Received: 18 January 2019 | Revised: 17 June 2019 | Accepted: 17 July 2019

DOI: 10.1002/nme.6173



RESEARCH ARTICLE

WILEY

# Numerical recipes for elastodynamic virtual element methods with explicit time integration

**Kyoungsoo Park**[1] | **Heng Chi**[2] | **Glaucio H. Paulino**[2]

[1]Department of Civil and Environmental Engineering, Yonsei University, Seoul, South Korea

[2]School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, Georgia

**Correspondence**
Kyoungsoo Park, Department of Civil and Environmental Engineering, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 120-749, South Korea.
Email: k-park@yonsei.ac.kr

**Funding information**
Ministry of Science, ICT and Future Planning, Grant/Award Number: NRF 2018R1A2B6007054; Ministry of Trade, Industry and Energy, Grant/Award Number: KETEP 20171510101910; US National Science Foundation (NSF), Grant/Award Number: 1624232; Raymond Allen Jones Chair at the Georgia Institute of Technology

**Summary**

We present a general framework to solve elastodynamic problems by means of the virtual element method (VEM) with explicit time integration. In particular, the VEM is extended to analyze nearly incompressible solids using the B-bar method. We show that, to establish a B-bar formulation in the VEM setting, one simply needs to modify the stability term to stabilize only the deviatoric part of the stiffness matrix, which requires no additional computational effort. Convergence of the numerical solution is addressed in relation to stability, mass lumping scheme, element size, and distortion of arbitrary elements, either convex or nonconvex. For the estimation of the critical time step, two approaches are presented, ie, the maximum eigenvalue of a system of mass and stiffness matrices and an effective element length. Computational results demonstrate that small edges on convex polygonal elements do not significantly affect the critical time step, whereas convergence of the VEM solution is observed regardless of the stability term and the element shape in both two and three dimensions. This extensive investigation provides numerical recipes for elastodynamic VEMs with explicit time integration and related problems.

**KEYWORDS**

convex and nonconvex elements, critical time step, elastodynamics, lumped mass matrix, virtual element method (VEM)

## 1 | INTRODUCTION

The virtual element method (VEM) has been recently developed to effectively and efficiently handle general polygonal and polyhedral elements.[1,2] Because of the flexibility and robustness in selecting element shapes, the VEM was employed to solve various partial differential equations of physical problems, such as elliptic, hyperbolic, and parabolic problems.[3-6] For engineering applications, elastostatic problems were investigated for small and finite strain conditions.[7-10] Computational results demonstrated that the VEM provided a more stable solution than the finite element method (FEM) for soft materials under a large deformation, and a locking-free behavior was observed for both nearly incompressible and incompressible materials.[9,10] Elastodynamic examples with nonconvex meshes were solved, and computational results illustrated that the VEM was able to consistently handle general nonconvex elements.[11] Furthermore, the VEM was also utilized to investigate fracture problems,[12-14] contact mechanics,[15] and topology optimization.[16,17]

In this study, two-dimensional (2D) and three-dimensional (3D) elastodynamic problems are comprehensively investigated in conjunction with the stability and convergence of a numerical solution for explicit time integration. In particular, the VEM is extended to investigate nearly incompressible solids using the B-bar method. For the estimation of the critical time step to obtain a stable VEM solution, two approaches are suggested, ie, the maximum eigenvalue of a local system

and an effective element length. Computational results demonstrate that both approaches provide a good approximation of the critical time step. Additionally, the effect of the stability term, the mass lumping scheme, and the element size and distortion on the critical time step is demonstrated for convex and nonconvex elements. Furthermore, convergence of the VEM solution is illustrated according to the stabilization scheme for both convex and nonconvex elements.

The remainder of this paper is organized as follows. The VEM formulation constructed using a linear space and its extension to the B-bar method are presented in Section 2. Section 3 explains mass lumping techniques and critical time-step approximation, in which we define an effective element length for convex and nonconvex elements. Then, critical time-step estimation and elastodynamic analysis are demonstrated in Sections 4 and 5, respectively. Finally, key findings of the present study are summarized in Section 6.

## 2 | BASIC ELASTODYNAMIC VEM FORMULATION

A solid domain ($\Omega$) has displacement boundary conditions on the displacement boundary ($\partial\Omega_{\mathbf{u}}$), and external tractions ($\mathbf{t}$) are applied on the traction boundary ($\partial\Omega_{\mathbf{t}}$). A weak form of elastodynamic problems is given as

$$\int_{\Omega} \mathbf{v} \cdot \rho_0 \ddot{\mathbf{u}} d\mathbf{x} + \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{v}) : \boldsymbol{\sigma}(\mathbf{u}) d\mathbf{x} = \int_{\partial\Omega_{\mathbf{t}}} \mathbf{v} \cdot \mathbf{t} d\mathbf{x} \quad \forall \mathbf{v} \in \mathcal{K}_0, \tag{1}$$

where $\mathbf{u}$ and $\ddot{\mathbf{u}}$ are displacement and acceleration vectors, respectively, and $\rho_0$ is the density of a material. The virtual displacement field ($\mathbf{v}$) satisfies the homogeneous essential boundary condition on the displacement boundary ($\partial\Omega_{\mathbf{u}}$), and $\mathcal{K}_0$ is a subset of the space of admissible displacement $\mathcal{K}$. In this study, the solid is assumed to be linear isotropic elastic, and thus, the stress tensor ($\boldsymbol{\sigma}$) is simply obtained from the linear elasticity tensor ($\mathbb{C}$) and the linearized strain tensor ($\boldsymbol{\epsilon}$), ie, $\boldsymbol{\sigma}(\mathbf{u}) = \mathbb{C}\boldsymbol{\epsilon}(\mathbf{u})$.

The domain ($\Omega$) is discretized into nonoverlapping elements. A generic element ($E$) in this discretization is assumed to be polygonal elements ($F$) in 2D and polyhedral elements ($P$) with planar faces in 3D. Within a discretized domain ($\Omega_h$), the global virtual element space ($\mathcal{K}_h$) is defined in relation with the local virtual element space $\mathcal{V}(E)$, which is expressed as

$$\mathcal{K}_h = \left\{ \mathbf{v}_h \in \mathcal{K} : \mathbf{v}_h|_E \in [\mathcal{V}(E)]^d, \forall E \in \Omega_h \right\}, \tag{2}$$

where $d$ is the dimension number and any admissible displacement field ($\mathbf{v}_h$) in $\mathcal{K}_h$ belongs to the local virtual space $[\mathcal{V}(E)]$ of an element ($E$). A 2D local virtual element space $[\mathcal{V}(F)]$ is first defined in a polygon ($F$), whereas a 3D local virtual element space $[\mathcal{V}(P)]$ of a polyhedron ($P$) is constructed from the 2D local space.[18] In this study, the linear virtual element space is considered, and the extension of linear to higher-order virtual elements can be found in the literature.[19] In the following subsections, 2D and 3D local virtual element spaces are defined, and two $L^2$-projection operators are explained for the discretization of the weak form. Then, a procedure to construct mass and stiffness matrices is described.

### 2.1 | Virtual element space in 2D

A preliminary 2D local virtual element space[18] is first introduced as

$$\tilde{\mathcal{V}}(F) = \left\{ v_h \in \mathcal{H}^1(F) : \Delta v_h \in \mathcal{P}(F) \text{ in } F, v_{h|e} \in \mathcal{P}(e) \, \forall \, e \in \partial F \right\}, \tag{3}$$

where $\Delta$ and $e$ denote the Laplacian operator and a generic edge of an element, respectively, and $\mathcal{P}(\cdot)$ is the space of linear polynomial functions. The preliminary space contains functions that satisfy first-order Laplacians in a polygon and linear variations on the element boundary ($\partial F$). Because $\mathcal{P}(F)$ is a subset of $\tilde{\mathcal{V}}(F)$, a projection operator ($\Pi^{\nabla}$) is defined by projecting functions in $\tilde{\mathcal{V}}(F)$ onto $\mathcal{P}(F)$, ie, $\Pi^{\nabla} v_h \in \mathcal{P}(F)$, such that

$$\int_F \nabla(\Pi^{\nabla} v_h) \cdot \nabla p \, d\mathbf{x} = \int_F \nabla v_h \cdot \nabla p \, d\mathbf{x} \qquad \forall p \in \mathcal{P}(F) \tag{4}$$

and

$$\sum_{\mathbf{x}_v \in F} \Pi^{\nabla} v_h(\mathbf{x}_v) = \sum_{\mathbf{x}_v \in F} v_h(\mathbf{x}_v), \tag{5}$$

where $\mathbf{x}_v$ is the position vector of a generic vertex ($v$) of $F$. A linear polynomial function $p$ consists of scaled monomials ($m_\alpha$), given as

$$p = \sum_{\alpha=1}^{3} a_\alpha m_\alpha, \tag{6}$$

where $a_\alpha$ are coefficients of scaled monomials and $p$ is a polynomial function. Scaled monomials ($m_\alpha$) are selected as

$$m_1 = 1, \quad m_2 = \frac{x - x_c}{h_F}, \quad m_3 = \frac{y - y_c}{h_F}, \tag{7}$$

where $(x_c, y_c)$ are the coordinates of the centroid of a polygon and $h_F$ is a length associated with polygonal element size. Based on the projection operator, the 2D local virtual element space is defined as

$$\mathcal{V}(F) = \left\{ v_h \in \tilde{\mathcal{V}}(F) \,:\, \int_F \left(\Pi^{\nabla} v_h - v_h\right) q \, d\mathbf{x} = 0 \quad \forall\, q \in \mathcal{P}(F) \right\}, \tag{8}$$

where $q$ is an arbitrary function that is in the linear polynomial space $\mathcal{P}(F)$.

One should note that the projection operator is uniquely evaluated using a complete set of degrees of freedom (DOFs) in 2D. The integration by parts of Equation (4) leads to

$$\int_F \nabla(\Pi^{\nabla} v_h) \cdot \nabla p \, d\mathbf{x} = \int_{\partial F} v_h \nabla p \cdot \mathbf{n}_e \, d\mathbf{x} - \int_F v_h \Delta p \, d\mathbf{x}, \tag{9}$$

where $\mathbf{n}_e$ is the unit normal vector on an edge. The first term on the right-hand side is exactly computed using the values of $v_h$ at $\mathbf{x}_v$, which correspond to DOFs, because $\nabla p$ is a constant and $v_h$ is a linear function on edges. The second term vanishes because $\Delta p$ is zero for a linear polynomial function.

## 2.2 | Virtual element space in 3D

A preliminary 3D local space[18] within a polyhedral element is defined as

$$\tilde{\mathcal{V}}(P) = \left\{ v_h \in \mathcal{H}^1(P) \,:\, \Delta v_h \in \mathcal{P}(P) \text{ in } P, v_h\big|_F \in \mathcal{V}(F) \,\forall\, F \in \partial P \right\}, \tag{10}$$

where functions on faces of a polyhedron are in the local space of $\mathcal{V}(F)$. Similarly to the 2D case, functions in $\tilde{\mathcal{V}}(P)$ are projected onto $\mathcal{P}(P)$, ie, $\Pi^{\nabla} v_h \in \mathcal{P}(P)$, such that

$$\int_P \nabla(\Pi^{\nabla} v_h) \cdot \nabla p \, d\mathbf{x} = \int_P \nabla v_h \cdot \nabla p \, d\mathbf{x} \quad \forall p \in \mathcal{P}(P) \tag{11}$$

and

$$\sum_{\mathbf{x}_v \in P} \Pi^{\nabla} v_h(\mathbf{x}_v) = \sum_{\mathbf{x}_v \in P} v_h(\mathbf{x}_v). \tag{12}$$

A linear polynomial space $\mathcal{P}(P)$ has four monomial basis functions, ie,

$$m_1 = 1, \quad m_2 = \frac{x - x_c}{h_P}, \quad m_3 = \frac{y - y_c}{h_P}, \quad m_4 = \frac{z - z_c}{h_P}, \tag{13}$$

where $h_P$ is a length associated with polyhedral element size. Finally, the 3D local virtual element space is given as

$$\mathcal{V}(P) = \left\{ v_h \in \tilde{\mathcal{V}}(P) \,:\, \int_F \left(\Pi^{\nabla} v_h - v_h\right) q d\mathbf{x} = 0 \,\forall\, q \in \mathcal{P}(P) \right\}. \tag{14}$$

Furthermore, the projection operator in 3D is also uniquely identified using the integration by parts and a complete set of DOFs in 3D, associated with the integration on the element boundary (ie, face), as explained in the 2D case.

## 2.3 | $L^2$-projection operators

For the construction of mass and stiffness matrices, two $L^2$-projection operators are employed, which are applicable for both 2D and 3D local virtual element spaces. The first projection operator ($\Pi^0$) is defined by projecting $v_h \in \mathcal{V}_k$ onto $\mathcal{P}(E)$, such that

$$\int_E \left(\Pi^0 v_h\right) p \, d\mathbf{x} = \int_E v_h p \, d\mathbf{x} \;\; \forall p \in \mathcal{P}(E). \tag{15}$$

The projection $\Pi^0 v_h$ can be exactly computed using DOFs and expressed in terms of DOFs. This is because, due to the definition of the local virtual element space (Equations (9) and (14)), the projection operator $\Pi^0$ is identical to the projection operator $\Pi^{\nabla}$ when the polynomial space is linear.[18]

The second projection operator ($\mathbf{\Pi}^0$) is defined by projecting $\nabla v_h$ onto a constant space $[\mathcal{P}_0(E)]^d$, such that

$$\int_E \mathbf{\Pi}^0 (\nabla v_h) \cdot \mathbf{p}_0 \, d\mathbf{x} = \int_E \nabla v_h \cdot \mathbf{p}_0 \, d\mathbf{x} \quad \forall \mathbf{p}_0 \in [\mathcal{P}_0(E)]^d, \tag{16}$$

where $\mathbf{p}_0$ is a vector of constant functions, ie,

$$\mathbf{p}_0 = \sum_{\alpha=1}^{n_{\mathbf{p}_0}} a_\alpha \mathbf{m}_\alpha. \tag{17}$$

Basis function vectors ($\mathbf{m}_\alpha$) are constant, which correspond to the unit base vectors, and the number of basis function vectors ($n_{\mathbf{p}_0}$) is the dimension number ($d$) in this study. The projection $\mathbf{\Pi}^0 \nabla v_h$ is exactly evaluated using DOFs and integration by parts. Integration by parts of Equation (16) results in

$$\int_E \mathbf{\Pi}^0 (\nabla v_h) \cdot \mathbf{p}_0 \, d\mathbf{x} = \int_{\partial E} v_h \mathbf{p}_0 \cdot \mathbf{n}_{\partial E} \, d\mathbf{x} - \int_E v_h \, \mathrm{div}(\mathbf{p}_0) \, d\mathbf{x}, \tag{18}$$

where $\mathbf{n}_{\partial E}$ is the unit normal vector to the element boundary. The first term on the right-hand side is simply the integration of a linear function on the element boundary using DOFs, whereas the second term is zero because $\mathbf{p}_0$ is constant. The integration on the element boundary is associated with the line and face integrals in 2D and 3D, respectively. One notes that the face integral in 3D can be evaluated by triangulation of faces.[7,18,20] Finally, we can show that, for any $v_h \in \mathcal{V}(E)$, the following relation holds between the two projections:

$$\nabla \left(\Pi^0 v_h\right) = \mathbf{\Pi}^0 (\nabla v_h). \tag{19}$$

## 2.4 | Mass matrix

For polygonal and polyhedral elements with $n$ nodes, the consistent mass matrix ($\mathbf{M}_E$) is constructed using the first $L^2$-projection ($\Pi^0$). It consists of the consistency and stability terms, given as

$$\mathbf{M}_E = \overline{\mathbf{M}}_{E,c} \otimes \mathbf{I}_d + \overline{\mathbf{M}}_{E,s} \otimes \mathbf{I}_d, \tag{20}$$

where $\mathbf{I}_d \in \mathbb{R}^{d\times d}$ is the identity matrix and $\otimes$ denotes the Kronecker product of two matrices ($\mathbf{A} \in \mathbb{R}^{i\times j}$, $\mathbf{B} \in \mathbb{R}^{k\times l}$), ie,

$$\mathbf{A}\otimes\mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1j}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{i1}\mathbf{B} & \cdots & a_{ij}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{ik\times jl}. \tag{21}$$

The first term on the right-hand side of Equation (20) is the consistency term of $\mathbf{M}_E$. It is expressed as[2]

$$[\overline{\mathbf{M}}_{E,c}]_{ij} = \int_E \rho_0(\Pi^0\phi_i)(\Pi^0\phi_j)d\mathbf{x} = \sum_{\alpha=1}^{n_p}\sum_{\beta=1}^{n_p}\left[(\mathbf{S}^0)^T\right]_{i\alpha}\int_E m_\alpha m_\beta d\mathbf{x}\,[\mathbf{S}^0]_{\beta j}, \tag{22}$$

where $\phi_i$ are implicit shape functions of $\mathcal{V}(E)$ and $n_p$ is the number of scaled monomial basis functions. The projection of implicit shape functions is expressed in terms of a set of scaled monomial functions, ie,

$$\Pi^0\phi_i = \sum_{\alpha=1}^{n_p} a_{\alpha i} m_\alpha, \tag{23}$$

where coefficients ($a_{\alpha i}$) correspond to the components of the matrix representation of the projection operator ($\mathbf{S}^0$). The coefficients can be exactly computed using integration by parts (eg, Equation (9)), as discussed previously. Additionally, in Equation (22), the integration of quadratic functions within an element is needed because linear scaled monomial functions are employed in this study. Finally, the stability term ($\overline{\mathbf{M}}_{E,s}$) is approximated as

$$\overline{\mathbf{M}}_{E,s} = \rho_0\,|E|\,(\mathbf{I}_n - \mathbf{P}^0)^T(\mathbf{I}_n - \mathbf{P}^0), \tag{24}$$

where $|E|$ denotes the volume of an element and $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix. Additionally, $\mathbf{P}^0$ is a matrix representation of the projection $\Pi^0$ expressed in terms of basis functions $\phi_j$,[2] ie,

$$\Pi^0\phi_i = \sum_{j=1}^{n} [\mathbf{P}^0]_{ij}\phi_j. \tag{25}$$

One should note that both $\mathbf{S}^0$ and $\mathbf{P}^0$ are matrix representations of the projection $\Pi^0$, but expressed in terms of different sets of basis functions.

## 2.5 | Stiffness matrix

The element stiffness matrix ($\mathbf{K}_E$) consists of a consistency term ($\mathbf{K}_{E,c}$) and a stability term ($\mathbf{K}_{E,s}$), ie, $\mathbf{K}_E = \mathbf{K}_{E,c} + \mathbf{K}_{E,s}$.[7] The second $L^2$-projection ($\boldsymbol{\Pi}^0$), which projects gradients of $v_h$, is utilized to construct the consistency term, whereas the first $L^2$-projection ($\Pi^0$) is used to approximate the stability term. Then, the consistency term of $\mathbf{K}_E$ is expressed as

$$\mathbf{K}_{E,c} = \int_E {\mathbf{B}_c}^T\mathbf{C}\mathbf{B}_c d\mathbf{x}, \tag{26}$$

where $\mathbf{C}$ is a matrix representation of the fourth-order elasticity tensor ($\mathbb{C}$). A matrix ($\mathbf{B}_c$) provides the relationship between strain and nodal displacements (DOFs), and thus, its components consist of the projection of the gradient of implicit shape functions. The projection of $\nabla\phi_i$ is expressed in terms of a set of constant basis function vectors, ie,

$$\boldsymbol{\Pi}^0(\nabla\phi_i) = \sum_{\alpha=1}^{n_{\mathbf{p}_0}} a_{\alpha i}\mathbf{m}_\alpha, \tag{27}$$

where coefficients ($a_{\alpha i}$) are exactly computed using Equation (18).

