Scientific Research Publishing

American Journal of Computational Mathematics, 2026, 16(2), 143-150
https://www.scirp.org/journal/ajcm
ISSN Online: 2161-1211
ISSN Print: 2161-1203

Economic Cascadic Tensor Multigrid Method Based on Finite Element Discretization for 3D Partial Differential Equations

Jingyu Huang, Chenliang Li*

School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin, China
Email: 635282837@qq.com, *chenliang_li@hotmail.com

How to cite this paper: Huang, J.Y. and Li, C.L. (2026) Economic Cascadic Tensor Multigrid Method Based on Finite Element Discretization for 3D Partial Differential Equations. *American Journal of Computational Mathematics*, **16**, 143-150.
https://doi.org/10.4236/ajcm.2026.162008

Received: May 3, 2026
Accepted: June 26, 2026
Published: June 29, 2026

Abstract

In this paper, based on Tucker product tensor form, we present an economic cascadic tensor multigrid method(ECTMG) for the 3D partial differential equations discretized by the finite element method. Meanwhile, we provide the convergence analysis of the new method. Finally, we verify the effectiveness of the new method through some numerical examples.

Keywords

Partial Differential Equations, Economic Cascadic Tensor Multigrid Method, Tucker Product, Finite Element Method

1. Introduction

The finite element method is widely applied in fields such as industrial equipment [1], biomedicine [2] [3], and new energy catalysis [4]. Partial differential equations are discretized into linear equation systems (1) through the finite element method [5].

$$Ax = b, \tag{1}$$

$$A = \left(\sum_{k=1}^{N} I_{J_N} \otimes \cdots \otimes I_{J_{k+1}} \otimes A^{(k)} \otimes I_{J_{k-1}} \otimes \cdots \otimes I_{J_1} \right), \quad A^{(k)} \in \mathbb{R}^{J_k \times J_k} \left(k = 1, \cdots, N \right),$$

$A^{(k)} \otimes I_{J_{k-1}}$ represent as follows

$$A^{(k)} \otimes I_{J_{k-1}} = \begin{bmatrix} a_{11} I_{J_{k-1}} & \cdots & a_{1J_2} I_{J_{k-1}} \\ \vdots & \ddots & \vdots \\ a_{1J_1} I_{J_{k-1}} & \cdots & a_{J_1 J_2} I_{J_{k-1}} \end{bmatrix}.$$

Chen and Lu [6] equivalently transform the linear system (1) into the following Sylvester tensor equation

$$\mathcal{X}\times_1 A^{(1)}+\mathcal{X}\times_2 A^{(2)}+\cdots+\mathcal{X}\times_N A^{(N)}=\mathcal{D}, \tag{2}$$

with $A^{(k)}\in\mathbb{R}^{J_k\times J_k}\left(k=1,\cdots,N\right)$ and $\mathcal{D}\in\mathbb{R}^{J_1\times\cdots\times J_N}$, $\mathcal{X}\in\mathbb{R}^{J_1\times J_2\times\cdots\times J_N}$. The mode-k product of $\mathcal{X}$ and $\mathcal{A}$ is denoted by $\mathcal{X}\times_k A^{(k)}$, the result is of size $J_1\times\cdots\times J_{k-1}\times J\times J_{k+1}\times\cdots\times J_N$, and its entries are defined by

$$\left(\mathcal{X}\times_k A^{(k)}\right)_{i_1\cdots i_{k-1}ji_{k+1}\cdots i_N}=\sum_{i_k=1}^{J_k}x_{i_1\cdots i_{k-1}i_k i_{k+1}\cdots i_N}a^{(k)}_{ji_k}.$$

Compared with Equation (1), Equation (2) reduces the storage space required for the solution process, thereby improving the solution efficiency. Grasedyck *et al.* [5] utilized the finite element method and expressed $A^{(i)}$ in Equation (2) as follows

$$A^{(i)}=\left(M^{(i)}\right)^{-1}\tilde{A}^{(i)}.$$

Here, $M^{(i)}$ represents the mass matrix, and $\tilde{A}^{(i)}$ is the coefficient matrix obtained by the finite element method.

