Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 356 (2019) 669–684
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration

Kyoungsoo Park[a], Heng Chi[b], Glaucio H. Paulino[b,*]

[a] Department of Civil and Environmental Engineering, Yonsei University, 50 Yonsei-ro Seodaemun-gu, Seoul 120-749, South Korea
[b] School of Civil and Environmental Engineering, Georgia Institute of Technology, 790 Atlantic Drive, Atlanta, GA 30332, United States

Received 7 February 2019; received in revised form 18 June 2019; accepted 20 June 2019
Available online xxxx

Highlights

- Exploring arbitrarily-shaped polygons/polyhedra in computational mechanics.
- The VEM is established for elastodynamics with explicit time integration.
- Novel diagonal matrix-based stabilization scheme improves VEM performance (and accuracy).
- The critical time step is evaluated using the maximum eigenvalue of a system of equations.

Abstract

While the literature on numerical methods (e.g. finite elements and, to a certain extent, virtual elements) concentrates on convex elements, there is a need to probe beyond this limiting constraint so that the field can be further explored and developed. Thus, in this paper, we employ the virtual element method for non-convex discretizations of elastodynamic problems in 2D and 3D using an explicit time integration scheme. In the formulation, a diagonal matrix-based stabilization scheme is proposed to improve performance and accuracy. To address efficiency, a critical time step is approximated and verified using the maximum eigenvalue of the local (rather than global) system. The computational results demonstrate that the virtual element method is able to consistently handle general nonconvex elements and even non-simply connected elements, which can lead to convenient modeling of arbitrarily-shaped inclusions in composites.

Keywords: Virtual element method (VEM); Elastodynamics; Nonconvex elements; Non-simply connected elements; Stabilization

1. Introduction

General polygonal and polyhedral elements have great potential in the field of computational mechanics [1]. Consideration of arbitrary element shape leads to great flexibility on domain discretization and associated computational schemes, e.g. easy treatment of representative volume elements, mesh refinement, coarsening and adaptivity. When the finite element method (FEM) is considered with arbitrarily shaped polygonal and polyhedral elements, the construction and integration of shape functions pose major challenges, especially in three dimensions

* Corresponding author.
E-mail address: paulino@gatech.edu (G.H. Paulino).

https://doi.org/10.1016/j.cma.2019.06.031
0045-7825/

(3D). For example, the Wachspress shape functions [2] are limited to convex polyhedra [3], while the shape functions from the Mean Value coordinates are applicable to ones with simplicial faces [4]. Alternatively, harmonic shape functions [5,6] and maximum-entropy shape functions [7,8] can be utilized for more general polyhedra, but, in practice, their values and gradients need to be computed numerically at every integration point of every element in the mesh, which can become cumbersome, especially in 3D. In terms of numerical integration, the shape functions of polygonal and polyhedral finite elements are typically non-polynomial — thus, standard Gauss quadrature will lead to consistency errors that do not vanish under mesh refinement [9]. As a result, integrating non-polynomial shape functions and their gradients within general polyhedra typically require numerous quadrature points [9], together with schemes to correct consistency errors [6,9–11].

The virtual element method (VEM) has been proposed (e.g., [12,13]) to consistently handle general polygonal and polyhedral elements. In contrast to the standard FEM, the VEM shape functions of each element are implicitly defined such that their projections onto the space of polynomials of suitable order can be exactly computed, and the approximation of the discrete bilinear form decomposed into *consistency* and *stability* terms. Because of the implicit shape functions, the VEM provides great flexibility on element shape, e.g., convex, non-convex, and non-simply connected [14]. The VEM has been successfully utilized to solve a large range of problems, including elliptic partial differential equations (PDEs) [15], parabolic PDEs [16,17], hyperbolic PDEs [18,19], elastostatics [20–24], fracture mechanics [25,26], topology optimization [27–29], Stokes and Navier–Stokes fluids [30–32], and others. For example, Beirão da Veiga et al. [30] defined a virtual element space so that the computed velocity field is divergence-free for Stokes and Navier–Stokes problems. Chi et al. [21] demonstrated that a displacement-based VEM formulation did not provide volumetric locking with a nearly incompressible material. Wriggers et al. [33] utilized VEM to efficiently handle non-matching meshes for contact problems. Because the VEM field is constantly expanding, then the aforementioned references are just an incomplete sample.

In this study, the VEM is tailored to consistently handle general nonconvex elements for elastodynamic problems with explicit time integration in both 2D and 3D. A diagonal matrix-based stabilization scheme is proposed to enhance the accuracy of the VEM solution, and a critical time step is approximated using the maximum eigenvalue of the local system. The computational results demonstrate that the VEM is able to consistently handle general nonconvex, and even non-simply connected (e.g., one with a hole in the interior) elements. A summary regarding the remainder of this paper follows. Section 2 addresses the basic VEM formulation, and Section 3 details the analytical estimation of the critical time step. Afterwards, Section 4 provides elastodynamic computational results. Finally, Section 5 presents concluding remarks, including the key findings of the present study.

2. Virtual element method formulation

Consider an elastic solid $\Omega \in \mathbb{R}^d$ with $\partial\Omega$ being its boundary and d being the problem dimension ($d = 2$ for 2D problems and $d = 3$ for 3D problems). A solid is subjected to an applied traction ($\mathbf{t}$) on part of its boundary ($\partial\Omega_{\mathbf{t}}$), and an applied displacement field ($\mathbf{u}_0$) on the other part ($\Omega_{\mathbf{u}}$), such that $\partial\Omega_{\mathbf{t}} \cup \partial\Omega_{\mathbf{u}} = \partial\Omega$ and $\partial\Omega_{\mathbf{t}} \cap \partial\Omega_{\mathbf{u}} = \emptyset$. Based on the principle of virtual work, the weak form of the governing equation for elastodynamic problems is given by

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

where $\mathbf{u}$ and $\ddot{\mathbf{u}}$ are the displacement and acceleration vectors, and ρ_0 is the material density. Additionally, $\mathbf{v}$ is a virtual displacement field that vanishes on $\partial\Omega_{\mathbf{u}}$ and $\mathcal{K}_0$ is a subset of the space of admissible displacements ($\mathcal{K}$). To compute the stress tensor ($\boldsymbol{\sigma}$), the linear isotropic elasticity tensor ($\mathbb{C}$) is employed with the linearized strain tensor ($\boldsymbol{\epsilon}$), i.e., $\boldsymbol{\sigma}(\mathbf{u}) = \mathbb{C}\boldsymbol{\epsilon}(\mathbf{u})$ as usual.

Then, a solid (Ω) is discretized into generic elements (E), i.e., non-overlapping polygons or polyhedra (with planar faces only), which leads to a discretized solid (Ω_h). A generic polyhedron (P) consists of four or more polygonal faces (F), while a generic polygon (F) consists of three or more edges (e). Notice that whenever we have definitions that are independent of dimension (d), we use E to denote a generic element in the mesh. Additionally, polyhedral meshes with planar faces may be generated using a few approaches [34–36], while polyhedral elements with non-planar faces can be handled by, for example, a subtriangulation technique [21].

To consistently handle arbitrarily nonconvex elements, e.g., the ones shown in Fig. 1, in the following subsections, the global and local virtual displacement spaces are first defined in conjunction with local projection operators. Based on those definitions, the stiffness and mass matrices of an element are then constructed, which includes the consistency and the stability terms.

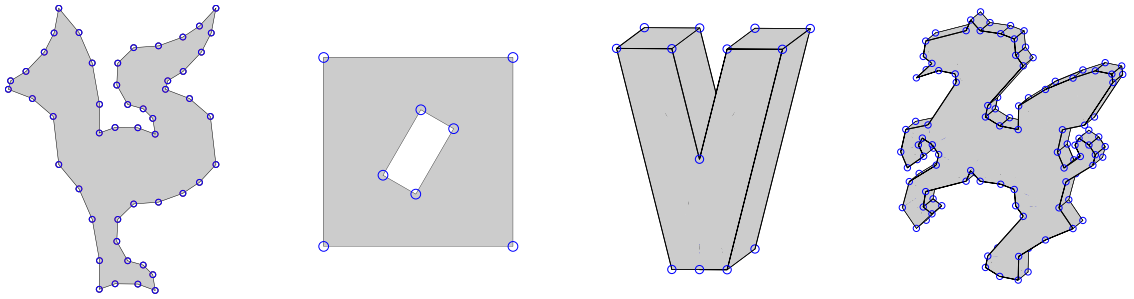


Fig. 1. Example of nonconvex polygonal and polyhedral elements: (a) bird polygon with $n_{nel} = 50$, where n_{nel} denotes the number of nodes per element, (b) matrix polygon where edges are non-simply connected, (c) "V" letter polyhedron with $n_{nel} = 16$, and (d) pegasus polyhedron with $n_{nel} = 132$.

2.1. Virtual element spaces in 2D and 3D

The global displacement space ($\mathcal{K}_h$) on Ω_h is defined as

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

The definition suggests that, within an element E, any admissible displacement field $\mathbf{v}_h$ in $\mathcal{K}_h$ belongs to a local virtual space $[\mathcal{V}_k(E)]^d$, which is assumed to contain all the polynomials of order k and whose definition is the focus of the remaining subsection. Following the technique proposed in [37], we first introduce the local virtual space $\mathcal{V}_k(F)$ in 2D and, building upon $\mathcal{V}_k(F)$, we then provide the definition of $\mathcal{V}_k(P)$ in 3D.

