and $h_3 = \frac{1}{2}$, the components of the vector $U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}$ are illustrated in Figures 3 and 4, where $\hat{U} := U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}$ for simplicity.

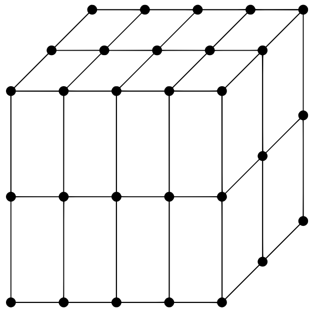

Figure 3: Grid points for $U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}$.

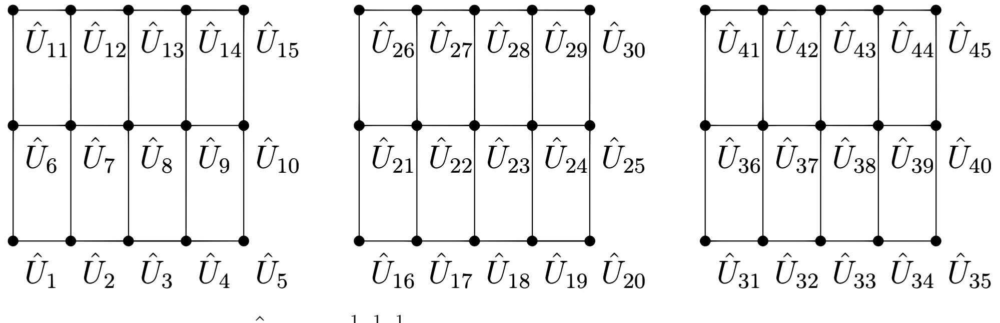
$\hat{U}_{11}$ $\hat{U}_{12}$ $\hat{U}_{13}$ $\hat{U}_{14}$ $\hat{U}_{15}$
$\hat{U}_{6}$ $\hat{U}_{7}$ $\hat{U}_{8}$ $\hat{U}_{9}$ $\hat{U}_{10}$
$\hat{U}_{1}$ $\hat{U}_{2}$ $\hat{U}_{3}$ $\hat{U}_{4}$ $\hat{U}_{5}$
$\hat{U}_{26}$ $\hat{U}_{27}$ $\hat{U}_{28}$ $\hat{U}_{29}$ $\hat{U}_{30}$
$\hat{U}_{21}$ $\hat{U}_{22}$ $\hat{U}_{23}$ $\hat{U}_{24}$ $\hat{U}_{25}$
$\hat{U}_{16}$ $\hat{U}_{17}$ $\hat{U}_{18}$ $\hat{U}_{19}$ $\hat{U}_{20}$
$\hat{U}_{41}$ $\hat{U}_{42}$ $\hat{U}_{43}$ $\hat{U}_{44}$ $\hat{U}_{45}$
$\hat{U}_{36}$ $\hat{U}_{37}$ $\hat{U}_{38}$ $\hat{U}_{39}$ $\hat{U}_{40}$
$\hat{U}_{31}$ $\hat{U}_{32}$ $\hat{U}_{33}$ $\hat{U}_{34}$ $\hat{U}_{35}$

Figure 4: Components of $\hat{U} := U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}$ on each layer along the $x_3$-direction. For example, $\hat{U}_{24} := U_{24}^{\frac{1}{4},\frac{1}{2},\frac{1}{2}} = U_{I(3,1,1)}^{\frac{1}{4},\frac{1}{2},\frac{1}{2}} = u(\frac{3}{4}, \frac{1}{2}, \frac{1}{2})$ by (3.13) and (3.14).

The vectors $U^{\frac{1}{2},\frac{1}{4},\frac{1}{2}} = \mathcal{T}_{(1,2)}(U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}})$ and $U^{\frac{1}{2},\frac{1}{2},\frac{1}{4}} = \mathcal{T}_{(1,3)}(U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}})$ can be obtained from $U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}$ with $\sigma = (1,2)$ and $(1,3)$ in (3.15), respectively. It is illustrated in Figures 5 and 6.

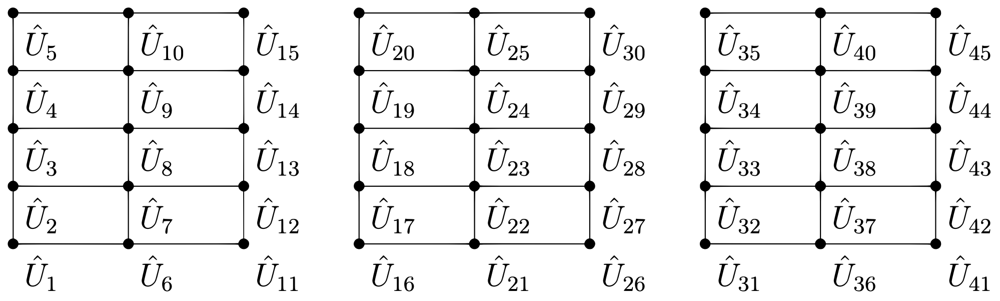
$\hat{U}_5$ $\hat{U}_{10}$ $\hat{U}_{15}$ $\hat{U}_{20}$ $\hat{U}_{25}$ $\hat{U}_{30}$ $\hat{U}_{35}$ $\hat{U}_{40}$ $\hat{U}_{45}$
$\hat{U}_4$ $\hat{U}_9$ $\hat{U}_{14}$ $\hat{U}_{19}$ $\hat{U}_{24}$ $\hat{U}_{29}$ $\hat{U}_{34}$ $\hat{U}_{39}$ $\hat{U}_{44}$
$\hat{U}_3$ $\hat{U}_8$ $\hat{U}_{13}$ $\hat{U}_{18}$ $\hat{U}_{23}$ $\hat{U}_{28}$ $\hat{U}_{33}$ $\hat{U}_{38}$ $\hat{U}_{43}$
$\hat{U}_2$ $\hat{U}_7$ $\hat{U}_{12}$ $\hat{U}_{17}$ $\hat{U}_{22}$ $\hat{U}_{27}$ $\hat{U}_{32}$ $\hat{U}_{37}$ $\hat{U}_{42}$
$\hat{U}_1$ $\hat{U}_6$ $\hat{U}_{11}$ $\hat{U}_{16}$ $\hat{U}_{21}$ $\hat{U}_{26}$ $\hat{U}_{31}$ $\hat{U}_{36}$ $\hat{U}_{41}$

Figure 5: On each layer along the $x_3$-direction, we obtain $U^{\frac{1}{2},\frac{1}{4},\frac{1}{2}}$ from $\hat{U} := U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}$ by (3.15). For example, $U^{\frac{1}{2},\frac{1}{4},\frac{1}{2}}_{I(0,3,1)} = u(0,\frac{3}{4},\frac{1}{2}) = u(\frac{3}{4},0,\frac{1}{2}) = U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}_{I(3,0,1)}$. That is, $U^{\frac{1}{2},\frac{1}{4},\frac{1}{2}}_{25} = U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}_{19} = \hat{U}_{19}$.