The structure of this paper is as follows. In Section 2, we discretize the high-dimensional partial differential equation by the finite element method, and give the convergence analysis of the new method. Section 3 shows and discusses the numerical results. Finally, in Section 4, we conclude this article with some content.

2. Main Result

2.1. Discretization of 3D PDE by Cubic Element

Let $\Omega\subset\mathbb{R}^3$ be a bounded Lipschitz domain with boundary $\partial\Omega=\Gamma_D\cup\Gamma_N$, $\Gamma_D\cap\Gamma_N=\varnothing$, and $\operatorname{meas}\left(\Gamma_D\right)>0$. Consider the following elliptic boundary value problem:

$$\begin{cases}-\nabla\cdot\left(A\nabla u\right)+\boldsymbol{b}\cdot\nabla u+cu=f & \text{in }\Omega,\\ u=0 & \text{on }\Gamma_D,\\ \left(A\nabla u\right)\cdot\boldsymbol{n}=g & \text{on }\Gamma_N,\end{cases}$$

where

- $A\left(x\right)\in\mathbb{R}^{9\times9}$ is symmetric and positive definite: there exist constants $0<\alpha\le\beta<\infty$ such that

$$\alpha\left\|\xi\right\|^2\le\xi^{\mathrm{T}}A\left(x\right)\xi\le\beta\left\|\xi\right\|^2\quad\forall\xi\in\mathbb{R}^3,\ \text{a.e. } x\in\Omega;$$

- $\boldsymbol{b}\left(x\right)\in\mathbb{R}^3$ is a bounded vector field; $c\left(x\right)\ge0$ is a bounded scalar;
- $f\in L^2\left(\Omega\right)$, $g\in L^2\left(\Gamma_N\right)$ are given data; $\boldsymbol{n}$ is the outward unit normal on Γ_N.

Define the function space $V=\left\{v\in H^1\left(\Omega\right):v|_{\Gamma_D}=0\right\}$. Multiplying the PDE by a test function $v\in V$ and integrating by parts yields the weak problem: find $u\in V$ such that

$$a\left(u,v\right)=\ell\left(v\right)\quad\forall v\in V,$$

with

$$\begin{cases} a(u,v)=\int_{\Omega}\left((A\nabla u)\cdot\nabla v+(\boldsymbol{b}\cdot\nabla u)v+cuv\right)\mathrm{d}x, \\ \ell(v)=\int_{\Omega} fv\mathrm{d}x+\int_{\Gamma_N} gv\mathrm{d}s. \end{cases}$$

Then its variational form is as follows

$$-\iiint_V\left[\nabla\cdot(A\nabla u)\right]v\mathrm{d}V+\iiint_V(\boldsymbol{b}\cdot\nabla u)v\mathrm{d}V+\iiint_V cuv\mathrm{d}V=\iiint_V fv\mathrm{d}V.$$

Thus, the complete weak form is obtained

$$\iiint_V\left(A\nabla u\cdot\nabla v+(\boldsymbol{b}\cdot\nabla u)v+cuv\right)\mathrm{d}V-\iint_{\partial V} v\frac{\partial u}{\partial n_A}\mathrm{d}S=\iiint_V fv\mathrm{d}V.$$

Let

$$N_x^{(l)},N_y^{(l)},N_z^{(l)}=2^{l+1},$$

$$h=\frac{1}{N_i^{(l)}+1}(i=x,y,z),$$

it can be obtained that

$$\Delta x=\frac{x_L-x_0}{N_x^{(l)}},\Delta y=\frac{y_L-y_0}{N_{ya}^{(l)}},\Delta z=\frac{z_L-z_0}{N_z^{(l)}}.$$

Figure 1 presents the cube element.

