Computer Methods in Applied Mechanics and Engineering 435 (2025) 117683

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Space-time virtual element method for elastodynamics: Theory, applications, and code development

Bing-Bing Xu *, Philipp Junker, Peter Wriggers

Institute of Continuum Mechanics, Leibniz Universität Hannover, Hannover, Germany

ARTICLE INFO

Keywords:
Virtual element method
Dynamics
Space time

ABSTRACT

In this work, a space-time virtual element method is presented for the discretization of the two-dimensional elastodynamics in a space-time cylinder. The basic idea of the space-time scheme is to treat time as an additional dimension. This approach has been extensively applied to numerous initial value problems based on the finite element method and virtual element method. However, there has been no report on the application of the space-time virtual element method in elastodynamics. The variational formulation and space-time bilinear format are derived based on Hamilton's principle to achieve this. The discretization space is obtained by the virtual element method in space and the upwind finite element method in time. The space-time mesh is obtained as a tensor product of space and time meshes, then the element stiffness matrix for the elastodynamic problem can be obtained based on the Kronecker product directly. Some two-dimensional examples solved by the developed space-time virtual element method are given to demonstrate the accuracy and stabilization. MATLAB codes for the space-time virtual element and finite element methods can be downloaded from https://github.com/Qinxiaoye/VEM-spcae-time-dynamic.

1. Introduction

The virtual element method (VEM) [1,2] is an extension of the finite element method (FEM) to general polygonal or polyhedral elements. Different from the finite element method and some other polygonal finite element methods, the virtual element method avoids the explicit calculation of shape functions in polygonal elements by defining a projection from the virtual element space to the polynomial space. Given the flexibility of the virtual element method in terms of element shape and degree of freedom definition, the method has been applied in a variety of different applications. Current applications include linear elastic mechanical problems [3–8], plate and shell problems [9–11], hyperelastic materials at finite deformations [12–15], elastoplastic mechanics problems [16], contact problems [17–20] and finite elastoplastic deformations [21–23]. For details, see the latest books of VEM in engineering [24].

The virtual element method is also developed for initial–boundary value problems where the governing equation is parabolic or hyperbolic. For example, transient heat conduction problems, wave problems [25], and dynamics problems [26–29]. In the above-mentioned VEM literature, the time-dependent problems are solved by a virtual element discretization in space with a time-stepping scheme in time (explicit time difference, Newmark method, etc.). With suitable parameter choices, the Newmark method is unconditionally stable. However, it is not a strictly symplectic method, which can result in phase delay during long-term dynamic simulations. For nonlinear problems, iterations are required at each time step of the Newmark method, resulting in low computational efficiency.

* Corresponding author.
E-mail address: bingbing.xu@ikm.uni-hannover.de (B.-B. Xu).

https://doi.org/10.1016/j.cma.2024.117683
Received 4 October 2024; Received in revised form 10 December 2024; Accepted 15 December 2024

Taking Hamilton's principle of stationary of action as a starting point for the solution of dynamic problem has a long history. First formulations can be found in Argyris et al. [30]. Due to the limited computational power at that time, many different incremental solution scheme like finite difference and the Newmark method were developed. Nowadays with powerful computers aid parallel computation technique, the use of Hamilton's principle has gained a revival in order to construct a method for solving time-dependent problems in a space–time scheme. The basic idea of the space–time format is to discretize time as a dimension with the time being an additional space variable. Due to the advantages of the space–time scheme, including energy conservation, ease of parallel computing, and adaptive meshing, this method has seen significant development over the past three decades [31–33]. Several numerical approaches are available for discretizing the space–time scheme, with the finite element method (FEM) being the most common. Up to now, the space–time finite element methods have been applied to dynamic problems [34–36], acoustic wave problems [37], elastic wave problems [38], fluid dynamics problems [39,40], etc. Recent work based on Hamilton's principle, thermo-mechanically coupled models can be obtained by evaluating of the stationarity conditions for all thermodynamic state variables. It is also possible to apply the space–time format for elastic wave propagation, visco-elasticity, and elastoplastic problems using FEM [41,42]. In addition to the finite element method, the isogeometric method has advantages for specific applications [43,44].

Since the virtual element method can use arbitrary polygonal meshes (which is conducive for adaptive techniques), different formats of space–time VEM have been used for the transient heat conduction equation [45,46], the dissipative wave equation [47], and the elastic wave equation [48]. However, up to now, there has been no report on the application of space–time VEM in elasto-dynamics problems. This paper designs and analyzes a space–time virtual element method for solving the 2D dynamic problem. In the virtual element method, since the projection operator is calculated based on the bilinear format, a "structured" numerical approach can be used to avoid the definition of additional degrees of freedom in time direction. In the "structured" numerical approach, the space–time mesh is obtained as a tensor product of space and time meshes. For this $2+1$ dimensional problem, the VEM for "space" discretization and FEM for "time" discretization are selected. The advantage of this technique is that the computational methods used to construct the projection operator in VEM for static problems can be directly applied. Furthermore, integration is performed on the two-dimensional polygonal mesh and the one-dimensional time mesh, enabling the final stiffness matrix to be obtained directly through the Kronecker product. To ensure stability of the space–time VEM, a time-upwind scheme is introduced.

This paper is divided into the following parts. The variational formulation of the dynamic problems based on Hamilton's principle is reviewed in Section 2. Besides in this section, the space–time bilinear format and time-upwind stabilization are also expanded. Then, the space–time virtual element method for elasto-dynamics is discussed in Section 3. The projection operator, stabilization term and the formation of residual and stiffness matrix are discussed. For comparison, the finite element method for space–time dynamics is summarized in Section 4. Numerical examples for two-dimensional elastodynamics are presented in Section 5. The conclusion and discussion will be provided in Section 6. The code, developed in this work can be downloaded from https://github.com/Qinxiaoye/VEM-spcae-time-dynamic.

2. Variational formulation

2.1. Variational formulation based on Hamilton's principle

In this work, the linear elasto-dynamic problem is considered with the space–time technique. Given a Lipschitz-bounded domain $\Omega \in \mathbb{R}^n$, where $n = 1, 2, 3$ and the space–time cylinder

$$Q := \Omega \times I \tag{1}$$

where time interval $I = (0, T)$, with $t \in I$, Hamilton's action functional is then formulated as integration of the momentum vector $\boldsymbol{p} := \rho \nabla_t \boldsymbol{u}$ along the path $[\boldsymbol{u}_0, \boldsymbol{u}_T]$, i.e.,

$$\mathcal{H} := \int_\Omega \int_{\boldsymbol{u}_0}^{\boldsymbol{u}_T} \boldsymbol{p} \cdot \,\mathrm{d}\boldsymbol{u}\,\mathrm{d}V = \int_\Omega \int_0^T \rho \nabla_t \boldsymbol{u} \cdot \nabla_t \boldsymbol{u}\,\mathrm{d}t\,\mathrm{d}V = \int_I \int_\Omega \rho \|\nabla_t \boldsymbol{u}\|^2\,\mathrm{d}V\,\mathrm{d}t = \int_I 2\mathcal{K}\,\mathrm{d}t, \tag{2}$$

where the definition of the temporal derivative $\frac{\mathrm{d}}{\mathrm{d}t} =: \nabla_t$ is used. The kinetic energy is introduced as

$$\mathcal{K} = \int_\Omega \frac{1}{2} \rho \|\nabla_t \boldsymbol{u}\|^2\,\mathrm{d}V. \tag{3}$$

Making use of the balance of energy (1st law of thermodynamics) yields

$$\int_I \mathcal{K}\,\mathrm{d}t + \int_I \mathcal{G}\,\mathrm{d}t = 0, \tag{4}$$

with the total potential $\mathcal{G}$ defined as

$$\mathcal{G} := \int_\Omega \Psi\,\mathrm{d}V - \int_\Omega \bar{\boldsymbol{f}} \cdot \boldsymbol{u}\,\mathrm{d}V - \int_{\Gamma_\sigma} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}\,\mathrm{d}\Omega, \tag{5}$$

where $\bar{\boldsymbol{f}}$ is the volume force and $\bar{\boldsymbol{t}}$ is the external pressure on the Neumann boundary Γ_σ. Besides,

$$\Psi := \frac{\lambda}{2} (\mathrm{tr}\,(\varepsilon))^2 + \mu \varepsilon : \varepsilon = \frac{1}{2} \varepsilon : \mathbb{D} : \varepsilon \tag{6}$$

is the free energy density with the elasticity tensor $\mathbb{D} \in \mathbb{R}^{d \times d \times d \times d}$,

$$\mathbb{D}_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu \left(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} \right), \tag{7}$$

where λ and μ are the Lame constant and shear modulus. The strain tensor is defined as

$$\varepsilon = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right) =: \nabla^{\mathrm{sym}} \boldsymbol{u} = \varepsilon(\boldsymbol{u}). \tag{8}$$

Then Hamilton's action function is reformulated as

$$\mathcal{H} = \int_I (\mathcal{K} - \mathcal{G})\, \mathrm{d}t. \tag{9}$$

Let us introduce a mixed formulation by introducing the vector of velocity $\boldsymbol{v}$ which is treated as independent variable. For such a mixed formulation, the kinetic energy is reformulated as

$$\mathcal{K} = \int_\Omega \rho \boldsymbol{v} \cdot \nabla_t \boldsymbol{u} - \frac{1}{2}\rho \|\boldsymbol{v}\|^2 \,\mathrm{d}\Omega, \tag{10}$$

following [42]. It is obvious that this formulation reduces to the original kinetic energy as soon as $\boldsymbol{v} \equiv \nabla_t \boldsymbol{u}$.

Then, Hamilton's principle of stationary action requires

$$\mathcal{H} = \mathcal{H}\left[\boldsymbol{v}, \boldsymbol{u}, \nabla^{\mathrm{sym}} \boldsymbol{u}, \nabla_t \boldsymbol{u}\right] \to \underset{\boldsymbol{v},\boldsymbol{u}}{\mathrm{stat}} \Leftrightarrow \delta\mathcal{H} = \delta_{\boldsymbol{v}}\mathcal{H} + \delta_{\boldsymbol{u}}\mathcal{H} = 0 \quad \forall \delta\boldsymbol{v}, \delta\boldsymbol{u}. \tag{11}$$

Due to the independence of $\delta\boldsymbol{v}$ and $\delta\boldsymbol{u}$, the stationarity condition can lead to

$$\begin{cases} \delta_{\boldsymbol{v}}\mathcal{H} = 0 \quad \forall \delta\boldsymbol{v} \\ \delta_{\boldsymbol{u}}\mathcal{H} = 0 \quad \forall \delta\boldsymbol{u} \end{cases} \tag{12}$$

with

$$\begin{cases} \delta_{\boldsymbol{v}}\mathcal{H} = \displaystyle\int_I \int_\Omega \rho\left(\nabla_t \boldsymbol{u} - \boldsymbol{v}\right) \cdot \delta\boldsymbol{v}\, \mathrm{d}V\, \mathrm{d}t = 0 \quad \forall \delta\boldsymbol{v} \\ \delta_{\boldsymbol{u}}\mathcal{H} = \displaystyle\int_I \int_\Omega \rho \boldsymbol{v} \cdot \nabla_t \delta\boldsymbol{u}\, \mathrm{d}V\, \mathrm{d}t - \int_I \left(\int_\Omega \left(\boldsymbol{\sigma} : \varepsilon(\delta\boldsymbol{u}) - \bar{\boldsymbol{f}} \cdot \delta\boldsymbol{u}\right) \mathrm{d}V - \int_{\Gamma_\sigma} \boldsymbol{t} \cdot \delta\boldsymbol{u}\, \mathrm{d}\Gamma \right) \mathrm{d}t = 0 \quad \forall \delta\boldsymbol{u} \end{cases} \tag{13}$$

where the stress $\boldsymbol{\sigma} = \frac{\partial \Psi}{\partial \varepsilon} = \mathbb{D} : \boldsymbol{\varepsilon}$ is applied. Employing integration by parts in time for the first term in $(13)_2$ transforms the stationarity conditions to

$$\begin{cases} \displaystyle\int_I \int_\Omega -\rho \boldsymbol{v} \cdot \delta\boldsymbol{v}\, \mathrm{d}V\, \mathrm{d}t + \int_I \int_\Omega \rho \nabla_t \boldsymbol{u} \cdot \delta\boldsymbol{v}\, \mathrm{d}V\, \mathrm{d}t = 0 \quad \forall \delta\boldsymbol{v} \\ \displaystyle\int_I \int_\Omega \rho \nabla_t \boldsymbol{v} \cdot \delta\boldsymbol{u}\, \mathrm{d}V\, \mathrm{d}t + \int_I \int_\Omega \boldsymbol{\sigma} : \delta\nabla^{\mathrm{sym}} \boldsymbol{u}\, \mathrm{d}V\, \mathrm{d}t = \int_I \left(\int_\Omega \bar{\boldsymbol{f}} \cdot \delta\boldsymbol{u}\, \mathrm{d}V + \int_{\Gamma_\sigma} \boldsymbol{t} \cdot \delta\boldsymbol{u}\, \mathrm{d}\Gamma \right) \mathrm{d}t \end{cases} \tag{14}$$

where the property $\delta\boldsymbol{u}|_0 = \delta\boldsymbol{u}|_T \equiv \boldsymbol{0}$ is utilized. Note that the initial condition (position, velocity) can be applied directly without any further modification.