To define the 2D local virtual space, we first introduce a preliminary local virtual space within a polygon (F) as

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

where $\mathcal{P}_k(\cdot)$ is the space of polynomial functions of order k and Δ stands for the Laplacian operator. By definition, the preliminary space $\tilde{\mathcal{V}}_k(F)$ contains functions that have kth order variations on the boundary of F and kth order Laplacians in the interior of F. Because $\mathcal{P}_k(F) \subset \tilde{\mathcal{V}}_k(F)$, one can define a projection operator of functions in $\tilde{\mathcal{V}}_k(F)$ onto $\mathcal{P}_k(F)$, i.e., $\Pi_k^{\nabla} v_h \in \mathcal{P}_k(F)$, such that

$$\int_F \nabla\left(\Pi_k^{\nabla} v_h\right) \cdot \nabla p_k d\mathbf{x} = \int_F \nabla v_h \cdot \nabla p_k d\mathbf{x} \qquad \forall p_k \in \mathcal{P}_k(F) \tag{4}$$

and

$$\begin{aligned} &\sum_{\mathbf{x}_v \in F} \Pi_k^{\nabla} v_h(\mathbf{x}_v) = \sum_{\mathbf{x}_v \in F} v_h(\mathbf{x}_v) && \text{if } k = 1 \\ &\int_F \Pi_k^{\nabla} v_h d\mathbf{x} = \int_F v_h d\mathbf{x} && \text{if } k \geq 2 \end{aligned} \tag{5}$$

where p_k is a polynomial function of order k, and $\mathbf{x}_v$ is the position vector of vertex v of F. The projection $\Pi_k^{\nabla} v_h$ is uniquely determined using integration by parts of Eq. (4),

$$\int_F \nabla\left(\Pi_k^{\nabla} v_h\right) \cdot \nabla p_k d\mathbf{x} = \int_{\partial F} v_h \nabla p_k \cdot \mathbf{n}_e d\mathbf{x} - \int_F v_h \Delta p_k d\mathbf{x} \tag{6}$$

where $\mathbf{n}_e$ is an outward normal vector on the element boundary, and the following three pieces of information are used:

$$\text{values of } v_h(\mathbf{x}_v) \qquad \forall \mathbf{x}_v \in F \tag{7}$$

$$\text{values of } v_h\left(\mathbf{x}_e^i\right) \qquad \forall e \in \partial F,\, i = \{1, 2, \ldots, k-1\} \tag{8}$$

$$\text{moment of } v_h, \int_F v_h p_{k-2} d\mathbf{x} \qquad \forall p_{k-2} \in \mathcal{P}_{k-2}(F) \tag{9}$$

where $\mathbf{x}_e^i$ is the ith integration point of the Gauss–Lobatto rule on the edge. In fact, the first term on the right hand side of Eq. (6) is determined using $v_h\left(\mathbf{x}_e^i\right)$ with the Gauss–Lobatto rule [Eq. (8)], and the second term is evaluated using the moments of v_h [Eq. (9)]. We note that the detailed calculation procedures of the projection operator can be found in the literature [13,38]. Finally, the local virtual space $[\mathcal{V}_k(F)]$ is defined as

$$\mathcal{V}_k(F) = \left\{ v_h \in \tilde{\mathcal{V}}_k(F) : \int_E \left(\Pi_k^{\nabla} v_h - v_h\right) q d\mathbf{x} = 0 \quad \forall q \in (\mathcal{P}_k / \mathcal{P}_{k-2}) \right\} \tag{10}$$

where $\mathcal{P}_k/\mathcal{P}_{k-2}$ indicates the polynomial space $\mathcal{P}_k(F)$ that is L^2 orthogonal to $\mathcal{P}_{k-2}(F)$. With the definition in Eq. (10), one can show that the pieces of information in Eqs. (7)–(9) constitute a complete set of degrees of freedom (DOFs) of $\mathcal{V}_k(F)$ [37].

The local virtual space $\mathcal{V}_k(P)$ in 3D can be constructed by assuming that the functions on every face $F \in \partial P$ belong to the local space $\mathcal{V}_k(F)$ defined in Eq. (10). Similarly, we first introduce a preliminary local virtual space $\tilde{\mathcal{V}}_k(P)$ as

$$\tilde{\mathcal{V}}_k(P) = \left\{ v_h \in \mathcal{H}^1(P) : \Delta v_h \in \mathcal{P}_k(P) \text{ in } P, v_h|_F \in \mathcal{V}_k(F) \ \forall F \in \partial P \right\}. \tag{11}$$

Then, the projection operator Π_k^{∇} from $\tilde{\mathcal{V}}_k(P)$ on to $\mathcal{P}_k(P)$ is defined in the same way as in the 2D case [17], i.e., Eqs. (4)–(6), and can be uniquely determined using the following four pieces of information: the three pieces of information, which are the same as those for the 2D case, i.e., Eqs. (7)–(9), plus another piece of information given by

$$\text{moment of } v_h, \int_P v_h p_{k-2} d\mathbf{x} \qquad \forall p_{k-2} \in \mathcal{P}_{k-2}(P) \tag{12}$$

Finally, the local virtual space in 3D $\mathcal{V}_k(P)$ can be formally defined by Eq. (10) by replacing F with P, and one can show that the aforementioned four pieces of information [Eqs. (7)–(9) plus Eq. (12)] constitute a complete set of degrees of freedom (DOFs) of $\mathcal{V}_k(P)$ [37].

2.2. L^2 Projection operators

For the discretization of the weak form of the governing equation, associated with the bilinear form, two L^2 projection operators are utilized. From now on, we will not distinguish between F and P, and will use E instead to denote a generic element in both 2D and 3D. For an arbitrary function $v_h \in \mathcal{V}_k(E)$, the first projection operator, $\Pi_k^0 : \mathcal{V}_k(E) \to \mathcal{P}_k(E)$, is defined as

$$\int_E \left(\Pi_k^0 v_h\right) p_k d\mathbf{x} = \int_E v_h p_k d\mathbf{x} \quad \forall p_k \in \mathcal{P}_k(E) \tag{13}$$

The projection $\Pi_k^0 v_h$ can be exactly computed using the degrees of freedom of v_h for $k \leq 2$ because the projection operator of Π_k^0 is the same as the projection operator of Π_k^{∇} when the polynomial order is less than or equal to 2 [37]. For cases of $k > 2$, the projection operator Π_k^0 will be different from Π_k^{∇}, but can be exactly computed using the DOFs of v_h and $\Pi_k^{\nabla} v_h$ [37]. Next, for the second projection operator, the gradient of v_h is projected to $[\mathcal{P}_{k-1}(E)]^d$, i.e., $\Pi_{k-1}^0 : [\mathcal{V}_k(E)]^d \to [\mathcal{P}_{k-1}(E)]^d$, and then the projection $\Pi_{k-1}^0 \nabla v_h$ is defined as

$$\int_E \left(\Pi_{k-1}^0 \nabla v_h\right) \cdot \mathbf{p}_{k-1} d\mathbf{x} = \int_E \nabla v_h \cdot \mathbf{p}_{k-1} d\mathbf{x} \quad \forall \mathbf{p}_{k-1} \in [\mathcal{P}_{k-1}(E)]^d \tag{14}$$

where $\mathbf{p}_{k-1}$ is a vector of $k-1$ order polynomials, which consist of scaled monomial functions of order $k-1$. The projection $\Pi_{k-1}^0 \nabla v_h$ is evaluated by applying integration by parts, which results in

$$\int_E \left(\Pi_{k-1}^0 \nabla v_h\right) \cdot \mathbf{p}_{k-1} d\mathbf{x} = \int_{\partial E} v_h \mathbf{p}_{k-1} d\mathbf{x} - \int_E v_h \operatorname{div}(\mathbf{p}_{k-1}) d\mathbf{x} \tag{15}$$

The first and second terms on the right hand side of Eq. (15) are exactly computable using the set of DOFs and an integration rule for polynomials on E.