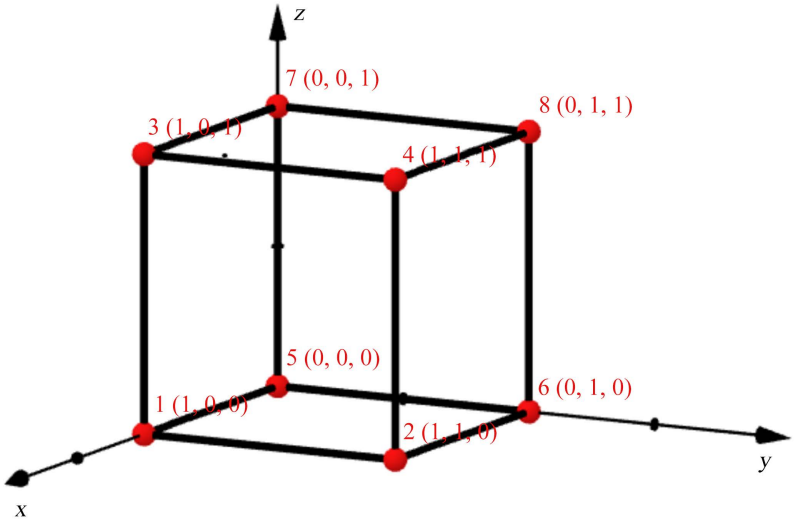

Figure 1. Cube element.

We write the following basic functions,

$$\begin{aligned} \hat{L}_1&=\xi(1-\eta)(1-\gamma), & \hat{L}_2&=\xi\eta(1-\gamma), \\ \hat{L}_3&=\xi(1-\eta)\gamma, & \hat{L}_4&=\xi\eta\gamma, \\ \hat{L}_5&=(1-\xi)(1-\eta)(1-\gamma), & \hat{L}_6&=(1-\xi)\eta(1-\gamma), \\ \hat{L}_7&=(1-\xi)(1-\eta)\gamma, & \hat{L}_8&=(1-\xi)\eta\gamma. \end{aligned}$$

And the coordinates are mapped as

$$x=\sum_{i=1}^{8} x_i \hat{L}_i(\xi,\eta,\gamma),$$
$$y=\sum_{i=1}^{8} y_i \hat{L}_i(\xi,\eta,\gamma),$$
$$z=\sum_{i=1}^{8} z_i \hat{L}_i(\xi,\eta,\gamma).$$

Furthermore, it can be concluded that the Jacobian matrix J is

$$J=\begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \gamma} & \frac{\partial y}{\partial \gamma} & \frac{\partial z}{\partial \gamma} \end{vmatrix}.$$

Therefore,

$$\begin{aligned} K_{ij} &= \iiint_{e_1} \nabla L_i \nabla L_j \nabla L_k \mathrm{d}x\mathrm{d}y\mathrm{d}z \\ &= \iiint_{\hat{e}_1} \nabla \hat{L}_i \nabla \hat{L}_j \nabla \hat{L}_k |J| \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\gamma. \end{aligned}$$

After assembling the stiffness matrix of the assembly unit, the linear system (1) can be obtained,

$$A_l x_l = b_l.$$

We have converted it equivalently to

$$\mathcal{X} \times_1 A_l^{(1)} + \mathcal{X} \times_2 A_l^{(2)} + \mathcal{X} \times_3 A_l^{(3)} = \mathcal{D}_l, \tag{3}$$

where, $A_l^{(i)} \in \mathbb{R}^{J_k \times J_k}$ ($k=1,2,3$) and $\mathcal{D}_l$ is the right-hand term, the structure of coefficient matrix $A_l^{(i)}$ is as follows

$$A_l^{(i)} = \left(M_l^{(i)}\right)^{-1} \tilde{A}_l^{(i)},$$

Here, $M_l^{(i)}$ represents the mass matrix, and $\tilde{A}_l^{(i)}$ is the coefficient matrix obtained by the finite element method.