2.2. Space–time bilinear format and time-upwind stabilization

Due to the introduction of the velocity $\boldsymbol{v}$ as independent variable, Eq. (14) is a two-field weak-form equation. The above formulation can be discretized by the finite element method [35,36] or virtual element method. Based on Eq. (14) the bilinear form can be obtained as

$$A(\boldsymbol{v}, \boldsymbol{u}, \delta\boldsymbol{v}, \delta\boldsymbol{u}) = \begin{pmatrix} A^{\boldsymbol{vv}}(\delta\boldsymbol{v}, \boldsymbol{v}) + A^{\boldsymbol{vu}}(\delta\boldsymbol{v}, \boldsymbol{u}) \\ A^{\boldsymbol{uv}}(\delta\boldsymbol{u}, \boldsymbol{v}) + A^{\boldsymbol{uu}}(\delta\boldsymbol{u}, \boldsymbol{u}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ l(\delta\boldsymbol{u}) \end{pmatrix} \tag{15}$$

where

$$\begin{aligned} A^{\boldsymbol{vv}}(\delta\boldsymbol{v}, \boldsymbol{v}) &= \int_0^T \int_\Omega -\rho \delta\boldsymbol{v} \cdot \boldsymbol{v}\, \mathrm{d}\Omega\, \mathrm{d}t, \\ A^{\boldsymbol{vu}}(\delta\boldsymbol{v}, \boldsymbol{u}) &= \int_0^T \int_\Omega \rho \delta\boldsymbol{v} \cdot \nabla_t(\boldsymbol{u})\, \mathrm{d}\Omega\, \mathrm{d}t, \\ A^{\boldsymbol{uv}}(\delta\boldsymbol{u}, \boldsymbol{v}) &= \int_0^T \int_\Omega \rho \delta\boldsymbol{u} \cdot \nabla_t \boldsymbol{v}\, \mathrm{d}\Omega\, \mathrm{d}t, \\ A^{\boldsymbol{uu}}(\delta\boldsymbol{u}, \boldsymbol{u}) &= \int_0^T \int_\Omega \boldsymbol{\varepsilon}(\delta\boldsymbol{u}) : \mathbb{D} : \boldsymbol{\varepsilon}(\boldsymbol{u})\, \mathrm{d}\Omega\, \mathrm{d}t, \end{aligned} \tag{16}$$

and

$$l(\delta\boldsymbol{u}) = \int_0^T \left(\int_\Omega \bar{\boldsymbol{f}} \cdot \delta\boldsymbol{u}\, \mathrm{d}\Omega + \int_{\Gamma_\sigma} \boldsymbol{t} \cdot \delta\boldsymbol{u}\, \mathrm{d}\Gamma \right) \mathrm{d}t. \tag{17}$$

Using the formulation based on Eq. (15) results for some applications to an unstable response. Thus, to ensure stability of the numerical solution, the time-upwind scheme introduced in [36] should be applied. This scheme replace the test functions $\delta\boldsymbol{u}$ and

$\delta\boldsymbol{v}$ by $\delta\boldsymbol{u}+\theta h\nabla_t(\delta\boldsymbol{u})$, $\delta\boldsymbol{v}+\theta h\nabla_t(\delta\boldsymbol{v})$, where θ is a positive constant and h is the element length in time direction. The discrete scheme in Eq. (15) is now reformulated as

$$\begin{aligned}
A^{\boldsymbol{vv}}(\delta\boldsymbol{v},\boldsymbol{v}) &= \int_0^T\int_\Omega -\rho\left(\delta\boldsymbol{v}+\theta h\nabla_t(\delta\boldsymbol{v})\right)\cdot\boldsymbol{v}\,\mathrm{d}\Omega\,\mathrm{d}t,\\
A^{\boldsymbol{vu}}(\delta\boldsymbol{v},\boldsymbol{u}) &= \int_0^T\int_\Omega \rho\left(\delta\boldsymbol{v}+\theta h\nabla_t(\delta\boldsymbol{v})\right)\cdot\nabla_t(\boldsymbol{u})\,\mathrm{d}\Omega\,\mathrm{d}t,\\
A^{\boldsymbol{uv}}(\delta\boldsymbol{u},\boldsymbol{v}) &= \int_0^T\int_\Omega \rho\left(\delta\boldsymbol{u}+\theta h\nabla_t(\delta\boldsymbol{u})\right)\cdot\nabla_t\boldsymbol{v}\,\mathrm{d}\Omega\,\mathrm{d}t,\\
A^{\boldsymbol{uu}}(\delta\boldsymbol{u},\boldsymbol{u}) &= \int_0^T\int_\Omega \boldsymbol{\varepsilon}\left(\delta\boldsymbol{u}+\theta h\nabla_t(\delta\boldsymbol{u})\right):\mathbb{D}:\boldsymbol{\varepsilon}(\boldsymbol{u})\,\mathrm{d}\Omega\,\mathrm{d}t,
\end{aligned}\tag{18}$$

and

$$l(\delta\boldsymbol{u}) = \int_0^T\left(\int_\Omega \bar{\boldsymbol{f}}\cdot\left(\delta\boldsymbol{u}+\theta h\nabla_t(\delta\boldsymbol{u})\right)\,\mathrm{d}\Omega+\int_{\Gamma_\sigma}\boldsymbol{t}\cdot\left(\delta\boldsymbol{u}+\theta h\nabla_t(\delta\boldsymbol{u})\right)\,\mathrm{d}\Gamma\right)\mathrm{d}t.\tag{19}$$

The selection of the parameter θ will be discussed in Section 5.

The virtual element method is used for discretization in the spatial dimension and the Lagrange interpolation (the interpolation function in the finite element method) is selected in the time dimension. Of course, the finite element method can also be used for the discretization in the spatial dimension as Section 4. The basic principle of the virtual element method will be introduced in the next.

3. Virtual element method for space–time dynamics

In this work, two-dimensional ($d=2$) elastodynamic problems under small strain assumption is considered. Compared with the finite element method, very arbitrary mesh division in the space domain is acceptable in VEM. Besides, since its shape function is projected into the polynomial space, only a single point integration (for $k=1$) in the space domain is required. These characteristics lead to great advantages in solving this space–time dynamics using VEM. Next, the basic analysis process in detail for Eqs. (14), (15) and (18) is given.

3.1. Mesh assumptions

In this work, the tensor-product-in-time set $\{\mathcal{T}_h\}$ is selected for the space–time cylinder. As shown in Fig. 1, the space domain Ω is divided into a lot of non-overlapping polygons $\{\mathcal{T}_h^{\boldsymbol{x}}\}$. A polygonal element is defined by the parameters area $|E|$, barycenter $\boldsymbol{x}_E=(x_E,y_E)^T$, and diameter h_E. The time interval $[0,T]$ is divided into N_T subintervals $\mathcal{K}_{t_n}:=(t_n,t_{n+1})$ with knots $0=t_0<t_1<\cdots<t_{N_T}=T$. Each element $\mathcal{K}$ in $\{\mathcal{T}_h\}$ can be written as $\mathcal{K}_{\boldsymbol{x}}\times\mathcal{K}_t$, where $\mathcal{K}_{\boldsymbol{x}}$ is the two-dimensional mesh in $\{\mathcal{T}_h^{\boldsymbol{x}}\}$.

3.2. Projection operators

The discrete form of the space domain Ω is focused firstly. To simplify and avoid misunderstanding, E is selected instead of $\mathcal{K}_{\boldsymbol{x}}$ for any 2D element in $\{\mathcal{T}_h^{\boldsymbol{x}}\}$.

For the use of the virtual element method in the space domain Ω, the definition of different projection operators in the virtual element method should be introduced firs. For the displacement vector $\boldsymbol{u}$ and velocity $\boldsymbol{v}$ used in Eq. (18), the components of the vector separately (similar to the finite element method) can be approximated, so the following $\mathcal{H}^1$ projection operator $\Pi^{\nabla}_{k,E}:\mathcal{H}^1\to\mathcal{P}_k(E)$ will be introduced as

$$a^E\left(w-\Pi^{\nabla}_{k,E}w,q_k\right)=0\quad q_k\in\mathcal{P}_k(E)\tag{20}$$

where k is the order of VEM (in this work the order is selected as $k=1$), w is any scalar, $a^E:\mathcal{H}^1\times\mathcal{H}^1\to\mathcal{R}$ is the local counterpart of the bilinear form $a(\bullet,\bullet)$, $a^E(p,q)=\int_E\nabla p\cdot\nabla q\,\mathrm{d}\Omega$, $\forall p,q\in\mathcal{H}^1$. To solve the above projection operator, additional conditions should be added to fix the constant in the definition (20)

$$\begin{cases}\dfrac{1}{N_E^V}\displaystyle\sum_{i=1}^{N_E^V}\Pi^{\nabla}_{k,E}w-w=0, & \text{for}\quad k=1,\\[2ex] \dfrac{1}{|E|}\left(\Pi^{\nabla}_{k,E}w-w\right)\mathrm{d}\Omega=0, & \text{for}\quad k>1,\end{cases}\tag{21}$$

where N_E^V is the number of vertices. Then, the actual definition of the virtual element space can be given by

$$\mathcal{V}_h^E:=\left\{v\in\mathcal{H}^1:w|_{\partial E}\in\mathcal{B}(\partial E),\Delta w\in\left(\mathcal{P}_{k-2}(E)\right)^3\right\},\tag{22}$$

where $\mathcal{B}(\partial E)$ is the boundary element space

$$\mathcal{B}(\partial E):=\left\{w\in C^0(\partial E):w_F\in\mathcal{P}_F,\forall F\in\partial E\right\}.\tag{23}$$

The degrees of freedom (DOFs) for the function space are selected as

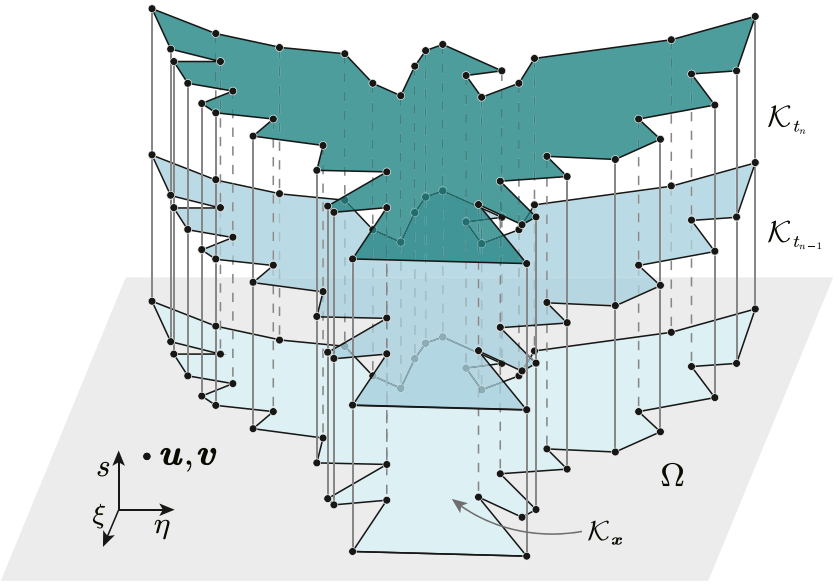

Fig. 1. First-order virtual element tensor-product-in-time meshes.

- the values of w_h at all N_E^V vertices of the polygon element E;
- for $k > 1$, the values of u_h at $k-1$ Gauss–Lobatto quadrature points on each edge $e \in \partial E$;
- for $k > 1$, the moments up to order $k-2$

$$\frac{1}{|E|}\int_E w_h q_\alpha \,\mathrm{d}\Omega, \quad \forall q_\alpha \in \mathcal{P}_{k-2}(E).$$

Expanding the variable v_h and its projection in different spaces, yields

$$w_h = \boldsymbol{\phi}^T \hat{\boldsymbol{w}}_h, \quad \Pi_{k,E}^{\nabla} w_h = \boldsymbol{m}_k^T \boldsymbol{\Pi}_{k,E*}^{\nabla} \hat{\boldsymbol{w}}_h = \boldsymbol{\phi}^T \boldsymbol{\Pi}_{k,E}^{\nabla} \hat{\boldsymbol{w}}_h, \tag{24}$$

where $\boldsymbol{\phi}$ and $\boldsymbol{m}_k$ are the basic functions for VEM space $\mathcal{V}_h^E$ and polynomial space $\mathcal{P}_k$, respectively; $\boldsymbol{\Pi}_{k,E*}$ is the matrix formulation of the Ritz projection operator, $\hat{\square}$ represents the nodal value. Substituting Eq. (24) into Eq. (21) and considering the degrees of freedom mentioned above, the $\mathcal{H}^1$ projection operator can be computed lastly.