For the approximation of the stability term, ie, $\mathbf{K}_{E,s} = \overline{\mathbf{K}}_{E,s} \otimes \mathbf{I}_d$, a diagonal matrix–based stabilization scheme is utilized, given as

$$\overline{\mathbf{K}}_{E,s} = (\mathbf{I}_n - \mathbf{P}^0)^T \mathbf{\Lambda} (\mathbf{I}_n - \mathbf{P}^0), \tag{28}$$

where $\mathbf{\Lambda}$ is a diagonal matrix to scale the stiffness of $\overline{\mathbf{K}}_{E,s}$. The components of the diagonal matrix are defined as[11,19]

$$\Lambda_{ii} = \max\left( [\mathbf{K}_{E,\mathrm{c}}]_{ii}, \alpha_0 \frac{\mathrm{tr}(\mathbf{C})}{n_{\mathbf{C}}} \right), \tag{29}$$

where $n_{\mathbf{C}}$ is the size of the elasticity matrix, eg, $n_{\mathbf{C}} = 3$ for 2D and $n_{\mathbf{C}} = 6$ for 3D. A nondimensional parameter ($\alpha_0$) can be selected in between 0 and 1 and provides a lower bound of the diagonal matrix ($\mathbf{\Lambda}$). The lower bound is defined because the diagonal terms of the consistency part ($\mathbf{K}_{E,\mathrm{c}}$) can be very small values, especially for nonconvex elements. In this study, $\alpha_0$ is selected as 1/3 and 1/9 for 2D and 3D, respectively.
Alternatively, a scalar-based stabilization scheme can be used, and $\overline{\mathbf{K}}_{E,s}$ is expressed as

$$\overline{\mathbf{K}}_{E,s} = \alpha_s (\mathbf{I}_n - \mathbf{P}^0)^T (\mathbf{I}_n - \mathbf{P}^0), \tag{30}$$

where $\alpha_s$ is a stiffness scaling parameter. Previously, $\mathrm{tr}(\mathbf{C})/n_{\mathbf{C}}$ was used as the scaling parameter.[9,20] Additionally, the present study proposes to use the maximum diagonal component of the consistent part for the scaling parameter, ie, $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$, in the scalar-based stabilization scheme.

## 2.6 | B-bar method

The presented VEM is extended to analyze nearly incompressible solids using the B-bar method.[21] In the B-bar method, the strain-displacement matrix ($\mathbf{B}$) is decomposed into deviatoric and dilatational parts, ie, $\mathbf{B} = \mathbf{B}_{\mathrm{dev}} + \mathbf{B}_{\mathrm{dil}}$. To avoid a volumetric locking behavior, the components of $\mathbf{B}_{\mathrm{dil}}$ are projected using the concept of selective integration and mean dilatation formulations, given as

$$\overline{B}_i = \frac{1}{|E|} \int_E B_i d\mathbf{x}, \tag{31}$$

where the components of $\mathbf{B}_{\mathrm{dil}}$ are replaced by $\overline{B}_i$, which leads to $\overline{\mathbf{B}}_{\mathrm{dil}}$. One should note that $\overline{B}_i$ can be exactly computed using $\mathbf{\Pi}^0(\nabla \phi_i)$ in our setting. The modified strain-displacement matrix ($\overline{\mathbf{B}} = \mathbf{B}_{\mathrm{dev}} + \overline{\mathbf{B}}_{\mathrm{dil}}$) is then utilized for the construction of the element stiffness matrix, which gives

$$\mathbf{K}_E = \int_E \overline{\mathbf{B}}^T \mathbf{C} \overline{\mathbf{B}} d\mathbf{x} = \int_E \mathbf{B}_{\mathrm{dev}}{}^T \mathbf{C} \mathbf{B}_{\mathrm{dev}} d\mathbf{x} + |E|\, \overline{\mathbf{B}}_{\mathrm{dil}}{}^T \mathbf{C} \overline{\mathbf{B}}_{\mathrm{dev}} + |E|\, \overline{\mathbf{B}}_{\mathrm{dev}}{}^T \mathbf{C} \overline{\mathbf{B}}_{\mathrm{dil}} + |E|\, \overline{\mathbf{B}}_{\mathrm{dil}}{}^T \mathbf{C} \overline{\mathbf{B}}_{\mathrm{dil}}, \tag{32}$$

where $\overline{\mathbf{B}}_{\mathrm{dev}}$ denotes the volume average of $\overline{\mathbf{B}}_{\mathrm{dev}} = \frac{1}{|E|} \int_E \mathbf{B}_{\mathrm{dev}} d\mathbf{x}$. Notice that, using the projection of the shape function gradients, we can compute $\overline{\mathbf{B}}_{\mathrm{dev}}$ exactly as well. The first term on the right-hand side of Equation (32) is defined as the stiffness matrix of the deviatoric part ($\mathbf{K}_{E,\mathrm{dev}}$), which can be approximated using the "consistency" and "stability" terms as

$$\mathbf{K}_{E,\mathrm{dev}} = \int_E \mathbf{B}_{\mathrm{dev}}{}^T \mathbf{C} \mathbf{B}_{\mathrm{dev}} d\mathbf{x} \approx |\mathrm{E}|\, \overline{\mathbf{B}}_{\mathrm{dev}}{}^T \mathbf{C} \overline{\mathbf{B}}_{\mathrm{dev}} + \mathbf{K}_{E,s(\mathrm{dev})}, \tag{33}$$

where $\mathbf{K}_{E,s(\mathrm{dev})}$ is the stability term associated with $\mathbf{K}_{E,\mathrm{dev}}$. Substituting Equation (33) in Equation (32) results in

$$\mathbf{K}_E \approx |E| \left( \overline{\mathbf{B}}_{\mathrm{dev}} + \overline{\mathbf{B}}_{\mathrm{dil}} \right)^T \mathbf{C} \left( \overline{\mathbf{B}}_{\mathrm{dev}} + \overline{\mathbf{B}}_{\mathrm{dil}} \right) + \mathbf{K}_{E,s(\mathrm{dev})}. \tag{34}$$

Note that, in the first term on the right-hand side of (34), the summation of $\overline{\mathbf{B}}_{\mathrm{dev}}$ and $\overline{\mathbf{B}}_{\mathrm{dil}}$ yields the volume average of the strain-displacement matrix ($\mathbf{B}_c$) in Equation (26), ie, $\mathbf{B}_c = \frac{1}{|E|} \int_E \mathbf{B} d\mathbf{x} = \overline{\mathbf{B}}_{\mathrm{dev}} + \overline{\mathbf{B}}_{\mathrm{dil}}$. Thus, this term recovers the consistency term in the standard VEM approximation. *The second term on the right-hand side of Equation* (34) *suggests that, to establish*

*a B-bar formulation in the VEM setting, one simply needs to modify the stability term to stabilize only the deviatoric part of the stiffness matrix ($\mathbf{K}_{E,\mathrm{dev}}$), which requires no additional computational effort with respect to the standard VEM formulation*. In this work, the stability term $\mathbf{K}_{E,s(\mathrm{dev})}$ is simply evaluated using the scalar-based stabilization scheme (Equation (30)) with the stiffness scaling constant ($\alpha_s$) of the shear modulus ($\mu$) because $\mathbf{K}_{E,\mathrm{dev}}$ corresponds to the deviatoric part.

## 3 | TIME INTEGRATION

The equation of motion is solved using the central difference method, which is a special case of the Newmark-$\beta$ method.[22] To achieve an efficient and stable computation in explicit time integration, a mass lumping scheme and critical time-step evaluation are essential, and thus, these are addressed in the following subsections.

### 3.1 | Mass lumping techniques

The consistent mass matrix is diagonalized for an efficient computation in explicit time integration. In this study, a diagonal mass matrix ($\mathbf{M}_E^D$) is obtained by scaling the diagonal terms of the consistent mass matrix,[23] named diagonal scaling, which is expressed as

$$\left[\mathbf{M}_E^D\right]_{ii} = [\mathbf{M}_E]_{ii} \frac{\rho_0 |E|}{\mathrm{tr}\,(\mathbf{M}_E)} \ \text{(no sum in } i\text{)}, \tag{35}$$

One should note that diagonal scaling always provides positive nodal masses because the diagonal terms of the consistent mass matrix are positive. For a computation of the lumped mass matrix in Equation (35), the numerical integration of second-order polynomials is needed.

Alternatively, the consistent mass matrix can be diagonalized using the row sum technique,[24,25] ie,

$$\left[\mathbf{M}_E^D\right]_{ii} = \sum_j [\mathbf{M}_E]_{ij}. \tag{36}$$

The diagonal mass matrix in Equation (36) can be exactly constructed without adopting numerical integrations. To show that, we will prove that the following two relations hold:

$$\sum_j [\overline{\mathbf{M}}_{E,c}]_{ij} = \int_E \rho_0 \Pi^0 \phi_i \mathrm{d}\mathbf{x} \tag{37}$$

and

$$\sum_j [\overline{\mathbf{M}}_{E,s}]_{ij} = \mathbf{0} \quad (\text{or, equivalently, } \overline{\mathbf{M}}_{E,s}\mathbf{1} = \mathbf{0}), \tag{38}$$

where $\mathbf{1} \in \mathbb{R}^n$ and $\mathbf{0} \in \mathbb{R}^n$ are vectors containing all 1 and 0 components, respectively. For the first relation (Equation (37)), taking the consistency matrix defined in Equation (22) and applying the row sum technique lead to

$$\sum_j [\overline{\mathbf{M}}_{E,c}]_{ij} = \sum_j \left[\int_E \rho_0 (\Pi^0 \phi_i)(\Pi^0 \phi_j) d\mathbf{x}\right] = \int_E \rho_0 (\Pi^0 \phi_i) \Pi^0 (\Sigma_j \phi_j) d\mathbf{x} = \int_E \rho_0 \Pi^0 \phi_i \mathrm{d}\mathbf{x}, \tag{39}$$

because of the partition-of-unity property of the basis function $\phi_i$, namely, $\sum_j \phi_j = 1$. The second relation (Equation (38)) is deduced from the definitions of $\overline{\mathbf{M}}_{E,s}$ and $\mathbf{P}^0$ in Equations (24) and (25), respectively. The definition of matrix $\mathbf{P}^0$ results in

$$\sum_{k=1}^{n} \Pi^0 \phi_i (\mathbf{x}_k) = \sum_{k=1}^{n} \sum_{j=1}^{n} [\mathbf{P}^0]_{ij} \phi_j (\mathbf{x}_k) = \sum_{k=1}^{n} \sum_{j=1}^{n} [\mathbf{P}^0]_{ij} \delta_{jk} = \sum_{k=1}^{n} [\mathbf{P}^0]_{ik}, \tag{40}$$

where $\delta_{ik}$ is the Kronecker delta symbol and $\mathbf{x}_k$ is the position vector of the $k$th node in an element. Because of the Kronecker delta property of the basis function $\phi_i$, ie, $\sum_{k=1}^{n} \Pi^0 \phi_i(\mathbf{x}_k) = \sum_{k=1}^{n} \delta_{ik} = 1$, the following property of $\mathbf{P}^0$ is obtained:

$$\mathbf{P}^0 \mathbf{1} = \mathbf{1}. \tag{41}$$

Using Equations (24) and (41), one obtains the following relations:

$$\overline{\mathbf{M}}_{E,s}\mathbf{1} = \rho_0 |E| \left(\mathbf{I}_n - \mathbf{P}^0\right)^T \left(\mathbf{I}_n - \mathbf{P}^0\right) \mathbf{1} = \rho_0 |E| \left(\mathbf{I}_n - \mathbf{P}^0\right)^T \left[\mathbf{I}_n \mathbf{1} - \mathbf{P}^0 \mathbf{1}\right] = \mathbf{0}. \tag{42}$$

Then, in order to compute the components in the diagonal mass matrix with the row sum technique, we only need to compute $\int_E \rho_0 \Pi^0 \phi_i d\mathbf{x}$ for all the vertices, which are all integrals of linear functions over the element $E$. As a consequence, no numerical integration is needed for the row sum technique because the integration of linear functions in general polygons (or polyhedra) can be computed using the centroid and area (or volume). However, the row sum technique does not guarantee positive nodal masses, even with the linear polynomial space in the VEM.

## 3.2 | Critical time step and effective element length

A critical time step is essential to achieve a stable solution in explicit time integration. Then, the critical time step ($\Delta t_{\text{cr}}$) for linear elastic problems is given as[25-27]

$$\Delta t_{\text{cr}} = \frac{2}{\omega_{\max}^G}, \tag{43}$$

where $\omega_{\max}^G$ is the maximum eigenfrequency of a global system matrix. The maximum eigenfrequency is obtained by solving an eigenvalue problem of a system, given as

$$\det(-\omega^2 \mathbf{M} + \mathbf{K}) = 0, \tag{44}$$

where $\mathbf{K}$ and $\mathbf{M}$ are the global stiffness and mass matrices, respectively, and $\omega$ is the eigenvalue of a system.

Because the calculation of the maximum eigenvalue for a large system is computationally expensive, the critical time step can be approximated using the maximum eigenvalue of a local system.[28,29] Because the maximum eigenvalue of a local system is greater than the maximum eigenvalue of a global system, an approximated critical time step ($\Delta \tilde{t}_{\text{cr}}$) is given as

$$\Delta \tilde{t}_{\text{cr}} = \frac{2}{\max\left(\omega_{\max}^E\right)} < \Delta t_{\text{cr}}, \tag{45}$$

where $\omega_{\max}^E$ is the maximum eigenvalue of an element. The maximum eigenvalue of an element is evaluated by solving the following equation:

$$\det(-\omega^2 \mathbf{M}_E + \mathbf{K}_E) = 0. \tag{46}$$

For every element in a mesh, $\omega_{\max}^E$ is computed, and then, the maximum of $\omega_{\max}^E$ is used to approximate $\Delta t_{\text{cr}}$.

Alternatively, the critical time step is approximated in relation with an element size and wave speed of a material.[30-32] A travel distance of wave for a given time shall not be greater than an element size, called the Courant condition, and thus, an approximated critical time step is expressed as

$$\Delta \tilde{t}_{\text{cr}} = \frac{\min(\ell_E)}{C_d} < \Delta t_{\text{cr}}, \tag{47}$$

where $\ell_E$ is an effective element length for an element and $C_d$ is the dilatational wave speed. For general convex and nonconvex elements in the VEM, an effective length of an element is defined as

$$\ell_E = \max\left(\ell_{e,\min}^E, \ell_{cn,\min}^E\right), \tag{48}$$

where $\ell^E_{e,\min}$ and $\ell^E_{cn,\min}$ are the minimum edge length and the minimum length from the centroid to nodes, respectively, of an element ($E$). For the standard FEM, as a rule of thumb, the minimum edge length is generally utilized for an effective element length. Then, in the following section, the effect of small edges in random polygons on the critical time step is demonstrated.

## 4 | CRITICAL TIME-STEP ESTIMATION

A critical time step is approximated using the maximum eigenvalue of a system (Equations (43)-(46)) and the effective element length (Equations (47) and (48)) for convex and nonconvex meshes. Four types of discretization are employed within a $1 \times 1$ rectangular domain, ie, centroidal Voronoi tessellation (CVT), random Voronoi tessellation (RVT), pegasus, and "VEM" meshes, as shown in Figure 1. The CVT and RVT meshes are generated using PolyMesher[33] with the number of elements of 1600. The $6 \times 6$ periodic patterns in the pegasus mesh are generated by using Escher's tessellation[34,35] and cropping elements to fit with the domain, as shown in Figure 1C. For the "VEM" mesh, each patch consists of seven elements, ie, four background elements and three letter elements. Material properties are arbitrarily selected. The elastic modulus and Poisson's ratio are 10 and 0.3, respectively, with the density of 2, which leads to the dilatational wave

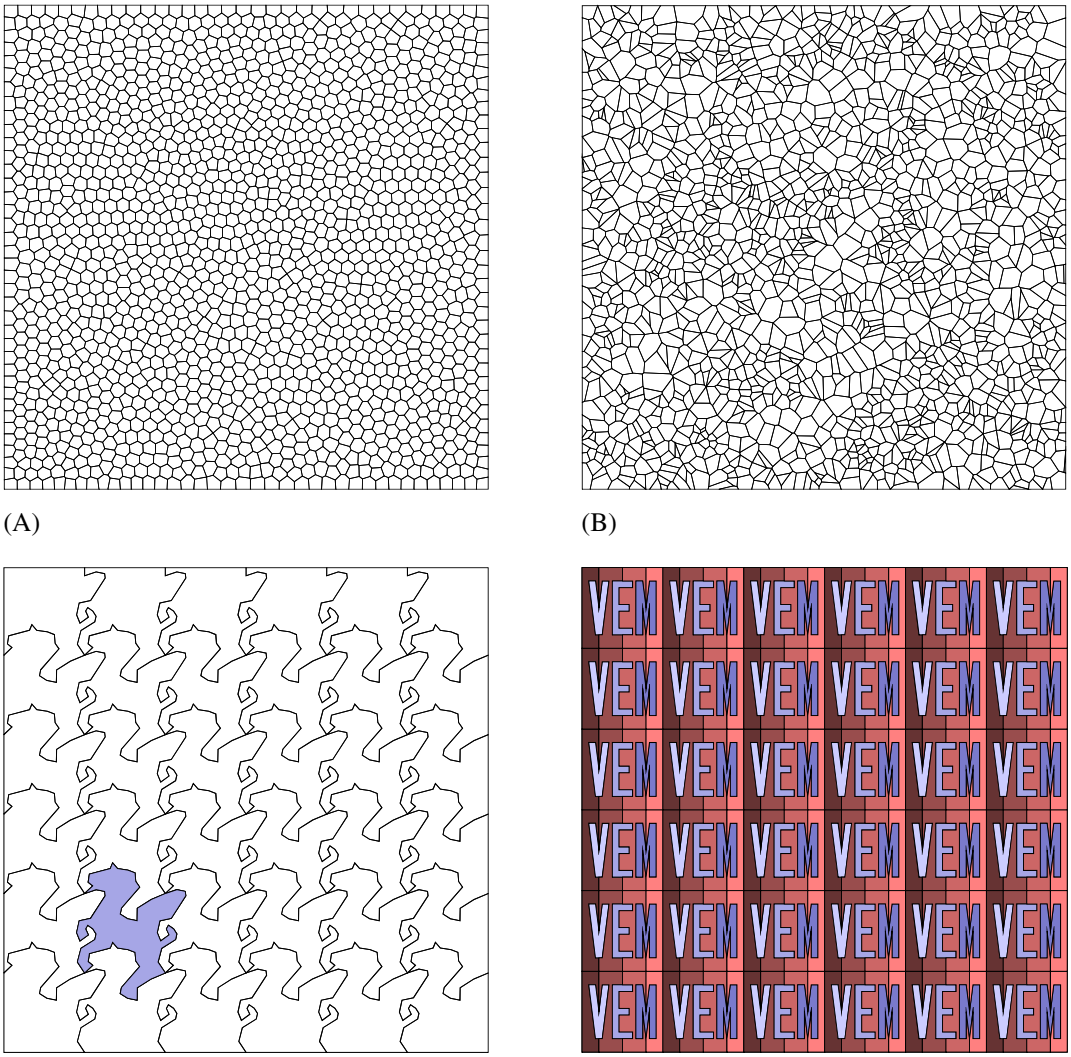



**FIGURE 1** Virtual element meshes for critical time-step approximation. A, Centroidal Voronoi tessellation mesh; B, Random Voronoi tessellation mesh; C, Pegasus mesh; D, "VEM" mesh—notice that there are seven colors (or shades in black and white) that correspond to each element (seven elements per "VEM" patch)

speed of 2.594. Additionally, the stability term is scaled using the diagonal terms of the consistency matrix with $\alpha_0 = 1/3$ (Equations (28) and (29)), whereas the mass matrix is diagonalized using the diagonal terms of the consistent mass matrix (Equation (35)).