2.3. Mass and stiffness matrices

From the virtual kinetic energy term in Eq. (1), the local consistent mass matrix is first constructed. Assuming element E has n nodes, we write the consistent mass matrix of E as

$$\mathbf{M}_E = \overline{\mathbf{M}}_E \otimes \mathbf{I}_d \tag{16}$$

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

$$\mathbf{A}\otimes\mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1j}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{i1}\mathbf{B} & \cdots & a_{ij}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{ik\times jl} \tag{17}$$

Then, $\overline{\mathbf{M}}_E$ is expressed as

$$\overline{\mathbf{M}}_E = \int_E \rho_0 \boldsymbol{\phi}^T\boldsymbol{\phi} d\mathbf{x} = \int_E \rho_0 \left(\Pi_k^0\boldsymbol{\phi}\right)^T\left(\Pi_k^0\boldsymbol{\phi}\right) d\mathbf{x} + \int_E \rho_0 \left(\boldsymbol{\phi}-\Pi_k^0\boldsymbol{\phi}\right)^T\left(\boldsymbol{\phi}-\Pi_k^0\boldsymbol{\phi}\right) d\mathbf{x} \tag{18}$$

where $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_n]$ and $\Pi_k^0\boldsymbol{\phi} = [\Pi_k^0\phi_1, \ldots, \Pi_k^0\phi_n]$ are row vectors of the shape functions (ϕ_i) and their projections. Employing the VEM, we approximate Eq. (18) as

$$\overline{\mathbf{M}}_E \approx \overline{\mathbf{M}}_{E,c} + \overline{\mathbf{M}}_{E,s} \tag{19}$$

where $\overline{\mathbf{M}}_{E,c}$ and $\overline{\mathbf{M}}_{E,s}$ are approximations of the first and second terms on the right-hand-side of Eq. (18), respectively. The first term on the right hand side of Eq. (18) is the consistency term ($\overline{\mathbf{M}}_{E,c}$) of the element mass matrix where the (i, j)th component of $\overline{\mathbf{M}}_{E,c}$ can be evaluated exactly as:

$$\left[\overline{\mathbf{M}}_{E,c}\right]_{ij} = \int_E \rho_0 \left(\Pi_k^0\phi_i\right)\left(\Pi_k^0\phi_j\right) d\mathbf{x} = \sum_{\alpha=1}^{n_{p_k}}\sum_{\beta=1}^{n_{p_k}} \left[\left(\mathbf{P}_k^0\right)^T\right]_{i\alpha} \int_E m_\alpha m_\beta d\mathbf{x} \left[\mathbf{P}_k^0\right]_{\beta j} \tag{20}$$

where m_α are scaled monomial functions and n_{p_k} is the number of scaled monomials of degree k. Additionally, $\mathbf{P}_k^0$ is the matrix representation of the projection operator Π_k^0. Additionally, in this study, a linear polynomial space is used on the element boundary, i.e., $k = 1$, and thus the scaled monomials are

$$\begin{aligned} &m_1 = 1, m_2 = \frac{x-x_c}{h_E}, m_3 = \frac{y-y_c}{h_E} && \text{for } d = 2 \\ &m_1 = 1, m_2 = \frac{x-x_c}{h_E}, m_3 = \frac{y-y_c}{h_E}, m_4 = \frac{z-z_c}{h_E} && \text{for } d = 3 \end{aligned} \tag{21}$$

where (x_c, y_c, z_c) is the position of the centroid of element E, and h_E is a characteristic size of E. The second term on the right hand side of Eq. (18) is the stability term ($\overline{\mathbf{M}}_{E,s}$), which can be approximated as

$$\overline{\mathbf{M}}_{E,s} = \rho_0 |E| \left(\mathbf{I}_n - \mathbf{P}_k^0\right)^T\left(\mathbf{I}_n - \mathbf{P}_k^0\right) \tag{22}$$

in which $|E|$ is the element volume, and $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix. *Notice that a numerical integration of polynomials over an element is needed for the construction of the consistent mass matrix*. To exactly integrate polynomials over a nonconvex domain, Green's integral formula [39] is utilized for 2D, and thus no sub-triangulation within an element is needed in this study. For the 3D case, the sub-tetrahedralization within an element is used to integrate polynomial functions while more general integration rules, such as the one by Chin et al. [40], could also be adopted. However, *if one employs a lumped mass matrix with a row-sum technique [41], the numerical integration can be avoided for linear virtual elements in both 2D and 3D.*

Next, based on the virtual strain energy term in Eq. (1), we construct the local stiffness matrix. Similarly to the local consistent mass matrix case, the local stiffness matrix of an element E also consists of a consistency term ($\mathbf{K}_{E,c}$) and a stability term ($\mathbf{K}_{E,s}$), i.e., $\mathbf{K}_E \approx \mathbf{K}_{E,c} + \mathbf{K}_{E,s}$ [23]. The consistency term, $\mathbf{K}_{E,c}$ is

$$\mathbf{K}_{E,c} = \int_E \mathbf{B}^T\mathbf{C}\mathbf{B} d\mathbf{x} \tag{23}$$

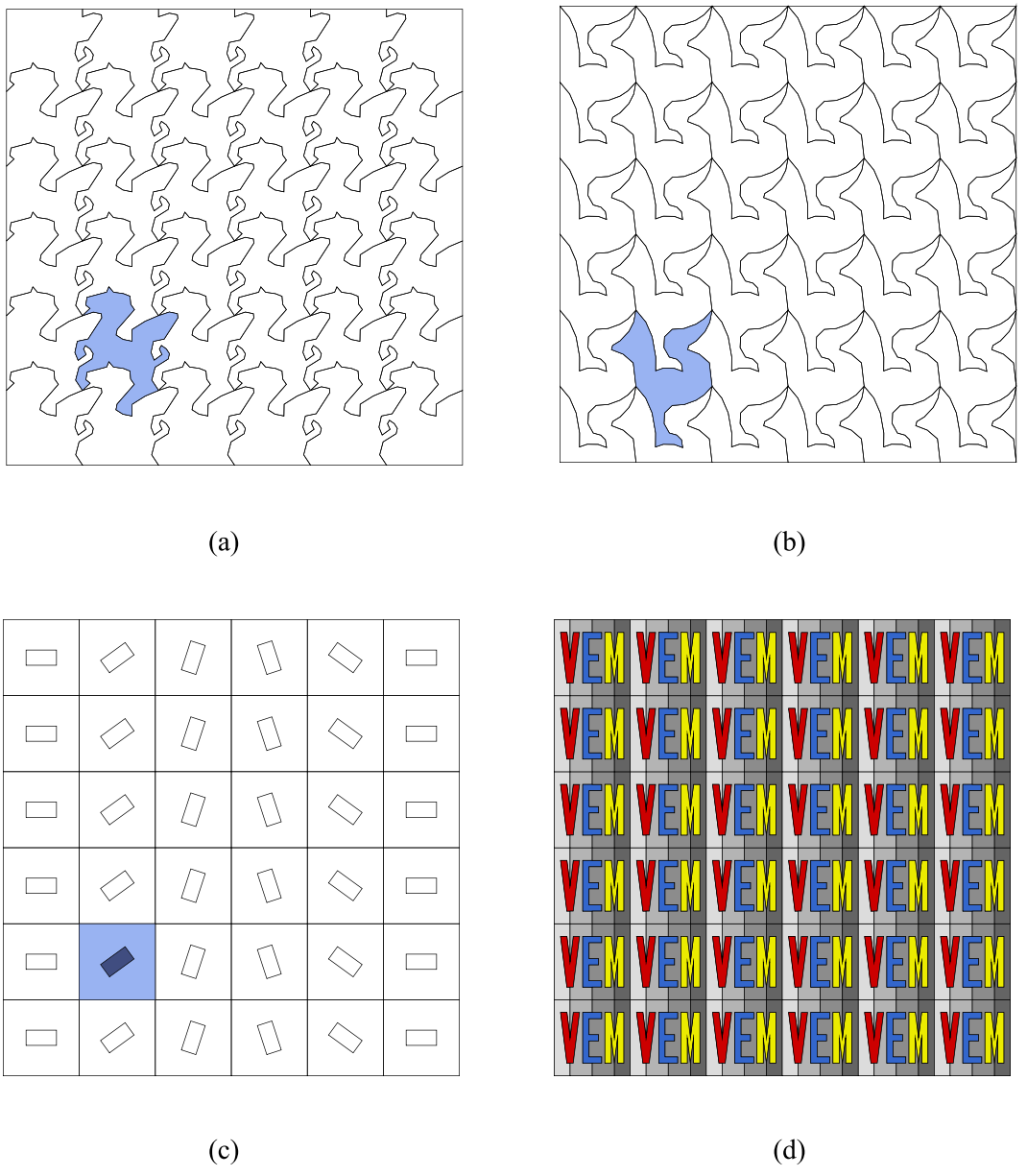


Fig. 2. Illustration of nonconvex meshes for the critical time step approximations: (a) pegasus mesh, (b) bird mesh, (c) particle mesh in which the reference particle is shaded dark blue and is surrounded by the matrix (shaded light blue), and (d) "VEM" mesh — there are 7 colors (or shades in B&W) which correspond to each element in the patch (7 elements per VEM patch) . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\mathbf{C}$ is a matrix representation of $\mathbb{C}$, and $\mathbf{B}$ provides the relation between displacement and strain, i.e.,

$$\mathbf{B} = \sum_{\alpha=1}^{d} \left[\Pi_{k-1}^{0} \nabla \boldsymbol{\phi} \right]_{\alpha} \otimes \overline{\mathbf{B}}_{\alpha} \tag{24}$$

in which the (i, j)th component of $\nabla \boldsymbol{\phi}$ is the partial derivative of the jth implicit shape function with respect to the ith coordinate, i.e., $[\nabla \boldsymbol{\phi}]_{ij} = \partial \phi_j / \partial x_i$. Then, $\Pi_{k-1}^{0} \nabla \boldsymbol{\phi}$ is the projection of each component of $\nabla \boldsymbol{\phi}$, and $\left[\Pi_{k-1}^{0} \nabla \boldsymbol{\phi} \right]_{\alpha}$

