Computer Methods in Applied Mechanics and Engineering 433 (2025) 117461

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Particle Virtual Element Method (PVEM): an agglomeration technique for mesh optimization in explicit Lagrangian free-surface fluid modelling

Cheng Fu [a], Massimiliano Cremonesi [a,*], Umberto Perego [a], Blaž Hudobivnik [b], Peter Wriggers [b]

[a] Department of Civil and Environmental Engineering, Politecnico di Milano, piazza Leonardo da Vinci, 32, Milano, 20133, Italy
[b] Institute for Continuum Mechanics, Leibniz Universität, An der Universität 1, building 8142, Hannover, 30823, Germany

ARTICLE INFO

Keywords:
Agglomeration
Virtual Element Method
PFEM
Mesh optimization
Free-surface flows

ABSTRACT

Explicit solvers are commonly used for simulating fast dynamic and highly nonlinear engineering problems. However, these solvers are only conditionally stable, requiring very small time-step increments determined by the characteristic length of the smallest, and often most distorted, element in the mesh. In the Lagrangian description of fluid motion, the computational mesh quickly deteriorates. To circumvent this problem, the Particle Finite Element Method (PFEM) creates a new mesh (e.g., through a Delaunay tessellation, based on node positions) when the current one becomes overly distorted. A fast and efficient remeshing technique is therefore of pivotal importance for an effective PFEM implementation in explicit dynamics. Unfortunately, the 3D Delaunay tessellation does not guarantee well-shaped elements, often generating zero- or near-zero-volume elements (slivers), which drastically reduce the stable time-step size. Available mesh optimization techniques have limited applicability due to their high computational cost when runtime remeshing is required. An innovative possibility to overcome this problem is the use of the Virtual Element Method (VEM), a variant of the finite element method that can make use of polyhedral elements of arbitrary shapes and number of nodes. This paper presents the formulation of a 3D first-order Particle Virtual Element Method (PVEM) for weakly compressible flows. Starting from a tetrahedral mesh, poorly shaped elements, such as slivers, are agglomerated to form polyhedral Virtual Elements (VEs) with a controlled characteristic length. This approach ensures full control over the minimum time-step size in explicit dynamics simulations, maintaining stability throughout the entire analysis.

1. Introduction

In Lagrangian finite element approaches to fluid dynamics problems, the computational mesh moves with the fluid particles providing an effective framework for handling rapidly evolving fluid domains, such as those in free-surface flows and fluid–structure interactions. Additionally, Lagrangian methods avoid the complexities associated with the nonlinearity of the convective term. However, one of the primary challenges in these methods is controlling mesh distortion [1], particularly in large-deformation problems like fluid flows, where the mesh can quickly deteriorate, leading to reduced computational accuracy and efficiency.

[*] Corresponding author.
E-mail addresses: cheg.fu@polimi.it (C. Fu), massimiliano.cremonesi@polimi.it (M. Cremonesi), umberto.perego@polimi.it (U. Perego), wriggers@ikm.uni-hannover.de (P. Wriggers).

https://doi.org/10.1016/j.cma.2024.117461
Received 2 August 2024; Received in revised form 9 October 2024; Accepted 10 October 2024

Among the Lagrangian mesh-based techniques for fluid modelling, the Particle Finite Element Method (PFEM) addresses mesh distortion during runtime by creating new ones through modifications of nodal connectivities, while preserving nodal positions. The PFEM has been successfully applied to a wide range of engineering applications, including free-surface flows [2,3], fluid–structure interaction problems [4–6], granular [7,8], multi-fluid flows [9,10], industrial forming processes [11,12] and thermal coupled simulations [13,14].