By observing the first three formulas of Eq. (18), another projection named L^2 orthogonal projection operator is also needed. The projection $\Pi_{k,E}^0 : L^2(E) \to \mathcal{P}_k$ is defined by

$$\left(w - \Pi_{k,E}^0 w, q_k\right)_E = 0, \quad \forall q_k \in \mathcal{P}_k(E). \tag{25}$$

The same degrees of freedom (DOFs) in $\mathcal{V}_h^E$ can be selected. Considering the enhanced VE space in [49], the L^2 orthogonal projection operators can be calculated. It can be checked that $\Pi_{k,E}^{\nabla} = \Pi_{k,E}^0$ for $k \leqslant 2$. In this paper, the case $k = 1$ is considered, so it is not necessary to calculate the L^2 orthogonal projection operators additionally.

3.3. Discrete bilinear forms in space domain

Since the tensor-product-in-time meshes $\mathcal{K} = \mathcal{K}_{\boldsymbol{x}} \times \mathcal{K}_t$ (virtual element in the space domain Ω and Lagrange interpolation in the time domain) is adopted, the bilinear format $a^E(\boldsymbol{p}_h, \boldsymbol{q}_h)$ of the elements in the space domain Ω can be calculated firstly, and then integrate over time to calculate the last bilinear format $A^E(\boldsymbol{p}_h, \boldsymbol{q}_h)$. In this part, the integral calculation in the space domain is considered.

First, for the general form of the scalar field equation, the approximations a_h^E, m_h^E are computable based on the projections defined above

$$a_h^E(p_h, q_h) := a^E(\Pi_{k,E}^{\nabla} p_h, \Pi_{k,E}^{\nabla} q_h) + S^E\left(\left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{\nabla}\right)\hat{\boldsymbol{p}}_h, \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{\nabla}\right)\hat{\boldsymbol{q}}_h\right), \tag{26}$$

$$m_h^E(p_h, q_h) := m^E(\Pi_{k,E}^0 p_h, \Pi_{k,E}^0 q_h) + R^E\left(\left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^0\right)\hat{\boldsymbol{p}}_h, \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^0\right)\hat{\boldsymbol{q}}_h\right), \tag{27}$$

where S^E and R^E are the stabilization bilinear forms, $\boldsymbol{I}$ is the identity matrix. It is easy to see that our bilinear format requires additional stabilization terms. Based on the above definition, the following useful integrations can be calculated as:

$$\begin{aligned} a_h^E(p,q) &= \int_{\mathcal{K}_x} \left(\nabla \Pi_{k,E}^{\nabla} p_h\right) \cdot \left(\nabla \Pi_{k,E}^{\nabla} q_h\right) \mathrm{d}\Omega + S^E\left(\left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{\nabla}\right)\hat{\boldsymbol{p}}_h, \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{\nabla}\right)\hat{\boldsymbol{q}}_h\right) \\ &= \hat{\boldsymbol{p}}_h^T \left[\left(\boldsymbol{\Pi}_{k,E*}^{\nabla}\right)^T \int_{\mathcal{K}_x} \nabla \boldsymbol{m}_k \nabla \boldsymbol{m}_k^T \,\mathrm{d}\Omega \boldsymbol{\Pi}_{k,E*}^{\nabla} + \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{\nabla}\right)^T \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{\nabla}\right)\right] \hat{\boldsymbol{q}}_h \\ &= \hat{\boldsymbol{p}}_h^T \left(\boldsymbol{K}_{\nabla}^c + \boldsymbol{K}_{\nabla}^s\right) \hat{\boldsymbol{q}}_h = \hat{\boldsymbol{p}}_h^T \boldsymbol{K}_{\nabla} \hat{\boldsymbol{q}}_h, \end{aligned} \tag{28}$$

$$\begin{aligned} m_h^E(p,q) &= \int_{\mathcal{K}_x} \left(\Pi_{k,E}^{0} p_h\right) \cdot \left(\Pi_{k,E}^{0} q_h\right) \mathrm{d}\Omega + R^E\left(\left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{0}\right)\hat{\boldsymbol{p}}_h, \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{0}\right)\hat{\boldsymbol{q}}_h\right) \\ &= \hat{\boldsymbol{p}}_h^T \left[\left(\boldsymbol{\Pi}_{k,E*}^{0}\right)^T \int_{\mathcal{K}_x} \boldsymbol{m}_k \boldsymbol{m}_k^T \,\mathrm{d}\Omega \boldsymbol{\Pi}_{k,E*}^{0} + |E| \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{0}\right)^T \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{0}\right)\right] \hat{\boldsymbol{q}}_h \\ &= \hat{\boldsymbol{p}}_h^T \left(\boldsymbol{K}_0^c + \boldsymbol{K}_0^s\right) \hat{\boldsymbol{q}}_h = \hat{\boldsymbol{p}}_h^T \boldsymbol{K}_0 \hat{\boldsymbol{q}}_h, \end{aligned} \tag{29}$$

where

$$\boldsymbol{K}_0^c := \left(\boldsymbol{\Pi}_{k,E*}^{0}\right)^T \int_{\mathcal{K}_x} \boldsymbol{m}_k \boldsymbol{m}_k^T \,\mathrm{d}\Omega \boldsymbol{\Pi}_{k,E*}^{0}, \quad \boldsymbol{K}_0^s := |E| \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{0}\right)^T \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{0}\right), \quad \boldsymbol{K}_0 := \boldsymbol{K}_0^c + \boldsymbol{K}_0^s, \tag{30}$$

$$\boldsymbol{K}_{\nabla}^c := \left(\boldsymbol{\Pi}_{k,E*}^{\nabla}\right)^T \int_{\mathcal{K}_x} \nabla \boldsymbol{m} \nabla \boldsymbol{m}^T \,\mathrm{d}\Omega \boldsymbol{\Pi}_{k,E*}^{\nabla}, \quad \boldsymbol{K}_{\nabla}^s := \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{\nabla}\right)^T \left(\boldsymbol{I} - \boldsymbol{\Pi}_{k,E}^{\nabla}\right), \quad \boldsymbol{K}_{\nabla} := \boldsymbol{K}_{\nabla}^c + \boldsymbol{K}_{\nabla}^s. \tag{31}$$

Next, the bilinear form for vector field $\boldsymbol{u}$ and $\boldsymbol{v}$ is considered. For example, the displacement can be written as the following form

$$\boldsymbol{u} = [u_x, u_y]^T, \quad \delta\boldsymbol{u} = [\delta u_x, \delta u_y]^T, \tag{32}$$

then the following bilinear form for velocity $\boldsymbol{v}$ can be calculated as

$$m_h^E(\delta\boldsymbol{v}, \boldsymbol{v}) = \int_{\mathcal{K}_x} \delta\boldsymbol{v} \cdot \boldsymbol{v} \,\mathrm{d}\Omega = \begin{bmatrix} \delta\hat{\boldsymbol{v}}_x^T & \delta\hat{\boldsymbol{v}}_y^T \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_0 & \\ & \boldsymbol{K}_0 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{v}}_x \\ \hat{\boldsymbol{v}}_y \end{bmatrix} = \delta\hat{\boldsymbol{v}}^T \boldsymbol{K}^0 \hat{\boldsymbol{v}}, \quad \boldsymbol{K}^0 := \begin{bmatrix} \boldsymbol{K}_0 & \\ & \boldsymbol{K}_0 \end{bmatrix}. \tag{33}$$

Using the Voigt notation ($\tilde{\square}$) for strain

$$\tilde{\boldsymbol{\varepsilon}}(\boldsymbol{u}) = \boldsymbol{A}\boldsymbol{d}(\boldsymbol{u}), \tag{34}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{d}(\boldsymbol{u}) = \begin{bmatrix} \dfrac{\partial u_x}{\partial x} & \dfrac{\partial u_x}{\partial y} & \dfrac{\partial u_y}{\partial x} & \dfrac{\partial u_y}{\partial y} \end{bmatrix}^T, \tag{35}$$

then

$$\begin{aligned} \boldsymbol{\Pi}_{k,E}^{\nabla} \boldsymbol{v}_h &= \begin{bmatrix} \boldsymbol{m}^T & \\ & \boldsymbol{m}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Pi}_{k,E*}^{\nabla} & \\ & \boldsymbol{\Pi}_{k,E*}^{\nabla} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}}_x^h \\ \hat{\boldsymbol{u}}_y^h \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}^T & \\ & \boldsymbol{\phi}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Pi}_{k,E}^{\nabla} & \\ & \boldsymbol{\Pi}_{k,E}^{\nabla} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{u}}_x^h \\ \hat{\boldsymbol{u}}_y^h \end{bmatrix} \\ &= \boldsymbol{M}^T \boldsymbol{\Pi}_*^{\nabla} \hat{\boldsymbol{u}}_h. \end{aligned} \tag{36}$$

Considering the definition before, the bilinear form for strain can be written as

$$\begin{aligned} a_h^E(\delta\boldsymbol{u}, \boldsymbol{u}) &= \int_{\mathcal{K}_x} \varepsilon(\delta\boldsymbol{u}) : \mathbb{D} : \varepsilon(\boldsymbol{u}) \,\mathrm{d}\Omega = \int_{\mathcal{K}_x} \boldsymbol{d}(\delta\boldsymbol{u})^T \boldsymbol{A}^T \tilde{\mathbb{D}} \boldsymbol{A} \boldsymbol{d}(\boldsymbol{u}) \,\mathrm{d}\Omega \\ &= \delta\hat{\boldsymbol{u}}^t \left(\left(\boldsymbol{\Pi}_*^{\nabla}\right)^T \int_{\mathcal{K}_x} \nabla \boldsymbol{M} \boldsymbol{A}^T \tilde{\mathbb{D}} \boldsymbol{A} \nabla \boldsymbol{M}^T \,\mathrm{d}\Omega \boldsymbol{\Pi}_*^{\nabla} + \gamma \begin{bmatrix} \boldsymbol{K}_{\nabla}^s & \\ & \boldsymbol{K}_{\nabla}^s \end{bmatrix} \right) \hat{\boldsymbol{u}} \\ &= \delta\hat{\boldsymbol{u}}^t \left(\boldsymbol{K}_{\varepsilon}^c + \gamma \boldsymbol{K}_{\varepsilon}^s\right) \hat{\boldsymbol{u}} = \delta\hat{\boldsymbol{u}}^t \boldsymbol{K}^{\epsilon} \hat{\boldsymbol{u}}, \end{aligned} \tag{37}$$

where γ in Eq. (37) is the stabilization parameter which can be selected as $\gamma = \frac{1}{4N_K}\mathrm{tr}(\boldsymbol{K}_{\varepsilon}^c)$. The stabilization term can be chosen in different ways, see the relevant work for details [4,24]. In general, for linear elastic problems, the choice of parameters in the stabilization term has little effect on the results. For linear elastic problems, the matrix $\boldsymbol{K}_{\varepsilon}^c$ can also obtained based on $\boldsymbol{K}_{\nabla}^c$. In actual

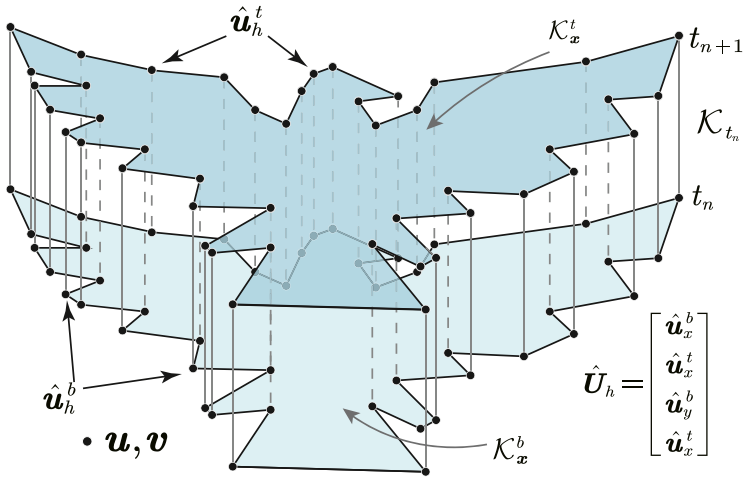

Fig. 2. First-order virtual element in space–time cylinder.

programming, the matrix can be calculated directly employing the form in Eq. (37). The matrices $\boldsymbol{K}_0, \boldsymbol{K}^0, \boldsymbol{K}^\epsilon$ (Eqs. (33)–(37)) can be used to calculate the stiffness matrix of the space–time dynamic problems.

3.4. Matrix form for the linear space–time elasticity

Since the space domain Ω is divided into a set of non-overlapping polygons $\{\mathcal{T}_h^{\boldsymbol{x}}\}$, the integrations of Eqs. (33)–(37) can be calculated based on the Gaussian-Legendre quadrature. For the first-order scheme discussed in this work, a single-point integration can be performed. The time interval $[0, T]$ is divided into N_T subintervals $\mathcal{K}_{t_n} := (t_n, t_{n+1})$ with knots $0 = t_0 < t_1 < \cdots < t_{N_T} = T$. Then, the space–time element can be obtained as $\mathcal{K} = \mathcal{K}_{\boldsymbol{x}} \times \mathcal{K}_t$ (as shown in Fig. 2).