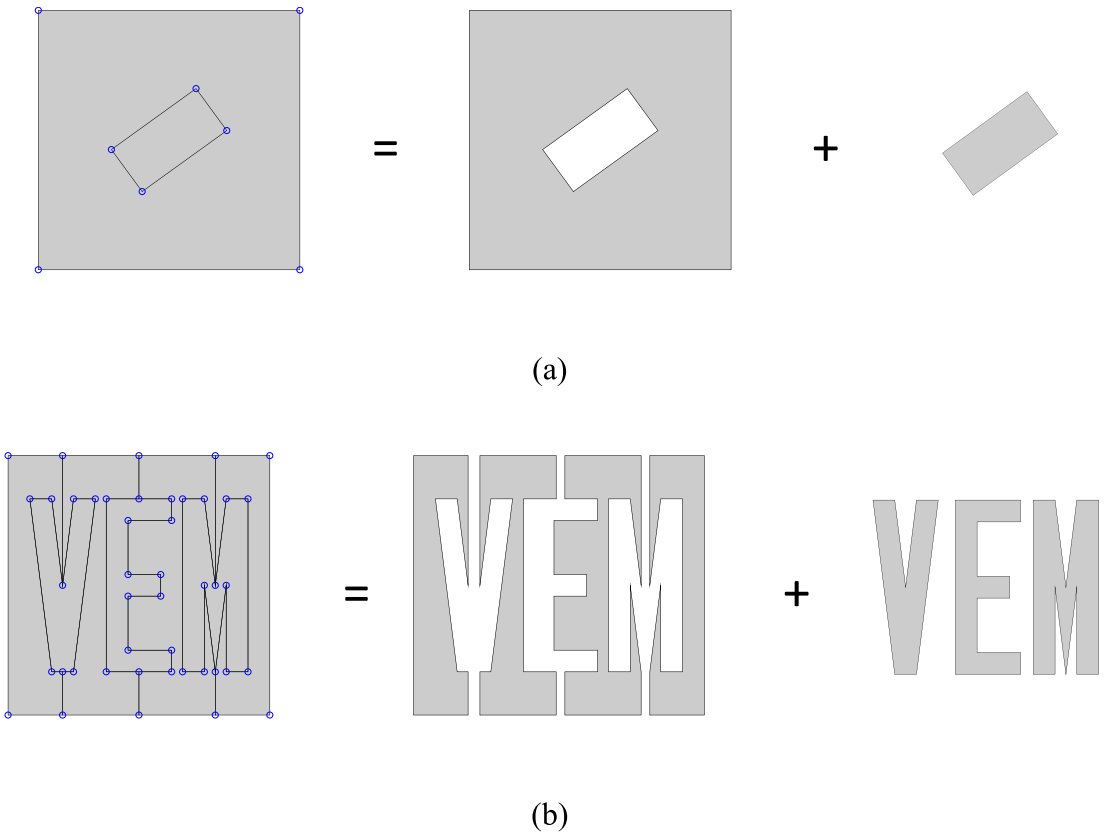


Fig. 3. Illustration of element patches: (a) Particle patch composed of 2 elements (i.e., matrix and particle elements), and (b) "VEM" patch displaying 7 elements (i.e., 4 background elements and 3 letter elements).

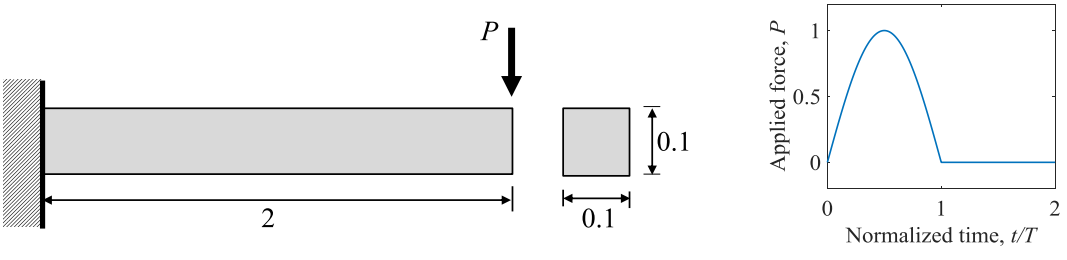


Fig. 4. Geometry and boundary conditions for the elastodynamic analysis.

denotes the αth row vector of $\Pi^0_{k-1}\nabla\boldsymbol{\phi}$, which corresponds to the gradient direction. Matrices $\overline{\mathbf{B}}_\alpha$ are

$$\overline{\mathbf{B}}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \overline{\mathbf{B}}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{for } d = 2$$

$$\overline{\mathbf{B}}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \overline{\mathbf{B}}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \overline{\mathbf{B}}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{for } d = 3 \tag{25}$$

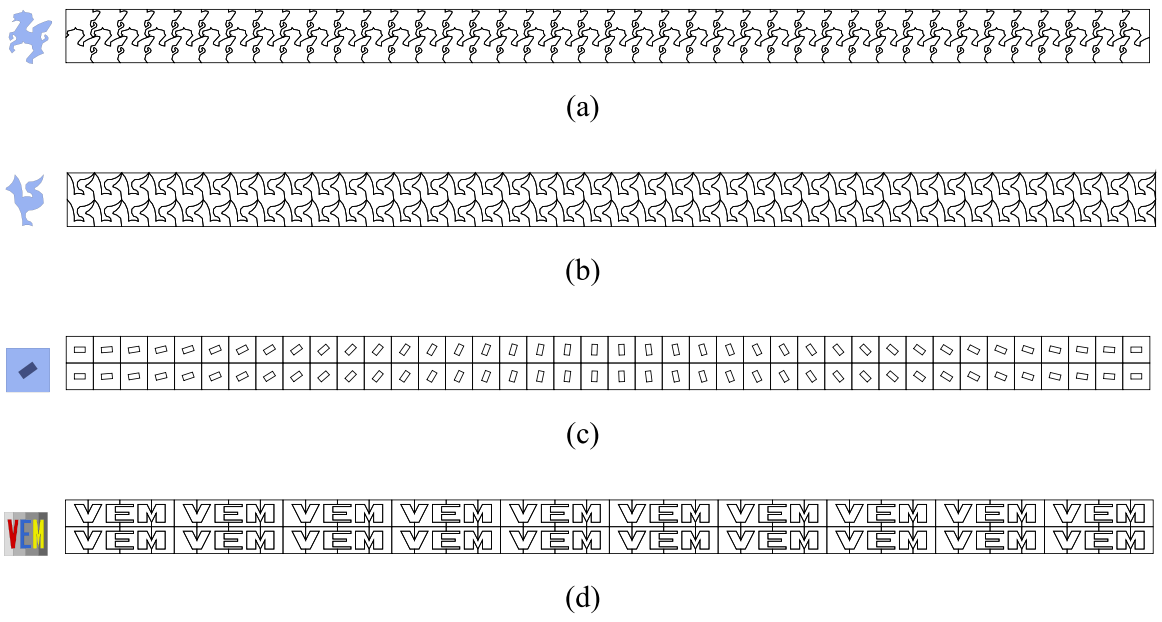


Fig. 5. Nonconvex meshes for the elastodynamic analysis: (a) pegasus mesh, (b) bird mesh, (c) particle mesh, and (d) "VEM" mesh.

The stability term of the element stiffness matrix is constructed as

$$\mathbf{K}_{E,s} = \overline{\mathbf{K}}_{E,s} \otimes \mathbf{I}_d \tag{26}$$

where $\overline{\mathbf{K}}_{E,s}$ is given as

$$\overline{\mathbf{K}}_{E,s} = \alpha_E \left(\mathbf{I}_d - \mathbf{P}_k^0\right)^T \left(\mathbf{I}_n - \mathbf{P}_k^0\right) \tag{27}$$

Previously, the stability term was scaled using a scalar value (α_E) such as the trace of the elasticity tensor [21,22]. Dassi and Mascotto [42] utilized the diagonal term of the consistency matrix to construct the stability term. In this study, to further improve the accuracy of computational results, a diagonal matrix ($\mathbf{\Lambda}$) is defined as

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

The components of the diagonal matrix are given as

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

where $n_{\mathbf{C}}$ is the size of the elasticity matrix, and α_0 is a non-dimensional parameter associated with a lower bound of the diagonal terms of $\mathbf{K}_{E,s}$. A lower bound of stiffness is specified using the elasticity matrix ($\mathbf{C}$) because stiffness obtained from $\left[\mathbf{K}_{\mathrm{E,c}}\right]_{ii}$ can be very small for a highly non-convex element with a larger number of nodes. In this study, α_0 is selected as $1/3$ and $1/9$ for 2D and 3D, respectively. We note that alternative selections of the stability term can be found in the literature [21,42].

3. Critical time steps

We discuss the estimation of the critical time step with respect to the efficiency of the resulting VEM framework. The critical time steps are estimated using the maximum eigenvalues of the global and local systems. Because solving an eigenvalue problem with a large system is computationally expensive, we explore the the use of the maximum eigenvalue of the local system to approximate the critical time step for explicit time integration with nonconvex meshes. A few examples illustrate the discussion.