To maintain good mesh quality, PFEM employs continuous remeshing using a combination of Delaunay tessellation and the 'alpha-shape' method [15]. However, the issue of mesh distortion becomes particularly intricate when employing an explicit time integration scheme, which is usually preferred in fast dynamics analyses. While in implicit methods, thanks to the unconditional stability, large time steps can be used even in the presence of poorly shaped elements, in conditionally stable explicit time integration schemes, small time steps, estimated by the smallest element characteristic length, are required. As a result, distorted elements can drastically reduce the stable time step, making the efficiency of runtime remeshing crucial. Unfortunately, while the 2D Delaunay tessellation guarantees optimal geometrical properties of the created triangles, the 3D version is less robust as it allows for the generation of zero- or almost zero-volume elements (slivers), leading to vanishing time-step increments.

Several mesh optimization algorithms have been proposed in the literature for the treatment of badly shaped elements (e.g. see [16–18]). While these methods have been successful in many applications, they are often computationally expensive when applied to runtime remeshing and are less effective when handling constrained boundary nodes. To overcome these difficulties, this work combines the PFEM with the Virtual Element Method (VEM) [19,20] resulting in a mesh of finite elements that includes, where needed, Virtual Elements (VEs) characterized by a larger characteristic length. The VEM can be interpreted as an extension of the classical Finite Element Method (FEM) allowing for the adoption of elements with an arbitrary number of nodes and shapes (polygons in 2D and polyhedra in 3D), possibly non-convex, conferring high flexibility to the meshing process.

The VEM was originally proposed by Beirão Da Veiga et al. [19,20] for the solution of Poisson's equations. Soon after, it has been extended to the analyses of linear elastic and inelastic problems [21–25]. Moreover, the VEM has been successfully adopted in a vast range of engineering applications such as elastodynamics [26–28], contact [29–32] and crack propagation [33,34], finite elasto-plastic deformation [35,36], incompressible solids [37–39] and Navier–Stokes problems [40,41].

The idea of exploiting the agglomeration capability of the VEM, originally proposed in [42] for solid problems, is here applied to fluid problems. Starting from an initial mesh of standard linear tetrahedral finite elements, larger polyhedral VEs with triangular faces are obtained through the agglomeration of badly shaped tetrahedra, achieving a good mesh quality to be used in explicit dynamics. This method offers a computationally efficient alternative to other smoothing techniques, effectively removing slivers and ensuring an optimal time step. The resulting mesh consists of standard tetrahedra and polyhedral VEs with triangular faces where needed. This variant of the PFEM, referred to as the Particle Virtual Element Method (PVEM), significantly increases the critical time-step size with minimal agglomeration cost, reducing the overall computational burden. By controlling the number of agglomerated elements, the method ensures a prescribed minimum characteristic length, enabling a nearly constant time-step size throughout the analysis. In particular, by tuning the number of elements to be agglomerated, a prescribed minimum characteristic length of the resulting agglomerated VE can be obtained, allowing for an almost constant time-step size throughout the analysis duration.

The paper is organized as follows. The governing equations for weakly compressible fluids are recalled in Section 2, together with the space and time discretization, whereas the PFEM is briefly introduced in Section 3. The VEM for fluids, formulated in the mixed velocity–pressure framework is discussed in Section 4, while the proposed PVEM agglomeration technique is presented in Section 5. The PVEM agglomeration is validated in Section 6 through 2D and 3D numerical examples. Finally, conclusions are drawn in Section 7.

2. Governing equations of weakly compressible fluids

Let $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x}, t)$ be the velocity field, $p = p(\boldsymbol{x}, t)$ the fluid pressure, $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{x}, t)$ the Cauchy stress tensor and ρ the fluid density, constant in time and uniform in all the domain. Being $\rho\boldsymbol{b}$ the fluid external body force per unit of volume and K the fluid bulk modulus, the Lagrangian Navier–Stokes equations in the time interval $[0, T]$ for an evolving fluid domain Ω^t are written as:

$$\rho \frac{d\boldsymbol{v}}{dt} = \nabla_x \cdot \boldsymbol{\sigma} + \rho\boldsymbol{b} \quad \text{in } \Omega^t \times [0, T] \tag{1}$$