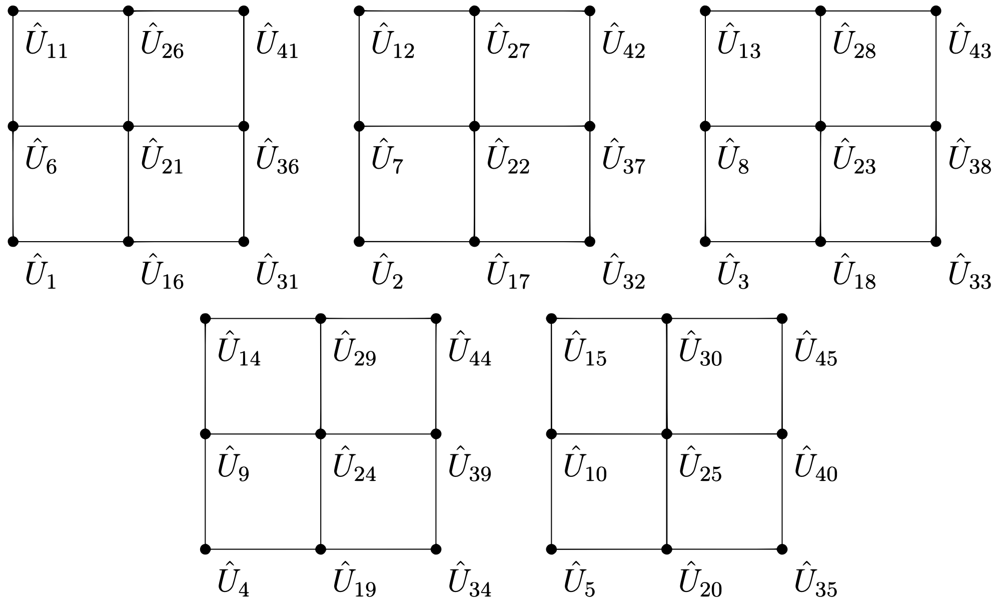
$\hat{U}_{11}$ $\hat{U}_{26}$ $\hat{U}_{41}$ $\hat{U}_{12}$ $\hat{U}_{27}$ $\hat{U}_{42}$ $\hat{U}_{13}$ $\hat{U}_{28}$ $\hat{U}_{43}$
$\hat{U}_6$ $\hat{U}_{21}$ $\hat{U}_{36}$ $\hat{U}_7$ $\hat{U}_{22}$ $\hat{U}_{37}$ $\hat{U}_8$ $\hat{U}_{23}$ $\hat{U}_{38}$
$\hat{U}_1$ $\hat{U}_{16}$ $\hat{U}_{31}$ $\hat{U}_2$ $\hat{U}_{17}$ $\hat{U}_{32}$ $\hat{U}_3$ $\hat{U}_{18}$ $\hat{U}_{33}$
$\hat{U}_{14}$ $\hat{U}_{29}$ $\hat{U}_{44}$ $\hat{U}_{15}$ $\hat{U}_{30}$ $\hat{U}_{45}$
$\hat{U}_9$ $\hat{U}_{24}$ $\hat{U}_{39}$ $\hat{U}_{10}$ $\hat{U}_{25}$ $\hat{U}_{40}$
$\hat{U}_4$ $\hat{U}_{19}$ $\hat{U}_{34}$ $\hat{U}_5$ $\hat{U}_{20}$ $\hat{U}_{35}$

Figure 6: On each layer along the $x_3$-direction, we get $U^{\frac{1}{2},\frac{1}{2},\frac{1}{4}}$ from $\hat{U} := U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}$ by (3.15). For example, $U^{\frac{1}{2},\frac{1}{2},\frac{1}{4}}_{I(0,2,1)} = u(0,1,\frac{1}{4}) = u(\frac{1}{4},1,0) = U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}_{I(1,2,0)}$. That is, $U^{\frac{1}{2},\frac{1}{2},\frac{1}{4}}_{16} = U^{\frac{1}{4},\frac{1}{2},\frac{1}{2}}_{12} = \hat{U}_{12}$.

The process of obtaining the vectors $U^{h_2,h_1,h_3} = U^{h_{(1,2)(1)},h_{(1,2)(2)},h_{(1,2)(3)}}$ or $U^{h_3,h_2,h_1} = U^{h_{(1,3)(1)},h_{(1,3)(2)},h_{(1,3)(3)}}$ from $U^{h_1,h_2,h_3}$ can be implemented by Algorithm 3.1 with $d = 3$, which requires $18N_{dof}$ operations.

# 4 The symmetrized two-scale finite element approximations

For the source problem (2.2) and eigenvalue problem (2.6), the two-scale finite element method has been developed [7, 12]. In this section, for (2.2) and (2.6) with symmetric solutions, Algorithm 3.1 will be combined with the two-scale finite element method.

Let $v$ be a symmetric function and $v_{h_1,h_2,\cdots,h_d} \in S^{h_1,h_2,\cdots,h_d}(\Omega)$ be a standard finite element approximation to $v$. As stated in Section 3, usually the grid points of $T^{h_1,h_2,\cdots,h_d}(\Omega)$ are numbered along the $x_1$-direction, the $x_2$-direction, $\cdots$, and the $x_d$-direction, consecutively. The values of $v_{h_1,h_2,\cdots,h_d}$ on these nodes are stored in a vector denoted by $V^{h_1,h_2,\cdots,h_d}$ with the size $N_{dof} := \prod_{i=1}^{d}(N_i+1)$.