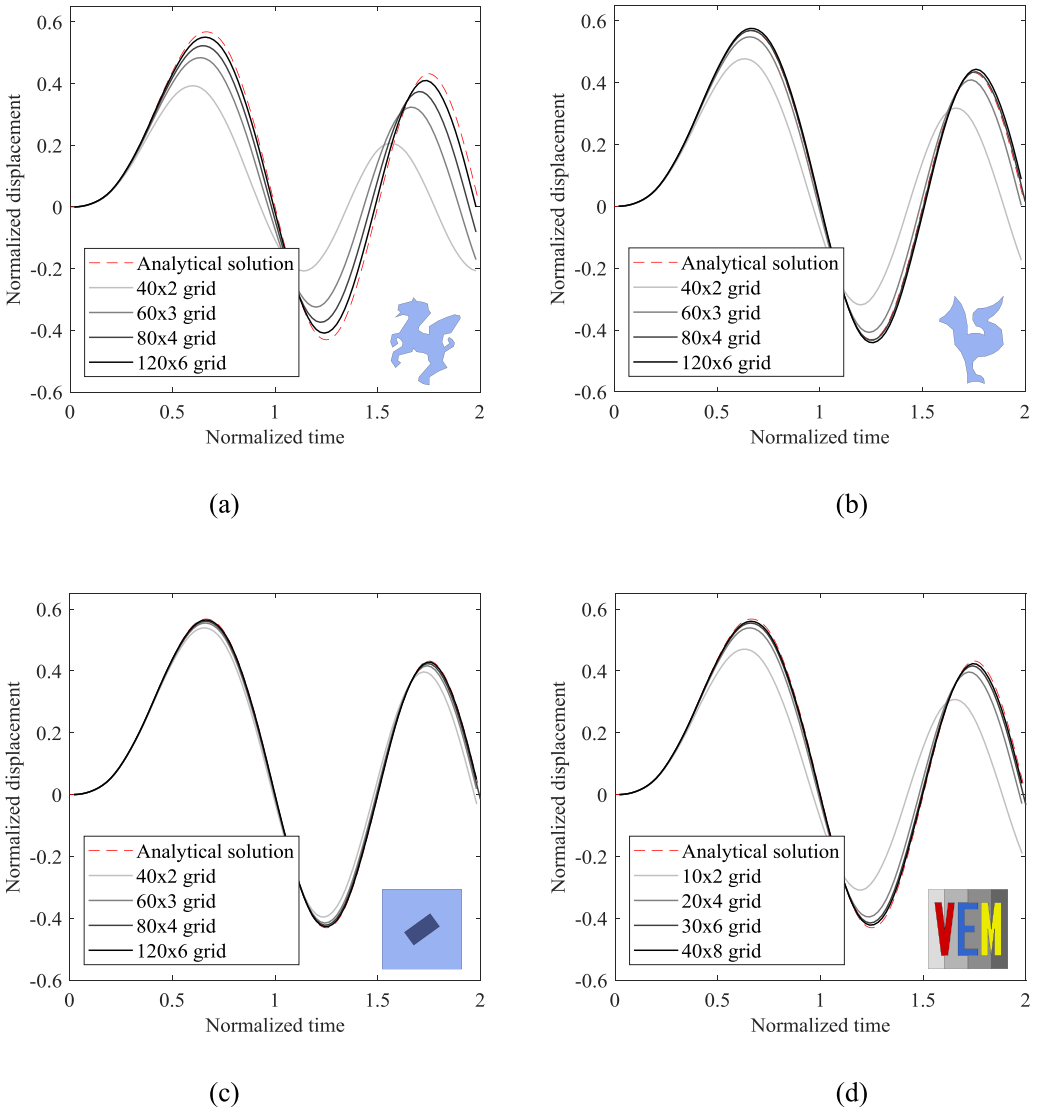


Fig. 6. Convergence for the nonconvex meshes using VEM with the diagonal matrix-based stabilization: (a) Pegasus mesh, (b) bird mesh, (c) particle mesh, and (d) "VEM" mesh.

3.1. Estimation of critical time steps

A critical time step is essential to achieve a stable solution in explicit time integration. For the estimation of a critical time step, the maximum eigenfrequency ($\omega_{\max}$) of a system is utilized, and thus a critical time step (Δt_{cr}) for linear elastic problems is given as [41],

$$\Delta t_{cr} = \frac{2}{\omega_{\max}} \tag{30}$$

For an accurate approximation of a critical time step, a global system of stiffness and mass matrices is used. Then, an eigenvalue problem of a global system is solved, i.e.,

$$\det\left(-\omega^2\mathbf{M} + \mathbf{K}\right) = 0 \tag{31}$$

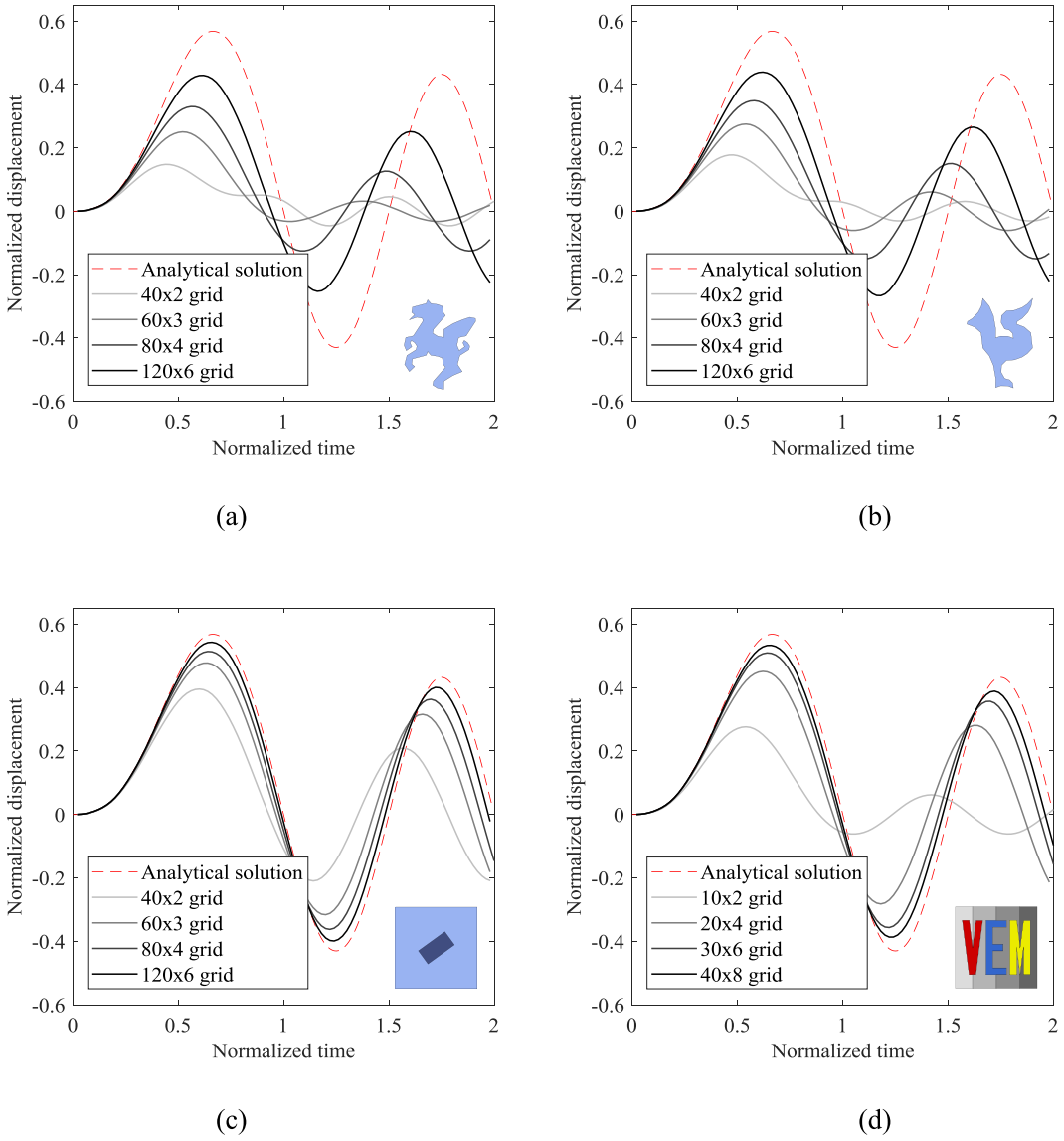


Fig. 7. Convergence for the nonconvex meshes using VEM with the scalar-based stabilization: (a) pegasus mesh, (b) bird mesh, (c) particle mesh, and (d) "VEM" mesh.

where $\mathbf{K}$ and $\mathbf{M}$ are the global stiffness and mass matrices, respectively, and ω is an eigenvalue of the system. Alternatively, because the calculation of the maximum eigenvalues for a large global system ($\omega^G_{\max}$) is computationally expensive, a critical time step is approximated by the maximum eigenvalue of a local system ($\omega^L_{\max}$) of an element stiffness matrix ($\mathbf{K}_E$) and an element mass matrix ($\mathbf{M}_E$), i.e., $\det\left(-\omega^2\mathbf{M}_E + \mathbf{K}_E\right) = 0$. Because the maximum eigenvalue of a local system is greater than the maximum eigenvalue of a global system, a critical time obtained from a local system is more conservative than a critical time step resulted from a global system [43,44].