2.2. Inverse of Mass Matrix M

According to [7], we can use the Stenger quadrature formula to approximate the inverse of the mass matrix M.

Let $M \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix, with all eigenvalues $\lambda > 0$. Consider the integral

$$\int_0^{\infty} \mathrm{e}^{tM} \mathrm{d}t.$$

For each eigenvalue λ, the integral is

$$\int_0^{\infty} \mathrm{e}^{-\lambda t} \mathrm{d}t.$$

Due to the relationship between the matrix exponential and the eigenvalues, we have

$$\int_0^{\infty} \mathrm{e}^{-tM}\mathrm{d}t = M^{-1}.$$

Let $p \in \mathbb{N}$, Step length is $h_{\mathrm{st}} = \pi^2/\sqrt{p}$, the integration nodes and weights are as follows

$$t_j = \ln\left(\mathrm{e}^{jh_{\mathrm{st}}} + \sqrt{1+\mathrm{e}^{2jh_{\mathrm{st}}}}\right), \quad j = -p, \cdots, p.$$

$$w_j = \frac{h_{\mathrm{st}}}{\sqrt{1+\mathrm{e}^{-2jh_{\mathrm{st}}}}}.$$

Then the integral is approximately equal to

$$\int_0^{\infty} \mathrm{e}^{-t\lambda}\mathrm{d}t \approx \sum_{j=-p}^{p} w_j \mathrm{e}^{-t_j\lambda}.$$

2.3. ECTMG Algorithm

Combining Sections 2.1 and 2.2, we present the following iterative method (**Algorithm 1**).

Algorithm 1. The calculation of $A^{(i)}$

Require: Symmetric positive definite matrix $M \in \mathbb{R}^{n\times n}$, integer $p > 0$, coefficient matrix $\tilde{A}^{(i)}$

$[V, D] \coloneqq \mathrm{eig}(M)$, $\lambda \coloneqq \mathrm{diag}(D)$, $h_{\mathrm{st}} \coloneqq \pi^2/\sqrt{p}$
for $j = -p, \ldots, p$ **do**
  $t_j \coloneqq \ln\left(e^{jh_{\mathrm{st}}} + \sqrt{1+e^{2jh_{\mathrm{st}}}}\right)$
  $w_j \coloneqq h_{\mathrm{st}}/\sqrt{1+e^{-2jh_{\mathrm{st}}}}$
end for
$S \coloneqq \sum_{j=-p}^{p} w_j \exp(-t_j\lambda)$
$M^{-1} \coloneqq VSV^T$
return $A^{(i)} \coloneqq M^{-1}\tilde{A}^{(i)}$

Similar to [7], we employ the following algorithm for the solution (**Algorithm 2**).

Algorithm 2. ECTMG algorithm [7].

Require: Given the maximum number of mesh grid L, initial guess value $\mathcal{X}_0$, the coefficient matrices $A_l^{(1)}, A_l^{(2)}, A_l^{(3)}$ of the Sylvester tensor equation and the right-hand side term $\mathcal{D}_l$.

On the coarsest grid level: $\mathcal{X}_1$: $\mathcal{X}_1 \coloneqq \mathscr{T}_1\mathcal{X}_0$.
for $l = 2, \ldots, L$ **do**
  $\mathcal{X}_l \coloneqq \mathcal{P}_{l-1}^{l}\mathcal{X}_{l-1}$;
  Perform m_l smoothing iterations on $\mathcal{X}_l$: $\mathcal{X}_l^{m_l} \coloneqq \mathscr{T}_l^{m_l}\mathcal{X}_l$;
  $\mathcal{X}_{l-1} \coloneqq \mathcal{X}_l^{m_l}$
end for
return $\mathcal{X} = \mathcal{X}_L$

With $\mathcal{P}_{l-1}^{l}$ is prologation [8], $\mathscr{T}_l^{m_l}$ is the smoother. And m_l represents the number of iterations, we similar to [9] provides the following selection rules:

When $d=2$

1) If $l>L_0$, then $m_l=\left\lfloor m_L\beta^{L-l}\right\rfloor$.