Under the current assumptions, the upper and lower surfaces ($\mathcal{K}_{\boldsymbol{x}}^b$ and $\mathcal{K}_{\boldsymbol{x}}^t$ as shown in Fig. 2) of the space–time element have the same shape, and therefore their shape functions are equal. If assumed a linear interpolation in the time dimension, the shape function of our space–time element $\mathcal{K}$ has the form

$$\boldsymbol{\phi}_{\mathcal{K}} = \begin{bmatrix} \boldsymbol{\phi}_{\mathcal{K}_{\boldsymbol{x}}^b} \boldsymbol{\phi}_{\mathcal{K}_t}(1) \\ \boldsymbol{\phi}_{\mathcal{K}_{\boldsymbol{x}}^t} \boldsymbol{\phi}_{\mathcal{K}_t}(2) \end{bmatrix}, \tag{38}$$

where $\boldsymbol{\phi}_{\mathcal{K}_{\boldsymbol{x}}^b} = \boldsymbol{\phi}_{\mathcal{K}_{\boldsymbol{x}}^t} =: \boldsymbol{\phi}_{\mathcal{K}_{\boldsymbol{x}}}$ is the shape function of our virtual element and $\boldsymbol{\phi}_{\mathcal{K}_t} := [\phi_{\mathcal{K}_t}(1), \phi_{\mathcal{K}_t}(2)]^T$ is the basic function of the one-dimensional line element in time-direction. The shape function is

$$\phi_{\mathcal{K}_t}(1) = \frac{1-\xi}{2}, \quad \phi_{\mathcal{K}_t}(2) = \frac{1+\xi}{2}. \tag{39}$$

The gradient of the tensor product shape function has the form as

$$\nabla_{\boldsymbol{x}} \boldsymbol{\phi}_{\mathcal{K}} = \begin{bmatrix} \nabla_{\boldsymbol{x}} \boldsymbol{\phi}_{\mathcal{K}_{\boldsymbol{x}}^b} \boldsymbol{\phi}_{\mathcal{K}_t}(1) \\ \nabla_{\boldsymbol{x}} \boldsymbol{\phi}_{\mathcal{K}_{\boldsymbol{x}}^t} \boldsymbol{\phi}_{\mathcal{K}_t}(2) \end{bmatrix}, \quad \nabla_t \boldsymbol{\phi}_{\mathcal{K}} = \begin{bmatrix} \boldsymbol{\phi}_{\mathcal{K}_{\boldsymbol{x}}^b} \nabla_t \boldsymbol{\phi}_{\mathcal{K}_t}(1) \\ \boldsymbol{\phi}_{\mathcal{K}_{\boldsymbol{x}}^t} \nabla_t \boldsymbol{\phi}_{\mathcal{K}_t}(2) \end{bmatrix} \tag{40}$$

The displacement $\boldsymbol{U}$ for the space–time element is given by

$$\boldsymbol{U}_h = \begin{bmatrix} \boldsymbol{\phi}_{\mathcal{K}}^T & \\ & \boldsymbol{\phi}_{\mathcal{K}}^T \end{bmatrix} \hat{\boldsymbol{U}}_h = \boldsymbol{\Phi}_{\mathcal{K}}^T \hat{\boldsymbol{U}}_h, \tag{41}$$

where

$$\boldsymbol{\Phi}_{\mathcal{K}}^T := \begin{bmatrix} \boldsymbol{\phi}_{\mathcal{K}}^T & \\ & \boldsymbol{\phi}_{\mathcal{K}}^T \end{bmatrix}, \quad \hat{\boldsymbol{U}}_h := \begin{bmatrix} \hat{\boldsymbol{u}}_x^b & \hat{\boldsymbol{u}}_x^t & \hat{\boldsymbol{u}}_y^b & \hat{\boldsymbol{u}}_y^t \end{bmatrix}^T. \tag{42}$$

The same interpolation function are used for the velocity $\boldsymbol{v}$.

Our discrete scheme is based on Eq. (18). For each space–time element $\mathcal{K}$, the element stiffness matrix $\boldsymbol{K}$ has the block form as

$$\boldsymbol{K} := \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{vv}} & \boldsymbol{K}_{\boldsymbol{vu}} \\ \boldsymbol{K}_{\boldsymbol{uv}} & \boldsymbol{K}_{\boldsymbol{uu}} \end{bmatrix} \tag{43}$$

According to the matrix calculated previously, the follow matrices can be obtained as

$$\begin{aligned}\boldsymbol{K}_{\boldsymbol{vv}} &= \rho^{-1}\int_{t_n}^{t_{n+1}}\int_{\mathcal{K}_x}\left(\boldsymbol{\Phi}_{\mathcal{K}}+\theta h\nabla_t\boldsymbol{\Phi}_{\mathcal{K}}\right)\boldsymbol{\Phi}_{\mathcal{K}}^T\,\mathrm{d}\Omega\,\mathrm{d}t\\ &= \rho^{-1}\begin{bmatrix}\int_{t_n}^{t_{n+1}}\boldsymbol{\phi}_{\mathcal{K}_t}\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}t\otimes\int_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}^T\,\mathrm{d}\Omega & \\ & \int_{t_n}^{t_{n+1}}\boldsymbol{\phi}_{\mathcal{K}_t}\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}t\otimes\int_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}^T\,\mathrm{d}\Omega\end{bmatrix}\\ &+ \rho^{-1}\theta h\begin{bmatrix}\int_{t_n}^{t_{n+1}}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}t\otimes\int_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}^T\,\mathrm{d}\Omega & \\ & \int_{t_n}^{t_{n+1}}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}t\otimes\int_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}^T\,\mathrm{d}\Omega\end{bmatrix},\end{aligned}\tag{44}$$

where $\otimes$ is the Kronecker product defined as $\boldsymbol{A}\otimes\boldsymbol{B}=(a_{ij}\boldsymbol{B})$, $A=(a_{ij})$. The integral term along the space domain in Eq. (44) was calculated previously as

$$\int_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}\boldsymbol{\phi}_{\mathcal{K}_x}^T\,\mathrm{d}\Omega=\boldsymbol{K}_0^c+\boldsymbol{K}_0^s=:\boldsymbol{K}_0,\tag{45}$$

where $\boldsymbol{K}_0^c$ and $\boldsymbol{K}_0^s$ are defined in Eq. (30).

Since the Lagrange interpolation is selected to represent $\boldsymbol{\phi}_{\mathcal{K}_t}$, the integral terms along the time direction can be calculated based on the Gaussian integration

$$\int_{t_n}^{t_{n+1}}\boldsymbol{\phi}_{\mathcal{K}_t}\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}t=\frac{h}{2}\int_{-1}^{1}\boldsymbol{\phi}_{\mathcal{K}_t}\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}\xi=\frac{h}{2}\sum_{i=1}^{2}w(i)\boldsymbol{\phi}_{\mathcal{K}_t}(\xi_i)\boldsymbol{\phi}_{\mathcal{K}_t}^T(\xi_i)=:\boldsymbol{L},\tag{46}$$

$$\int_{t_n}^{t_{n+1}}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}t=\frac{h}{2}\int_{-1}^{1}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}\xi=\frac{h}{2}\sum_{i=1}^{2}w(i)\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}(\xi_i)\boldsymbol{\phi}_{\mathcal{K}_t}^T(\xi_i)=:\boldsymbol{Q},\tag{47}$$

$$\int_{t_n}^{t_{n+1}}\boldsymbol{\phi}_{\mathcal{K}_t}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}t=\frac{h}{2}\int_{-1}^{1}\boldsymbol{\phi}_{\mathcal{K}_t}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}\xi=\frac{h}{2}\sum_{i=1}^{2}w(i)\boldsymbol{\phi}_{\mathcal{K}_t}(\xi_i)\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}^T(\xi_i)=:\boldsymbol{R},\tag{48}$$

$$\int_{t_n}^{t_{n+1}}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}t=\frac{h}{2}\int_{-1}^{1}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}^T\,\mathrm{d}\xi=\frac{h}{2}\sum_{i=1}^{2}w(i)\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}(\xi_i)\nabla_t\boldsymbol{\phi}_{\mathcal{K}_t}^T(\xi_i)=:\boldsymbol{W},\tag{49}$$

where h is the size of the virtual space–time element in the time direction, and $w(i)$ is the weight of the Gaussian integration.

If define any matrix $\boldsymbol{K}^{any}$ as a block matrix as follows

$$\boldsymbol{K}^{any}=\begin{bmatrix}\boldsymbol{K}_{\mathrm{I,I}}^{any} & \boldsymbol{K}_{\mathrm{I,II}}^{any}\\ \boldsymbol{K}_{\mathrm{II,I}}^{any} & \boldsymbol{K}_{\mathrm{II,II}}^{any}\end{bmatrix},\tag{50}$$

the matrix $\boldsymbol{K}_{\boldsymbol{vv}}$ can also obtained as

$$\boldsymbol{K}_{\boldsymbol{vv}}=\rho^{-1}\begin{bmatrix}\boldsymbol{L}\otimes\boldsymbol{K}_{\mathrm{I,I}}^0 & \\ & \boldsymbol{L}\otimes\boldsymbol{K}_{\mathrm{II,II}}^0\end{bmatrix}+\rho^{-1}\theta h\begin{bmatrix}\boldsymbol{Q}\otimes\boldsymbol{K}_{\mathrm{I,I}}^0 & \\ & \boldsymbol{Q}\otimes\boldsymbol{K}_{\mathrm{II,II}}^0\end{bmatrix},\tag{51}$$

where $\boldsymbol{K}^0:=\mathrm{diag}(\boldsymbol{K}_0,\boldsymbol{K}_0)$ as defined in Eq. (33).

Besides, the other stiffness matrices can be calculated as follows

$$\boldsymbol{K}_{\boldsymbol{vu}}=\begin{bmatrix}\boldsymbol{R}\otimes\boldsymbol{K}_{\mathrm{I,I}}^0 & \\ & \boldsymbol{R}\otimes\boldsymbol{K}_{\mathrm{II,II}}^0\end{bmatrix}+\theta h\begin{bmatrix}\boldsymbol{W}\otimes\boldsymbol{K}_{\mathrm{I,I}}^0 & \\ & \boldsymbol{W}\otimes\boldsymbol{K}_{\mathrm{II,II}}^0\end{bmatrix},\tag{52}$$

$$\boldsymbol{K}_{\boldsymbol{uv}}=\begin{bmatrix}\boldsymbol{R}\otimes\boldsymbol{K}_{\mathrm{I,I}}^0 & \\ & \boldsymbol{R}\otimes\boldsymbol{K}_{\mathrm{II,II}}^0\end{bmatrix}+\theta h\begin{bmatrix}\boldsymbol{W}\otimes\boldsymbol{K}_{\mathrm{I,I}}^0 & \\ & \boldsymbol{W}\otimes\boldsymbol{K}_{\mathrm{II,II}}^0\end{bmatrix},\tag{53}$$

$$\boldsymbol{K}_{\boldsymbol{uu}}=\begin{bmatrix}\boldsymbol{L}\otimes\boldsymbol{K}_{\mathrm{I,I}}^{\varepsilon} & \boldsymbol{L}\otimes\boldsymbol{K}_{\mathrm{I,II}}^{\varepsilon}\\ \boldsymbol{L}\otimes\boldsymbol{K}_{\mathrm{II,I}}^{\varepsilon} & \boldsymbol{L}\otimes\boldsymbol{K}_{\mathrm{II,II}}^{\varepsilon}\end{bmatrix}+\theta h\begin{bmatrix}\boldsymbol{Q}\otimes\boldsymbol{K}_{\mathrm{I,I}}^{\varepsilon} & \boldsymbol{Q}\otimes\boldsymbol{K}_{\mathrm{I,II}}^{\varepsilon}\\ \boldsymbol{Q}\otimes\boldsymbol{K}_{\mathrm{II,I}}^{\varepsilon} & \boldsymbol{Q}\otimes\boldsymbol{K}_{\mathrm{II,II}}^{\varepsilon}\end{bmatrix},\tag{54}$$

where $\boldsymbol{K}^0$ and $\boldsymbol{K}^{\epsilon}$ are given in Eqs. (33) and (37). By assembling all elements and solving the systems of equation, the displacement $\boldsymbol{u}$ at different times can be obtained directly without any time integration format. Science the tensor-product-in-time meshes is selected, the calculation of stiffness matrix $\boldsymbol{K}$ is very efficient (only need to compute some integrals $\{\mathcal{T}_h^x\}$ of the elements in the 2D computational domain and then compute the overall stiffness matrix via the Kronecker product).

In order to clarify the analysis process, the flow chart is given, as shown in Fig. 3.

4. Finite element method for space–time dynamics

The form of Eqs. (51)–(54) provides an unified format, so it can be used to derive a finite element method with the tensor-product-in-time meshes for the calculation of the space–time dynamics. Similar to the virtual element method, the calculation of matrix $\boldsymbol{K}^0$ and $\boldsymbol{K}^{\epsilon}$ is needed based on the finite element method for the 2D elements in the computational domain Ω, then the stiffness matrix can be calculated based on Eqs. (51)–(54). In this work, the first-order quadrilateral element is choosed for discretization in the space domain. In the time direction, the two-node linear elements are used.