## 4.1 | Selection of effective element length

The critical time step is evaluated using the following three approaches, ie, the maximum eigenvalue of a global system, the maximum eigenvalue of a local system, and the effective element length. For each element, $\ell^E_{e,\min}$ and $\ell^E_{cn,\min}$ are calculated, and the histogram for each length is plotted in Figures 2A and 2B for the CVT and RVT meshes. Additionally, a representative element length is also calculated using the maximum eigenvalue of a global system and the maximum eigenvalue of a local system, which are given as $\ell^G_\omega = 2C_d/\omega^G_{\max}$ and $\ell^L_\omega = 2C_d/\max(\omega^E_{\max})$, respectively. Figure 2 demonstrates that $\ell^L_\omega$ and $\min(\ell_E)$ provide a good approximation of $\ell^G_\omega$ for the CVT and RVT meshes.

For the four meshes, the approximated critical time steps are summarized in Table 1. Additionally, the critical time step is also calculated using $\min(\ell^E_{e,\min})$ and $\min(\ell^E_{cn,\min})$ for the comparison. Similarly, the maximum eigenvalue of a local system ($\omega^E_{\max}$) and the effective element length ($\ell_E$) provide a good approximation of the critical time step obtained from the maximum eigenvalue of a global system for all the mesh types. The critical time step obtained from $\min(\ell_E)$ is generally similar to the step calculated from $\min(\ell^E_{cn,\min})$. However, one notes that $\min(\ell^E_{cn,\min})$ leads to the zero time step for the "VEM" mesh because the centroid of an element in the "VEM" mesh corresponds to the nodal location. Furthermore, $\Delta t_{\text{cr}}$ obtained from the minimum edge length is one order of magnitude lower than $\Delta t_{\text{cr}}$ resulted from the maximum eigenvalue, especially for CVT and RVT meshes, as shown in Table 1.

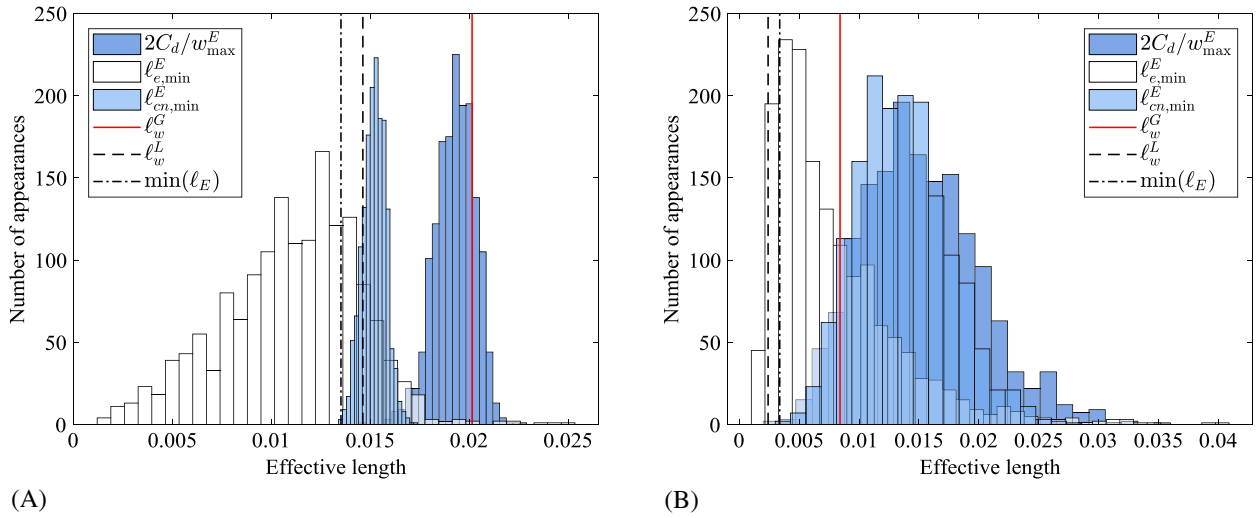



**FIGURE 2** Histogram of the representative element length from $2C_d/w^E_{\max}$, the minimum edge length of an element ($\ell^E_{e,\min}$), and the minimum centroid-to-node length of an element ($\ell^E_{e,\min}$) for (A) the centroidal Voronoi tessellation mesh and (B) the random Voronoi tessellation mesh

**TABLE 1** Estimated critical time step for the centroidal Voronoi tessellation (CVT), random Voronoi tessellation (RVT), pegasus, and "VEM" meshes

| | CVT mesh | RVT mesh | Pegasus mesh | "VEM" mesh |
|---|---|---|---|---|
| $2/\omega^G_{\max}$ | $8.227 \times 10^{-3}$ | $3.433 \times 10^{-3}$ | $10.938 \times 10^{-3}$ | $6.032 \times 10^{-3}$ |
| $2/\max\left(\omega^E_{\max}\right)$ | $5.973 \times 10^{-3}$ | $0.981 \times 10^{-3}$ | $10.254 \times 10^{-3}$ | $5.482 \times 10^{-3}$ |
| $\min(\ell_E)/C_d$ | $5.212 \times 10^{-3}$ | $1.300 \times 10^{-3}$ | $3.973 \times 10^{-3}$ | $2.677 \times 10^{-3}$ |
| $\min\left(\ell^E_{cn,\min}\right)/C_d$ | $5.187 \times 10^{-3}$ | $1.300 \times 10^{-3}$ | $3.973 \times 10^{-3}$ | 0.0000 |
| $\min\left(\ell^E_{e,\min}\right)/C_d$ | $0.673 \times 10^{-3}$ | $0.518 \times 10^{-3}$ | $3.328 \times 10^{-3}$ | $2.677 \times 10^{-3}$ |

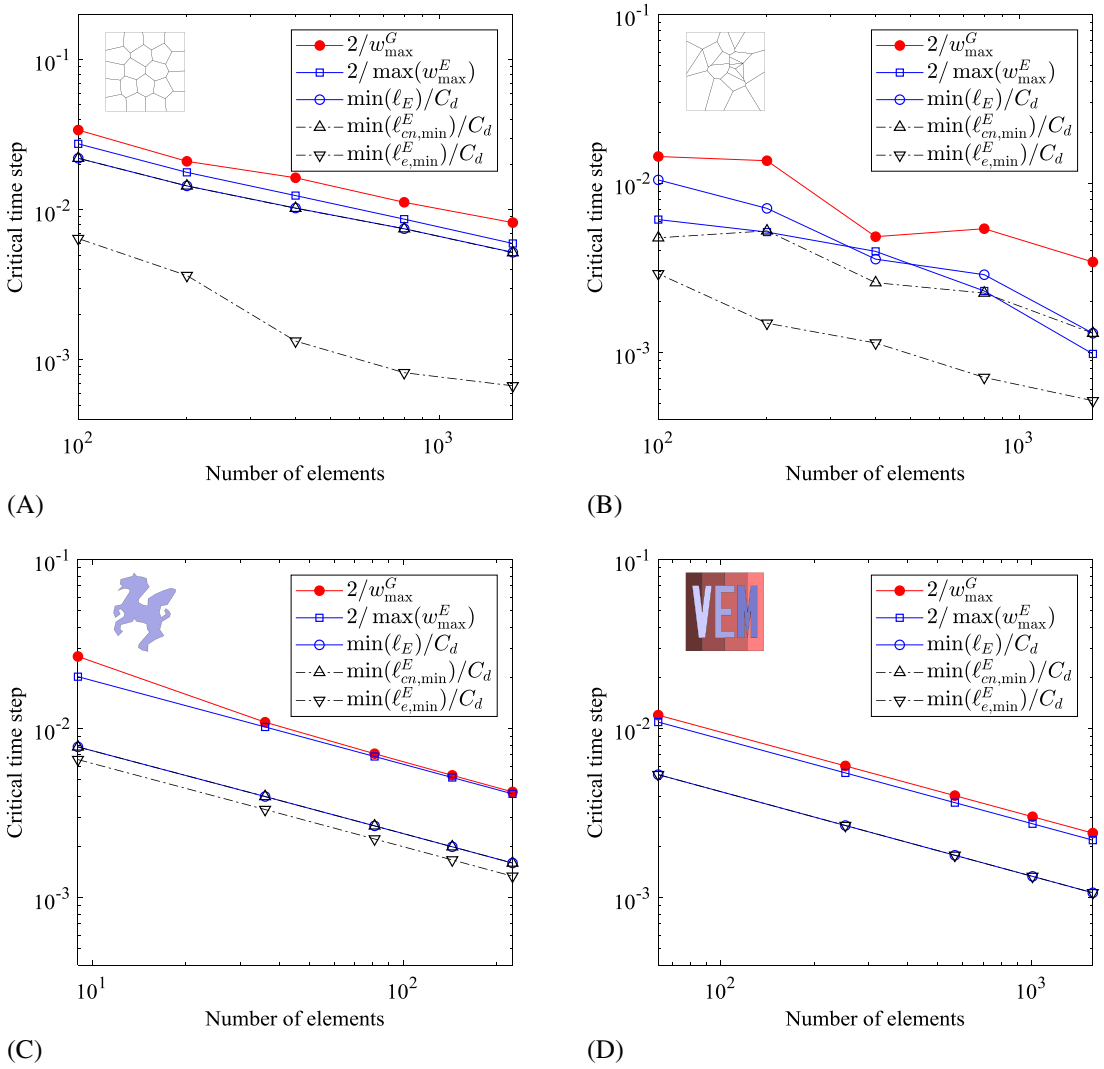



**FIGURE 3** Estimation of the critical time step according to the number of elements for (A) the centroidal Voronoi tessellation mesh, (B) the random Voronoi tessellation mesh, (C) the pegasus mesh, and (D) the "VEM" mesh [Colour figure can be viewed at wileyonlinelibrary.com]

## 4.2 | Effect of element size

The change in the critical time step is investigated according to the element size. The critical time step is estimated using the five approaches, ie, $\omega^G_{\max}$, $\omega^E_{\max}$, $\min(\ell_E)$, $\min(\ell^E_{cn,\min})$, and $\min(\ell^E_{e,\min})$. For each mesh type, five meshes are generated, ie, 25 meshes in total. The numbers of elements for the CVT and RVT meshes are 100, 200, 400, 800, and 1600. For the pegasus and "VEM" meshes, $3\times3$, $6\times6$, $9\times9$, $12\times12$, and $15\times15$ patterns are generated. One notes that a larger number of elements corresponds to a smaller element size. Figure 3 demonstrates that the critical time step decreases while the number of elements increases, as expected. In general, $\omega^E_{\max}$ and $\ell_E$ provide a good approximation of the critical time step according to decreasing the element size. One notes that $\min(\ell^E_{e,\min})$ is similar to $\min(\ell_E)$ for the nonconvex meshes (pegasus and "VEM" meshes) but one order of magnitude lower than $\min(\ell_E)$ for the convex meshes (ie, CVT and RVT). Therefore, in this study, relatively short edges in the CVT and RVT meshes do not significantly impact the critical time step in an elastodynamic VEM.

## 4.3 | Comparison between VEM and FEM

The critical time steps obtained from the VEM are compared with the time step resulted from the FEM. For the construction of shape functions for convex polygons in the FEM, mean value coordinates are employed.[36] The critical time step is estimated using the maximum eigenvalue of a global system ($\omega^G_{\max}$) for the CVT and RVT meshes. The critical time of the

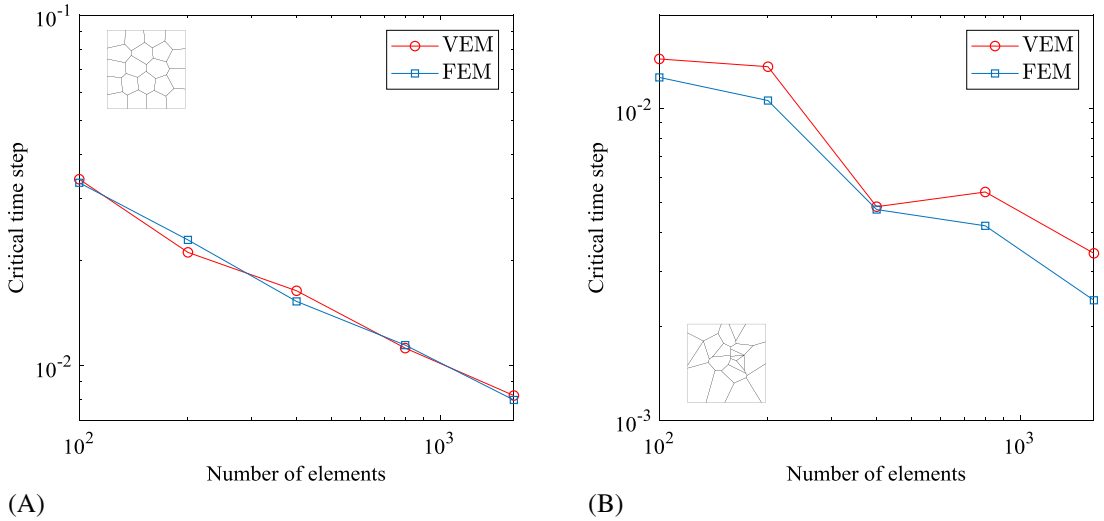



**FIGURE 4** Comparison of the critical time step between the virtual element method (VEM) and the finite element method (FEM). A, Centroidal Voronoi tessellation mesh; B, Random Voronoi tessellation mesh [Colour figure can be viewed at wileyonlinelibrary.com]

FEM is similar to that of the VEM for the CVT meshes (Figure 4A), whereas the critical time step of the VEM is slightly greater than that of the FEM for the RVT meshes (Figure 4B). Thus, the VEM can provide a smaller computational cost than the FEM for elastodynamic problems, especially when mesh quality is relatively low.

## 4.4 | Effect of stability term

The effect of the stabilization scheme on the critical time step is investigated. The four cases of the stabilization method are used, ie, the matrix-based stabilization with $\alpha_0 = 1/3$, the matrix-based stabilization with $\alpha_0 = 0$, the scalar-based stabilization with $\alpha_s = \mathrm{tr}(\mathbf{C})/n_{\mathbf{C}}$, and the scalar-based stabilization with $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$. Then, the critical time steps are evaluated using the maximum eigenvalue of a global system for the RVT and pegasus meshes, as shown in Figure 5. The computational results illustrate that the change in the critical time is not sensitive according to the choice of the stabilization scheme for convex meshes, eg, the RVT mesh. For nonconvex meshes, eg, the pegasus mesh, the critical time step changes with respect to the stabilization scheme. Although the diagonal matrix–based stabilization scheme with $\alpha_0 = 0$ provides the largest critical time step, the accuracy of the computational results can be lower than that in the other cases, as discussed in Section 5.1.

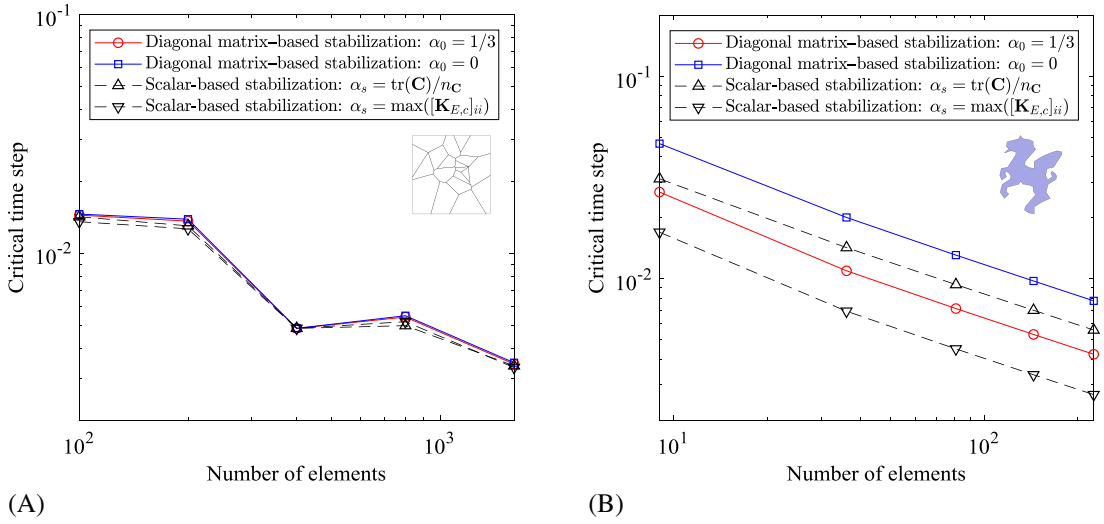



**FIGURE 5** Effect of the stability term on the critical time step. A, Random Voronoi tessellation mesh; B, Pegasus mesh [Colour figure can be viewed at wileyonlinelibrary.com]

## 4.5 | Lumped mass matrix

The diagonal scaling method is compared with the row sum technique for the estimation of the critical time step. The maximum eigenvalues of global and local systems are used to estimate the critical time step for the RVT and pegasus meshes. The effect of the mass lumping scheme on the critical time step is not significant for the convex mesh (eg, Figure 6A). The slight differences in the critical time step between diagonal scaling and the row sum technique are observed for the nonconvex mesh (eg, Figure 6B), although the evaluated critical time steps are within the same order of magnitude. Such difference is resulted from different nodal mass distributions with respect to the mass lumping technique. For the RVT mesh, the nodal mass distribution with the row sum technique is similar to that with diagonal scaling, as shown in Figure 7A, and thus, the estimated critical time steps are similar to each other. However, for the pegasus mesh, diagonal scaling provides a more uniform nodal mass distribution than the row sum technique, and the minimum nodal mass with diagonal scaling is greater than that with the row sum technique (see Figure 7B). Therefore, diagonal scaling provides a larger critical time step than the row sum technique, whereas the accuracy of the computational result with diagonal scaling is equivalent to or better than that with the row sum technique, as discussed in Section 5.2.