$$\frac{dp}{dt} + K(\nabla_x \cdot \boldsymbol{v}) = 0 \quad \text{in } \Omega^t \times [0, T] \tag{2}$$

where ∇_x is the gradient operator. Eq. (2) establishes a linear relation between the pressure rate and the rate of the fluid infinitesimal volumetric strain, consistent with the assumption of a weakly compressible fluid. The Cauchy stress tensor $\boldsymbol{\sigma}$ can be decomposed into its isotropic and deviatoric parts:

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau} \tag{3}$$

where $\boldsymbol{I}$ denotes the identity tensor and $\boldsymbol{\tau}$ the deviatoric stress tensor. $\boldsymbol{\tau}$ is related to the deviatoric strain rate $\dot{\boldsymbol{\epsilon}}$, defined as

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}(\nabla_x \boldsymbol{v} + \nabla_x \boldsymbol{v}^T) - \frac{1}{3}(\nabla_x \cdot \boldsymbol{v})\boldsymbol{I} \tag{4}$$

by the fluid viscosity $\mu(\dot{\boldsymbol{\epsilon}})$:

$$\boldsymbol{\tau} = 2\mu(\dot{\boldsymbol{\epsilon}})\dot{\boldsymbol{\epsilon}} \tag{5}$$

To complete the problem definition, the set of Eqs. (1) and (2) have to be complemented with appropriate initial and boundary conditions. Denoting $\Gamma^t = \partial\Omega^t$ the fluid boundary at time t, subdivided into non-overlapping Dirichlet Γ^t_D and Neumann Γ^t_N parts, the following conditions are imposed:

$$\boldsymbol{v}(\boldsymbol{x},t) = \bar{\boldsymbol{v}}(\boldsymbol{x},t) \quad \text{on } \Gamma^t_D \tag{6}$$

$$\boldsymbol{\sigma}(\boldsymbol{x},t)\cdot\boldsymbol{n} = \boldsymbol{h}(\boldsymbol{x},t) \quad \text{on } \Gamma^t_N \tag{7}$$

being $\bar{\boldsymbol{v}}(\boldsymbol{x},t)$ a given velocity function, $\boldsymbol{h}(\boldsymbol{x},t)$ the applied surface tractions and $\boldsymbol{n}$ the outward normal to the boundary Γ^t_N. The initial conditions can be expressed as:

$$\boldsymbol{v}(\boldsymbol{x},t=0) = \boldsymbol{v}_0(\boldsymbol{x}) \quad \text{in } \Omega^0 \tag{8}$$

$$p(\boldsymbol{x},t=0) = p_0(\boldsymbol{x}) \quad \text{in } \Omega^0 \tag{9}$$

where $\boldsymbol{v}_0$ and p_0 are given functions defining the values of velocity and pressure at the initial time.

2.1. Space discretization

A Galerkin finite element approach is adopted for the space discretization of Eqs. (1)–(2). Let S^v and S^p be the spaces of admissible functions for velocities and pressure satisfying homogeneous boundary conditions on Γ^t_D and Γ^t_N, respectively. The weak form of the momentum conservation can be obtained by multiplying Eq. (1) by a vector of test functions $\boldsymbol{w} \in S^v$. Integrating the obtained equations and exploiting Green's formula, the following expression is obtained:

$$\int_{\Omega^t} \rho\boldsymbol{w}\cdot\frac{d\boldsymbol{v}}{dt}d\Omega = \int_{\Omega^t} p(\nabla_x\cdot\boldsymbol{w})d\Omega - \int_{\Omega^t}\nabla_x\boldsymbol{w} : \boldsymbol{\tau}d\Omega + \int_{\Omega^t}\boldsymbol{w}\cdot\rho\boldsymbol{b}\,d\Omega + \int_{\Gamma^t}\boldsymbol{w}\cdot\boldsymbol{h}d\Gamma \quad \forall\boldsymbol{w}\in S^v \tag{10}$$