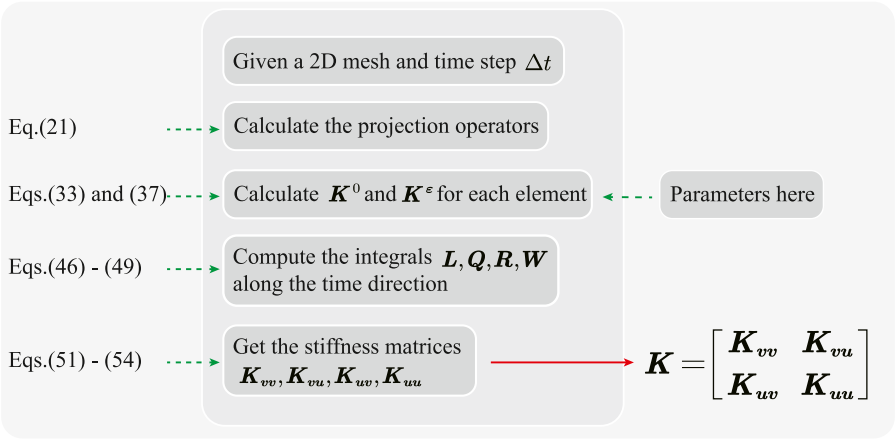

Fig. 3. Flowchart of solving dynamic problems in space–time format.

Table 1
The total degrees of freedom used by the space–time format for different time steps and different meshes.

Time step	Mesh 1	Mesh 2	Mesh 3
$\Delta t = 0.2$	31 824	40 800	42 840
$\Delta t = 0.1$	63 024	80 800	84 840

With the shape function $\boldsymbol{\phi}_{\text{FEM}}$ of the finite element method, then the matrix $\boldsymbol{K}^0$ and $\boldsymbol{K}^\epsilon$ can be calculated as

$$\boldsymbol{K}^0 = \int_{\mathcal{K}_x} \begin{bmatrix}\boldsymbol{\phi}_{\text{FEM}} & \\ & \boldsymbol{\phi}_{\text{FEM}}\end{bmatrix} \begin{bmatrix}\boldsymbol{\phi}_{\text{FEM}} & \\ & \boldsymbol{\phi}_{\text{FEM}}\end{bmatrix}^T \mathrm{d}\Omega, \tag{55}$$

$$\boldsymbol{K}^\epsilon = \int_{\mathcal{K}_x} \boldsymbol{B}^T \hat{\mathbb{D}} \boldsymbol{B}\, \mathrm{d}\Omega, \tag{56}$$

where

$$\boldsymbol{B} := \begin{bmatrix} \frac{\partial \boldsymbol{\phi}_{\text{FEM}}^T}{\partial x} & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \frac{\partial \boldsymbol{\phi}_{\text{FEM}}^T}{\partial y} \\ \frac{\partial \boldsymbol{\phi}_{\text{FEM}}^T}{\partial y} & \frac{\partial \boldsymbol{\phi}_{\text{FEM}}^T}{\partial x} \end{bmatrix}. \tag{57}$$

Substituting Eqs. (56) and (57) into Eqs. (52) to (54), yields the stiffness matrix. Compared with the time integration format in the traditional finite element method, this format has high computational efficiency and is simple to program. The relevant finite element code can be found in https://github.com/Qinxiaoye/VEM-spcae-time-dynamic or https://www.vemhub.com/code.

5. Numerical examples

5.1. Transversal beam vibration

The purpose of the first example is to study the effect of the stability parameter θ on the results in the space–time format. As shown in Fig. 4, a cantilever beam is considered with the shape as $L \times H = 5 \times 1$. The left side is fixed and two different vertical loads $P(t)$ are applied on the right side (see Fig. 4). The first type is a constant load $P = P_0$. Another one is a half sine load $P = P_0 \sin(\frac{\pi t}{5})$ for $t \leq 5$. The material parameters are assumed as $E = 200000$, $\nu = 0.3$.

Since the computational domain Ω is discretized using the virtual element method, polygonal meshes (including non-convex meshes) can be used as shown in Fig. 4. In this example, the finite element method is selected for comparison. In addition to using the space–time approach, the Newmark method is also employed to solve the dynamics problem for comparison. For different time steps and different meshes, the total degrees of freedom used in the space–time scheme are listed in Table 1.

Firstly, a constant load $P = P_0 = 100$ is assumed. To test the influence of the stabilization parameter θ, the virtual element method with mesh 1 (given in Fig. 4) is used. The density is chosen as $\rho = 100$. The element length in the time direction is selected as $\Delta t = 0.2$, and the total calculation time is 10s. The reference solution is obtained by the Newmark time integration method with a time step of $\Delta t = 0.01$. For different stabilization parameters $\theta \in \{0, 0.01, 0.1, 0.4, 0.8\}$, the y-direction displacements over time of the lower right node are given in Fig. 5. Within the given range, the parameter θ has little effect on the result. Generally speaking,

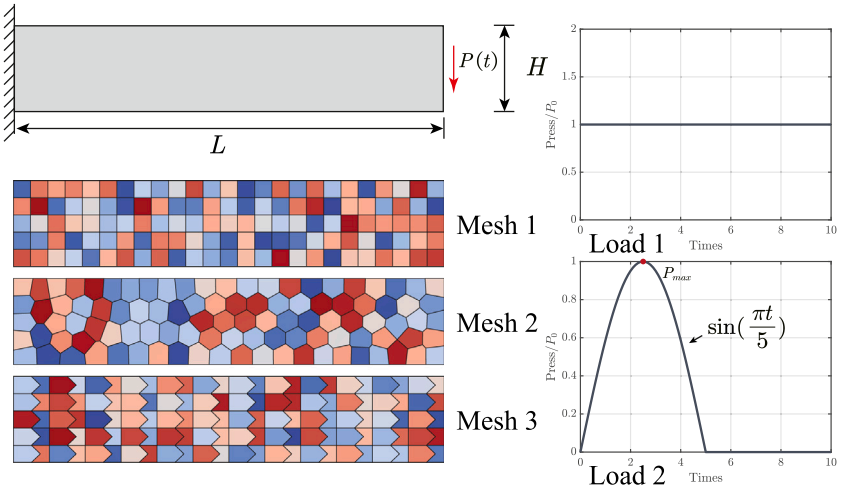

Fig. 4. Transversal beam vibration, meshes and applied loads.

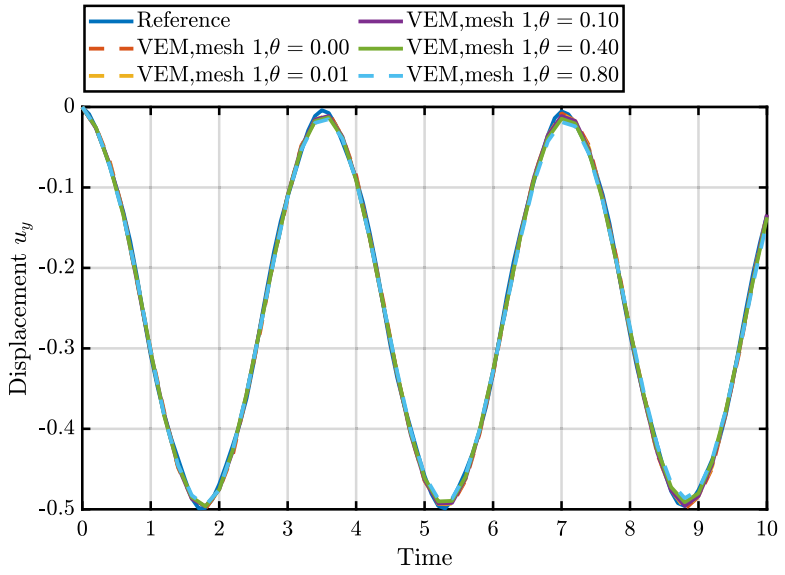

Fig. 5. Displacement over time response at point $(5, 0)$.

a larger θ is more stable, but the solution will be damped, leading to energy lost and a reduced displacement. At the same time, a smaller θ yields better results. Therefore, $\theta = 0.01$ is chosen in the following analysis.

Since Newmark is not a strictly energy-preserving scheme, energy drift may occur in the Newmark integration method for long-term integration. To test the space–time scheme of the virtual element method, the density of the beam is selected as $\rho = 10$. The element length in the time direction is selected as $\Delta t = 0.1$, and the total calculation time is 10s. At the same time, the Newmark method with $\Delta t = 0.1$ is used in FEM and VEM to calculate the displacements. The y-direction displacements of the lower right node are given in Fig. 6. It can be seen that compared to the Newmark method, the space–time scheme is free of phase error when using the same time step.

In addition to constant loads, the time-varying load condition can also be introduced. As shown in Fig. 4, a half sine load is assumed as $P = P_0 \sin(\frac{\pi t}{5})$, $P_0 = 100$ for $t \leq 5$ ($P_0 = 0$ for $t > 5$). The element length in time direction is selected as $\Delta t = 0.1$, and the total calculation time is 10s. Very small time step ($\Delta t = 0.01$) is used in finite element method for comparison. The y-direction displacements of the lower right node are depicted in Fig. 7. It can be seen that under the current circumstances, the space–time VEM can still obtain sufficiently accurate results, while the results obtained by the Newmark method are more sensitive to the time step requirements.

Computational efficiency is an important topic of our concern. For the same space–time framework, the VEM and FEM are selected for comparison. For quadrilateral mesh, when the time step is $\Delta t = 0.2$, VEM takes 2.46 s and FEM takes 2.57 s. When the

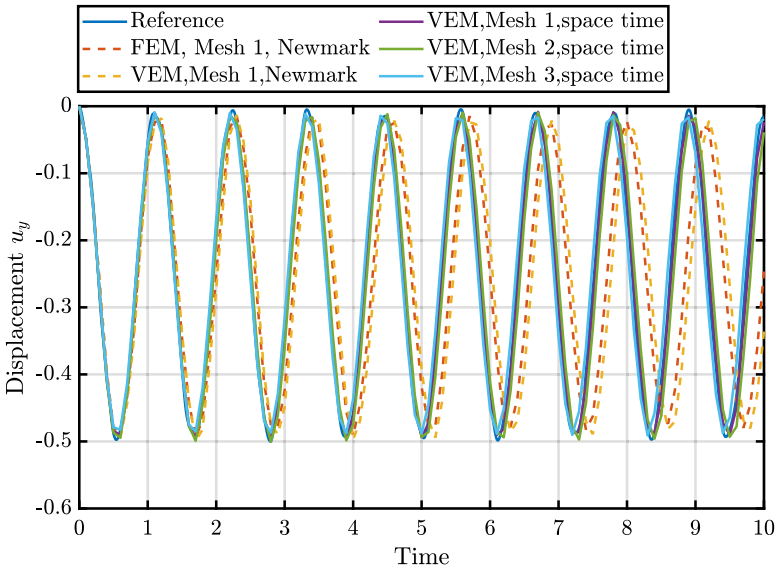

Fig. 6. Displacement over time response at point $(5, 0)$, constant load.

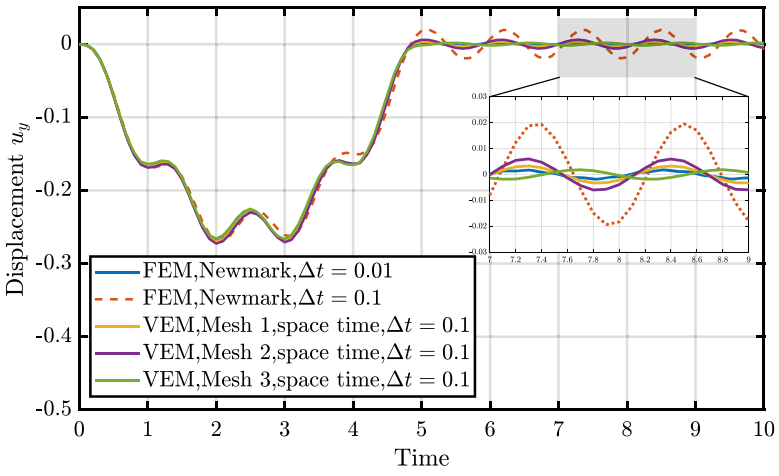

Fig. 7. Displacement over time response at point $(5, 0)$, time-varying load.

time step is $\Delta t = 0.1$, VEM takes 5.33 s and FEM takes 5.76 s. It can be seen that the efficiency of the virtual element method is slightly faster than that of the finite element method. Since the space–time scheme increases the number of degrees of freedom of the problem, it accounts for a large proportion of the time spent on solving linear algebraic equations. This can be observed in both VEM and FEM. When the time step is $\Delta t = 0.1$, VEM takes 3.503 s and FEM takes 4.13 s to solve the equation.

The contour plots of displacement u_y and velocity v_y obtained by space–time VEM with different meshes and different densities are illustrated in Figs. 8 and 9, respectively.

5.2. Cook membrane problem

In this example, the Cook's membrane is considered as shown in Fig. 10. The relevant dimensions are $L = 48$, $H_1 = 44$, $H_2 = 16$. The left side of the membrane is fixed and a constant vertical load $q_y = 100$ is applied at the right side. In this example, two different types of meshes including regular meshes and polygonal meshes are used (shown in Fig. 10). The material parameters are assumed as $E = 200000$, $\nu = 0.3$ and $\rho = 10$.

To test the convergence of the space–time VEM, a mesh refinement study with the element division of $2^N \times 2^N$ for $N = 2, 3, 4$ is given. For the space–time VEM, the mesh length in the time direction is selected as $h = \Delta t = 0.2$ and $h = \Delta t = 0.1$. Besides, the Newmark method is also used for comparison. For different time steps and different meshes, the total degrees of freedom used in the space–time scheme are listed in Table 2.