Define a bijection $G : \mathbb{C}^{N_{dof}} \to S^{h_1,h_2,\cdots,h_d}(\Omega)$ satisfying

$$G(V^{h_1,h_2,\cdots,h_d}) = \sum_{i=1}^{N_{dof}} V_i^{h_1,h_2,\cdots,h_d}\phi_i, \qquad \forall V^{h_1,h_2,\cdots,h_d} \in \mathbb{C}^{N_{dof}},$$

where $V_i^{h_1,h_2,\cdots,h_d}$ is the $i$-th component of $V^{h_1,h_2,\cdots,h_d}$ and $\phi_i$ is the Lagrangian basis of $S^{h_1,h_2,\cdots,h_d}(\Omega)$ with $i = 1, 2, \cdots, N_{dof}$. Obviously we have $v_{h_1,h_2,\cdots,h_d} = G(V^{h_1,h_2,\cdots,h_d})$. For any $\sigma \in \mathrm{Sym}(d)$, there holds

$$v_{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}} = G(V^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}).$$

Invoking (3.6) we obtain

$$\mathcal{T}_\sigma(V^{h_1,h_2,\cdots,h_d}) = V^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}, \quad \forall V^{h_1,h_2,\cdots,h_d} \in \mathbb{C}^{N_{dof}}.$$

The process of obtaining $V^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}$ from $V^{h_1,h_2,\cdots,h_d}$ can be implemented by a symmetrization algorithm (Algorithm 3.1). Define a bijection $R_\sigma : S^{h_1,h_2,\cdots,h_d}(\Omega) \to S^{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}(\Omega)$ satisfying

$$R_\sigma = G \circ \mathcal{T}_\sigma \circ G^{-1}. \tag{4.1}$$

We have

$$v_{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}} = R_\sigma(v_{h_1,h_2,\cdots,h_d}), \quad \forall v_{h_1,h_2,\cdots,h_d} \in S^{h_1,h_2,\cdots,h_d}(\Omega). \tag{4.2}$$

It is seen from (4.1) and (4.2) that the critical process of getting $R_\sigma(v_{h_1,h_2,\cdots,h_d})$ from $v_{h_1,h_2,\cdots,h_d}$ is to calculate $\mathcal{T}_\sigma(V^{h_1,h_2,\cdots,h_d})$ from $V^{h_1,h_2,\cdots,h_d}$ by Algorithm 3.1, which requires $2d^2 N_{dof}$ operations with $N_{dof} := \prod_{i=1}^{d}(N_i+1)$. While computing $v_{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}$ by the standard finite element method usually demands $MN_{dof}$ operations with an amount $M \gg 2d^2$, due to creating stiffness matrix and solving linear algebraic systems. Consequently, if $v_{h_1,h_2,\cdots,h_d}$ has been calculated, then obtaining $v_{h_{\sigma(1)},h_{\sigma(2)},\cdots,h_{\sigma(d)}}$ through Algorithm 3.1 is much more efficient than the standard finite element method. From this observation, some efficient algorithms for (2.2) and (2.6) with symmetric solutions can be established.

## 4.1 The source problem

For the source problem (2.2) with symmetric solution, we propose a symmetrized two-scale finite element method to reduce computational cost by combining Algorithm 3.1

with the two-scale finite element method in [7, 12]. Let $h, H \in (0,1)$ and assume that $H/h$ is a positive integer. Let $w_{h\alpha+H(\mathbf{e}-\alpha)} \in S_0^{h\alpha+H(\mathbf{e}-\alpha)}(\Omega)(\mathbf{0} \leq \alpha \leq \mathbf{e})$, and define

$$B^h_{H\mathbf{e}} w_{h\mathbf{e}} = \sum_{i=1}^{d} w_{H\hat{\mathbf{e}}_i + h\mathbf{e}_i} - (d-1) w_{H\mathbf{e}}.$$

For instance, $B^h_{H,H,H} w_{h,h,h} = w_{h,H,H} + w_{H,h,H} + w_{H,H,h} - 2w_{H,H,H}$, for $d = 3$. Invoking (4.2), a symmetrized two-scale finite element method for the linear source problem (2.2) with symmetric solution is described in Algorithm 4.1.

**Algorithm 4.1.** *1. Solve (2.4) on a coarse grid: find $P_{H\boldsymbol{e}}u \in S_0^{H\boldsymbol{e}}(\Omega)$ such that*

$$a(P_{H\boldsymbol{e}}u, v) = (f, v) \quad \forall v \in S_0^{H\boldsymbol{e}}(\Omega).$$

*2. Solve (2.4) on a grid which is fine in the first coordinate direction: find $P_{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}u \in S_0^{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}(\Omega)$ such that*

$$a(P_{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}u, v) = (f, v) \quad \forall v \in S_0^{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}(\Omega).$$

*Get $R_{(1,i)}(P_{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}u) \in S_0^{h\mathrm{e}_i+H\hat{\mathrm{e}}_i}(\Omega),\ i \in \mathbb{Z}_d/\{1\}$ from $P_{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}u$ through Algorithm 3.1.*

*3. Set*

$$B^h_{H\boldsymbol{e}} P_{h\boldsymbol{e}}u = P_{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}u + \sum_{i=2}^{d} R_{(1,i)}(P_{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}u) - (d-1)P_{H\boldsymbol{e}}u.$$

For $d = 2$ or 3, there holds the following error estimates for the symmetrized two-scale finite element solution.

**Theorem 4.1.** *Assume that $\partial_{x_l} a_{ij} \in W^{1,\infty}(\Omega)$ and $\partial_{x_l} b_l \in L^\infty(\Omega)$ for $i, j, l \in \mathbb{Z}_d$. If $u \in H_0^1(\Omega) \cap W^{G,4}(\Omega)$ and $B^h_{H\boldsymbol{e}} P_{h\boldsymbol{e}}u$ is obtained from Algorithm 4.1, then*

$$\|u - B^h_{H\boldsymbol{e}} P_{h\boldsymbol{e}}u\|_{1,\Omega} \lesssim (h + H^3)\|u\|_{W^{G,4}(\Omega)},$$
$$\|u - B^h_{H\boldsymbol{e}} P_{h\boldsymbol{e}}u\|_{0,\Omega} \lesssim (h^2 + H^4)\|u\|_{W^{G,4}(\Omega)}.$$

*Proof.* Invoking Theorem 3.3 and Lemma 4.1 in [15], we have

$$\|P_{h\mathbf{e}}u - B^h_{H\mathbf{e}} P_{h\mathbf{e}}u\|_{1,\Omega} \lesssim (h^2 + H^3)\|u\|_{W^{G,4}(\Omega)}. \tag{4.3}$$

Then imitating the proof of Theorem 2.5 in [11], we complete the proof. □

**Remark 4.1.** *We may see from Algorithm 4.1 that the symmetrized two-scale finite element method is quite applicable for the source problem* (2.1) *with symmetric solution. $R_{(1,i)}(P_{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}u)$ can be obtained from $P_{h\mathrm{e}_1+H\hat{\mathrm{e}}_1}u$ through Algorithm 3.1. The computational cost of Algorithm 3.1 is far smaller than computing $P_{h\mathrm{e}_i+H\hat{\mathrm{e}}_i}u$ by the standard finite element method. For example, when $d = 3$, the amount of operations for Algorithm 3.1 to get $P_{H,h,H}u = R_{(1,2)}(P_{h,H,H}u)$ from $P_{h,H,H}u$ is only $18N_{dof}$ with $N_{dof} := (n-1)(N-1)(N-1)$. While computing $P_{H,h,H}u$ by the standard finite element method usually requires $MN_{dof}$ operations with $M \gg 18$, due to creating stiffness matrix and solving linear algebraic systems. Hence the computational cost is reduced significantly.*