3.2. Examples

For the estimation of the critical time steps, four types of nonconvex meshes are utilized, i.e., pegasus, bird, particle and "VEM" meshes, as shown in Fig. 2. Six by six repetitive patterns are placed within a unit square

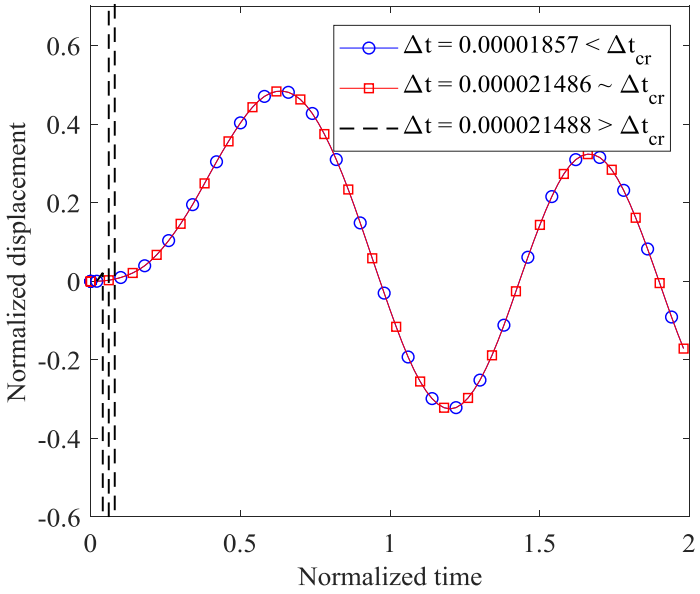


Fig. 8. Verification of the critical time step.

Table 1
Mesh statistics and estimated critical time steps for the pegasus, bird, particle and "VEM" meshes.

	Pegasus mesh	Bird mesh	Particle mesh	"VEM" mesh
Number of elements	36	48	72	252
Number of nodes	986	877	193	1399
$2/\omega^G_{\max}$	0.7823×10^{-3}	0.6705×10^{-3}	2.3557×10^{-3}	0.4329×10^{-3}
$2/\max(\omega^L_{\max})$	0.7333×10^{-3}	0.4051×10^{-3}	1.0104×10^{-3}	0.3936×10^{-3}

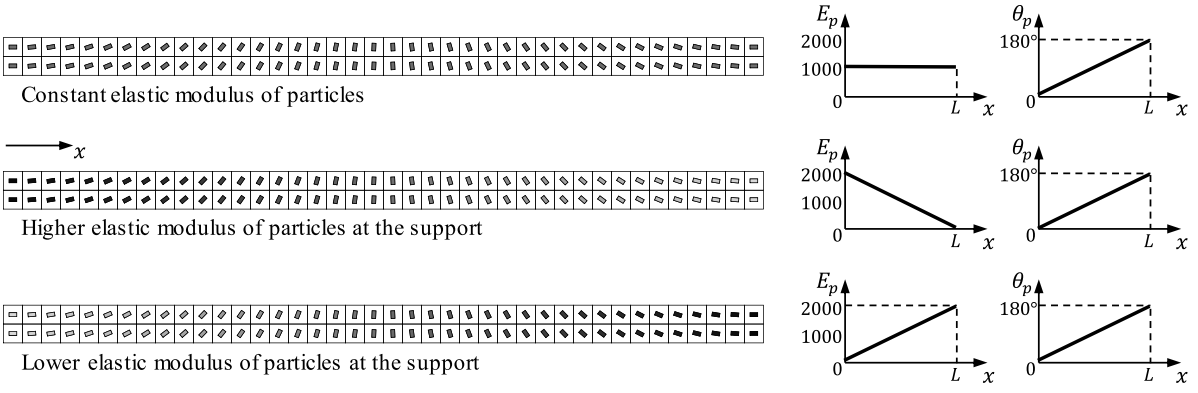
domain. The repetitive patterns of the pegasus and bird meshes are generated using Escher's tessellation[1] [45], and the patterns on the square boundary are cropped to fit with the square domain. For the particle mesh [Fig. 2(c)], a pattern consists of two elements, i.e., a square element with a rectangular hole and a rectangular element to fill the rectangular hole [see Fig. 3(a)]. Then, a rectangular hole and a rectangular element are rotated form 0^o to 180^o along the horizontal direction. For the "VEM" mesh [Fig. 2(d)], one patch consists of seven elements, i.e., three elements of letters and four elements of background, as shown in Fig. 3(b). The elastic modulus and the Poisson's ratio are assumed as 1000 and 0.25, respectively, with the density being unit. Additionally, the consistent mass matrix is diagonalized by scaling the diagonal terms of the consistent mass matrix, named as a special lumping technique [41], which provides a positive nodal mass for nonconvex elements.

The critical time steps are estimated using the maximum eigenvalues of the global and local systems, and the corresponding results are summarized in Table 1. The critical time steps obtained from the maximum eigenvalue of the local system are about 43∼94% of those obtained from the global system. Because solving an eigenvalue problem with a large system is computationally intensive, the use of the maximum eigenvalue of the local system can be a good candidate to approximate the critical time step of nonconvex meshes in the explicit time integration.

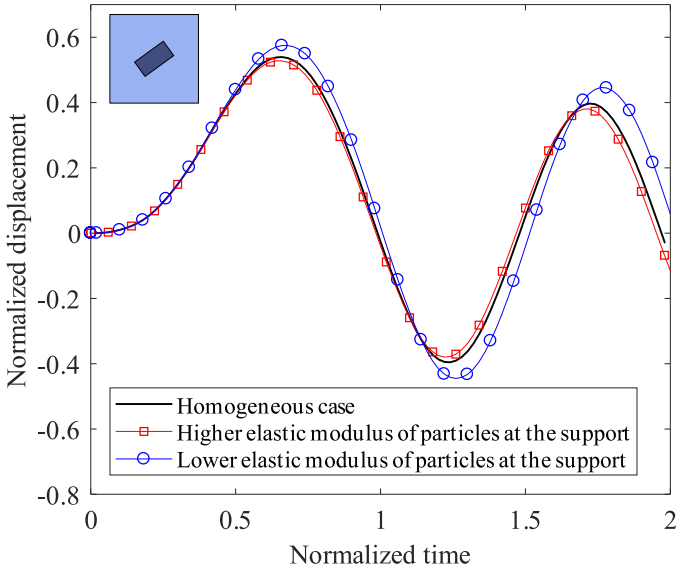
4. Elasto-dynamic examples

The use of nonconvex meshes for elastodynamics is verified using the presented VEM framework. For a representative elastodynamic example, the geometry and boundary conditions are illustrated in Fig. 4. The cantilever has a length (L) of 2 with the rectangular cross section of 0.1 by 0.1. A concentrated force, $P(t) = P_0 \sin(\pi t)$, is applied at the tip of the cantilever for $0 \leq t \leq 1$, and then the force is set to zero for $t > 1$. The amplitude of

[1] The reader is referred to the official website of the great artist M.C. Escher: http://www.mcescher.com/.



(a)



(b)

Fig. 9. (a) Variation of elastic modulus of particles, and (b) computational result with inclusions.

the sinusoidal force (P_0) is 1, and the elastic modulus (E) and Poisson's ratio (ν) are 1000 and 0.25, respectively, with the density of 0.016308. Additionally, the time step is estimated on the basis of the maximum eigenvalue of a local system, and the central difference method is utilized for the explicit time integration.

4.1. Two-dimensional example

Four types of nonconvex meshes are generated within the rectangular domain (e.g., Fig. 5). For each mesh type, four virtual element meshes are generated while increasing the number of the mesh patterns. For the pegasus, bird, and particle meshes, $40 \times 2, 60 \times 3, 80 \times 4$, and 120×6 patterns are generated, while $10 \times 2, 20 \times 4, 30 \times 6$, and 40×8 patterns are generated for the "VEM" meshes.

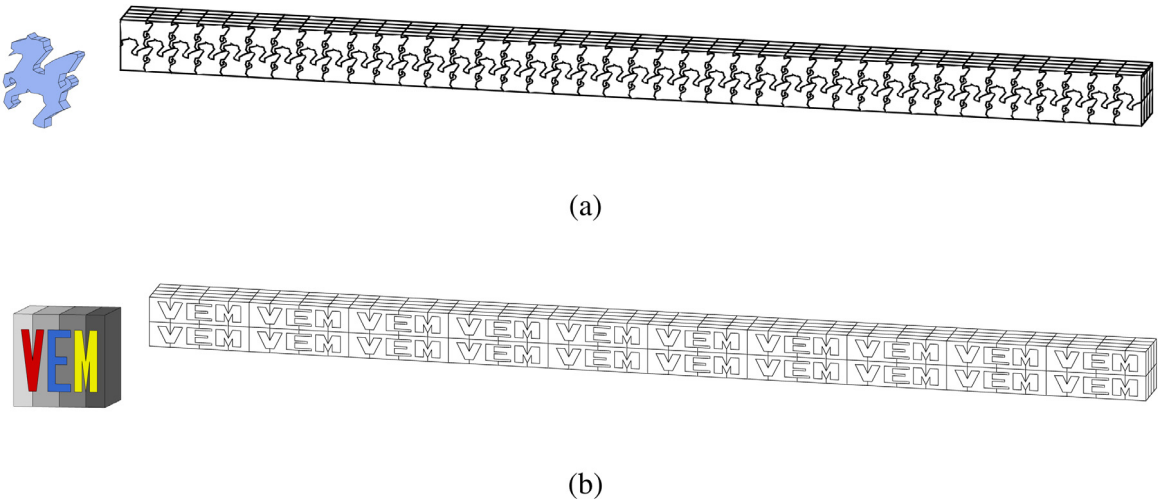


Fig. 10. Extruded nonconvex meshes for three-dimensional analysis: (a) pegasus mesh, and (b) "VEM" mesh.