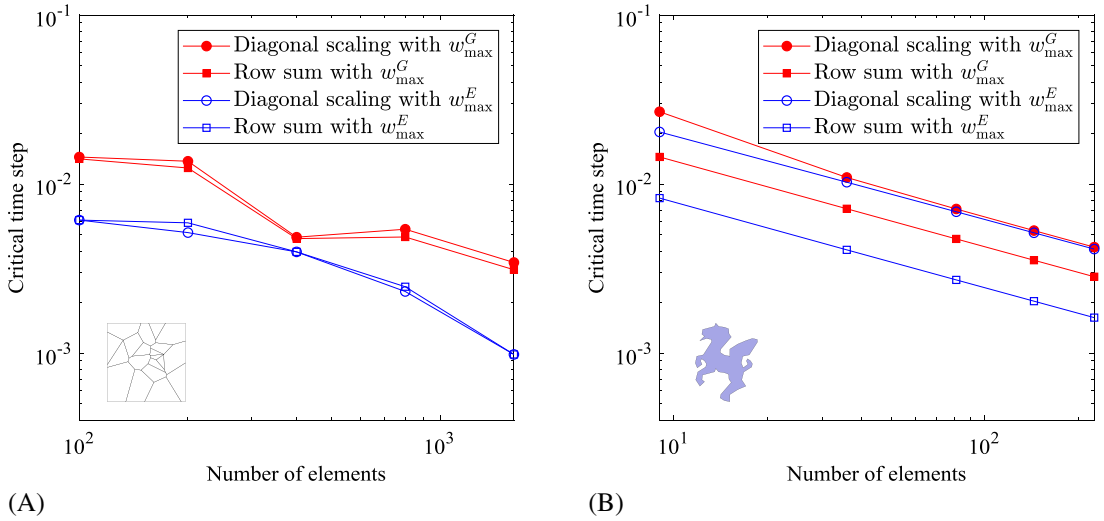



**FIGURE 6** Critical time-step comparison between diagonal scaling and the row sum technique. A, Random Voronoi tessellation mesh; B, Pegasus mesh [Colour figure can be viewed at wileyonlinelibrary.com]

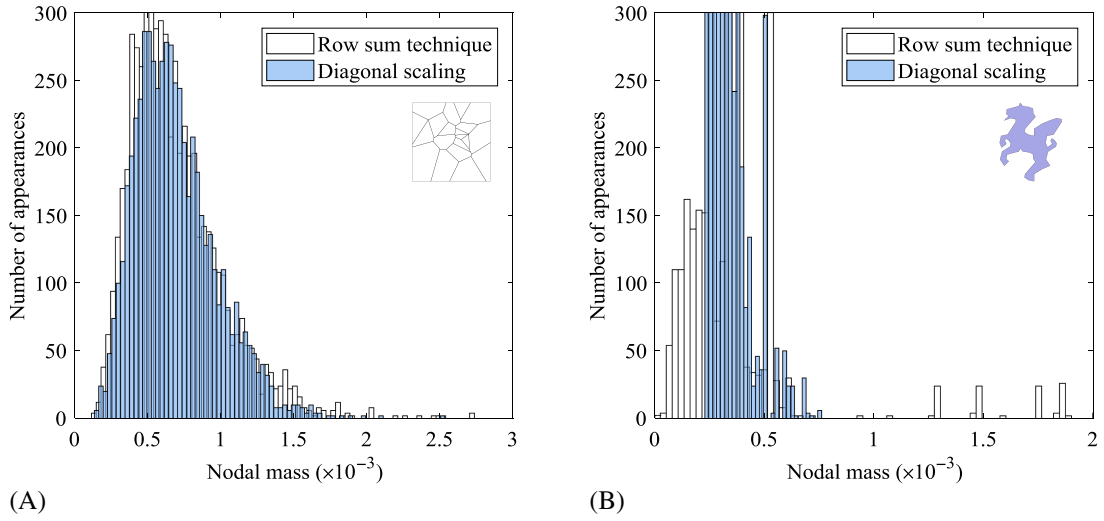



**FIGURE 7** Nodal mass comparison between diagonal scaling and the row sum technique. A, Random Voronoi tessellation mesh with 1600 elements; B, Pegasus mesh with the $15 \times 15$ pattern

## 4.6 | Element distortion

To investigate the effect of element distortion on the critical time step in the VEM, a mesh is stretched either from 1 to 5 (ie, elongation) or from 1 to 0.2 (ie, compression). For the elongation case, the size of the initial domain is $1 \times 0.2$, and the CVT mesh with 800 elements and the pegasus mesh with the $15 \times 3$ pattern are generated, as shown in Figures 8A and 8B, respectively. The meshes with the stretch ($\lambda$) of 5 are illustrated in Figures 8C and 8D. While the stretch increases as 2, 3, 4, and 5, the critical time step is evaluated using $\omega^G_{\max}$, $\omega^E_{\max}$, $\min(\ell_E)$, $\min(\ell^E_{cn,\min})$, and $\min(\ell^E_{e,\min})$. Figure 9 demonstrates

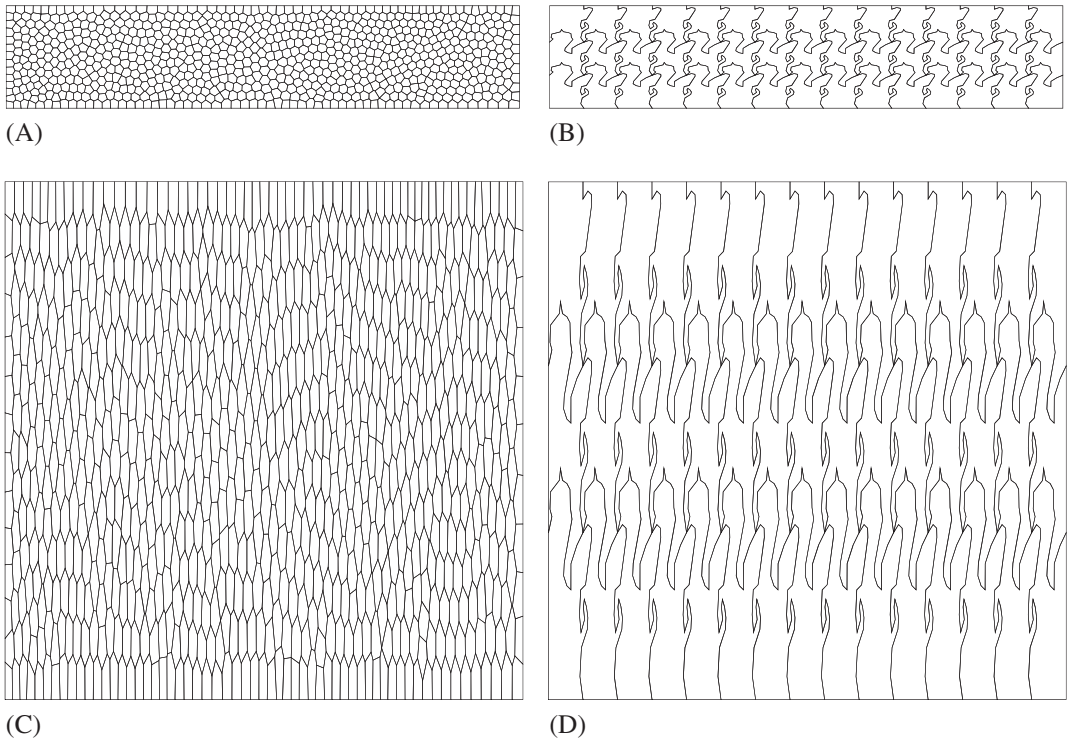



**FIGURE 8** Element distortion under elongation. A, Original centroidal Voronoi tessellation (CVT) mesh with 800 elements; B, Original pegasus mesh with the $15 \times 3$ pattern; C, Distorted CVT mesh with $\lambda = 5$; D, Distorted pegasus mesh with $\lambda = 5$

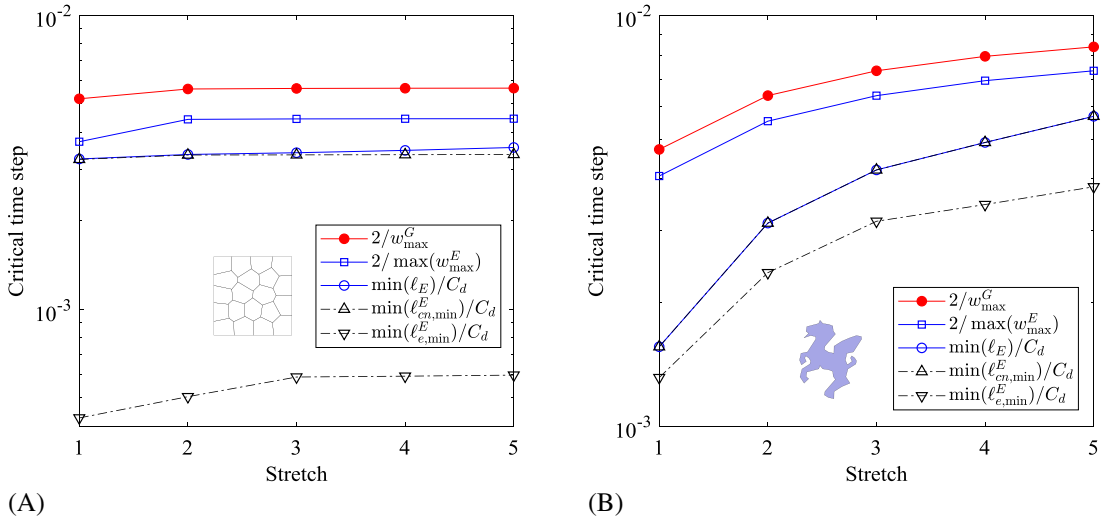



**FIGURE 9** Change in the critical time step according to stretch. A, Centroidal Voronoi tessellation mesh; B, Pegasus mesh [Colour figure can be viewed at wileyonlinelibrary.com]

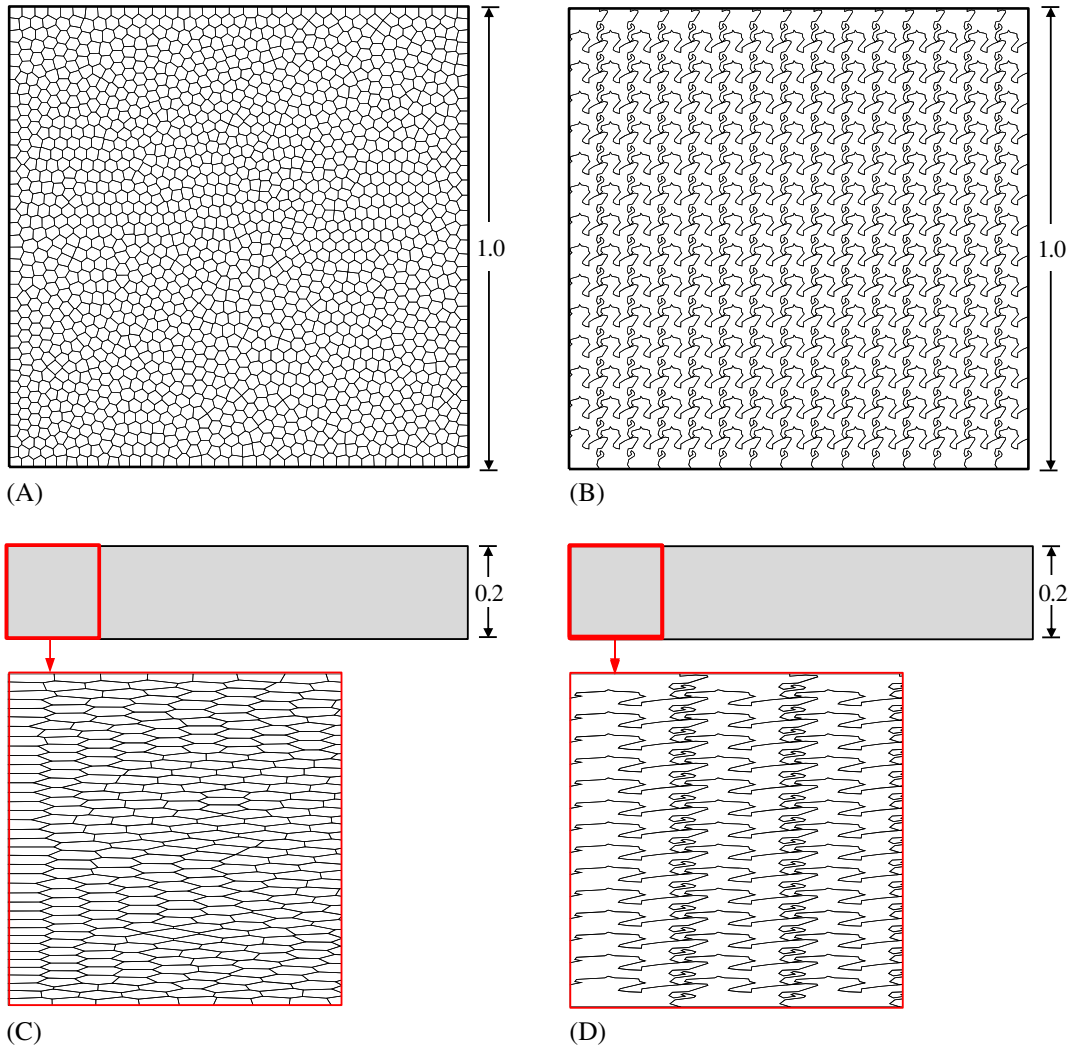



**FIGURE 10** Element distortion under compression. A, Original centroidal Voronoi tessellation (CVT) mesh with 1600 elements in a square domain; B, Original pegasus mesh with the $15 \times 15$ pattern in a square domain; C, Zoomed-in view of the distorted CVT mesh with $\lambda = 0.2$; D, Zoomed-in view of the distorted pegasus mesh with $\lambda = 0.2$ [Colour figure can be viewed at wileyonlinelibrary.com]

that the critical time step tends to increase while the stretch increases because the element size increases with the increase in stretch.

For the compression case, the $1 \times 1$ square domain is discretized into the CVT mesh with 1600 elements and the pegasus mesh with the $15 \times 15$ pattern, as shown in Figures 10A and 10B, respectively. Then, the stretches of 0.8, 0.6, 0.4, and 0.2 are introduced in the domain, and the example of the meshes with $\lambda = 0.2$ is shown in Figures 10C and D. The evaluated critical time steps with $\omega^G_{\max}$, $\omega^E_{\max}$, $\min(\ell_E)$, $\min(\ell^E_{cn,\min})$, and $\min(\ell^E_{e,\min})$ are plotted in Figure 11. While the stretch decreases from 1 to 0.2, the element size and the element quality decrease, and thus, the critical time step decreases, as expected.

## 5 | ELASTODYNAMIC EXAMPLES

To verify the elastodynamic VEM, four computational examples are employed, ie, swinging plate, 2D cantilever, 3D cantilever, and Cook's beam, with compressible and nearly incompressible materials. In this computation, the time step is determined using the maximum eigenvalue of a local system, and consistent units are employed.

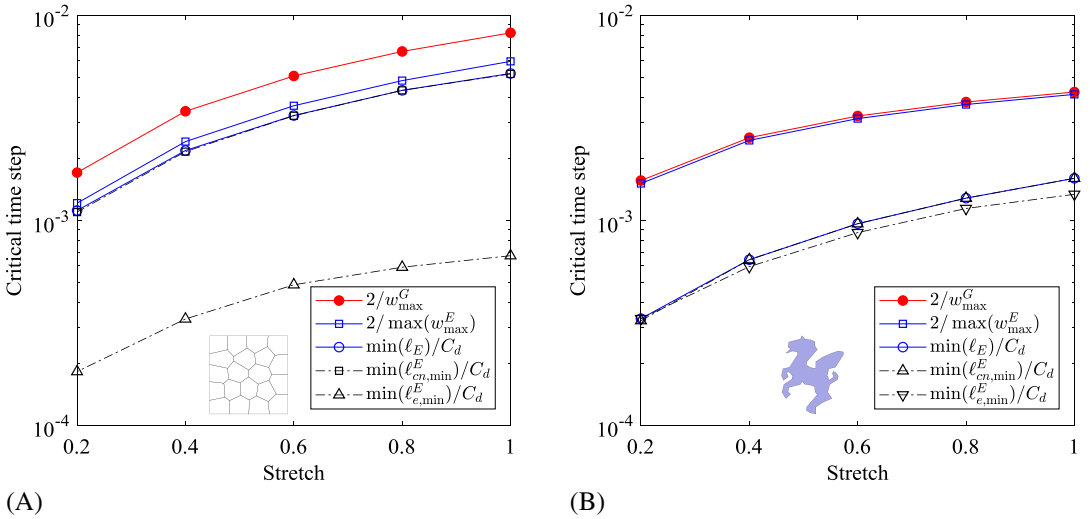



**FIGURE 11** Change in the critical time step according to decreasing the stretch. A, Centroidal Voronoi tessellation mesh; B, Pegasus mesh [Colour figure can be viewed at wileyonlinelibrary.com]

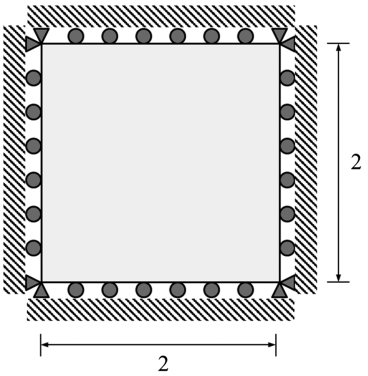



**FIGURE 12** Geometry of a swinging plate example

## 5.1 | Swinging plate

A swinging plate example[37-39] is utilized to verify the rate of convergence for the elastodynamic VEM. In a square domain of $2 \times 2$, the displacement along the four edges is fixed along the edge-normal direction, as shown in Figure 12. The domain is discretized into four types of meshes, ie, CVT, RVT, pegasus, and "VEM" meshes (see Figure 1). Then, the analytical solution is given as

$$\mathbf{u}(\mathbf{x}, t) = u_0 \sin(\omega t) \begin{bmatrix} -\sin\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right) \\ \cos\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_2\right) \end{bmatrix} \tag{49}$$

when the initial displacement and velocity boundary conditions are imposed based on Equation (49). The displacement amplitude of $u_0$ is selected as 0.01, and $\omega$ is the angular frequency, ie, $\omega = \frac{\pi}{2}\sqrt{2\mu/\rho_0}$. The density ($\rho_0$) is 1100, and the elastic modulus is $1.7 \times 10^7$ with the plane strain condition. Poisson's ratio ($\nu$) is selected as 0.45 and 0.49994, which correspond to compressible and nearly incompressible materials, respectively. For the error estimation of both convex and nonconvex elements, the $L^2$-type displacement error measure[9] is defined as

$$\epsilon_{0,\mathbf{u}}(t) = \left[\sum_E \frac{|E|}{n} \sum_{\mathbf{x}_v \in E} (\mathbf{u}(\mathbf{x}_v, t) - \mathbf{u}_h(\mathbf{x}_v, t)) \cdot (\mathbf{u}(\mathbf{x}_v, t) - \mathbf{u}_h(\mathbf{x}_v, t))\right]^{\frac{1}{2}}, \tag{50}$$

where $\mathbf{u}(\mathbf{x}_v, t)$ and $\mathbf{u}_h(\mathbf{x}_v, t)$ are the exact and approximated displacements on vertex $\mathbf{x}_v$ at time $t$, respectively. In this study, the error measure is estimated at $t = 0.001$.