**Remark 4.2.** *Combining the two-scale finite element method with the postprocessing technique, the postprocessed two-scale finite element method and the two-scale postprocessed finite element method have been proposed in [11, 15], in which the interpolation postprocessing operator $\Pi_{2\mathbf{h}}$ for uniform tensor product grids $T^{\mathbf{h}}(\Omega)$ is used. For the linear source problem* (2.1) *with symmetric solution, these methods can also be combined with Algorithm 3.1 similarly to reduce computational cost further.*

## 4.2 The eigenvalue problem

Similar to Section 4.1, for the eigenvalue problem (2.6) with symmetric solution, we combine Algorithm 3.1 with the two-scale finite element method in [7, 12] to propose the symmetrized two-scale finite element method (Algorithm 4.2).

**Algorithm 4.2.** *1. Solve (2.7) on a coarse grid: find* $(\lambda_{He}, u_{He}) \in \mathbb{R} \times S_0^{He}(\Omega)$ *satisfying* $\|u_{He}\|_{0,\Omega} = 1$ *and*

$$a(u_{He}, v) = \lambda_{He}(u_{He}, v) \qquad \forall v \in S_0^{He}(\Omega).$$

*2. Solve (2.7) on a grid which is fine in the first coordinate direction:*
*find* $(\lambda_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}, u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}) \in \mathbb{R} \times S_0^{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}(\Omega)$ *satisfying* $\|u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}\|_{0,\Omega} = 1$ *and*

$$a(u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}, v) = \lambda_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}(u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}, v) \qquad \forall v \in S_0^{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}(\Omega).$$

*For* $i \in \mathbb{Z}_d/\{1\}$, *set* $\lambda_{h\mathbf{e}_i+H\hat{\mathbf{e}}_i} = \lambda_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}$ *and obtain* $R_{(1,i)}(u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}) \in S_0^{h\mathbf{e}_i+H\hat{\mathbf{e}}_i}(\Omega)$ *from* $u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1} \in S_0^{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}(\Omega)$ *through Algorithm 3.1.*

*3. Set*

$$B_{He}^h \lambda_{he} = \sum_{i=1}^{d} \lambda_{h\mathbf{e}_i+H\hat{\mathbf{e}}_i} - (d-1)\lambda_{He},$$

$$B_{He}^h u_{he} = u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1} + \sum_{i=2}^{d} R_{(1,i)}(u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}) - (d-1)u_{He}.$$

From (4.2) we have $u_{h\mathbf{e}_i+H\hat{\mathbf{e}}_i} = R_{(1,i)}(u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1})$. For $d = 2$ or 3, there holds the following result, which is a refinement of the results in [7, 12].

**Theorem 4.2.** *Let* $(\lambda_{He}, u_{He})$ *and* $(\lambda_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}, u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1})$ *be eigenpairs of* (2.7) *that approximate the same exact eigenpair* $(\lambda, u)$. *If* $\partial_{x_l} a_{ij} \in W^{1,\infty}(\Omega)$ *for* $i, j, l \in \mathbb{Z}_d$ *and* $u \in H_0^1(\Omega) \cap W^{G,4}(\Omega)$, *then*

$$\|u - B_{He}^h u_{he}\|_{1,\Omega} \lesssim h + H^3, \tag{4.4}$$

$$|\lambda - B_{He}^h \lambda_{he}| \lesssim h^2 + H^4. \tag{4.5}$$

*Proof.* Combining Lemma 5.2 with Theorem 3.3 in [15], we obtain

$$\|B_{H\mathbf{e}}^h u_{h\mathbf{e}} - u_{h\mathbf{e}}\|_{1,\Omega} \lesssim h^2 + H^3. \tag{4.6}$$

Thus we arrive at (4.4). By Lemma 2.1, we have

$$B_{H\mathbf{e}}^h \lambda_{h\mathbf{e}} - \lambda_{h\mathbf{e}} = \lambda(u, P_{h\mathbf{e}}u - B_{H\mathbf{e}}^h P_{h\mathbf{e}}u) + \mathcal{O}(H^4).$$

Applying Theorem 4.1, we complete the proof. □

**Remark 4.3.** *For the eigenvalue problem with symmetric eigenfunction, Algorithm 3.1 is combined with the two-scale finite element method to obtain new and more efficient algorithm (Algorithm 4.2). Algorithm 3.1 can also be combined with the postprocessed two-scale finite element method in [11] and the two-scale postprocessed finite element method in [15]. Besides, some two-scale finite element discretizations for the nonlinear eigenvalue problems have been proposed and analyzed in [9]. Consequently, for the nonlinear eigenvalue problems with symmetric eigenfunctions, the two-scale finite element methods in [9] can also be combined with Algorithm 3.1 to get new algorithms.*

# 5 Numerical examples

In this section, several numerical examples, including the electronic structure calculations, are presented to illustrate the efficiency of our approaches. To optimize the cost of the computation, we choose $h = H^2$ approximately.

**Example 1** Consider a source problem with a singular coefficient:

$$\begin{cases} -\Delta u - \dfrac{1}{\sqrt{x_1^2 + x_2^2 + x_3^2}} u + x_1 x_2 x_3 u = f \quad in \quad \Omega = (0,1)^3, \\ \qquad\qquad\qquad\qquad\qquad\qquad u = 0 \quad on \quad \partial\Omega \end{cases}$$

with $f$ chosen so that the exact solution is

$$u = 2x_1x_2x_3(1-x_1)(1-x_2)(1-x_3)\mathrm{e}^{x_1+x_2+x_3}.$$

Let $L = 1.0$ be the side length of $\Omega$.

For this linear source problem with symmetric solution, the numerical results are presented in Figures 7-8 and Tables 1-2. Figure 7 supports the convergence results of Theorem 4.1. Time consumptions of different methods are presented in Figure 8. It is shown that the two-scale finite element method is more efficient than the standard finite element method. Algorithm 4.1 reduces the time consumption further compared with the two-scale finite element method, which is illustrated in more details in Tables 1-2. All of these results show that for this linear source problem with symmetric solution, considering both accuracy and computational cost, the symmetrized two-scale finite element method (Algorithm 4.1) is the preferred method.