2) If $l\le L_0$, $m_l=\left\lfloor m_*^{\frac{1}{2}}\left(L-\left(2-\varepsilon_0\right)l\right)\kappa_l\right\rfloor$.

When $d\ge 3$

1) If $l>L_0$, then $m_l=\left\lfloor m_L\beta^{L-l}\right\rfloor$.

2) If $l\le L_0$, there are two cases:

a) If $\left(2-\varepsilon_0\right)L_0\le L$, then $m_l=\left\lfloor m_*^{\frac{1}{2}}\left(L-\left(2-\varepsilon_0\right)l\right)\kappa_l\right\rfloor$.

b) If $\left(2-\varepsilon_0\right)L_0>L$, there exists a positive integer $L'<L_0$ such that $\left(2-\varepsilon_0\right)L'\le L$, for all $l\le L'$, choose $m_l=\left\lfloor m_*\left(L-\left(2-\varepsilon_0\right)l\right)\kappa_l\right\rfloor$.

Remark Let $\kappa_l=h_l^{-2}$, and ε_0 be a positive constant in the interval [0, 1].

3. Numerical Example

In this section, we verify the effectiveness of the ECTMG algorithm through some numerical examples. All tests will be done with configuration: 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70 GHz 2.69 GHz. Let CPU(s) represent the iteration time. $\varepsilon_0=0.1$.

We define the error between the exact solution $\mathcal{X}^*$ and the numerical solution $\mathcal{X}^k$ as follows

$$E\left(\mathcal{X}^*-\mathcal{X}^k\right)=\max_{i_1,i_2,\cdots,i_N}\left|x^*_{i_1,i_2,\cdots,i_N}-x^k_{i_1,i_2,\cdots,i_N}\right|,$$

Obviously, we have $E\left(\mathcal{X}^*-\mathcal{X}^k\right)=\left\|vec\left(\mathcal{X}^*-\mathcal{X}^k\right)\right\|_\infty$. In the table of numerical examples in the following text, the symbol † indicates insufficient memory during the algorithm's execution.

Example 1. ([8]) *Consider the following Poisson equation*

$$\begin{aligned}-\triangle u&=f \quad \text{in} \quad V=[0,1]^3\\ u&=0 \quad \text{on} \quad \partial V.\end{aligned}$$

Taking the partition step length as $h=\frac{1}{n+1}$, using the hexahedral element, we can obtain the Sylvester tensor Equation (1), with $A^{(k)}$ in Equation (2) as follows

$$A^{(k)}=\left(M^{(k)}\right)^{-1}\tilde{A}^{(k)},$$

with $\tilde{A}^{(k)}$ as follows

$$\tilde{A}^{(k)}=\frac{2}{9}\begin{bmatrix}4&-2&0\\-2&4&-2\\0&-2&4\end{bmatrix}.$$

and $M^{(k)}$ as follows

$$M^{(k)}=\frac{1}{4}\begin{bmatrix}4&1&0\\1&4&1\\0&1&4\end{bmatrix}.$$

Set $V_l = 2 \Big/ \sum_{k=1}^{3} \left(\alpha_l^k \beta_l^k \right)$, where α_l^k and β_l^k are the maximum and minimum eigenvalues of matrix $A_l^{(k)}$, respectively. Select the right-hand side tensor $\mathcal{B}$ such that the solution to the corresponding 3-order Poisson equation is $u = \frac{1}{3} \cdot \prod_{k=1}^{3} \left(x_k - x_k^2 \right)$, where $\left(x_k \right)_i = i/(n+1)$, $i = 1, 2, \cdots, n$. Table 1 and Table 2 lists the numerical results of BiCG_BTF and ECTMG(BiCG_BTF) algorithms.