The convergence of the computational results for the nonconvex meshes is demonstrated in Fig. 6. The displacement (u_p) and time (t) are nondimensionalized as $u_p EI/P_0 L^3$ and t/T, respectively, where I is the second moment of area and T is the natural period of the cantilever. For the pegasus, "VEM" and particle meshes [Fig. 6(a), (c) and (d)], the computational results show monotonic convergence to the analytical solution. However, the bird mesh [Fig. 6(b)] does not display monotonic convergence, which may result from an error cancellation in the coarse discretization. For instance, when the stability term is underestimated, the corresponding solution becomes flexible while the discrete solution is generally stiffer than analytical solution. Thus, an error cancellation is expected. One should note that the contribution of the stability term is relatively large for a large element size, while vanishing under mesh refinement. Therefore, if one further refines the bird mesh, then monotonic convergence is expected.

Additionally, the computational results with the scalar-based stabilization scheme [Eq. (27)] are illustrated in Fig. 7 for comparison. Fig. 7 demonstrates monotonic convergence of the computational results to the analytical solution for all meshes, while the differences between the computational results and the analytical solution are quite large, especially for the pegasus and bird meshes. In summary, the diagonal matrix-based stabilization scheme [Eqs. (28)–(29)] significantly improves the solution quality of the VEM for nonconvex meshes while a monotonic convergence is not guaranteed because of error cancellation.

The critical time step is computationally verified using the pegasus mesh with the 60×3 grid. The maximum eigenvalues of the global system and the local system are 93079.6 and 107696.4, respectively, which leads to the corresponding critical time step of 2.1487×10^{-5} and 1.8571×10^{-5}. Then, three computational simulations are performed with Δt of 1.857×10^{-5}, 2.1486×10^{-5} and 2.1488×10^{-5}. Fig. 8 shows that the computational result diverges when the time step is slightly larger than the critical time step obtained from the maximum eigenvalue of the global system, which confirms the validity of the critical time step estimation for nonconvex meshes using VEM.

Additionally, the cantilever example with inclusions is solved using the particle mesh of 10×2 patterns. The matrix material is described by elements with a hole, and the elastic modulus of the matrix is 1000. Each particle is represented by an element to fill the hole, and three cases of the elastic modulus of particles (E_p) are employed, i.e., one constant case, and two graded cases. For the constant case, the elastic modulus of particle is the same as the elastic modulus of the matrix, which leads to a homogeneous material. For the graded cases, the elastic modulus varies linearly from 0 at one end to 2000 at the other end, which results in the stronger or weaker elastic modulus of particles at the support, as shown in Fig. 9(a). Fig. 9(b) illustrates the computational results of the three cases. When the weaker elastic modulus of particles is introduced at the support region, the corresponding displacement increases, as expected. The displacement of the homogeneous case is in between the two graded cases in this study.

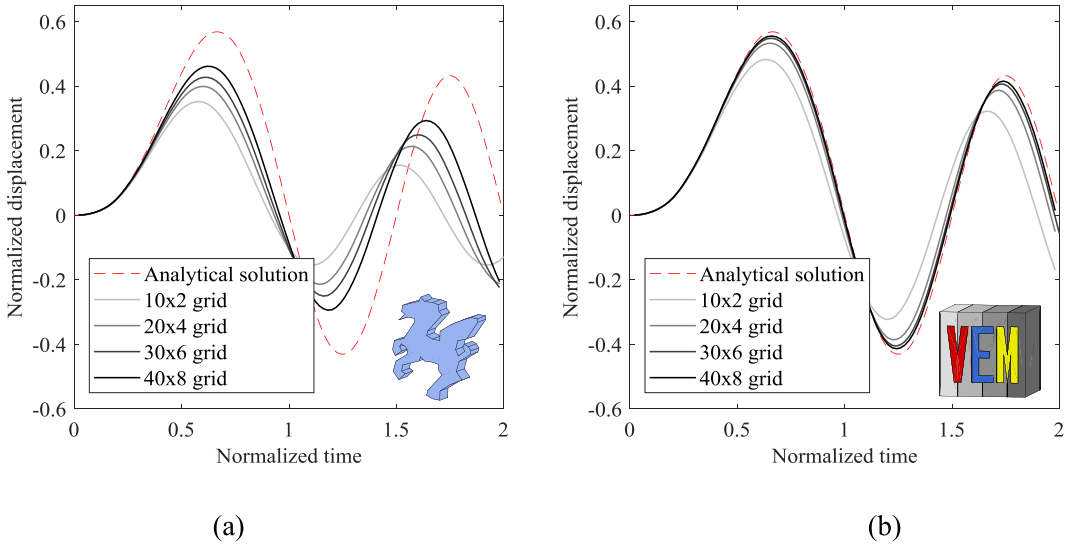


Fig. 11. Convergence of the three-dimensional elastodynamic analysis with the diagonal matrix-based stabilization: (a) pegasus mesh, and (b) "VEM" mesh.

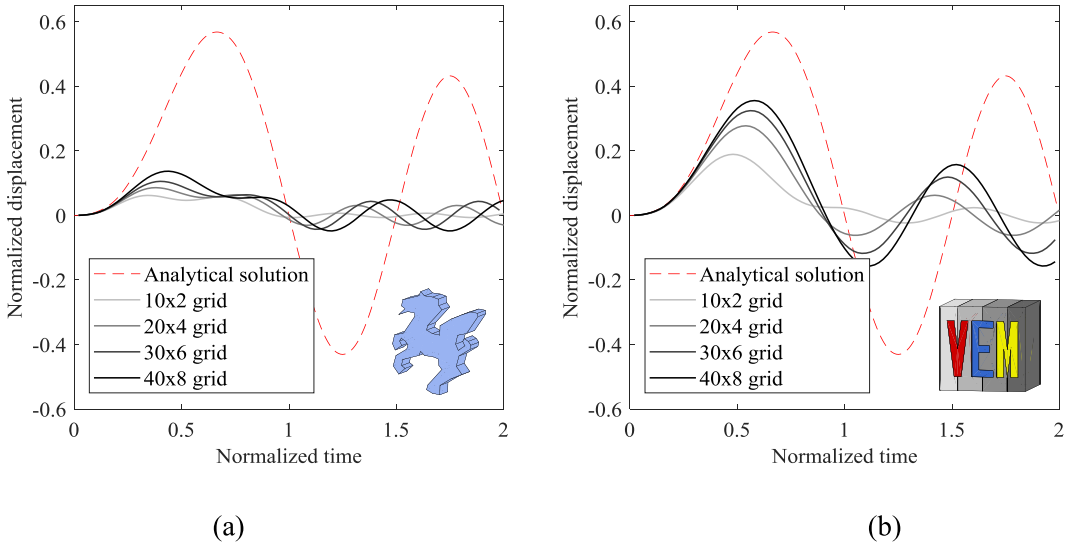


Fig. 12. Convergence of the three-dimensional elastodynamic analysis with the scalar-based stabilization: (a) pegasus mesh, and (b) "VEM" mesh.

4.2. Three-dimensional example

For the three-dimensional analysis, the pegasus and "VEM" meshes are extruded along the out-of-plane direction, as shown in Fig. 10. The number of layers along the out-of-plane direction is fixed as four, while the number of patterns increases as in the 2D case. The computational results with the diagonal matrix-based stabilization and the scalar-based stabilization are plotted in Figs. 11 and 12, respectively. Both results display convergence to the analytical solution, while the diagonal matrix-based stabilization provides more accurate results than the scalar-based stabilization, as in the 2D example.

5. Concluding remarks

This paper shows the relevance of arbitrarily-shaped polygons/polyhedra for the elastodynamic VEM with explicit time integration. A diagonal matrix-based stabilization scheme is proposed using the consistency part of the stiffness matrix and the constitutive matrix, which is shown to improve the computational results for both 2D and 3D cases. In the present VEM formulation, while no numerical integration is needed to construct the stiffness matrix for nonconvex meshes with first-order approximations (i.e. linear elements); numerical integration of polynomial functions is needed for the construction of the consistent mass matrix, which is performed using Green's integral formula. A lumped mass matrix is obtained by scaling the diagonal terms of the consistent mass matrix, which leads to positive nodal masses for nonconvex elements. To estimate the critical time steps for the VEM with nonconvex meshes, the maximum eigenvalues are utilized. For instance, the maximum eigenvalue of the global system results in an accurate approximation of the critical time step, while the maximum eigenvalue of the local system provides a conservative estimation of the critical time step and an efficient solution scheme.