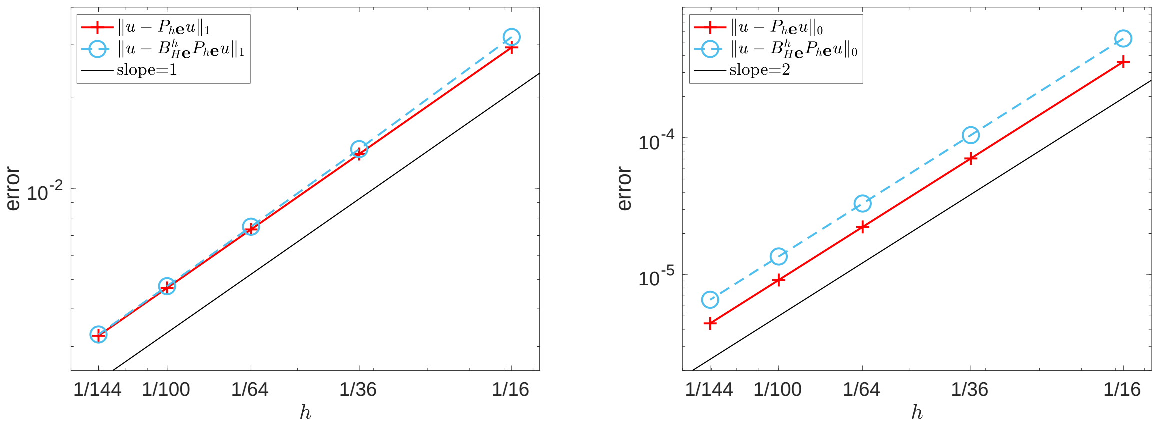

Figure 7: (Example 1) The convergence curves of solutions in $H^1$ norm and $L^2$ norm by the standard FEM and Algorithm 4.1.

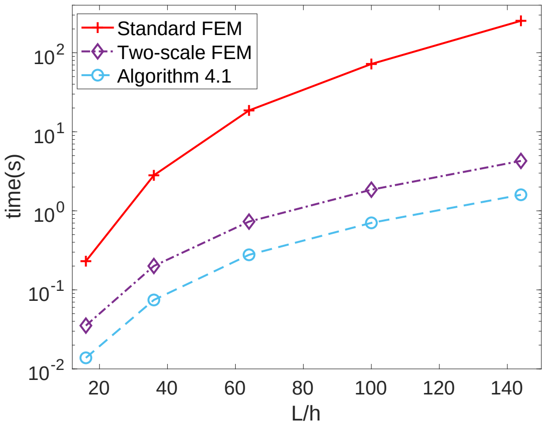

Figure 8: (Example 1) The time consumptions of the three methods.

Table 1

Example 1: The running time (s) by the two-scale FEM.

| $\frac{L}{h}\times\frac{L}{H}\times\frac{L}{H}$ | $P_{H\mathbf{e}}u$ | $P_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}u$ | $P_{h\mathbf{e}_2+H\hat{\mathbf{e}}_2}u$ | $P_{h\mathbf{e}_3+H\hat{\mathbf{e}}_3}u$ | $B^h_{H\mathbf{e}}P_{h\mathbf{e}}u$ | Total |
|---|---|---|---|---|---|---|
| $16\times4\times4$ | 0.0030 | 0.0104 | 0.0110 | 0.0104 | 0.0006 | 0.0354 |
| $36\times6\times6$ | 0.0106 | 0.0609 | 0.0624 | 0.0614 | 0.0052 | 0.2005 |
| $64\times8\times8$ | 0.0364 | 0.2250 | 0.2220 | 0.2192 | 0.0281 | 0.7307 |
| $100\times10\times10$ | 0.0754 | 0.5574 | 0.5484 | 0.5596 | 0.1104 | 1.8512 |
| $144\times12\times12$ | 0.1005 | 1.2790 | 1.2835 | 1.2941 | 0.3264 | 4.2835 |

Table 2
Example 1: The running time (s) by Algorithm 4.1.

| $\frac{L}{h}\times\frac{L}{H}\times\frac{L}{H}$ | $P_{H\mathbf{e}}u$ | $P_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}u$ | $P_{h\mathbf{e}_2+H\hat{\mathbf{e}}_2}u, P_{h\mathbf{e}_3+H\hat{\mathbf{e}}_3}u$ | $B^h_{H\mathbf{e}}P_{h\mathbf{e}}u$ | Total |
|---|---|---|---|---|---|
| $16\times4\times4$ | 0.0030 | 0.0104 | 0.0001 | 0.0003 | 0.0138 |
| $36\times6\times6$ | 0.0106 | 0.0609 | 0.0004 | 0.0026 | 0.0745 |
| $64\times8\times8$ | 0.0364 | 0.2250 | 0.0027 | 0.0135 | 0.2776 |
| $100\times10\times10$ | 0.0754 | 0.5574 | 0.0136 | 0.0581 | 0.7045 |
| $144\times12\times12$ | 0.1005 | 1.2790 | 0.0461 | 0.1708 | 1.5964 |

**Example 2** Consider a source problem:

$$\begin{cases} -\sum_{i,j=1}^{3}\frac{\partial}{\partial x_j}(a_{ij}\frac{\partial u}{\partial x_i})+\sum_{i=1}^{3}b_i\frac{\partial u}{\partial x_i}+u=f \quad in \quad \Omega=(1,2)^3, \\ \qquad\qquad\qquad\qquad u=0 \quad on \quad \partial\Omega, \end{cases}$$

where

$$A=(a_{ij})=\begin{pmatrix} \mathrm{e}^{x_1} & 1 & 1 \\ 1 & \mathrm{e}^{x_2} & 1 \\ 1 & 1 & \mathrm{e}^{x_3} \end{pmatrix}, b_1=b_2=b_3=0.001,$$

and $f$ is chosen so that we have the symmetric exact solution as follows

$$u=(1-x_1)(2-x_1)(1-x_2)(2-x_2)(1-x_3)(2-x_3)(\sin\sqrt{x_1x_2x_3})\mathrm{e}^{x_1+x_2+x_3}.$$

Let $L=1.0$ be the side length of $\Omega$.

The numerical results are presented in Figures 9-10 and Tables 3-4 for the nonsymmetric source problem with symmetric solution. Similar to Example 1, Figure 9 also supports the convergence results of Theorem 4.1. It is shown that the symmetrized two-scale finite element method (Algorithm 4.1) is still better than the other methods.