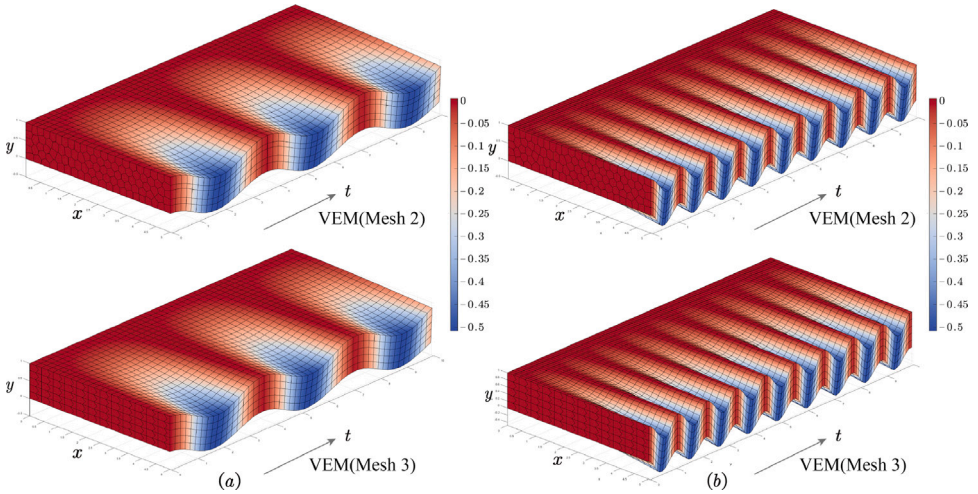

Fig. 8. Contour plots of displacement u_y obtained by space–time VEM with different meshes, (a) $\rho = 100$, (b) $\rho = 10$.

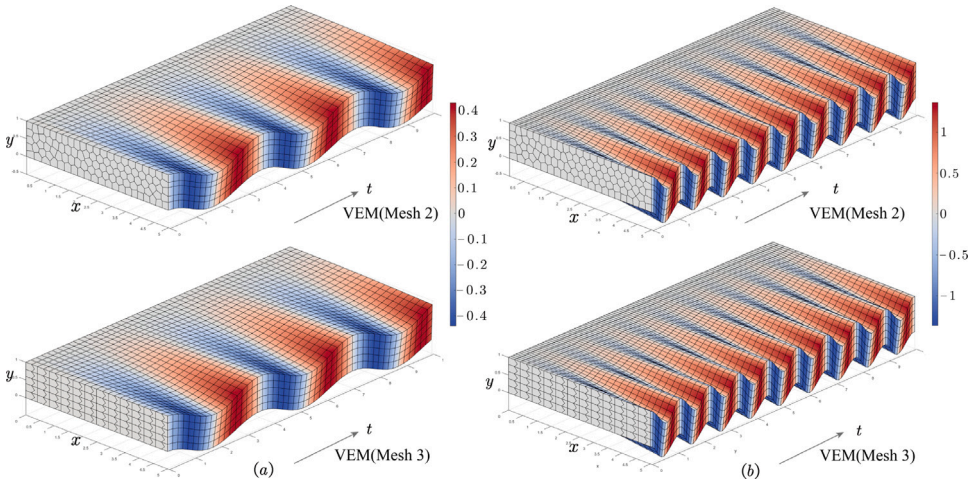

Fig. 9. Contour plots of velocity v_y obtained by space–time VEM with different meshes, (a) $\rho = 100$, (b) $\rho = 10$.

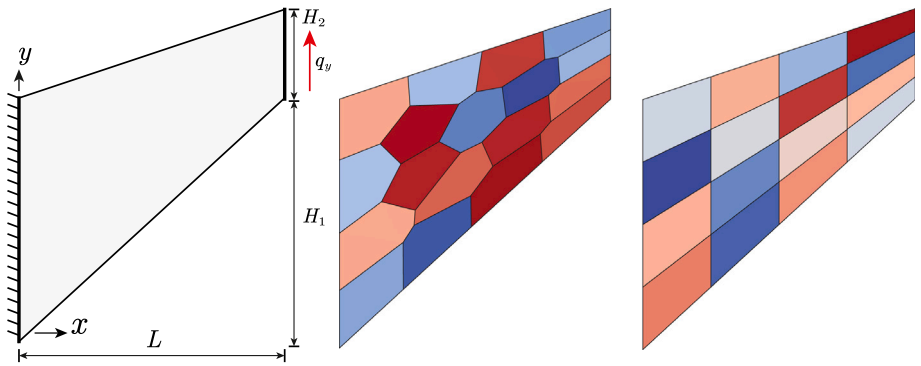

Fig. 10. Cook's membrane problem, polygon mesh and regular mesh.

Table 2
The total degrees of freedom used by the space–time format for different time steps and different meshes (poly is for polygon mesh and quad is for quadrilateral/regular mesh).

Time step	$N = 2$		$N = 3$		$N = 4$	
	Poly	Quad	Poly	Quad	Poly	Quad
$\Delta t = 0.2$	7752	5100	24 684	16 524	95 064	58 956
$\Delta t = 0.1$	15 352	10 100	48 884	32 724	188 264	116 756

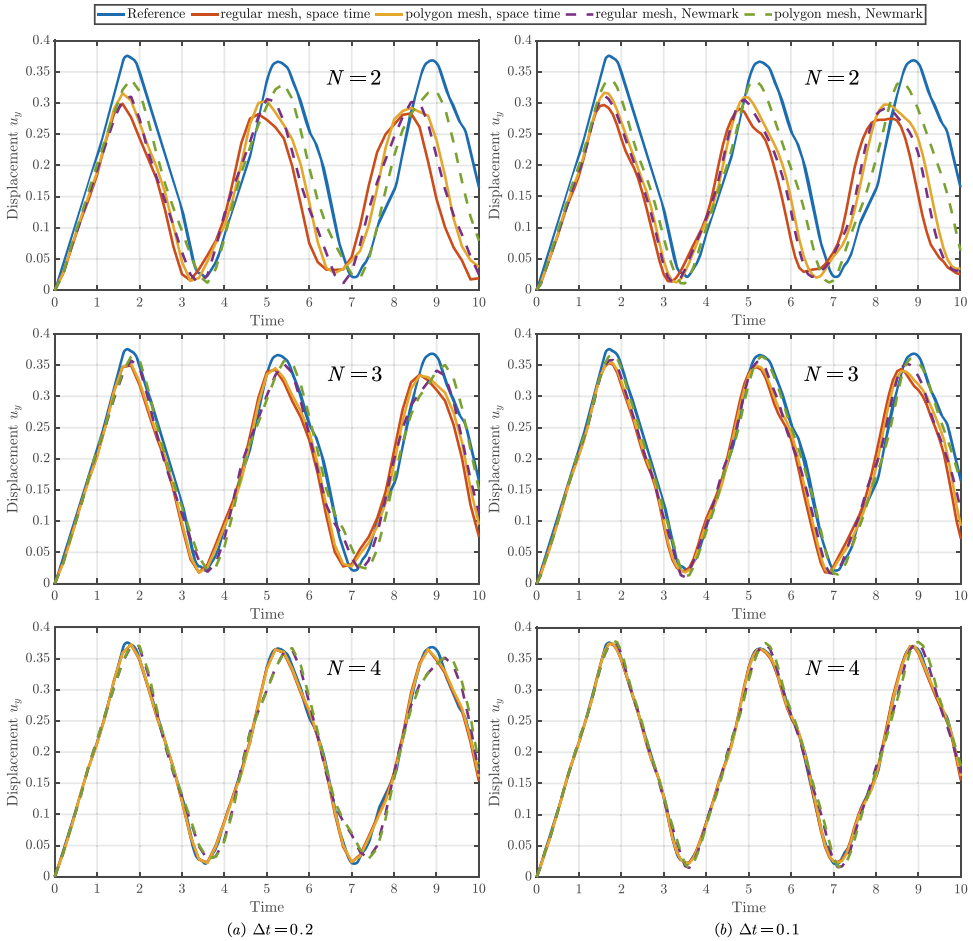

Fig. 11. Convergence study of displacement u_y for the top right corner over time response for Cook membrane problem, N is the element division.

The associated convergence study of the maximum displacement u_y for the top right corner is given in Fig. 11. It can be observed that as the mesh size decreases, the results obtained by space–time VEM converge to the reference solution. For $N = 4$, the displacements agree very well with the reference solution even for $\Delta t = 0.2$.

The contour plots of displacement in the y direction and the von Mises stress for the space–time domain ($N = 4, \Delta t = 0.2$) are given in Fig. 12. For $N = 4$, the contour plots of the von Mises stress obtained by space–time VEM for different meshes are depicted in Fig. 13.

5.3. Center oblique crack plate

In this example, a rectangular plate with an arbitrarily oriented central crack is considered as illustrated in Fig. 14. The geometrical parameters of the plate are $2 \times H = 40$, $2 \times L = 20$, and the crack length is $2a = 5$. The uniform traction $P = 100$ is given on the top and the bottom of the plate. The material parameters are assumed as $E = 200000$, $\nu = 0.3$, $\rho = 10$. The central crack can be rotated to an arbitrary angle α and in this work, different angles are selected as $\alpha = 0°, 20°, 40°, 60°, 80°$. This is an advantage of VEM since the element at interfaces can have arbitrary number of nodes. Thus the rotated inner part of the mesh in Fig. 14 yields always a continuous discretization.

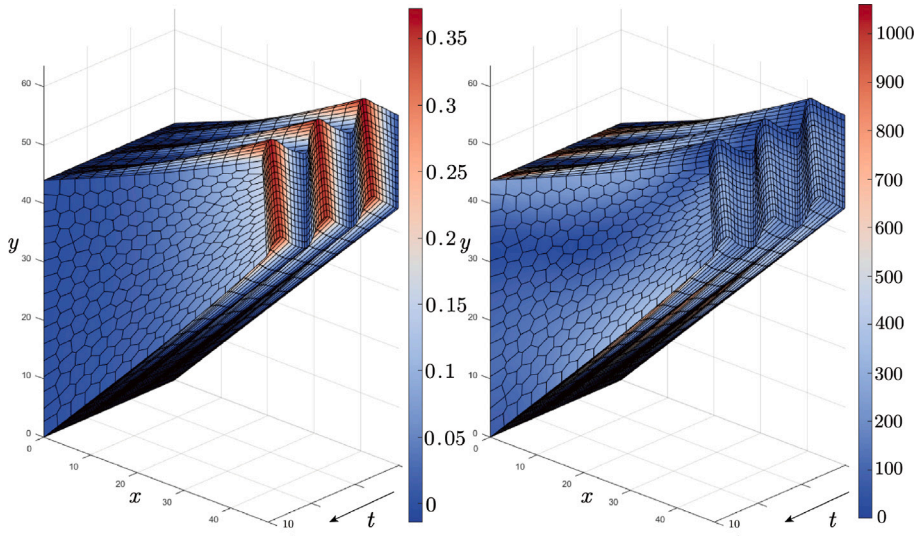

Fig. 12. Contour plots of the displacement u_y and von Mises stress for $N = 4$ and $\Delta t = 0.2$ (scaling factor is 10).

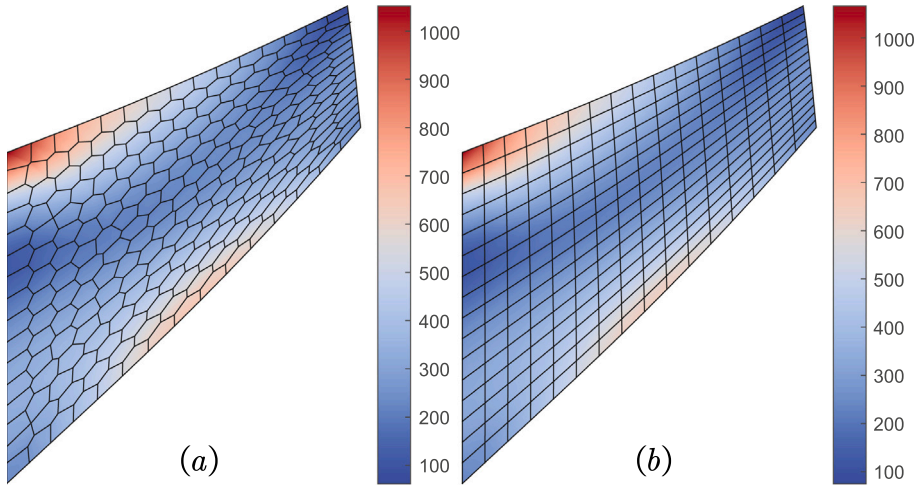

Fig. 13. Contour plots of the stresses for different meshes at $t = 9$ (scaling factor is 10), (a) polygonal mesh, (b) regular mesh.

Table 3
The total degrees of freedom used by the space–time format for different meshes (different angles).