Acknowledgments

KP acknowledges the supports from the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning, South Korea (Grant Number: 2018R1A2B6007054), and from the Korea Institute of Energy Technology Evaluation and Planning (KETEP) funded by the Ministry of Trade, Industry & Energy, South Korea (Grant Number: 20174030201480). HC and GHP acknowledge support from the US National Science Foundation (NSF) under grant #1624232 (formerly #1437535), and the support from the Raymond Allen Jones Chair at the Georgia Institute of Technology, United States. The authors thank Prof. Beirao da Veiga for suggesting investigation of the diagonal matrix-based stabilization.

References

[1] K. Hormann, N. Sukumar, Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics, CRC Press, New York, 2018.
[2] E.L. Wachspress, A Rational Finite Element Basis, Academic Press, New York, 1975.
[3] M. Floater, A. Gillette, N. Sukumar, Gradient bounds for Wachspress coordinates on polytopes, SIAM J. Numer. Anal. 52 (1) (2014) 515–532.
[4] M.S. Floater, G. Kós, M. Reimers, Mean value coordinates in 3D, Comput. Aided Geom. Design 22 (2005) 623–631.
[5] S. Martin, P. Kaufmann, M. Botsch, M. Wicke, M. Gross, Polyhedral finite elements using harmonic basis functions, Comput. Graph. Forum 27 (5) (2008) 1521–1529.
[6] J.E. Bishop, A displacement-based finite element formulation for general polyhedra using harmonic shape functions, Internat. J. Numer. Methods Engrg. 97 (2014) 1–31.
[7] N. Sukumar, Quadratic maximum-entropy serendipity shape functions for arbitrary planar polygons, Comput. Methods Appl. Mech. Engrg. 263 (2013) 27–41.
[8] K. Hormann, N. Sukumar, Maximum entropy coordinates for arbitrary polytopes, Comput. Graph. Forum 27 (5) (2008) 1513–1520.
[9] C. Talischi, G.H. Paulino, Addressing integration error for polygonal finite elements through polynomial projections: A patch test connection, Math. Models Methods Appl. Sci. 24 (8) (2014) 1701–1727.
[10] G. Manzini, A. Russo, N. Sukumar, New perspective on polygonal and polyhedral finite element method, Math. Models Methods Appl. Sci. 24 (2014) 1665–1699.
[11] C. Talischi, A. Pereira, I.F.M. Menezes, G.H. Paulino, Gradient correction for polygonal and polyhedral finite elements, Internat. J. Numer. Methods Engrg. 102 (2015) 728–747.
[12] L. Beirão Da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L.D. Marini, A. Russo, Basic principles of Virtual Element Methods, Math. Models Methods Appl. Sci. 23 (1) (2013) 199–214.
[13] L. Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo, The hitchhiker's guide to the virtual element method, Math. Models Methods Appl. Sci. 24 (8) (2014) 1541–1573.
[14] L. Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo, Serendipity Nodal VEM spaces, Comput. & Fluids 141 (2016) 2–12.
[15] L. Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo, Virtual Element Method for general second-order elliptic problems on polygonal meshes, Math. Models Methods Appl. Sci. 26 (4) (2016) 729–750.
[16] G. Vacca, L. Beirão da Veiga, Virtual element methods for parabolic problems on polygonal meshes, Numer. Methods Partial Differential Equations 31 (2015) 2110–2134.
[17] L. Beirão da Veiga, F. Dassi, A. Russo, High-order virtual element method on polyhedral meshes, Comput. Math. Appl. 74 (5) (2017) 1110–1122.
[18] G. Vacca, Virtual element methods for hyperbolic problems on polygonal meshes, Comput. Math. Appl. 74 (5) (2017) 882–898.
[19] D. Adak, E. Natarajan, S. Kumar, Virtual element method for semilinear hyperbolic problems on polygonal meshes, Int. J. Comput. Math. 96 (5) (2019) 971–991.

[20] E. Artioli, S. de Miranda, C. Lovadina, L. Patruno, A family of virtual element methods for plane elasticity problems based on the Hellinger–Reissner principle, Comput. Math. Appl. 340 (2018) 978–999.
[21] H. Chi, L. Beirão da Veiga, G.H. Paulino, Some basic formulations of the virtual element method (VEM) for finite deformations, Comput. Methods Appl. Mech. Engrg. 318 (2017) 148–192.
[22] H. Chi, L. Beirão da Veiga, G.H. Paulino, A simple and effective gradient recovery scheme and a posteriori error estimator for the Virtual Element Method (VEM), Comput. Methods Appl. Mech. Engrg. 347 (2019a) 21–58.
[23] A.L. Gain, C. Talischi, G.H. Paulino, On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes, Comput. Methods Appl. Mech. Eng. 282 (2014) 132–160.
[24] P. Wriggers, B.D. Reddy, W. Rust, B. Hudobivnik, Efficient virtual element formulations for compressible and incompressible finite deformations, Comput. Mech. 60 (2) (2017) 253–268.
[25] F. Aldakheel, B. Hudobivnik, A. Hussein, P. Wriggers, Phase-field modeling of brittle fracture using an efficient virtual element scheme, Comput. Methods Appl. Mech. Engrg. 341 (2018) 443–466.
[26] M.F. Benedetto, A. Caggiano, G. Etse, Virtual elements and zero thickness interface-based approach for fracture analysis of heterogeneous materials, Comput. Methods Appl. Mech. Engrg. 338 (2018) 41–67.
[27] P.F. Antonietti, M. Bruggi, S. Scacchi, M. Verani, On the virtual element method for topology optimization on polygonal meshes: A numerical study, Comput. Math. Appl. 74 (5) (2017) 1091–1109.
[28] A.L. Gain, G.H. Paulino, L.S. Duarte, I.F.M. Menezes, Topology optimization using polytopes, Comput. Methods Appl. Mech. Engrg. 293 (2015) 411–430.
[29] H. Chi, A. Pereira, I.F.M. Menezes, G.H. Paulino, Virtual element method (VEM)-based topology optimization: an integrated framework, Struct. Multidiscip. Optim. (2019) http://dx.doi.org/10.1007/s00158-019-02268-w.
[30] L. Beirão da Veiga, C. Lovadina, G. Vacca, Divergence free Virtual Elements for the Stokes problem on polygonal meshes, Math. Model. Numer. Anal. 51 (2017) 509–535.
[31] L. Beirão da Veiga, C. Lovadina, G. Vacca, Virtual elements for the Navier–Stokes problem on polygonal meshes, SIAM J. Numer. Anal. 56 (3) (2018) 1210–1242.
[32] G. Wang, F. Wang, L. Chen, Y. He, A divergence free weak virtual element method for the Stokes–Darcy problem on general meshes, Comput. Methods Appl. Mech. Engrg. 344 (2019) 998–1020.
[33] P. Wriggers, W.T. Rust, B.D. Reddy, A virtual element method for contact, Comput. Mech. 58 (6) (2016) 1039–1050.
[34] M.S. Ebeida, S.A. Mitchell, Uniform random voronoi meshes, in: Quadros W. R. (Ed.), Proceedings of the 20th International Meshing Roundtable, Springer, Berlin, Heidelberg, 2011.
[35] A. Abdelkader, C.L. Bajaj, M.S. Ebeida, A.H. Mahmoud, S.A. Mitchell, J.D. Owens, A.A. Rushdi, Vorocrust illustrated: theory and challenges, in: B. Speckmann, C.D. Tóth (Eds.), 34th International Symposium on Computational Geometry, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2018.
[36] A. Abdelkader, C.L. Bajaj, M.S. Ebeida, A.H. Mahmoud, S.A. Mitchell, J.D. Owens, A.A. Rushdi, VoroCrust: Voronoi meshing without clipping. arXiv:1902.08767 [cs.GR], 2019.
[37] B. Ahmad, A. Alsaedi, F. Brezzi, L.D. Marini, A. Russo, Equivalent projectors for virtual element methods, Comput. Math. Appl. 66 (3) (2013) 376–391.
[38] O.J. Sutton, The virtual element method in 50 lines of MATLAB, Numer. Algorithms 75 (2017) 1141–1159.
[39] A. Sommariva, M. Vianello, Product Gauss cubature over polygons based on Green's integration formula, BIT Numer. Math. 47 (2007) 441–453.
[40] E.B. Chin, J.B. Lasserre, N. Sukumar, Numerical integration of homogeneous functions on convex and nonconvex polygons and polyhedra, Comput. Mech. 56 (6) (2015) 967–981.
[41] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publications, New York, 2000.
[42] F. Dassi, L. Mascotto, Exploring high-order three dimensional virtual elements: Bases and stabilizations, Comput. Math. Appl. 75 (9) (2018) 3379–3401.
[43] J.I. Lin, An element eigenvalue theorem and its application for stable time steps, Comput. Methods Appl. Mech. Engrg. 73 (1989) 283–294.
[44] J.I. Lin, Bounds on eigenvalues of finite element systems, Internat. J. Numer. Methods Eng. 32 (5) (1991) 957–967.
[45] G.H. Paulino, A.L. Gain, Bridging art and engineering using Escher-based virtual elements, Struct. Multidiscip. Optim. 51 (2015) 867–883.