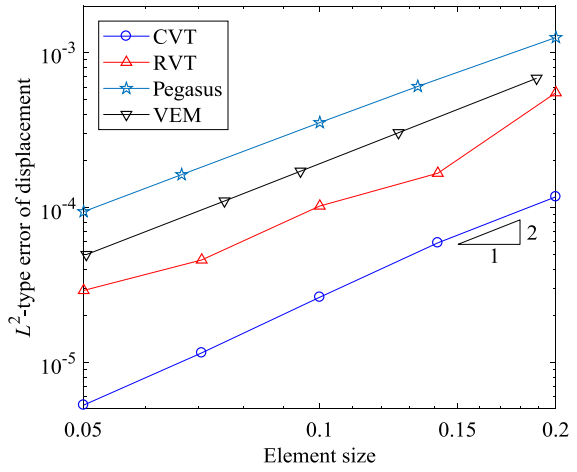



**FIGURE 13** Rate of convergence according to the mesh type for the compressible material ($\nu = 0.45$). CVT, centroidal Voronoi tessellation; RVT, random Voronoi tessellation [Colour figure can be viewed at wileyonlinelibrary.com]

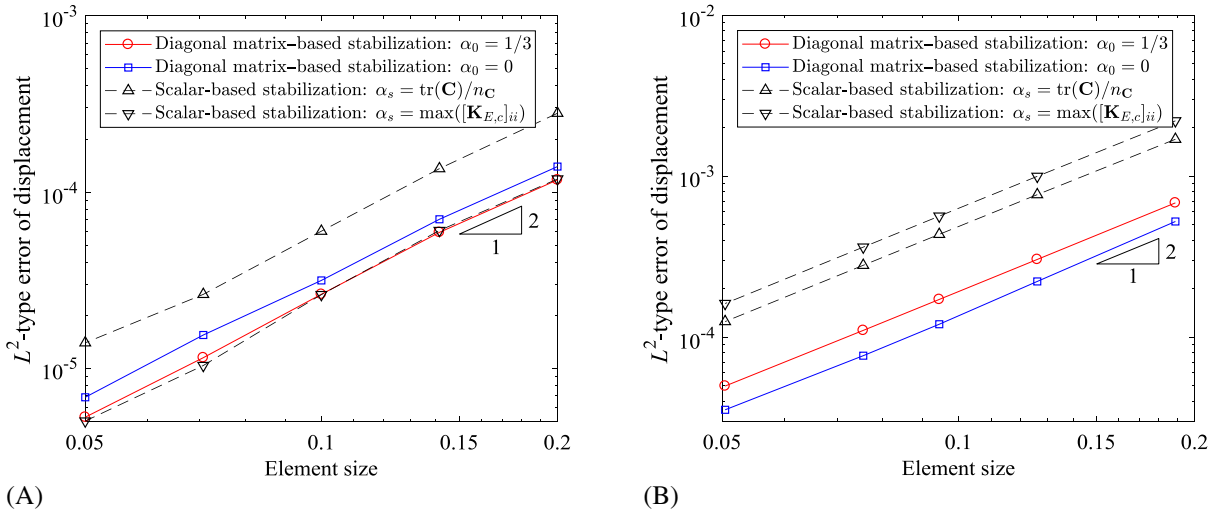



**FIGURE 14** Rate of convergence with respect to the stability terms for a compressible material. A, Centroidal Voronoi tessellation mesh; B, "VEM" mesh [Colour figure can be viewed at wileyonlinelibrary.com]

For the compressible material ($\nu = 0.45$), the $L^2$-type displacement error is evaluated under mesh refinement, and the optimal rate of convergence of 2 is achieved for all the mesh types, as shown in Figure 13. The diagonal matrix–based stabilization scheme is utilized with $\alpha_0 = 1/3$. Additionally, the effect of the stabilization scheme is investigated using the CVT and "VEM" meshes. The four cases of the stabilization method are utilized, ie, (a) diagonal matrix–based stabilization with $\alpha_0 = 1/3$, (b) diagonal matrix–based stabilization with $\alpha_0 = 0$, (c) scalar-based stabilization with $\alpha_s = \mathrm{tr}(\mathbf{C})/n_{\mathbf{C}}$, and (d) scalar-based stabilization with $\alpha_s = \max([\mathbf{K}_{E,c}]_{ii})$. Figure 14 demonstrates that the VEM provides the optimal rate of convergence regardless of the stabilization schemes, whereas the accuracy of the solution depends on the stabilization scheme in this study. Additionally, to investigate the influence of the stability term, the strain energy associated with the consistency part ($\mathbf{K}_{E,c}$) and the stability part ($\mathbf{K}_{E,s}$) is evaluated using the CVT mesh (see Figure 15). The strain energy associated with $\mathbf{K}_{E,s}$ decreases under mesh refinement, and it is a few orders of magnitude lower than the energy associated with $\mathbf{K}_{E,c}$.

The computational results of the VEM are compared with those of the FEM for the CVT and RVT meshes. In the FEM, shape functions of polygonal elements are constructed using the mean value coordinates.[36] In the VEM, the diagonal matrix–based stabilization scheme with $\alpha_0 = 1/3$ is utilized. The computational results demonstrate that the VEM provides slightly more accurate results than the FEM in this example, as shown in Figure 16.

When Poisson's ratio is selected as 0.49995, a material is nearly incompressible. Thus, the B-bar method is employed with the scalar-based stabilization scheme ($\alpha_s = \mu$), as discussed in Section 2.6. Figure 17 demonstrates that the VEM

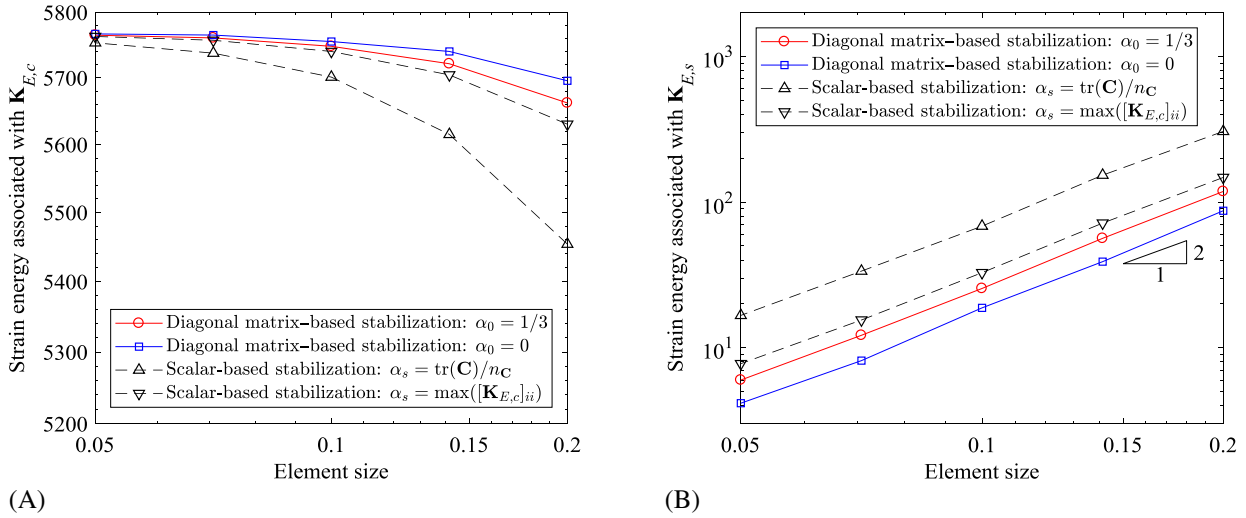



**FIGURE 15** Strain energy associated with (A) the consistency term ($\mathbf{K}_{E,c}$) and (B) the stability term ($\mathbf{K}_{E,s}$) [Colour figure can be viewed at wileyonlinelibrary.com]

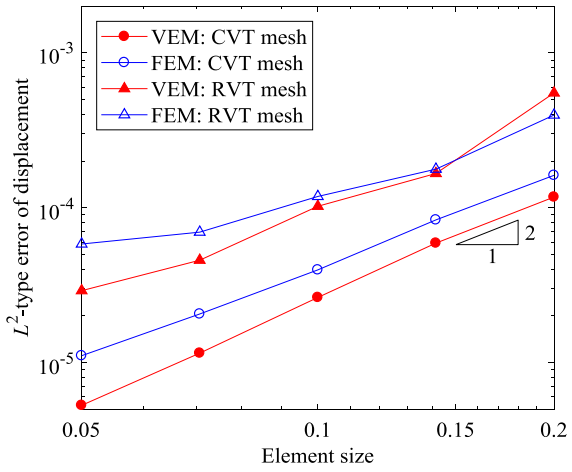



**FIGURE 16** Comparison between the finite element method (FEM) and the virtual element method (VEM) for the $L^2$-type displacement error. CVT, centroidal Voronoi tessellation; RVT, random Voronoi tessellation [Colour figure can be viewed at wileyonlinelibrary.com]

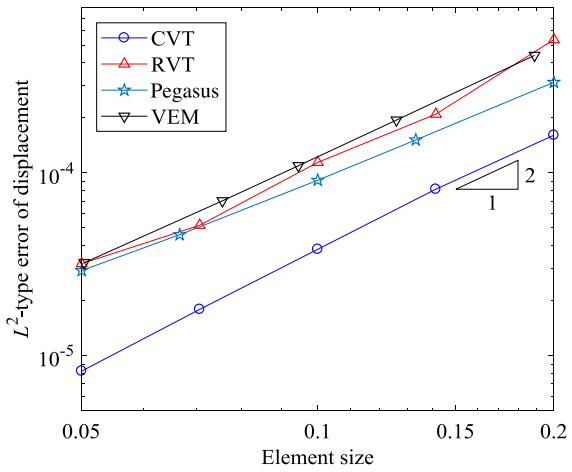



**FIGURE 17** Rate of convergence according to the mesh type for the nearly incompressible material ($\nu = 0.49995$) [Colour figure can be viewed at wileyonlinelibrary.com]

with the B-bar method results in the rate of convergence of 2, whereas the CVT mesh leads to the most accurate result in this study. The magnitude of the displacement vector at $t = 0.001$ is plotted in Figure 18. Although highly nonconvex elements are employed (ie, pegasus mesh), a smooth displacement field is achieved. Additionally, the VEM without the B-bar method is also employed for comparison purposes. The four stabilization schemes are utilized, ie, $\alpha_0 = 1/3$, $\alpha_0 = 0$, $\alpha_s = \mathrm{tr}(\mathbf{C})/n_{\mathbf{C}}$, and $\alpha_s = \max([\mathbf{K}_{E,c}]_{ii})$, and the CVT meshes are employed. Figure 19 demonstrates that the

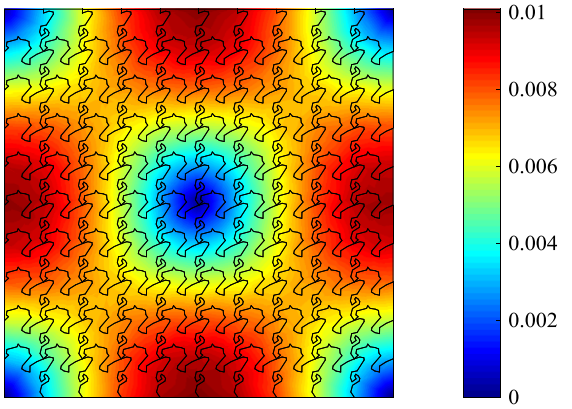



**FIGURE 18** Magnitude of the displacement vector at $t = 0.001$

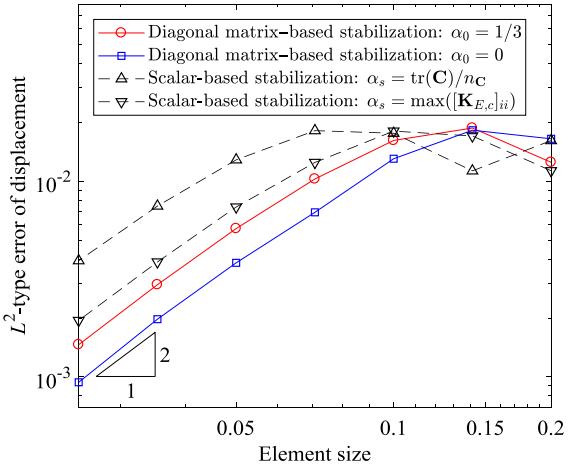



**FIGURE 19** Effect of the stabilization scheme on the rate of convergence for the nearly incompressible material ($\nu = 0.49995$) [Colour figure can be viewed at wileyonlinelibrary.com]

VEM without the B-bar method also reaches the optimal rate of convergence regardless of the stabilization scheme when the element size is smaller than a certain level. In addition, the diagonal matrix–based stabilization scheme tends to provide more accurate results than the scalar-based stabilization scheme in this study when the B-bar method is not utilized.

## 5.2 | Cantilever example

The length of the cantilever is 2 with a cross section of $0.1 \times 0.1$, as shown in Figure 20. The elastic modulus, Poisson's ratio, and density are 100, 0.25, and 0.1, respectively, and the plane stress condition is assumed. A sinusoidal load ($P$) is applied at the tip of the cantilever, ie, $P(t) = \sin(\pi t/T)$, for $0 \le t \le T$, where $T$ is the period of the cantilever (ie, $T = 7.831$), and the force is set to zero for $t \ge T$.

The cantilever is discretized into three types of meshes, eg, CVT, RVT, and pegasus meshes, as shown in Figure 21. For each mesh type, the convergence of the VEM is investigated in conjunction with the stabilization scheme. The four cases of the stabilization method are employed, ie, (a) diagonal matrix–based stabilization with $\alpha_0 = 1/3$, (b) diagonal matrix–based stabilization with $\alpha_0 = 0$, (c) scalar-based stabilization with $\alpha_s = \mathrm{tr}(\mathbf{C})/n_{\mathbf{C}}$, and (d) scalar-based stabilization with $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$. One notes that the computational results with the diagonal matrix–based stabilization with $\alpha_0 = 1$ are similar to those with the scalar-based stabilization with $\alpha_s = \mathrm{tr}(\mathbf{C})/n_{\mathbf{C}}$ because $\mathrm{tr}(\mathbf{C})/n_{\mathbf{C}}$ is greater than $[\boldsymbol{K}_{E,c}]_{ii}$ in most cases.

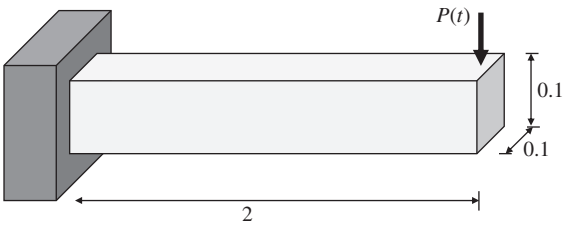



**FIGURE 20** Geometry and boundary conditions for a three-dimensional elastodynamic problem

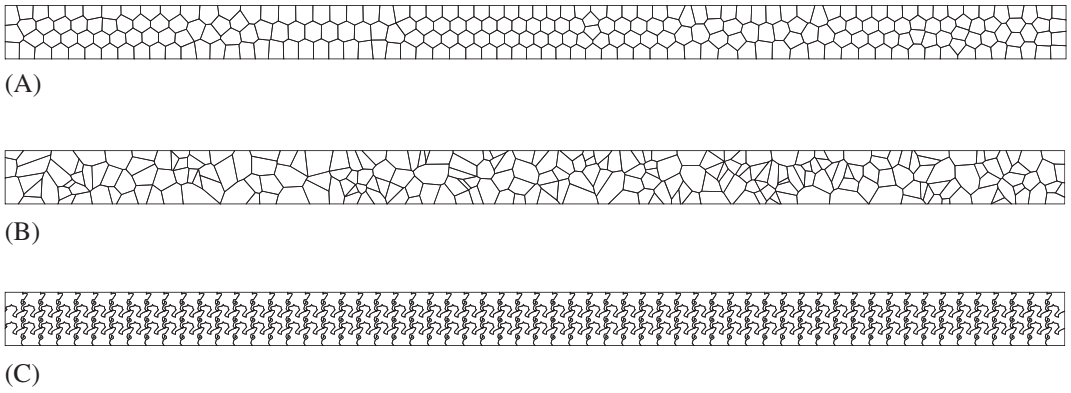



**FIGURE 21** Example of two-dimensional meshes for the cantilever example. A, Centroidal Voronoi tessellation mesh; B, Random Voronoi tessellation mesh; C, Pegasus mesh

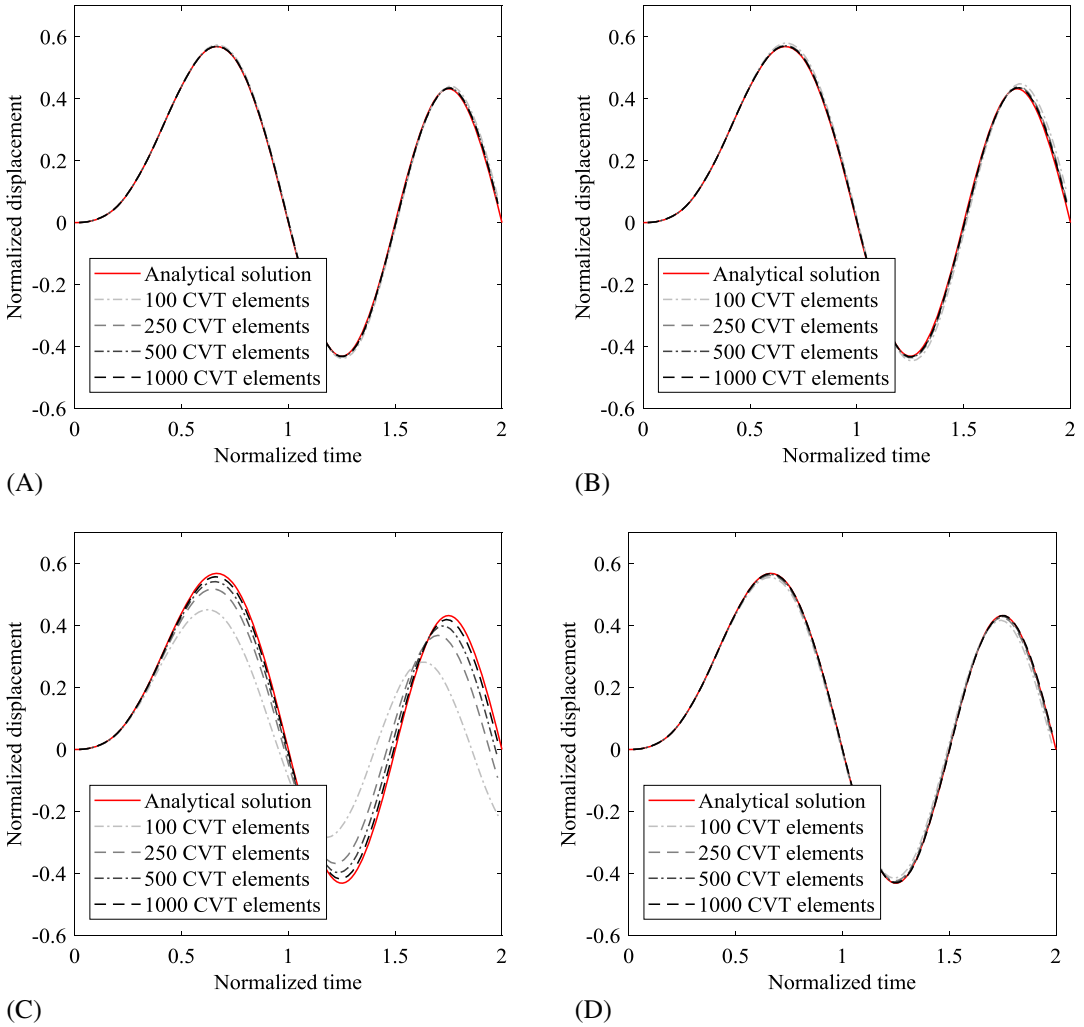



**FIGURE 22** Convergence of the centroidal Voronoi tessellation (CVT) meshes with respect to the stabilization schemes. A, Diagonal matrix–based stabilization with $\alpha_0 = 1/3$; B, Diagonal matrix–based stabilization with $\alpha_0 = 0$; C, Scalar-based stabilization with $\alpha_s = \mathrm{tr}(\mathbf{C})/n_{\mathbf{C}}$; D, Scalar-based stabilization with $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$

For the convex meshes (ie, CVT and RVT meshes), the VEM solution is almost on top of the analytical solution when the nondimensional constant $\alpha_0$ is in a range of $0 \leq \alpha_0 \leq 1/3$ for the diagonal matrix–based stabilization, as shown in Figures 22A,B and 23A,B. For the scalar-based stabilization, the scaling parameter ($\alpha_s$) based on $\boldsymbol{K}_{E,c}$ provides a more accurate solution than $\alpha_s$ based on the elasticity tensor. In summary, the diagonal matrix–based stabilization scheme provides a good approximation of the VEM, and the nondimensional parameter ($\alpha_0$) associated with a lower bound of $\boldsymbol{\Lambda}$ does not significantly affect the accuracy of the VEM solution for convex meshes. Furthermore, regardless of $\alpha_0$ and $\alpha_s$, the computational results demonstrate the convergence of the VEM solution to the analytical solution.