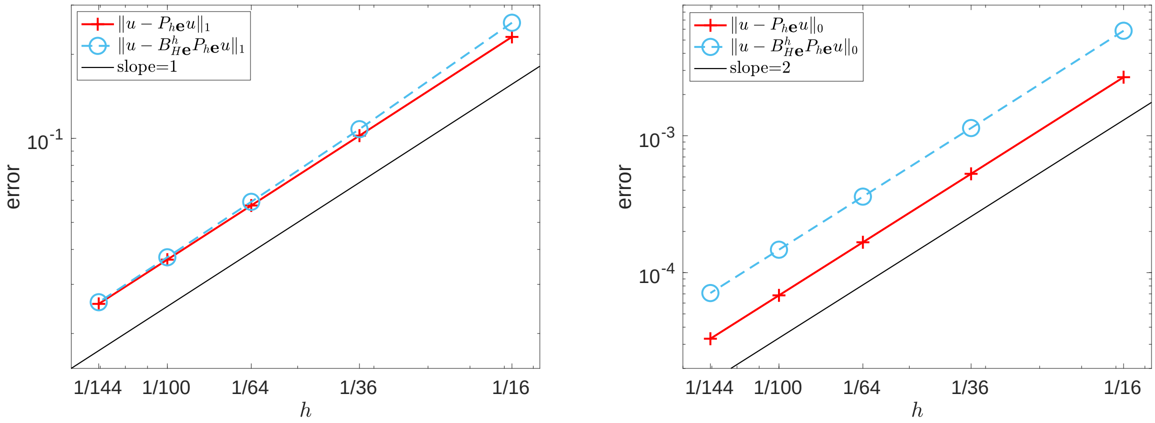

Figure 9: (Example 2) The convergence curves of solutions in $H^1$ norm and $L^2$ norm by the standard FEM and Algorithm 4.1.

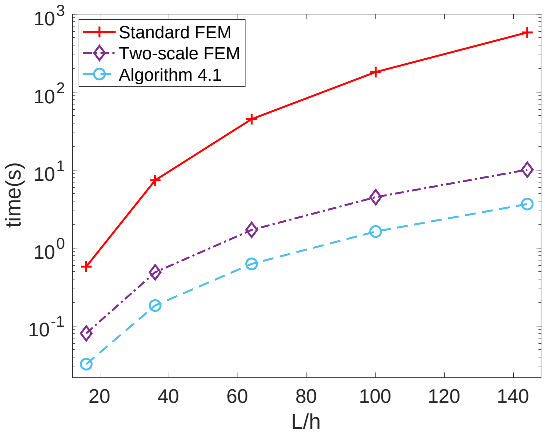

Figure 10: (Example 2) The time consumptions of the three methods.

Table 3
Example 2: The running time (s) by the two-scale FEM.

| $\frac{L}{h} \times \frac{L}{H} \times \frac{L}{H}$ | $P_{H\mathbf{e}}u$ | $P_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}u$ | $P_{h\mathbf{e}_2+H\hat{\mathbf{e}}_2}u$ | $P_{h\mathbf{e}_3+H\hat{\mathbf{e}}_3}u$ | $B^h_{H\mathbf{e}}P_{h\mathbf{e}}u$ | Total |
|---|---|---|---|---|---|---|
| $16 \times 4 \times 4$ | 0.0062 | 0.0257 | 0.0243 | 0.0236 | 0.0008 | 0.0806 |
| $36 \times 6 \times 6$ | 0.0292 | 0.1510 | 0.1539 | 0.1506 | 0.0052 | 0.4899 |
| $64 \times 8 \times 8$ | 0.0677 | 0.5428 | 0.5385 | 0.5349 | 0.0275 | 1.7114 |
| $100 \times 10 \times 10$ | 0.1424 | 1.4185 | 1.4174 | 1.4132 | 0.1095 | 4.5010 |
| $144 \times 12 \times 12$ | 0.2577 | 3.1907 | 3.1658 | 3.1772 | 0.3328 | 10.1242 |

Table 4
Example 2: The running time (s) by Algorithm 4.1.

| $\frac{L}{h} \times \frac{L}{H} \times \frac{L}{H}$ | $P_{H\mathbf{e}}u$ | $P_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}u$ | $P_{h\mathbf{e}_2+H\hat{\mathbf{e}}_2}u, P_{h\mathbf{e}_3+H\hat{\mathbf{e}}_3}u$ | $B^h_{H\mathbf{e}}P_{h\mathbf{e}}u$ | Total |
|---|---|---|---|---|---|
| $16 \times 4 \times 4$ | 0.0062 | 0.0257 | 0.0001 | 0.0005 | 0.0325 |
| $36 \times 6 \times 6$ | 0.0292 | 0.1510 | 0.0006 | 0.0028 | 0.1836 |
| $64 \times 8 \times 8$ | 0.0677 | 0.5428 | 0.0031 | 0.0137 | 0.6273 |
| $100 \times 10 \times 10$ | 0.1424 | 1.4185 | 0.0133 | 0.0570 | 1.6312 |
| $144 \times 12 \times 12$ | 0.2577 | 3.1907 | 0.0478 | 0.1748 | 3.6710 |

**Example 3** Consider a model for the quantum harmonic oscillator:

$$(-\frac{1}{2}\Delta + \frac{1}{2}r^2)u = \lambda u \quad \mathit{in}\ \mathbb{R}^3.$$

The minimum eigenvalue $\lambda = 1.5$ and the associated eigenfunction $u = \pi^{-\frac{3}{4}}\mathrm{e}^{-r^2/2}$. Since the eigenfunctions decay exponentially, in our calculation, the following eigenvalue problem is considered.

$$\begin{cases} (-\frac{1}{2}\Delta + \frac{1}{2}r^2)u = \lambda u & \mathit{in}\ \Omega, \\ \qquad\qquad\qquad u = 0 & \mathit{on}\ \partial\Omega, \end{cases}$$

where $\Omega = [-5.0, 5.0]^3$ and $r = (x^2+y^2+z^2)^{\frac{1}{2}}$. Let $L = 10.0$ be the side length of $\Omega$. The numerical results of the minimum eigenvalue and associated eigenfunction are presented in Figures 11-12 and Tables 5-6. Figure 11 supports the convergence results of Theorem 4.2. For this linear eigenvalue problem with symmetric solution, our results illustrate the efficiency of the symmetrized two-scale finite element method (Algorithm 4.2).