Angle α	$\alpha = 0$	$\alpha = 20$	$\alpha = 40$	$\alpha = 60$	$\alpha = 80$
d.o.f.	372 096	372 912	374 136	372 504	373 320

The space–time scheme is used and the size of the mesh in the time direction is selected as $\Delta t = 0.02$. Again, the Newmark method with the time step $\Delta t = 0.01$ is used for comparison. For different angles, the total number of nodes are different. The total degrees of freedom for different angles are listed in Table 3. For different angles, the displacements u_y over time at the top right corner are illustrated in Fig. 15. The displacements obtained by the space–time VEM ($\Delta t = 0.02$) agree well with the displacements obtained by the Newmark method ($\Delta t = 0.01$). The contour plots of von Mises stresses and displacements u_y are depicted in Figs. 16 and 17 for different angles. The von Mises stresses at $t = 0.8$ for different α are illustrated in Fig. 18.

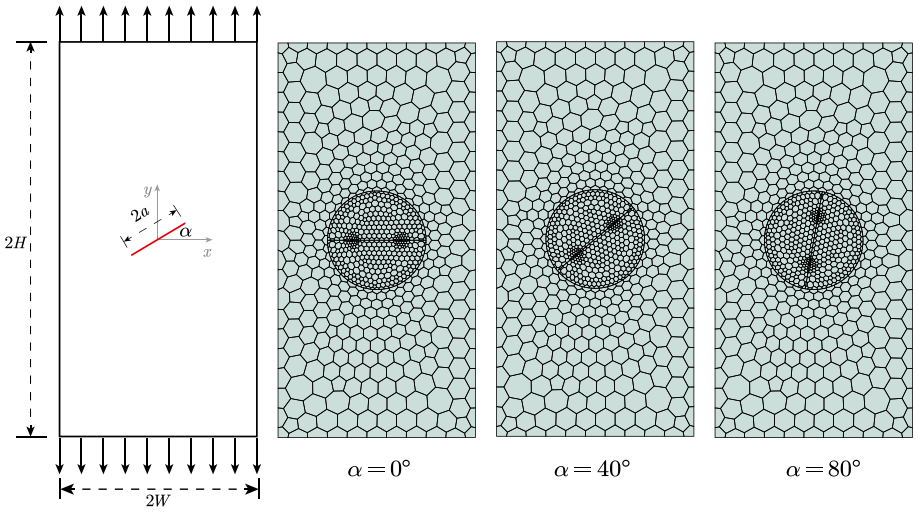

Fig. 14. Geometry and meshes of a rectangular plate with an arbitrarily oriented central crack.

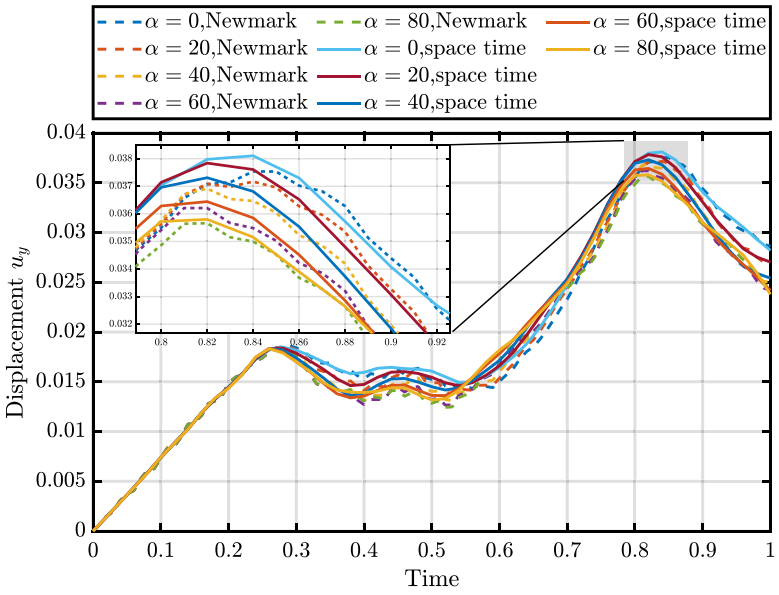

Fig. 15. Displacement u_y for the top right corner over time response for different α.

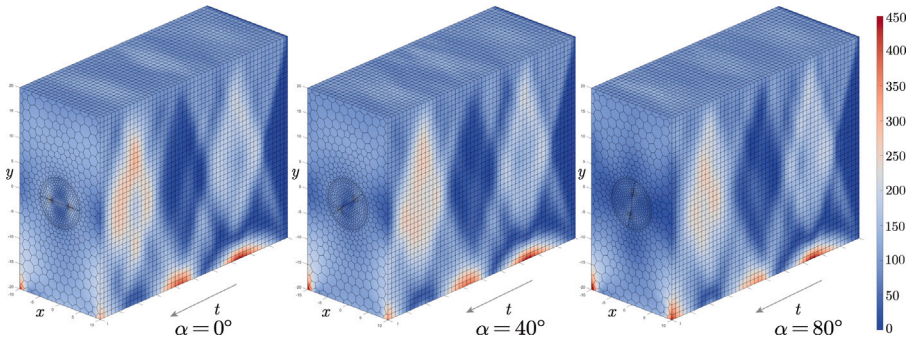

Fig. 16. Contour plots of von Mises stresses over time response for different α.

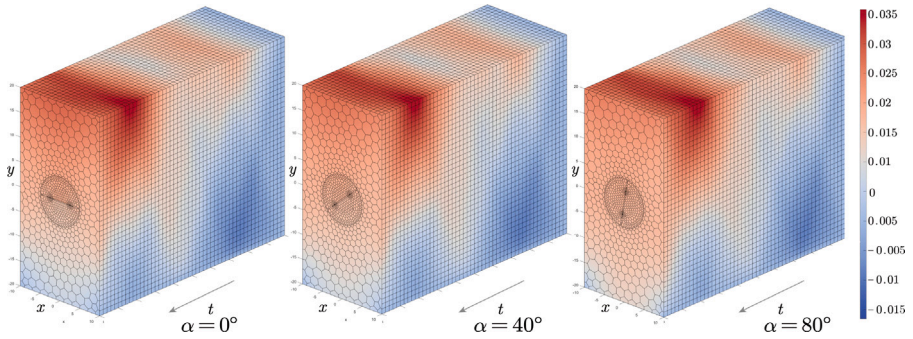

Fig. 17. Displacements u_y over time response for different α.

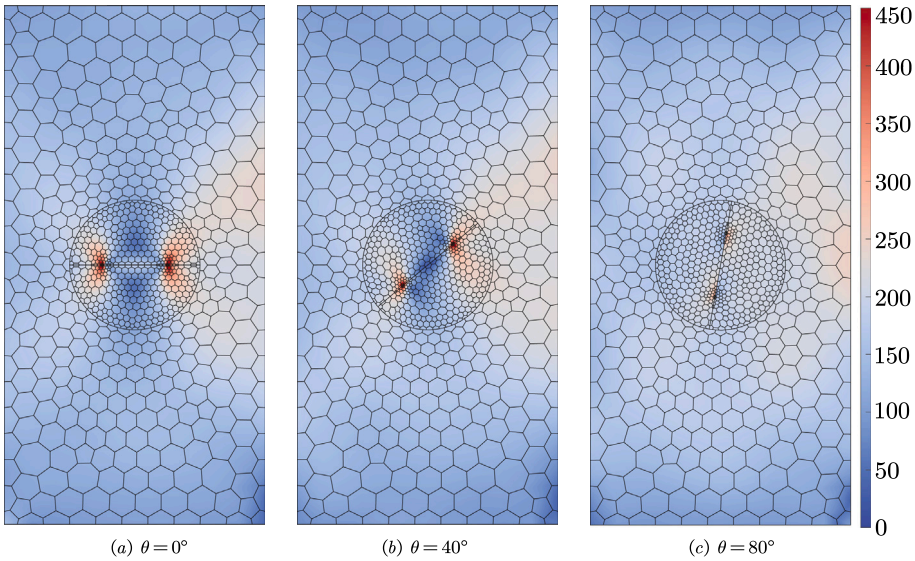

Fig. 18. Contour plots of von Mises stresses at $t = 0.8$ for different α.

5.4. Arbitrary geometry considering body force

For the last example, a two-dimensional animal-shaped subjected to volume forces is considered. The geometry and polygonal VEM mesh are shown in Fig. 19. In this work, downward and rightward body forces $g = 10$ are considered. The material parameters are assumed as $E = 200000$, $\nu = 0.3$, $\rho = 50$.

The space–time virtual element method is used with the mesh size $h = \Delta t = 2$ in time direction. The total calculation time is $t = 80$. The total degrees of freedom is 222 876. For the space–time domain $\Omega \times I$, the von Mises stresses for two different body forces are given in Fig. 20. Besides, the von Mises stresses for two different body forces at $t = 80$ are illustrated in Fig. 21.

6. Conclusion

In this work, a space–time virtual element method is presented for the discretization of two-dimensional elastodynamics based on Hamilton's principle of stationary action. In our space–time scheme, the virtual element method is used in the space dimension and the finite element method is used in the time dimension. Compared with other space–time formulation, the "structured" numerical approach avoids the introduction of additional degrees of freedom in the time direction and simplifies the calculations. Besides, arbitrary polygonal elements can be used in the space dimension, thus enhancing the flexibility of meshing. The results undering the advantages of high accuracy and stability for the new space–time formulation. For large-scale problems, the computational efficiency may be slow, so parallel technology and adaptive mesh are needed. The space–time scheme will be extended to nonlinear problems in the future. The virtual element method code and finite element method code involved in this work can be found in https://github.com/Qinxiaoye/VEM-spcae-time-dynamic.

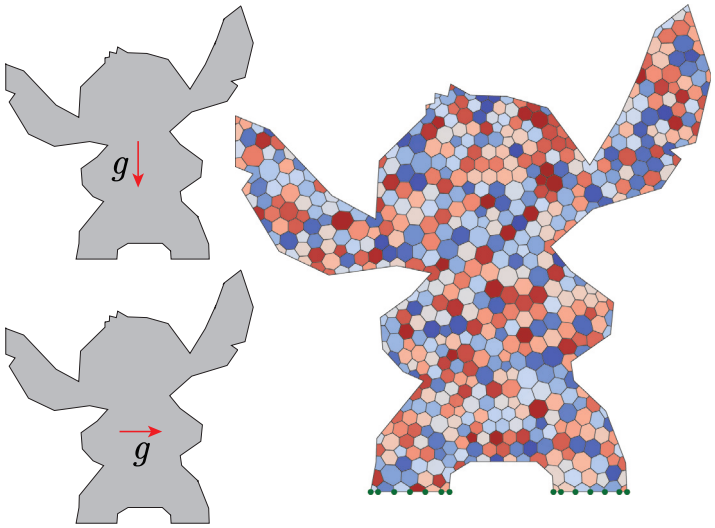

Fig. 19. Geometry and mesh of the animal shape plates.

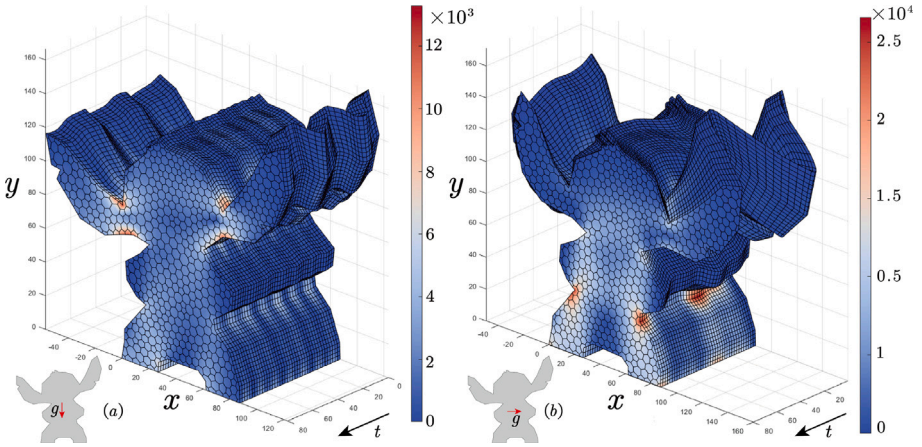

Fig. 20. Contour plots of von Mises stresses for different body force.

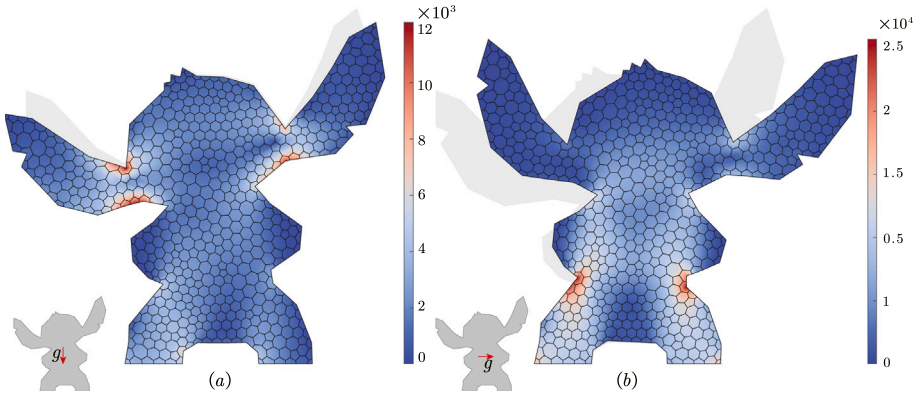

Fig. 21. Contour plots of von Mises stresses for different body force at $t = 80$.