For the nonconvex mesh, ie, pegasus mesh, the convergence of computational results is shown in Figure 24. The diagonal matrix–based stabilization with $\alpha_0 = 1/3$ provides the most accurate results in this study. When the lower bound is not specified in the diagonal matrix–based stabilization ($\alpha_0 = 0$), the result provides a relatively large error (Figure 24B) because some diagonal terms of $\boldsymbol{K}_{E,c}$ for a nonconvex element are very small. In this example, the minimum of $[\boldsymbol{K}_{E,c}]_{ii}$ is a few orders of magnitude lower than the maximum of $[\boldsymbol{K}_{E,c}]_{ii}$. Thus, the accuracy of the VEM solution with nonconvex meshes is sensitive with respect to the stabilization scheme than with convex meshes. Additionally, Figure 24D demonstrates that the mesh grid of $60 \times 3$ provides a more accurate solution than the mesh grid of $120 \times 6$ because of error cancellation at the relatively coarse discretization. If one further refines the pegasus mesh, one can observe the convergence of computational results under mesh refinement.

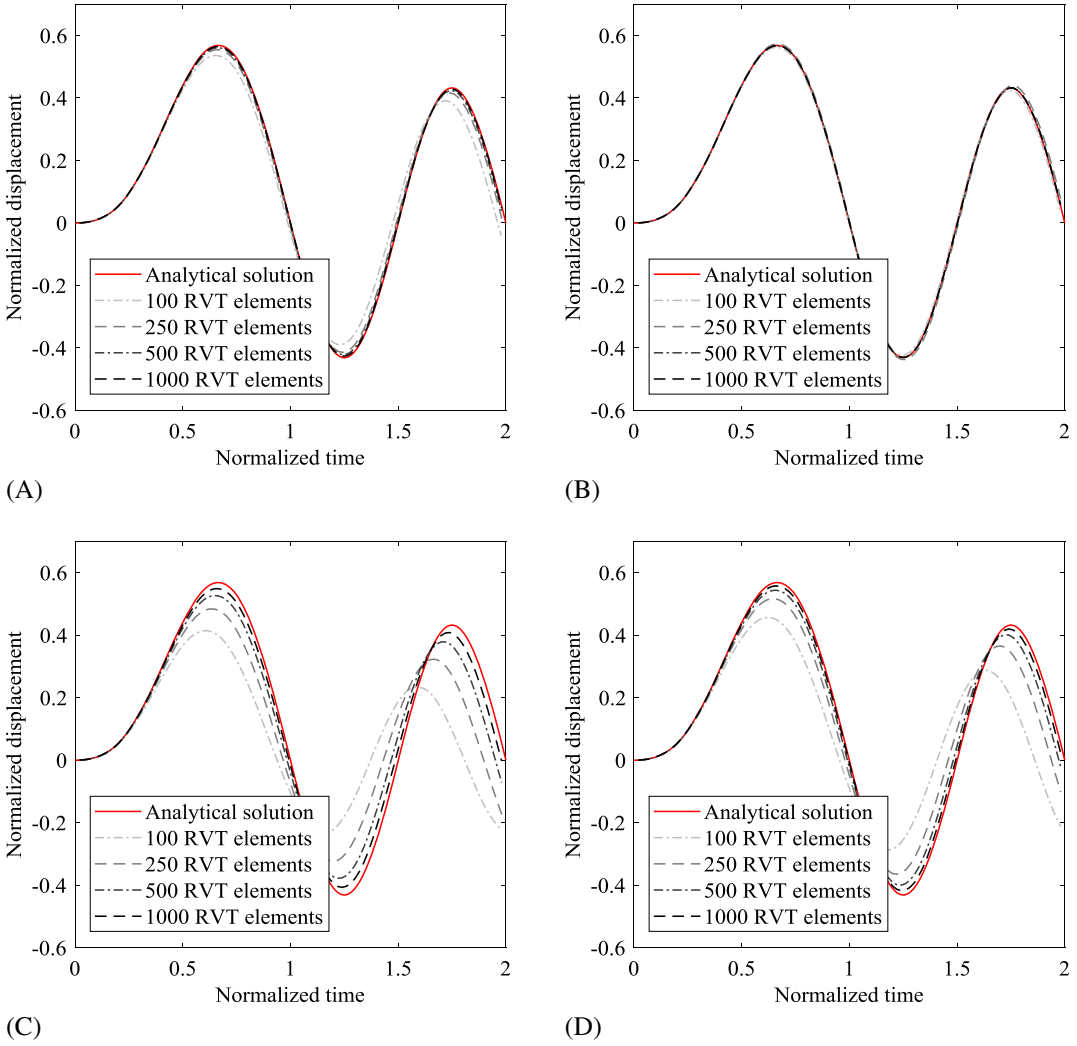



**FIGURE 23** Convergence of the random Voronoi tessellation (RVT) meshes with respect to the stabilization schemes.
A, Diagonal matrix–based stabilization with $\alpha_0 = 1/3$; B, Diagonal matrix–based stabilization with $\alpha_0 = 0$; C, Scalar-based stabilization with $\alpha_s = \text{tr}(\mathbf{C})/n_{\mathbf{C}}$; D, Scalar-based stabilization with $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$

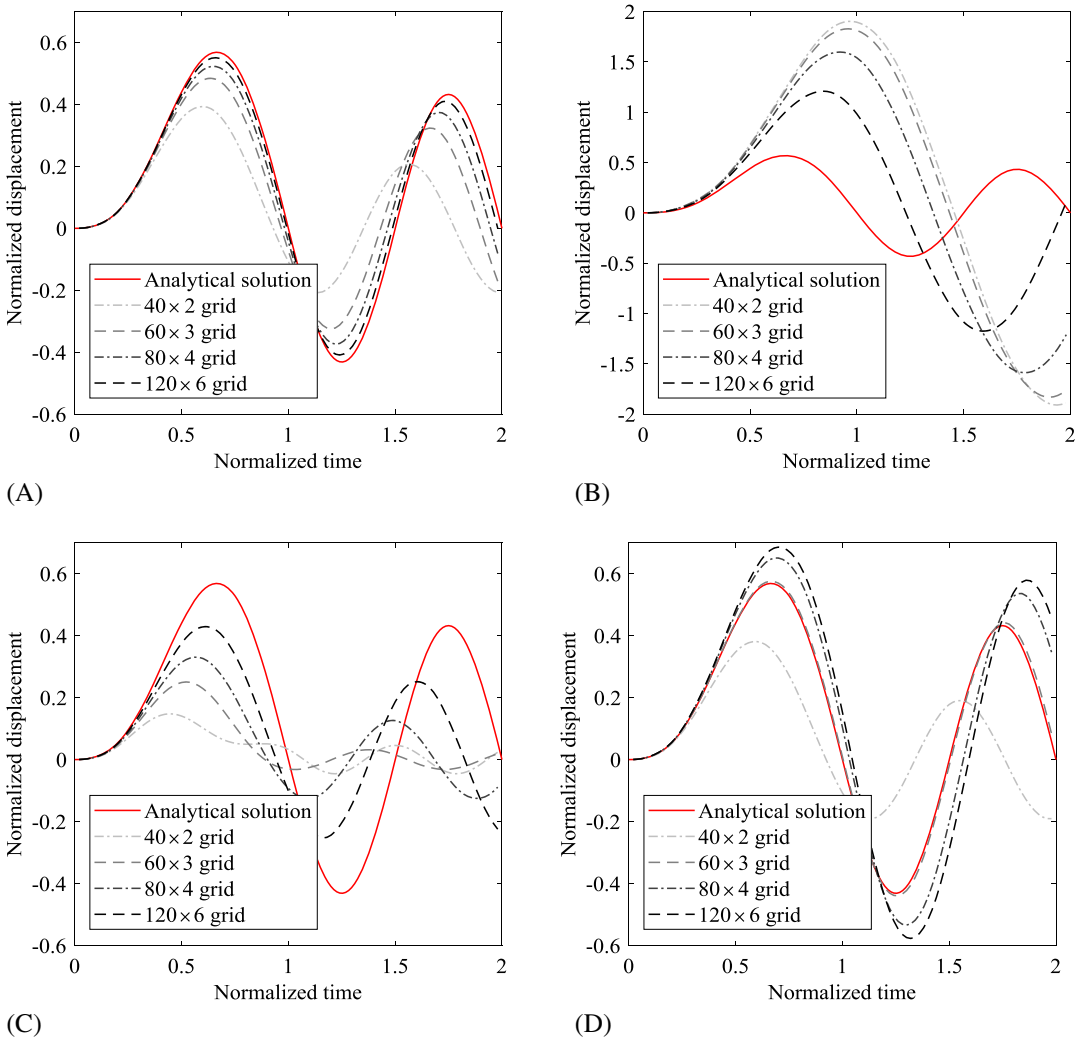



**FIGURE 24** Convergence of the pegasus meshes with respect to the stabilization schemes. A, Diagonal matrix–based stabilization with $\alpha_0 = 1/3$; B, Diagonal matrix–based stabilization with $\alpha_0 = 0$; C, Scalar-based stabilization with $\alpha_s = \mathrm{tr}(\mathbf{C})/n_{\mathbf{C}}$; D, Scalar-based stabilization with $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$

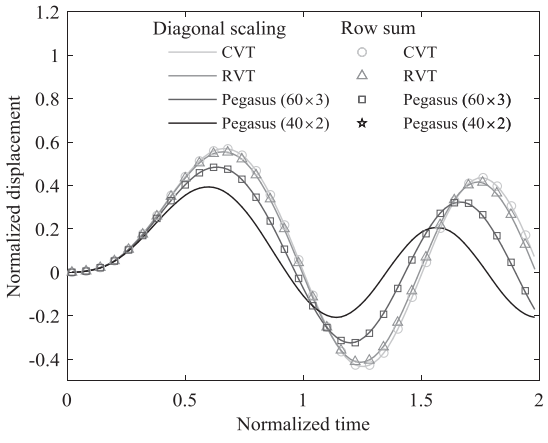



**FIGURE 25** Comparison of computational results according to the mass lumping technique. CVT, centroidal Voronoi tessellation; RVT, random Voronoi tessellation

The effect of the mass lumping techniques on computational results is investigated by comparing diagonal scaling (Equation (35)) with the row sum technique (Equation (36)). Four meshes are used in this study, ie, the CVT mesh with 250 elements, the RVT mesh with 250 elements, the pegasus mesh with $60 \times 3$ patches, and the pegasus mesh with $40 \times 2$ patches. Computational results with diagonal scaling are indicated as solid lines, whereas computational results with the row sum technique are denoted by markers (see Figure 25). These results are on top of each other for the CVT, RVT, and $60 \times 3$ pegasus meshes. However, the row sum technique provides a diverged solution for the $40 \times 2$ pegasus mesh, and thus, the result is not available in Figure 25. This is because the row sum technique does not guarantee the positive nodal mass for arbitrary nonconvex elements even with the linear polynomial space. For example, the histogram of nodal masses for the $40 \times 2$ pegasus mesh is illustrated in Figure 26. The row sum technique leads to negative nodal masses, whereas diagonal scaling provides positive nodal masses. Additionally, the distribution of nodal masses with the row sum technique is wider than that with diagonal scaling, as shown previously.

The computational results of the VEM are compared with those of the FEM for the CVT and RVT mesh types, as shown in Figures 27A and 27B, respectively. For each mesh type, two meshes are generated with numbers of elements of 100 and 250. For VEM analysis, the diagonal matrix–based stabilization with $\alpha_0 = 1/3$ is employed, and the results are plotted with solid markers. In the FEM, mean value coordinates[36] are utilized to construct shape functions, and the results are plotted with empty markers. Figure 27 demonstrates that the VEM provides a slightly more accurate solution than the FEM in this study.

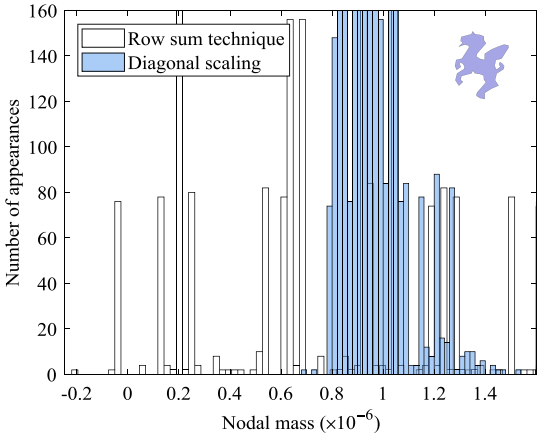



**FIGURE 26** Nodal mass comparison between diagonal scaling and the row sum technique with the $40 \times 2$ pegasus mesh

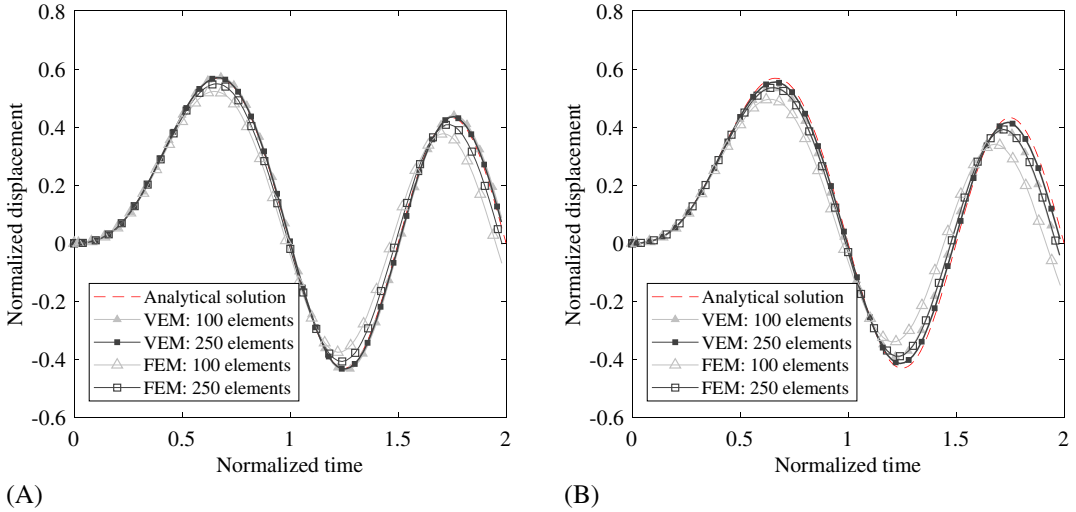



**FIGURE 27** Computational result comparison between the virtual element method (VEM) and the finite element method (FEM) for (A) the centroidal Voronoi tessellation mesh and (B) the random Voronoi tessellation mesh [Colour figure can be viewed at wileyonlinelibrary.com]

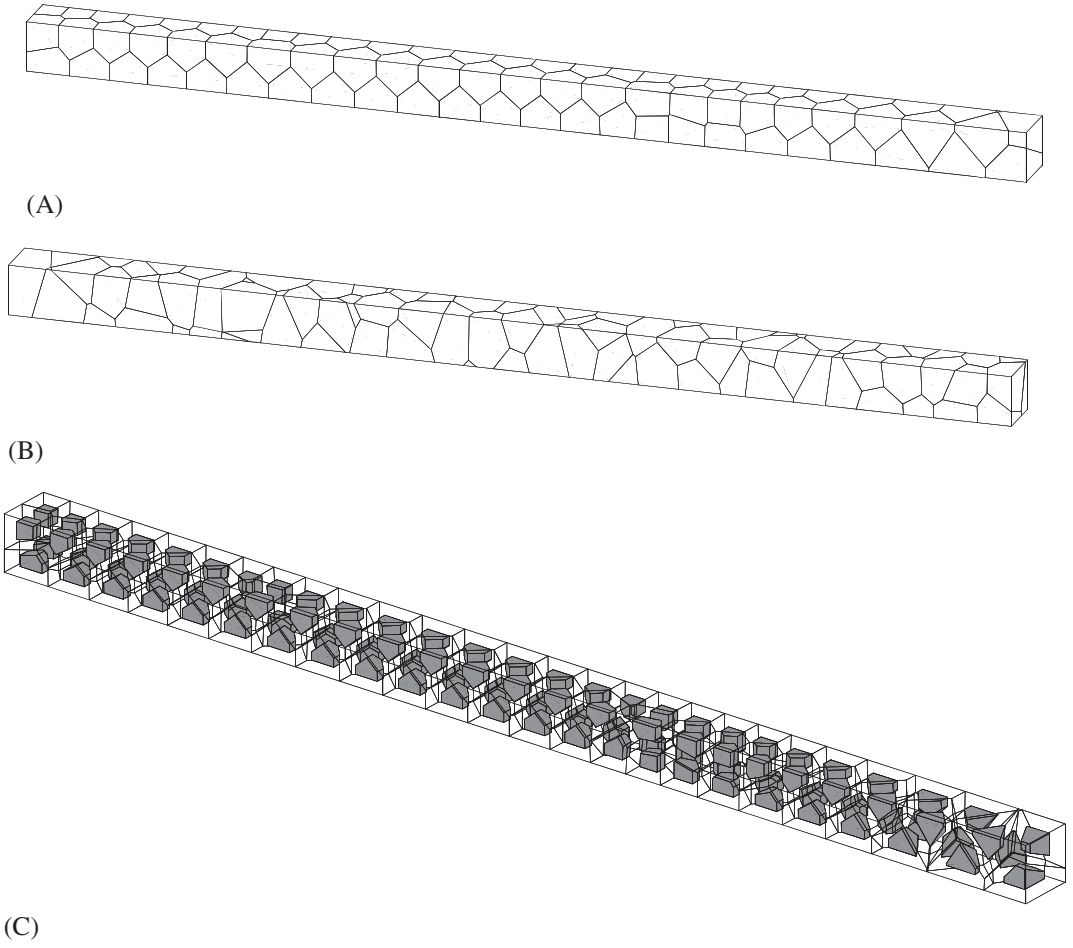



**FIGURE 28** Example of three-dimensional meshes for the cantilever example. A, Centroidal Voronoi tessellation mesh; B, Random Voronoi tessellation mesh; C, Particle mesh