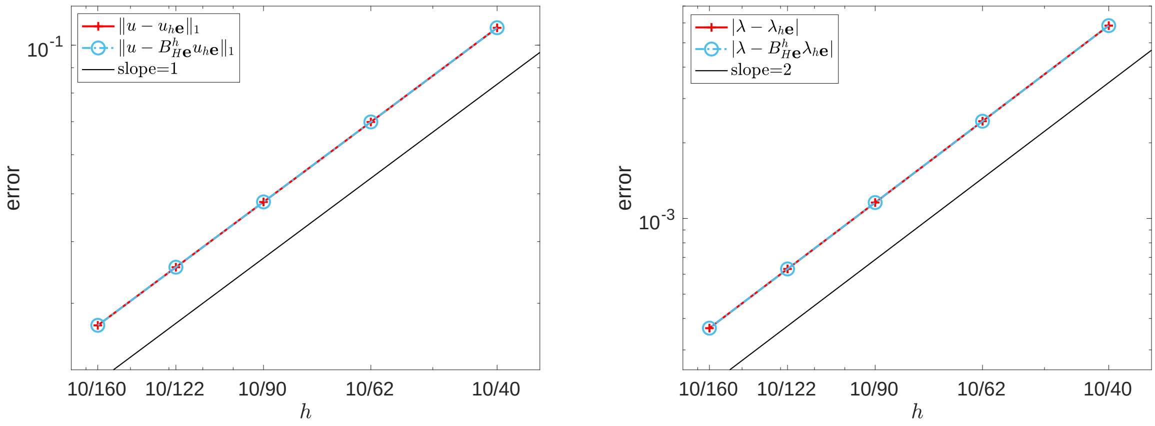

Figure 11: (Example 3) The convergence curves of eigenvalue and eigenfunction by the standard FEM and Algorithm 4.2.

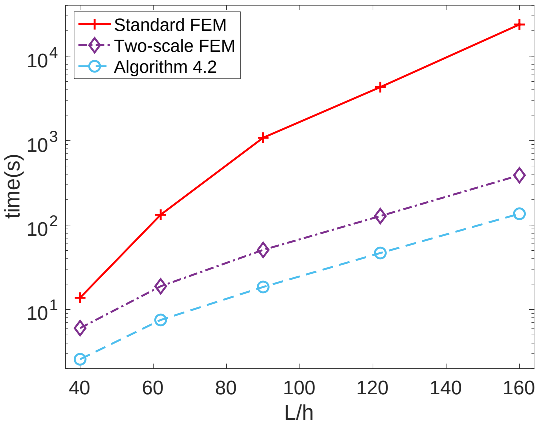

Figure 12: (Example 3) The time consumptions of the three methods.

Table 5
Example 3: The running time (s) by the two-scale FEM.

| $\frac{L}{h} \times \frac{L}{H} \times \frac{L}{H}$ | $u_{H\mathbf{e}}$ | $u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}$ | $u_{h\mathbf{e}_2+H\hat{\mathbf{e}}_2}$ | $u_{h\mathbf{e}_3+H\hat{\mathbf{e}}_3}$ | $B^h_{H\mathbf{e}}u_{h\mathbf{e}}$ | $B^h_{H\mathbf{e}}\lambda_{h\mathbf{e}}$ | Total |
|---|---|---|---|---|---|---|---|
| $40 \times 20 \times 20$ | 0.7060 | 1.7362 | 1.6241 | 1.8173 | 0.1405 | 0.0001 | 6.0242 |
| $62 \times 25 \times 25$ | 1.6529 | 5.3517 | 5.1389 | 6.1617 | 0.5373 | 0.0001 | 18.8426 |
| $90 \times 30 \times 30$ | 3.5823 | 13.2750 | 13.4933 | 18.8052 | 1.6285 | 0.0001 | 50.7844 |
| $122 \times 35 \times 35$ | 7.4273 | 35.3173 | 41.0032 | 40.0038 | 4.0751 | 0.0001 | 127.8268 |
| $160 \times 40 \times 40$ | 13.1982 | 113.4475 | 105.7242 | 146.7036 | 9.8352 | 0.0016 | 388.9103 |

Table 6
Example 3: The running time (s) by Algorithm 4.2.

| $\frac{L}{h} \times \frac{L}{H} \times \frac{L}{H}$ | $u_{H\mathbf{e}}$ | $u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}$ | $u_{h\mathbf{e}_2+H\hat{\mathbf{e}}_2}, u_{h\mathbf{e}_3+H\hat{\mathbf{e}}_3}$ | $B^h_{H\mathbf{e}}u_{h\mathbf{e}}$ | $B^h_{H\mathbf{e}}\lambda_{h\mathbf{e}}$ | Total |
|---|---|---|---|---|---|---|
| $40 \times 20 \times 20$ | 0.7060 | 1.7362 | 0.0009 | 0.1338 | 0.0001 | 2.5770 |
| $62 \times 25 \times 25$ | 1.6529 | 5.3517 | 0.0029 | 0.5145 | 0.0001 | 7.5221 |
| $90 \times 30 \times 30$ | 3.5823 | 13.2750 | 0.0118 | 1.6162 | 0.0001 | 18.4854 |
| $122 \times 35 \times 35$ | 7.4273 | 35.3173 | 0.0341 | 3.8900 | 0.0001 | 46.6688 |
| $160 \times 40 \times 40$ | 13.1982 | 113.4475 | 0.0953 | 9.3284 | 0.0003 | 136.0697 |

**Example 4** Consider the Kohn-Sham equation for the helium atom:

$$\left(-\frac{1}{2}\Delta - \frac{2}{|r|} + \int \frac{\rho(r')}{|r-r'|}dr' + V_{xc}\right) u = \lambda u \quad in\ \mathbb{R}^3,$$

with $\|u\|_{0,\mathbb{R}^3} = 1$, where $|r| = (x^2+y^2+z^2)^{\frac{1}{2}}$ and $\rho = 2|u|^2$. In our computation, we solve the following nonlinear eigenvalue problem [5]: find $(\lambda, u) \in \mathbb{R} \times H_0^1(\Omega)$ such that $\|u\|_{0,\Omega} = 1$ and

$$\begin{cases} \left(-\frac{1}{2}\Delta - \frac{2}{|r|} + \int_\Omega \frac{\rho(r')}{|r-r'|}dr' + V_{xc}\right) u = \lambda u \quad in\ \Omega, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad u = 0 \qquad on\ \partial\Omega, \end{cases} \tag{5.1}$$

where $\Omega = [-5.0, 5.0]^3$. We choose $V_{xc}(\rho) = -\frac{3}{2}\alpha(\frac{3}{\pi}\rho)^{1/3}$ with $\alpha = 0.77298$. Let $L = 10.0$ be the side length of $\Omega$.

For this nonlinear eigenvalue problem, it has been shown in [9] that the two-scale finite element method is more economic than the standard finite element method. Because the first eigenfunction of (5.1) is symmetric, the two-scale finite element discretization proposed in [9] can be also combined with Algorithm 3.1. To illustrate this observation, numerical results are presented in Figure 13 and Table 7. One can see that by combining the idea of Algorithm 3.1 with the two-scale finite element method, the time consumption is reduced further.