CRediT authorship contribution statement

Bing-Bing Xu: Writing – original draft, Funding acquisition, Data curation. **Philipp Junker:** Writing – review & editing. **Peter Wriggers:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Bing-Bing Xu reports financial support was provided by Alexander von Humboldt Foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The first and last authors are grateful for the support provided by the Alexander von Humboldt Foundation, Germany.

Data availability

No data was used for the research described in the article.

References

[1] L. Beirao da Veiga, A. Cangiani, G. Manzini, L. Marini, A. Russo, Basic principles of virtual element methods, Math. Models Methods Appl. Sci. 23 (2012) 199–214, http://dx.doi.org/10.1142/S0218202512500492.
[2] L. Beirao da Veiga, F. Brezzi, L. Marini, A. Russo, Mixed virtual element methods for general second order elliptic problems on polygonal meshes, ESAIM Math. Model. Numer. Anal. 26 (2014) 727–747, http://dx.doi.org/10.1051/m2an/2015067.
[3] L. Beirao da Veiga, F. Brezzi, Virtual elements for linear elasticity problems, SIAM J. Numer. Anal. 51 (2) (2013) 794–812, http://dx.doi.org/10.1137/120874746.
[4] A. Gain, C. Talischi, G. Paulino, On the virtual element method for three-dimensional elasticity problems on arbitrary polyhedral meshes, Comput. Methods Appl. Mech. Engrg. 282 (2013) 132–160, http://dx.doi.org/10.1016/j.cma.2014.05.005.
[5] E. Artioli, L. Beirao da Veiga, C. Lovadina, E. Sacco, Arbitrary order 2D virtual elements for polygonal meshes: Part I, elastic problem, Comput. Mech. 60 (2017) 727–747, http://dx.doi.org/10.1007/s00466-017-1404-5.
[6] F. Dassi, C. Lovadina, M. Visinoni, A three-dimensional Hellinger-Reissner virtual element method for linear elasticity problems, Comput. Methods Appl. Mech. Engrg. 364 (2020) 112910, http://dx.doi.org/10.1016/j.cma.2020.112910.
[7] M. Mengolini, M. Benedetto, A. Aragón, An engineering perspective to the virtual element method and its interplay with the standard finite element method, Comput. Methods Appl. Mech. Engrg. 350 (2019) 995–1023, http://dx.doi.org/10.1016/j.cma.2019.02.043.
[8] A. Ortiz-Bernardin, R. Silva-Valenzuela, S. Salinas-Fernández, N. Hitschfeld, S. Luza, B. Rebolledo, A node-based uniform strain virtual element method for compressible and nearly incompressible elasticity, Internat. J. Numer. Methods Engrg. 124 (2022) http://dx.doi.org/10.1002/nme.7189.
[9] J. Zhao, B. Zhang, C. Shaochun, S. Mao, The morley-type virtual element for plate bending problems, J. Sci. Comput. 76 (2018) http://dx.doi.org/10.1007/s10915-017-0632-3.
[10] K. Sze, X. Liu, S. Lo, Popular benchmark problems for geometric nonlinear analysis of shells, Finite Elem. Anal. Des. 40 (11) (2004) 1551–1569, http://dx.doi.org/10.1016/j.finel.2003.11.001.
[11] T. Wu, P. Pimenta, P. Wriggers, On triangular virtual elements for Kirchhoff-Love shells, Arch. Appl. Mech. (2024) 1–34, http://dx.doi.org/10.1007/s00419-024-02591-9.
[12] H. Chi, L.B. da Veiga, G. Paulino, Some basic formulations of the virtual element method (VEM) for finite deformations, Comput. Methods Appl. Mech. Engrg. 318 (2016) 995–1023, http://dx.doi.org/10.1016/j.cma.2016.12.020.
[13] P. Wriggers, B. Reddy, W. Rust, B. Hudobivnik, Efficient virtual element formulations for compressible and incompressible finite deformations, Comput. Mech. 60 (2017) 995–1023, http://dx.doi.org/10.1007/s00466-017-1405-4.
[14] D. van Huyssteen, B. Reddy, A virtual element method for isotropic hyperelasticity, Comput. Methods Appl. Mech. Engrg. 367 (2020) 113134, http://dx.doi.org/10.1016/j.cma.2020.113134.
[15] B.-B. Xu, W.-L. Fan, P. Wriggers, High-order 3D virtual element method for linear and nonlinear elasticity, Comput. Methods Appl. Mech. Engrg. 431 (2024) 117258, http://dx.doi.org/10.1016/j.cma.2024.117258.
[16] B.-B. Xu, Y.-F. Wang, P. Wriggers, Stabilization-free virtual element method for 2D elastoplastic problems, Internat. J. Numer. Methods Engrg. (2024) http://dx.doi.org/10.1002/nme.7490.
[17] P. Wriggers, W. Rust, B. Reddy, A virtual element method for contact, Comput. Mech. 58 (2016) 995–1023, http://dx.doi.org/10.1007/s00466-016-1331-x.
[18] F. Aldakheel, B. Hudobivnik, E. Artioli, L. Beirao da Veiga, P. Wriggers, Curvilinear virtual elements for contact mechanics, Comput. Methods Appl. Mech. Engrg. 372 (2020) 113394, http://dx.doi.org/10.1016/j.cma.2020.113394.
[19] W. Shen, M. Ohsaki, J. Zhang, A 2-dimentional contact analysis using second-order virtual element method, Comput. Mech. 70 (2022) 995–1023, http://dx.doi.org/10.1007/s00466-022-02165-y.
[20] M. Cihan, B. Hudobivnik, J. Korelc, P. Wriggers, A virtual element method for 3D contact problems with non-conforming meshes, Comput. Methods Appl. Mech. Engrg. 402 (2022) 115385, http://dx.doi.org/10.1016/j.cma.2022.115385.
[21] P. Wriggers, B. Hudobivnik, A low order virtual element formulation for finite elasto-plastic deformations, Comput. Methods Appl. Mech. Engrg. 327 (2017) 4702–4725, http://dx.doi.org/10.1016/j.cma.2017.08.053.
[22] B. Hudobivnik, F. Aldakheel, P. Wriggers, A low order 3D virtual element formulation for finite elasto–plastic deformations, Comput. Mech. 63 (2019) 4702–4725, http://dx.doi.org/10.1007/s00466-018-1593-6.
[23] M. Cihan, B. Hudobivnik, F. Aldakheel, P. Wriggers, 3D mixed virtual element formulation for dynamic elasto-plastic analysis, Comput. Mech. 68 (2021) 1–18, http://dx.doi.org/10.1007/s00466-021-02010-8.
[24] P. Wriggers, F. Aldakheel, B. Hudobivnik, Virtual Element Methods in Engineering Sciences, Springer International Publishing, 2024, http://dx.doi.org/10.1007/978-3-031-39255-9.

[25] F. Dassi, A. Fumagalli, I. Mazzieri, A. Scotti, G. Vacca, A virtual element method for the wave equation on curved edges in two dimensions, J. Sci. Comput. 90 (2021) http://dx.doi.org/10.1007/s10915-021-01683-w.
[26] K. Park, H. Chi, G. Paulino, On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration, Comput. Methods Appl. Mech. Engrg. 356 (2019) 669–684, http://dx.doi.org/10.1016/j.cma.2019.06.031.
[27] K. Park, H. Chi, G. Paulino, Numerical recipes for elastodynamic virtual element methods with explicit time integration, Internat. J. Numer. Methods Engrg. 121 (2019) 1–31, http://dx.doi.org/10.1002/nme.6173.
[28] M. Cihan, B. Hudobivnik, F. Aldakheel, P. Wriggers, Virtual element formulation for finite strain elastodynamics, Comput. Model. Eng. Sci. 129 (2021) 1151–1180, http://dx.doi.org/10.32604/cmes.2021.016851.
[29] N. Sukumar, M. Tupek, Virtual elements on agglomerated finite elements to increase the critical time step in elastodynamic simulations, Internat. J. Numer. Methods Engrg. 123 (2022) 4702–4725, http://dx.doi.org/10.1002/nme.7052.
[30] J.H. Argyris, D.W. Scharpf, Finite elements in time and space, Aeronaut. J. 73 (708) (1969) 1041–1044, http://dx.doi.org/10.1017/S0001924000051198.
[31] P. Jamet, Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain, SIAM J. Numer. Anal. 15 (5) (1978) 912–928, http://dx.doi.org/10.1137/0715059.
[32] I. Babuska, T. Janik, The h-p version of the finite element method for parabolic equations. Part I. The p-version in time, Numer. Methods Partial Differential Equations 5 (4) (1989) 363–399, http://dx.doi.org/10.1002/num.1690050407.
[33] T.J. Hughes, J.R. Stewart, A space-time formulation for multiscale phenomena, J. Comput. Appl. Math. 74 (1) (1996) 217–229, http://dx.doi.org/10.1016/0377-0427(96)00025-8.
[34] T.J. Hughes, G.M. Hulbert, Space-time finite element methods for elastodynamics: Formulations and error estimates, Comput. Methods Appl. Mech. Engrg. 66 (3) (1988) 339–363, http://dx.doi.org/10.1016/0045-7825(88)90006-0.
[35] C. Hesch, S. Schuss, M. Dittmann, S. Eugster, M. Favino, R. Krause, Variational space–time elements for large-scale systems, Comput. Methods Appl. Mech. Engrg. 326 (2017) 541–572, http://dx.doi.org/10.1016/j.cma.2017.08.020.
[36] S. Schuss, S. Glas, C. Hesch, Non-linear space-time elasticity, Internat. J. Numer. Methods Engrg. 124 (2023) http://dx.doi.org/10.1002/nme.7194.
[37] P. Bansal, A. Moiola, I. Perugia, C. Schwab, Space–time discontinuous Galerkin approximation of acoustic waves with point singularities, IMA J. Numer. Anal. 41 (3) (2020) 2056–2109, http://dx.doi.org/10.1093/imanum/draa088.
[38] P.F. Antonietti, I. Mazzieri, F. Migliorini, A space-time discontinuous Galerkin method for the elastic wave equation, J. Comput. Phys. 419 (2020) 109685, http://dx.doi.org/10.1016/j.jcp.2020.109685.
[39] T.J. Hughes, L.P. Franca, G.M. Hulbert, A new finite element formulation for computational fluid dynamics: VIII. The galerkin/least-squares method for advective-diffusive equations, Comput. Methods Appl. Mech. Engrg. 73 (2) (1989) 173–189, http://dx.doi.org/10.1016/0045-7825(89)90111-4.
[40] F. Shakib, T.J. Hughes, A new finite element formulation for computational fluid dynamics: Ix. Fourier analysis of space-time Galerkin/least-squares algorithms, Comput. Methods Appl. Mech. Engrg. 87 (1) (1991) 35–58, http://dx.doi.org/10.1016/0045-7825(91)90145-V.
[41] P. Junker, D. Balzani, An extended hamilton principle as unifying theory for coupled problems and dissipative microstructure evolution, Contin. Mech. Thermodyn. 33 (2021) http://dx.doi.org/10.1007/s00161-021-01017-z.
[42] P. Junker, T. Wick, Space-time variational material modeling: a new paradigm demonstrated for thermo-mechanically coupled wave propagation, visco-elasticity, elasto-plasticity with hardening, and gradient-enhanced damage, Comput. Mech. 73 (2023) 1–38, http://dx.doi.org/10.1007/s00466-023-02371-2.
[43] U. Langer, S.E. Moore, M. Neumüller, Space-time isogeometric analysis of parabolic evolution problems, Comput. Methods Appl. Mech. Engrg. 306 (2016) 342–363, http://dx.doi.org/10.1016/j.cma.2016.03.042.
[44] G. Loli, G. Sangalli, P. Tesini, High-order spline upwind for space–time isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 417 (2023) 116408, http://dx.doi.org/10.1016/j.cma.2023.116408.
[45] S. Gómez, L. Mascotto, I. Perugia, Design and performance of a space–time virtual element method for the heat equation on prismatic meshes, Comput. Methods Appl. Mech. Engrg. 418 (2024) 116491, http://dx.doi.org/10.1016/j.cma.2023.116491.
[46] S. Gomez, L. Mascotto, A. Moiola, I. Perugia, Space-time virtual elements for the heat equation, SIAM J. Numer. Anal. 62 (1) (2024) 199–228, http://dx.doi.org/10.1137/22M154140X.
[47] P.F. Antonietti, F. Bonizzoni, M. Verani, A cVEM-DG space-time method for the dissipative wave equation, Comput. Math. Appl. 152 (2023) 341–354, http://dx.doi.org/10.1016/j.camwa.2023.10.022.
[48] P. Wriggers, P. Junker, On a space-time implementation of the wave equation using virtual elements, Comput. Mech. (2024) 1–15, http://dx.doi.org/10.1007/s00466-024-02556-3.
[49] B. Ahmad, A. Alsaedi, F. Brezzi, L. Marini, A. Russo, Equivalent projectors for virtual element methods, Comput. Math. Appl. 66 (2013) 376–391, http://dx.doi.org/10.1016/j.camwa.2013.05.015.