Denoting by q a scalar test function, with $q \in S^p$, the weak form of the pressure Eq. (2) is written as:

$$\int_{\Omega^t} q\frac{dp}{dt}d\Omega + \int_{\Omega^t} qK_f(\nabla_x\cdot\boldsymbol{v})d\Omega = 0 \quad \forall q \in S^p \tag{11}$$

The unknown velocity and pressure fields are modelled over each finite element in terms of their nodal values, gathered in vectors $\boldsymbol{V}(t)$ and $\boldsymbol{P}(t)$, respectively:

$$\boldsymbol{v}(\boldsymbol{x},t) = \boldsymbol{N}^v(\boldsymbol{x})\boldsymbol{V}(t) \tag{12}$$

$$\boldsymbol{p}(\boldsymbol{x},t) = \boldsymbol{N}^p(\boldsymbol{x})\boldsymbol{P}(t) \tag{13}$$

where $\boldsymbol{N}^v(\boldsymbol{x})$ and $\boldsymbol{N}^p(\boldsymbol{x})$ contain linear shape functions. At a given instant t, the semi-discretized momentum and pressure equations read:

$$\boldsymbol{M}_v\frac{d\boldsymbol{V}}{dt} = -\boldsymbol{K}_\mu\boldsymbol{V} + \boldsymbol{D}^T\boldsymbol{P} + \boldsymbol{F}_{ext} \tag{14}$$

$$\boldsymbol{M}_p\frac{d\boldsymbol{P}}{dt} = -K\boldsymbol{D}\boldsymbol{V} \tag{15}$$

where $\boldsymbol{M}_v$ is the fluid mass matrix, $\boldsymbol{M}_p$ is a volume-like matrix, $\boldsymbol{K}_\mu$ is the viscosity matrix, $\boldsymbol{D}$ is the matrix discretization of the divergence operator and the vector $\boldsymbol{F}_{ext}$ contains equivalent nodal external forces.

The use of equal order interpolation for velocity and pressure violates the LBB (Ladyzhenskaya–Babuška–Brezzi) condition, leading to possible spurious pressure oscillations. Therefore, the formulation has to be properly stabilized. In the present work, following the Direct Pressure Stabilization (DPS) method [43,44], a new term based on the L2 projection of the linear pressure field onto a constant has been added to Eq. (2)

$$\int_{\Omega^t} q\frac{dp}{dt}d\Omega + \int_{\Omega^t_f} qK_f(\nabla_x\cdot\boldsymbol{v})d\Omega + \int_{\Omega^t}(p-\tilde{\Pi}_p)(q-\tilde{\Pi}_q)d\Omega = 0 \quad \forall q \in S^p \tag{16}$$

where the projection operator $\tilde{\Pi}$ is defined by the following condition:

$$\int_{\Omega^t}\tilde{\Pi}_q(p-\tilde{\Pi}_p)d\Omega = 0 \tag{17}$$

The projection operator applied to the linear pressure field provides a constant value in each element:

$$\tilde{\Pi}_p = \tilde{\boldsymbol{N}}^p\boldsymbol{P}_e \tag{18}$$

where $\tilde{\boldsymbol{N}}^p$ are constant shape functions over the element. Applying the pressure spatial discretization, the stabilizing term in Eq. (16) evaluated for each element e reads:

$$\int_{\Omega^t_e}(\boldsymbol{N}^{p^T}\boldsymbol{N}^p - \tilde{\boldsymbol{N}}^{p^T}\tilde{\boldsymbol{N}}^p)d\Omega\,\boldsymbol{P}_e = \boldsymbol{S}_e\boldsymbol{P}_e \tag{19}$$