Table 1. Numerical results for $n = 255 \times 255 \times 255$.

Algorithm	CPU (s)	$\mathbb{E}\left(\mathcal{X}^* - \mathcal{X}_L \right)$	IT
BiCG_BTF	663.32	2.20×10^{-7}	994
ECTMG (BiCG_BTF)	142.81	4.72×10^{-7}	(3, 205, 256, 1024, 4096, 512, 64)

Remark: In the BiCG_BTF algorithm, IT represents the total number of iterations required for the entire algorithm process; in the ECTMG algorithm, it is the number of iterations m_l for the corresponding level.

Table 2. Numerical results for $n = 511 \times 511 \times 511$.

Algorithm	CPU (s)	$\mathbb{E}\left(\mathcal{X}^* - \mathcal{X}_L \right)$	IT
BiCG_BTF	14076.29	4.96×10^{-7}	1958
ECTMG (BiCG_BTF)	790.07	5.78×10^{-7}	(3, 269, 256, 1024, 15,360, 1920, 240, 30)

As can be seen from Table 1 and Table 2, through the linear system (2) discretized by the finite element method, the ECTMG algorithm can still efficiently solve the problem and maintain the same accuracy as BiCG_BTF, which verifies the feasibility of the finite element discretized equation for this solution framework.

4. Conclusion

Based on the finite element method, we discretize the three-dimensional partial differential equations and solves it using the economic cascadic tensor multigrid method. The numerical results show that the new method is effective.

Acknowledgements

We acknowledge the financial support from the Natural Science Foundation of China under Grant 12161027 and the Science and Technology Project of Guangxi (Guike AD25069086).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Zhao, Y., Chang, J.C. and Liang, J.L. (2024) Simulation of Temperature Field of Rotary Kiln Burner Based on Finite Element Method. *Applied Mathematics Journal of Chinese Universities*, **39**, 51-63. (In Chinese)

[2] Wu, H.X., Liu, X.Y., Wang, T.Y., *et al.* (2026) Finite Element Simulation of Scoliosis with Muscle Unit Introduction: Verification of Correction Effect under Bidirectional Load. *Chinese Journal of Tissue Engineering Research*, **30**, 2172-2181. (In Chinese)

[3] Gao, X. and Xing, W.H. (2023) Application of Finite Element Analysis in Spine Surgery. *Chinese Journal of Tissue Engineering Research*, **27**, 2921-2927. (In Chinese)

[4] Nie, Q.H., Song, X.D., Liu, S., *et al.* (2025) Advances in Application of Finite Element Numerical Simulations to Electrocatalytic Reduction Systems of Energy-Related Small Molecules. *Clean Coal Technology*, **31**, 1-13. (In Chinese)

[5] Grasedyck, L. (2004) Existence and Computation of Low Kronecker-Rank Approximations for Large Linear Systems of Tensor Product Structure. *Computing*, **72**, 247-265. https://doi.org/10.1007/s00607-003-0037-z

[6] Chen, Z. and Lu, L. (2012) A Projection Method and Kronecker Product Preconditioner for Solving Sylvester Tensor Equations. *Science China Mathematics*, **55**, 1281-1292. https://doi.org/10.1007/s11425-012-4363-5

[7] Huang, J.Y. and Li, C.L. (2026) An Economic Cascadic Tensor Multigrid Method for Solving High Dimensional Elliptic Linear Partial Differential Problems. arXiv preprint arXiv: 2606. 25643.

[8] Ballani, J. and Grasedyck, L. (2013) A Projection Method to Solve Linear Systems in Tensor Format. *Numerical Linear Algebra with Applications*, **20**, 27-43. https://doi.org/10.1002/nla.1818

[9] Shi, Z., Xu, X. and Huang, Y. (2007) Economical cascadic multigrid method (ECMG). *Science in China Series A*: *Mathematics*, **50**, 1765-1780. https://doi.org/10.1007/s11425-007-0127-z