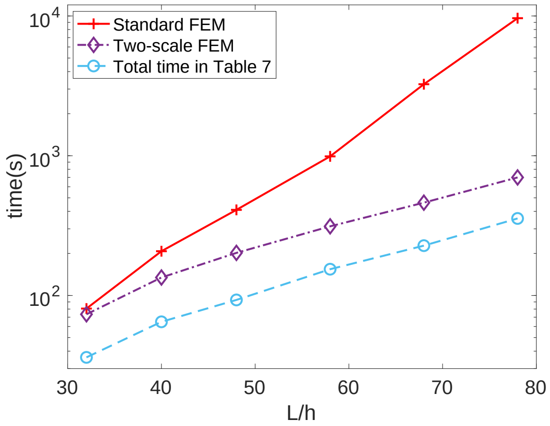

Figure 13: (Example 4) The time consumptions of the three methods.

Table 7
Example 4: The running time (s) by combining the two-scale FEM [9] with Algorithm 3.1.

| $\frac{L}{h}\times\frac{L}{H}\times\frac{L}{H}$ | $u_{H\mathbf{e}}$ | $u_{h\mathbf{e}_1+H\hat{\mathbf{e}}_1}$ | $u_{h\mathbf{e}_2+H\hat{\mathbf{e}}_2}, u_{h\mathbf{e}_3+H\hat{\mathbf{e}}_3}$ | $u^h_{H\mathbf{e}}$ | $\lambda^h_{H\mathbf{e}}$ | Total |
|---|---|---|---|---|---|---|
| $32\times 18\times 18$ | 9.62 | 18.71 | 0.0006 | 0.16 | 7.55 | 36.10 |
| $40\times 20\times 20$ | 15.11 | 34.77 | 0.0012 | 0.29 | 14.55 | 64.84 |
| $48\times 22\times 22$ | 19.87 | 46.39 | 0.0015 | 0.54 | 26.05 | 93.00 |
| $58\times 24\times 24$ | 27.12 | 79.33 | 0.0028 | 0.91 | 46.61 | 154.25 |
| $68\times 26\times 26$ | 36.50 | 112.12 | 0.0042 | 1.47 | 76.90 | 227.41 |
| $78\times 28\times 28$ | 49.10 | 187.78 | 0.0062 | 2.22 | 116.20 | 355.92 |

# 6 Conclusions

In this paper, a symmetrization algorithm is developed to perform a vector transformation associated with symmetric functions. By combining the symmetrization algorithm with the two-scale finite element methods, some symmetrized two-scale finite element methods are proposed for the elliptic source and eigenvalue problems with symmetric solutions. Numerical analyses and numerical results show that the symmetrized two-scale finite element methods save computational cost significantly compared with the corresponding two-scale finite element methods. The symmetrization algorithm can also be combined with the postprocessed two-scale finite element method to obtain efficient numerical methods [11, 15]. Besides, some two-scale finite element discretizations have been developed for a class of integral equation and integrodifferential equation [3, 14, 17], which can be also combined with the symmetrization algorithm to get more efficient algorithms. Moreover, we believe that our symmetrization algorithm is significant for solving higher dimensional problems on tensor product domains.

# Acknowledgments

The authors thank Professor Huajie Chen for enlightening discussions. P. Hou was partially supported by the National Natural Science Foundation of China (grant 11971066). F. Liu was partially supported by the National Natural Science Foundation of China (grant 11771467) and the disciplinary funding of Central University of Finance and Economics. A. Zhou was partially supported by the National Key R & D Program of China under grants 2019YFA0709600 and 2019YFA0709601.

# References

[1] I. Babuska and J.E. Osborn. Eigenvalue problems. In *Handbook of Numerical Analysis*, volume II, pages 641–787. North-Holland, Amsterdam, 1991.

[2] M. Bachmayr, G. Dusson, and C. Ortner. Polynomial approximation of symmetric functions. *arXiv*, 2109.14771, 2021.

[3] H. Chen, F. Liu, N. Reich, C. Winter, and A. Zhou. Two-scale finite element discretizations for integrodifferential equations. *J. Integ. Equ. Appl.*, 23:351–381, 2011.

[4] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 4 of *Studies in Mathematics and its Applications.* North-Holland Publishing Co., Amsterdam, 1978.

[5] X. Dai and A. Zhou. Three-scale finite element discretizations for quantum eigenvalue problems. *SIAM J. Numer. Anal.*, 46:295–324, 2008.

[6] J. Fang, X. Gao, and A. Zhou. A symmetry-based decomposition approach to eigenvalue problems. *J. Sci. Comput.*, 57:638–669, 2013.

[7] X. Gao, F. Liu, and A. Zhou. Three-scale finite element eigenvalue discretizations. *BIT*, 48(3):533–562, 2008.

[8] J. Han, Y. Li, L. Lin, J. Lu, J. Zhang, and L. Zhang. Polynomial approximation of symmetric functions. *arXiv*, 1912.01765, 2022.

[9] P. Hou and F. Liu. Two-scale finite element discretizations for nonlinear eigenvalue problems in quantum physics. *Adv. Comput. Math.*, 47:59, 2021.

[10] D. Joyner. *Adventures in group theory: Rubik's cube, Merlin's machine, and other mathematical toys.* Johns Hopkins University Press, second edition, 2008.

[11] F. Liu, M. Stynes, and A. Zhou. Postprocessed two-scale finite element discretizations, part I. *SIAM J. Numer. Anal.*, 49:1947–1971, 2011.

[12] F. Liu and A. Zhou. Two-scale finite element discretizations for partial differential equations. *J. Comput. Math.*, 24:373–392, 2006.

[13] F. Liu and A. Zhou. Localizations and parallelizations for two-scale finite element discretizations. *Commun. Pure Appl. Anal.*, 6(3):757–773, 2007.

[14] F. Liu and A. Zhou. Two-scale Boolean Galerkin discretizations for Fredholm integral equations of the second kind. *SIAM J. Numer. Anal.*, 45:296–312, 2007.

[15] F. Liu and J. Zhu. Two-scale sparse finite element approximations. *Sci. China Math.*, 59(4):789–808, 2016.

[16] C. Pflaum and A. Zhou. Error analysis of the combination technique. *Numer. Math.*, 84:327–350, 1999.

[17] Y. Xu and A. Zhou. Fast Boolean approximation methods for solving integral equations in high dimensions. *J. Integ. Equ. Appl.*, 16:83–110, 2004.