When a 2D 3-nodes triangular element e is considered, $\tilde{N}^p$ is defined as $\tilde{\boldsymbol{N}}^p = [1/3, 1/3, 1/3]$ leading to a stabilization matrix $\boldsymbol{S}_e$ defined as follows:

$$\int_{\Omega_e^t} (\boldsymbol{N}^{p^T} \boldsymbol{N}^p - \tilde{\boldsymbol{N}}^{p^T} \tilde{\boldsymbol{N}}^p) d\Omega \, \boldsymbol{P}_e = \boldsymbol{S}_e \boldsymbol{P}_e = \frac{\Omega_e}{36} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \boldsymbol{P}_e \tag{20}$$

Ω_e being the element area. Finally, the mass conservation Eq. (15) is rewritten as

$$\boldsymbol{M}_p \frac{d\boldsymbol{P}}{dt} = -K_f \boldsymbol{D}\boldsymbol{V} - \frac{1}{\omega} \boldsymbol{S}\boldsymbol{P} \tag{21}$$

where ω is a stabilization parameter with the dimension of a time, typically chosen as the current time-step size.

2.2. Time discretization

The discretization in time of Eqs. (10)–(11) is obtained by dividing the time domain $[0, T]$ into N time steps $\Delta t^{n+1} = t^{n+1} - t^n$, such that $T = \sum_{n=1}^{N} \Delta t^n$. Adopting an explicit time integration with the Central Difference Scheme (CDS), the mid-step velocity $\boldsymbol{V}^{n+\frac{1}{2}}$ is computed as:

$$\boldsymbol{V}^{n+\frac{1}{2}} = \boldsymbol{V}^n + \frac{1}{2} \boldsymbol{A}^n \Delta t^{n+1} \tag{22}$$

being $\boldsymbol{A}^n$ the vector of nodal accelerations computed at the previous step. Using the mid-step velocity, the nodal displacements $\boldsymbol{U}$ are then evaluated:

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \boldsymbol{V}^{n+\frac{1}{2}} \Delta t^{n+1} \tag{23}$$

and the coordinates of the nodes are updated. Once the new configuration at t^{n+1} is known, pressures can be obtained by solving the pressure equation:

$$\boldsymbol{M}_p \boldsymbol{P}^{n+1} = \boldsymbol{M}_p \boldsymbol{P}^n - \boldsymbol{S}^n \boldsymbol{P}^n - \Delta t^{n+1} K_f \boldsymbol{D} \boldsymbol{V}^{n+\frac{1}{2}} \tag{24}$$

where $\boldsymbol{S}^n$ is the stabilization matrix defined in the previous section and the stabilization parameter ω in (21) has been set equal to Δt^{n+1}. Then the vector of nodal resultant forces $\boldsymbol{F}^{n+1}$ (the right-hand side of Eq. (14)) and the diagonal mass matrix $\boldsymbol{M}_v^{n+1}$ are computed in the updated configuration, allowing for the solution of the momentum equation to find the new nodal accelerations $\boldsymbol{A}^{n+1}$:

$$\boldsymbol{A}^{n+1} = (\boldsymbol{M}_v^{n+1})^{-1} \boldsymbol{F}^{n+1} \tag{25}$$

Finally, the end-step velocities can be computed as:

$$\boldsymbol{V}^{n+1} = \boldsymbol{V}^{n+\frac{1}{2}} + \frac{1}{2} \boldsymbol{A}^{n+1} \Delta t^{n+1} \tag{26}$$

Due to the conditional stability of the CDS, the time-step size Δt^{n+1} is defined adaptively to guarantee the respect of the Courant–Friedrichs–Lewy (CFL) condition [45]:

$$\Delta t^{n+1} = C_N \min_e \left(\frac{h_e^{n+1}}{c} \right) \tag{27}$$

where h_e denotes the characteristic length (e.g. radius of the circumcircle in 2D or circumsphere in 3D) of the eth element, c is the speed of the dilatational wave in the fluid and C_N represents the Courant number, assumed to be $C_N = 0.9$.