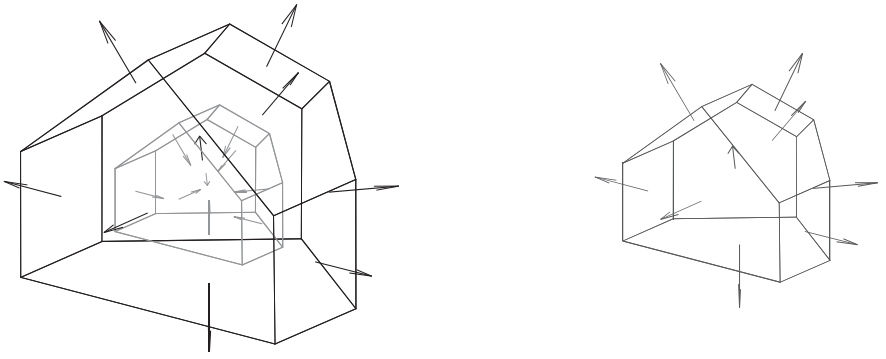


**FIGURE 29** Elements and their normal vectors for the particle mesh

## 5.3 | Three-dimensional example

For 3D analysis, the cantilever example (Figure 20) is solved using convex and nonconvex polyhedral elements. Three types of meshes are used, ie, CVT, RVT, and particle meshes, as shown in Figure 28. The CVT and RVT meshes are generated using PolyMesher3D.[40] The particle mesh is generated by creating empty holes within the CVT mesh. The particle mesh consists of two types of polyhedra, ie, one with an empty hole and another to fill the empty hole, as shown in Figure 29. Four meshes are generated for each mesh type. The numbers of elements are 100, 250, 500, and 1000 for the CVT and RVT meshes, and they are 200, 500, 1000, and 2000 for the particle meshes. Four stabilization schemes are compared among each other, ie, diagonal matrix–based stabilization with $\alpha_0 = 1/9$ (case I), diagonal matrix–based stabilization with $\alpha_0 = 0$ (case II), scalar-based stabilization with $\alpha_s = \text{tr}(\mathbf{C})/n_{\mathbf{C}}$ (case III), and scalar-based stabilization with $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$ (case IV).

For each mesh type, the convergence of the VEM solution is investigated according to the cases of the stabilization scheme, as shown in Figures 30-32. For all mesh types, the stabilization method with cases I, II, and IV provides a more accurate approximation than the stabilization method with case III. Additionally, the results of case III lead to the monotonic convergence of the VEM solution, whereas those of the other cases are not monotonic. Nonmonotonic convergence

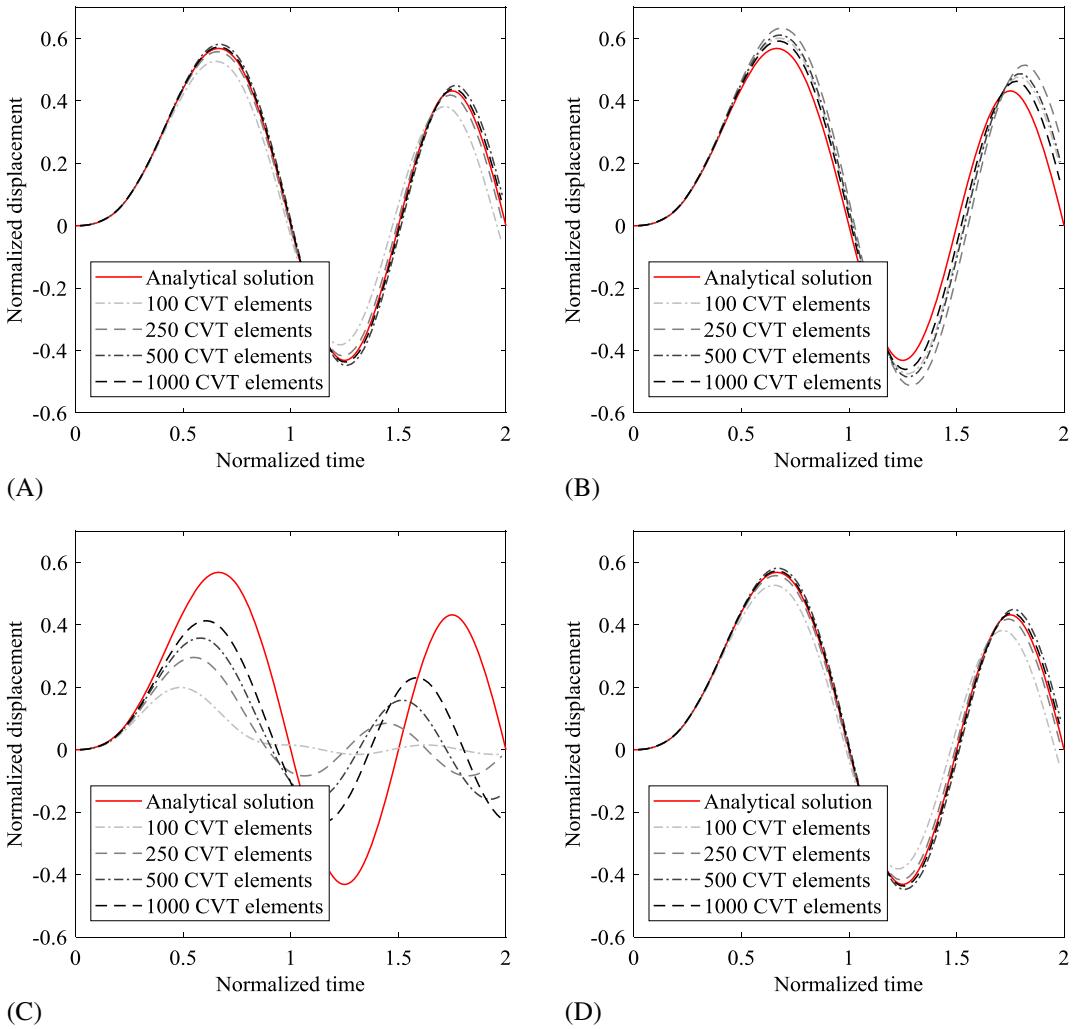



**FIGURE 30** Convergence of the three-dimensional centroidal Voronoi tessellation (CVT) meshes according to the stabilization schemes. A, Diagonal matrix–based stabilization with $\alpha_0 = 1/9$; B, Diagonal matrix–based stabilization with $\alpha_0 = 0$; C, Scalar-based stabilization with $\alpha_s = \text{tr}(\mathbf{C})/n_{\mathbf{C}}$; D, Scalar-based stabilization with $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$

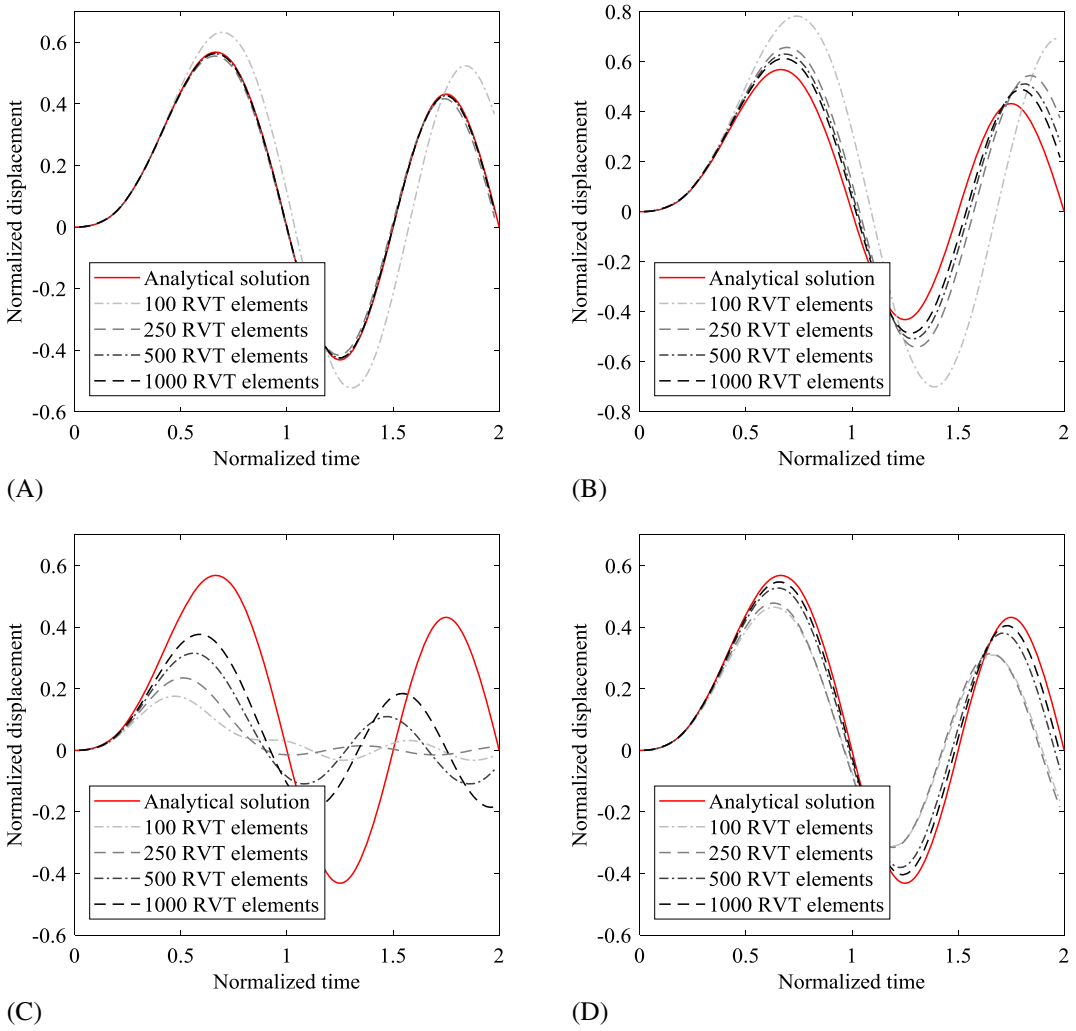



**FIGURE 31** Convergence of the three-dimensional random Voronoi tessellation (RVT) meshes according to the stabilization schemes. A, Diagonal matrix–based stabilization with $\alpha_0 = 1/9$; B, Diagonal matrix–based stabilization with $\alpha_0 = 0$; C, Scalar-based stabilization with $\alpha_s = \text{tr}(\mathbf{C})/n_{\mathbf{C}}$; D, Scalar-based stabilization with $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$

is resulted from error cancellation associated with the stability term. For example, when the lower bound of the scaling term ($\alpha_0$) is set to zero (case II), the approximated $\overline{\mathbf{K}}_{E,s}$ may be lower than the exact $\overline{\mathbf{K}}_{E,s}$, and thus, the VEM solution is more flexible than the analytical solution, as shown in Figures 30B, 31B, and 32B. On the other hand, for stabilization case III, the approximated $\overline{\mathbf{K}}_{E,s}$ may be stiffer than the exact $\overline{\mathbf{K}}_{E,s}$, and thus, the VEM solution provides a smaller deformation than the analytical solution, as shown in Figures 30C, 31C, and 32C.

A cantilever with inclusions is investigated using the particle mesh with 200 elements. The matrix is represented by elements with a hole, whereas each particle is described by one element to fill the hole. The elastic modulus of the matrix is fixed at 100, whereas the five cases of the elastic modulus of particles ($E_p$) are defined, ie, three constant cases and two graded cases, as shown in Figure 33A. For the constant cases, the elastic modulus is 1, 100, and 10 000. For the graded case, the elastic modulus of a particle is defined by a function value at the centroid of a particle. Two exponentially varying functions are used, ie, $E_p(x) = 10\,000\exp(-4.6052x)$ and $E_p(x) = \exp(4.6052x)$, which lead to the stronger and weaker elastic moduli of particles at the support, respectively. Then, the displacement-time results are shown in Figure 33B. When the smaller elastic modulus is defined at the support region, the corresponding displacement is larger than the displacement in the homogeneous case.

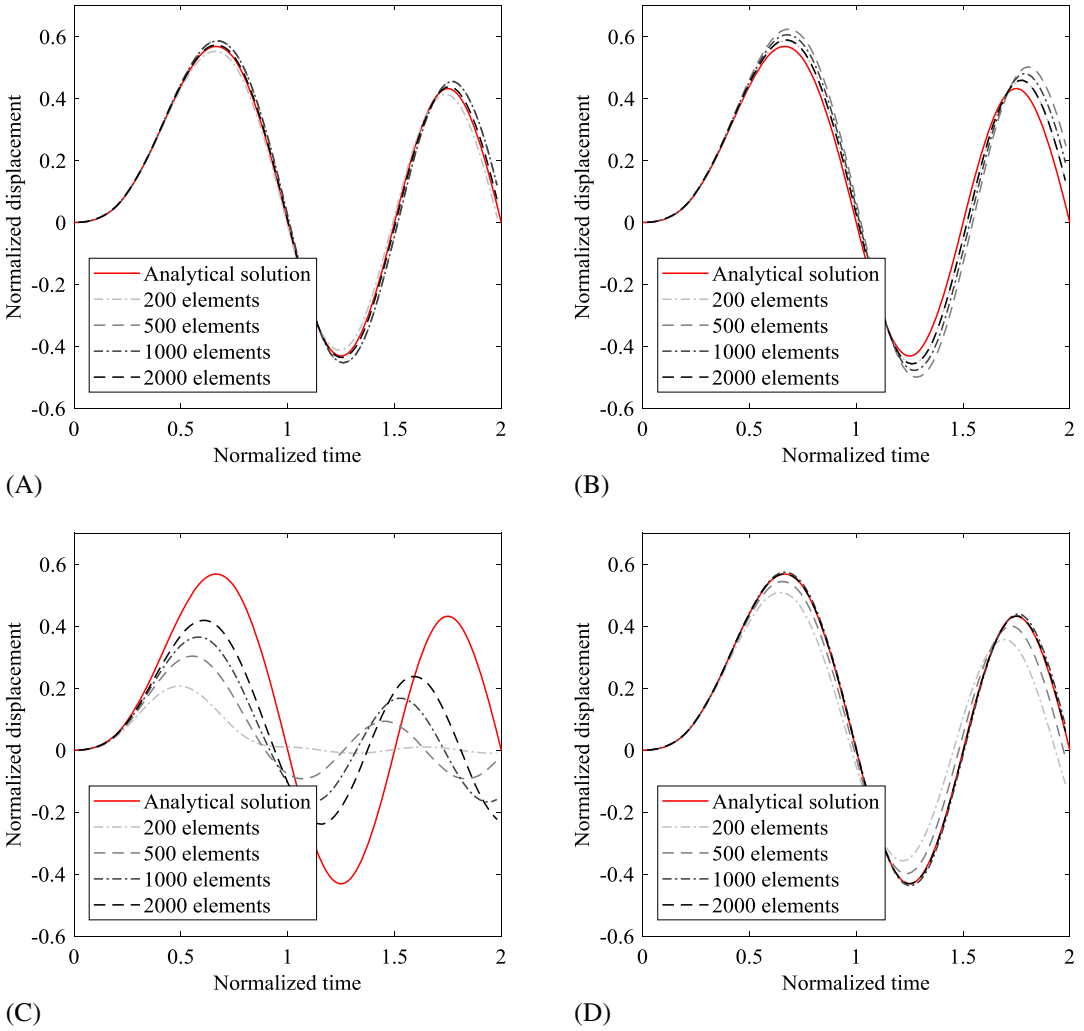
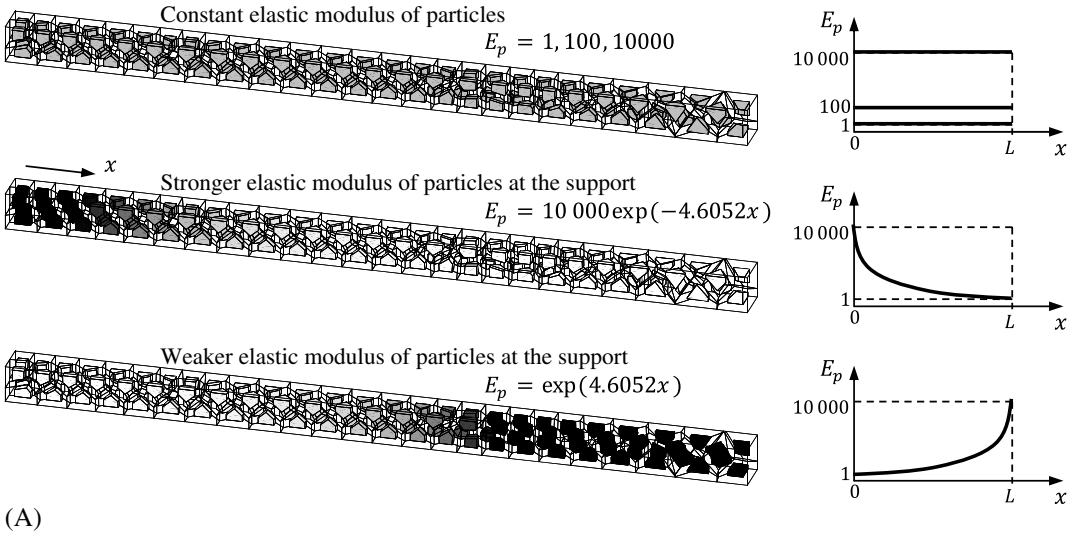



**FIGURE 32** Convergence of the three-dimensional particle meshes according to the stabilization schemes. A, Diagonal matrix–based stabilization with $\alpha_0 = 1/9$; B, Diagonal matrix–based stabilization with $\alpha_0 = 0$; C, Scalar-based stabilization with $\alpha_s = \text{tr}(\mathbf{C})/n_{\mathbf{C}}$; D, Scalar-based stabilization with $\alpha_s = \max([\boldsymbol{K}_{E,c}]_{ii})$

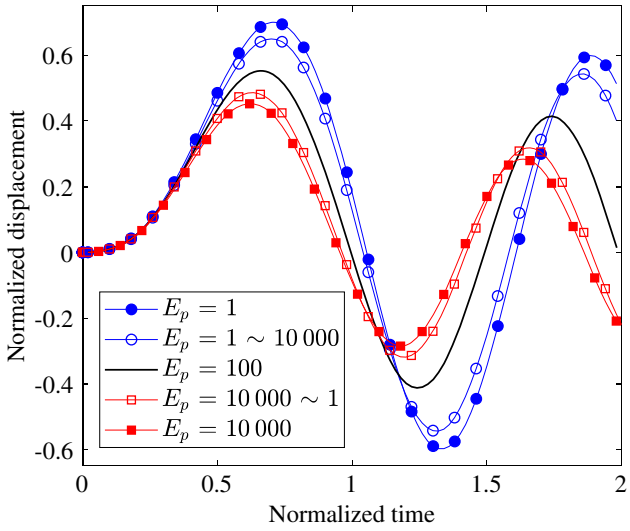
Additionally, the period of the stiffer cantilever at the support is shorter than that of the homogeneous cantilever, as expected.

## 5.4 | Cook's beam with nearly incompressible solids

Wave propagation in Cook's beam with a nearly incompressible material is investigated. The geometry of the beam is shown in Figure 34, and a uniformly distributed traction of $P(t)$ is applied along the right edge, ie, $P(t) = \sin(\pi t/5)$. The domain is discretized into CVT meshes, and four meshes are generated with numbers of elements of 200, 400, 800, and 1600. The elastic modulus and Poisson's ratio are selected as 250 and 0.49995, respectively, with the plane strain condition, and the density is 1. Because of near incompressibility, the B-bar method is utilized using the scalar-based stabilization scheme ($\alpha_s = \mu$). Then, the VEM with the B-bar method is compared with the standard VEM (ie, without the B-bar method), as shown in Figure 35. The two approaches provide the converged solution of the vertical displacement, whereas the VEM with the B-bar method provides a more accurate solution than the standard VEM for nearly incompressible solids. Additionally, the horizontal and vertical stresses at $t = 5$ are plotted in Figures 36A and 36B, respectively. The number of elements is 400, and the smooth stress fields are obtained for nearly incompressible solids.




(A)




(B)

**FIGURE 33** A, Five cases of elastic moduli of particles along the longitudinal direction; B, Computational results of each case [Colour figure can be viewed at wileyonlinelibrary.com]

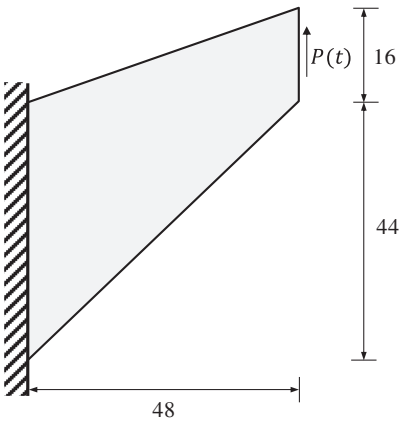



**FIGURE 34** Geometry of Cook's beam example

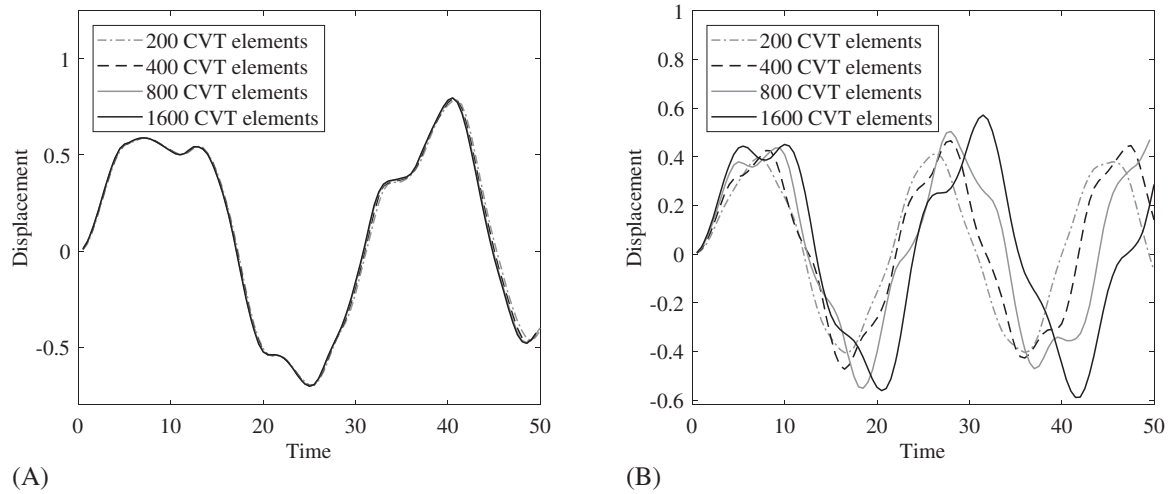



**FIGURE 35** Convergence of the vertical displacement under mesh refinement. A, Virtual element method (VEM) with the B-bar method; B, Standard VEM (ie, without the B-bar method). CVT, centroidal Voronoi tessellation

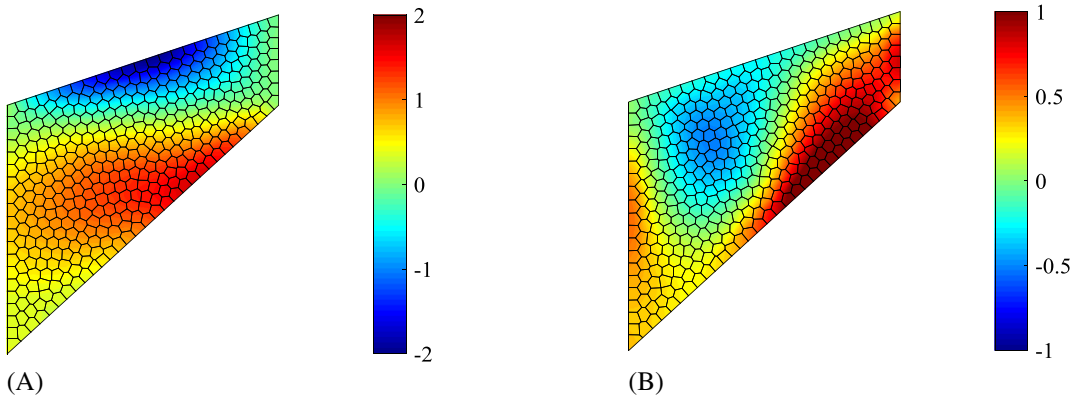



**FIGURE 36** Stress fields at $t = 5$ with nearly incompressible solids. A, Horizontal stress; B, Vertical stress

## 6 | CONCLUSIONS

The VEM is utilized to solve 2D and 3D elastodynamic problems with explicit time integration. Convergence and stability of the numerical solution are addressed for arbitrary convex/nonconvex polygonal and polyhedral elements. The key findings from our numerical recipes for an elastodynamic VEM with explicit time integration are summarized as follows.

- Regardless of the stability term and element shapes, the VEM provides an optimal rate of convergence under mesh refinement for elastodynamic problems.
- For the analysis of nearly incompressible solids, the VEM with the B-bar method is suggested. To construct a B-bar formulation in the VEM setting, the stability term is modified to stabilize only the deviatoric part of the stiffness matrix, which requires no additional computational effort. Computational results demonstrate that the VEM with the B-bar method provides more accurate results than the standard VEM (ie, without the B-bar method).
- To approximate the critical time step in an elastodynamic VEM with explicit time integration, two approaches are suggested, ie, the maximum eigenvalue of a local system and the effective element length. These two approaches conservatively estimate the critical time step obtained from the maximum eigenvalue of a global system, whereas the estimated time steps are within the same order of magnitude.

- The effective element length for polygonal elements is determined using the minimum element edge length and the minimum length from the centroid to the nodes of an element. Computational results demonstrate that small edges in polygons do not significantly affect the critical time step for convex elements, ie, CVT and RVT.
- In general, the diagonal matrix–based stabilization schemes with $\alpha_0 = 1/3$ for 2D and with $\alpha_0 = 1/9$ for 3D problems provide the most accurate results in this study. According to the choice of the stability term in the stiffness matrix, nonconvex meshes are more sensitive to the accuracy of the VEM solution than convex meshes.
- Diagonal scaling provides positive nodal masses in the VEM. However, the row sum technique does not guarantee positive nodal masses with nonconvex elements, even for the linear polynomial space. Numerical instability is observed when negative nodal mass appears with the row sum technique. Additionally, the nodal mass distribution with the row sum technique is wider than the distribution with diagonal scaling.
- For the computation of the consistent mass matrix, as well as the lumped mass matrix obtained from the diagonal scaling technique, a second-order numerical integration of polynomials within a polygon (or polyhedron) is needed for the linear VEM. When the row sum technique is used to construct the lumped mass matrix, a first-order integration is needed. The first-order integration can be efficiently achieved by a one-point rule, with the integration point and weight being the centroid and volume of each element, respectively.
- The computational results of the VEM are compared with those of the FEM for convex elements (ie, CVT and RVT). The VEM with the diagonal matrix–based stabilization can provide more accurate results than the FEM for CVT and RVT meshes. For regular meshes (eg, CVT meshes), the critical time step is similar for both the VEM and the FEM. For irregular meshes (eg, RVT meshes), the VEM can achieve a larger critical time step as compared to the FEM.

## ACKNOWLEDGEMENTS

KP acknowledges support from the National Research Foundation of Korea through the Basic Science Research Program funded by the Ministry of Science, ICT and Future Planning under grant 2018R1A2B6007054 and from the Korea Institute of Energy Technology Evaluation and Planning funded by the Ministry of Trade, Industry and Energy under grant 20171510101910. HC and GHP acknowledge support from the National Science Foundation under grant 1624232 (formerly grant 1437535) and from the Raymond Allen Jones Chair at the Georgia Institute of Technology.

## ORCID


*Kyoungsoo Park* https://orcid.org/0000-0001-6243-5016

## REFERENCES

1. Beirão da Veiga L, Brezzi F, Cangiani A, Manzini G, Marini LD, Russo A. Basic principles of virtual element methods. *Math Models Methods Appl Sci*. 2013;23(1):199-214. https://doi.org/10.1142/S0218202512500492
2. Beirão da Veiga L, Brezzi F, Marini LD, Russo A. The hitchhiker's guide to the virtual element method. *Math Models Methods Appl Sci*. 2014;24(8):1541-1573. https://doi.org/10.1142/S021820251440003X
3. Vacca G, Beirão da Veiga L. Virtual element methods for parabolic problems on polygonal meshes. *Numer Methods Partial Differ Equ*. 2015;31:2110-2134. https://doi.org/10.1002/num
4. Beirão da Veiga L, Brezzi F, Marini LD, Russo A. Virtual element method for general second-order elliptic problems on polygonal meshes. *Math Models Methods Appl Sci*. 2016;26(4):729-750. https://doi.org/10.1142/S0218202516500160
5. Vacca G. Virtual element methods for hyperbolic problems on polygonal meshes. *Comput Math Appl*. 2017;74(5):882-898. https://doi.org/10.1016/j.camwa.2016.04.029
6. Adak D, Natarajan E, Kumar S. Virtual element method for semilinear hyperbolic problems on polygonal meshes. *Int J Comput Math*. 2018;96(5):971-991. https://doi.org/10.1080/00207160.2018.1475651
7. Gain AL, Talischi C, Paulino GH. On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Comput Methods Appl Mech Eng*. 2014;282:132-160. https://doi.org/10.1016/j.cma.2014.05.005
8. Beirão da Veiga L, Lovadina C, Mora D. A virtual element method for elastic and inelastic problems on polytope meshes. *Comput Methods Appl Mech Eng*. 2015;295:327-346. https://doi.org/10.1016/j.cma.2015.07.013
9. Chi H, Beirão da Veiga L, Paulino GH. Some basic formulations of the virtual element method (VEM) for finite deformations. *Comput Methods Appl Mech Eng*. 2017;318:148-192. https://doi.org/10.1016/j.cma.2016.12.020
10. Wriggers P, Reddy BD, Rust W, Hudobivnik B. Efficient virtual element formulations for compressible and incompressible finite deformations. *Computational Mechanics*. 2017;60(2):253-268. https://doi.org/10.1007/s00466-017-1405-4
11. Park K, Chi H, Paulino GH. On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration. *Comput Methods Appl Mech Eng*. 2018. Under review.

12. Aldakheel F, Hudobivnik B, Hussein A, Wriggers P. Phase-field modeling of brittle fracture using an efficient virtual element scheme. *Comput Methods Appl Mech Eng*. 2018;341:443-466. https://doi.org/10.1016/j.cma.2018.07.008
13. Benedetto MF, Caggiano A, Etse G. Virtual elements and zero thickness interface-based approach for fracture analysis of heterogeneous materials. *Comput Methods Appl Mech Eng*. 2018;338:41-67. https://doi.org/10.1016/j.cma.2018.04.001
14. Benedetto MF, Berrone S, Borio A, Pieraccini S, Scialò S. A hybrid mortar virtual element method for discrete fracture network simulations. *J Comput Phys*. 2016;306:148-166. https://doi.org/10.1016/j.jcp.2015.11.034
15. Wriggers P, Rust WT, Reddy BD. A virtual element method for contact. *Computational Mechanics*. 2016;58(6):1039-1050. https://doi.org/10.1007/s00466-016-1331-x
16. Antonietti PF, Bruggi M, Scacchi S, Verani M. On the virtual element method for topology optimization on polygonal meshes: a numerical study. *Comput Math Appl*. 2017;74(5):1091-1109. https://doi.org/10.1016/j.camwa.2017.05.025
17. Chi H, Pereira A, Menezes IFM, Paulino GH. Virtual element method (VEM)-based topology optimization: an integrated framework, structural and multidisciplinary optimization. *Struct Multidiscip Optim* 2019. In press. https://doi.org/10.1007/s00158-019-02268-w
18. Ahmad B, Alsaedi A, Brezzi F, Marini LD, Russo A. Equivalent projectors for virtual element methods. *Comput Math Appl*. 2013;66(3):376-391. https://doi.org/10.1016/j.camwa.2013.05.015
19. Beirão da Veiga L, Dassi F, Russo A. High-order virtual element method on polyhedral meshes. *Comput Math Appl*. 2017;74(5):1110-1122. https://doi.org/10.1016/j.camwa.2017.03.021
20. Chi H, Beirão da Veiga L, Paulino GH. A simple and effective gradient recovery scheme and a posteriori error estimator for the virtual element method (VEM). *Comput Methods Appl Mech Eng* 2018;347:21-58. https://doi.org/10.1016/j.cma.2018.08.014
21. Hughes TJR. Generalization of selective integration procedures to anisotropic and nonlinear media. *Int J Numer Methods Eng*. 1980;15:1413-1418.
22. Newmark NM. A method of computation for structural dynamics. *J Eng Mech*. 1959;85(7):67-94. https://doi.org/0.1016/j.compgeo.2015.08.008
23. Hinton E, Rock T, Zienkiewicz OC. A note on mass lumping in the finite element time domain method. *Earthq Eng Struct Dyn*. 1976;4:245-249. https://doi.org/10.1109/TAP.2005.863159
24. Bathe K-J. *Finite Element Procedures*. Upper Saddle River, NJ: Prentice-Hall; 1996.
25. Hughes TJR. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Mineola, NY: Dover Publications; 2000.
26. Leech JW, Hsu PT, Mack EW. Stability of a finite-difference method for solving matrix equations. *AIAA Journal*. 1965;3(11):2172-2173. https://doi.org/10.2514/3.3342
27. Goudreau GL, Taylor RL. Evaluation of numerical integration methods in elastodynamics. *Comput Methods Appl Mech Eng*. 1972;2(1):69-97. https://doi.org/10.1016/0045-7825(73)90023-6
28. Lin JI. An element eigenvalue theorem and its application for stable time steps. *Comput Methods Appl Mech Eng*. 1989;73:283-294.
29. Lin JI. Bounds on eigenvalues of finite element systems. *Int J Numer Methods Eng*. 1991;32(5):957-967. https://doi.org/10.1002/nme.1620320503
30. Courant R, Friedrichs K, Lewy H. Über die partiellen Differenzengleiehungen der Physik. *Mathematische Annalen*. 1928;100(1):32-74. https://doi.org/10.1007/BF01448839
31. Mullen R, Belytschko T. An analysis of an unconditionally stable explicit method. *Comput Struct*. 1983;16(6):691-696. https://doi.org/10.1016/0045-7949(83)90060-3
32. Askes H, Rodríguez-Ferran A, Hetherington J. The effects of element shape on the critical time step in explicit time integrators for elasto-dynamics. *Int J Numer Methods Eng*. 2015;101:809-824. https://doi.org/10.1002/nme.4819
33. Talischi C, Paulino GH, Pereira A, Menezes IFM. PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab. *Struct Multidiscip Optim*. 2012;45(3):309-328. https://doi.org/10.1007/s00158-011-0706-z
34. Escher MC. The official website. http://www.mcescher.com/
35. Paulino GH, Gain AL. Bridging art and engineering using Escher-based virtual elements. *Struct Multidiscip Optim*. 2015;51:867-883. https://doi.org/10.1007/s00158-014-1179-7
36. Floater MS. Mean value coordinates. *Comput Aided Geom Des*. 2003;20:19-27. https://doi.org/10.1016/j.cagd.2005.06.004
37. Haider J, Lee CH, Gil AJ, Bonet J. A first-order hyperbolic framework for large strain computational solid dynamics: an upwind cell centred total Lagrangian scheme. *Int J Numer Methods Eng*. 2017;109(3):407-456. https://doi.org/10.1002/nme.5293
38. Lee CH, Gil AJ, Bonet J. Development of a stabilised Petrov-Galerkin formulation for conservation laws in Lagrangian fast solid dynamics. *Comput Methods Appl Mech Eng*. 2014;268:40-64. https://doi.org/10.1016/j.cma.2013.09.004
39. Scovazzi G, Carnes B, Zeng X, Rossi S. A simple, stable, and accurate linear tetrahedral finite element for transient, nearly, and fully incompressible solid dynamics: a dynamic variational multiscale approach. *Int J Numer Methods Eng*. 2016;106:799-839. https://doi.org/10.1002/nme
40. Thedin RS, Pereira A, Menezes IFM, Paulino GH. Polyhedral mesh generation and optimization for finite element computations. *Paper presented at: XXXV Ibero-Latin American Congress on Computational Methods in Engineering*; 2014; Fortaleza, Brazil.


**How to cite this article:** Park K, Chi H, Paulino GH. Numerical recipes for elastodynamic virtual element methods with explicit time integration. *Int J Numer Methods Eng*. 2020;121:1–31. https://doi.org/10.1002/nme.6173