3. Particle Finite Element Method (PFEM)

The PFEM is a Lagrangian mesh-based finite element method particularly effective for solving fluid flow problems characterized by free surfaces, breaking waves and evolving interfaces [1–3,46]. Thanks to its ability to inexpensively track free surfaces and interfaces throughout the analysis, the method is particularly appealing for fluid–structure interaction problems [4–6]. In the Lagrangian description, the finite element mesh follows the movements of the fluid so that the PFEM requires the generation of a new mesh whenever the current one becomes too distorted. The remeshing process has to be fast since it is performed frequently during the analysis. Moreover, it has to be accurate to provide a correct description of the fluid domain, tracking the rapid evolution of the boundaries. Remeshing is performed through application of the Delaunay tessellation [47], which changes continuously the nodal connectivities preserving nodal positions. All physical properties and variables are stored at nodal level so that no information is lost when the current mesh is deleted. To avoid expensive mapping from the old mesh to the new one, linear elements (triangles in 2D and tetrahedra in 3D) with vertex nodes are used [1]. Fig. 1-b shows an example of 2D Delaunay tessellation of the cloud of points in Fig. 1-a. However, the Delaunay algorithm generates the convex hull containing a given set of nodes, see Fig. 1-b, which generally does not correspond to the real fluid domain. To overcome this issue, the alpha-shape method, a technique to remove the

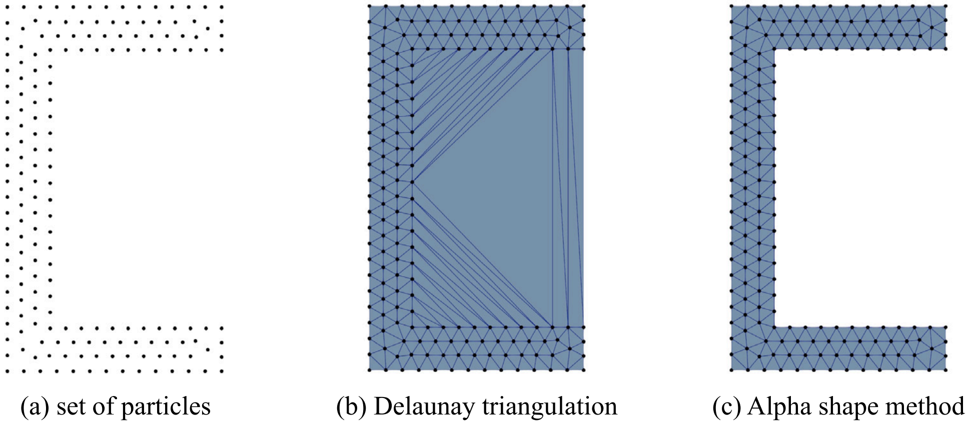


Fig. 1. Mesh generation in 2D PFEM. Starting from a cloud of fluid particles (a), a convex domain is created by Delaunay tessellation (b). The correct fluid domain is then recovered through the application of the alpha-shape method that removes non-physical/distorted elements (c).

unphysical elements, is applied [1,3,15]. According to the alpha-shape method, a geometrical distortion index α_e is introduced at the element level:

$$\alpha_e = \frac{R_e}{h_{mean}} \tag{28}$$

where R_e denotes the radius of the circumcircle (2D) or circumsphere (3D) to the element e, whereas h_{mean} represents the mesh characteristic length, chosen as the average of the minimum element edges in the initial mesh. To remove the distorted elements from the Delaunay mesh, a threshold value $\bar{\alpha}$ is introduced. An element is considered overly distorted if it has a distortion index larger than the user-defined threshold, that is:

$$\alpha_e \geq \bar{\alpha} \qquad \text{or} \qquad R_e \geq \bar{\alpha} h_{mean} \tag{29}$$

Elements satisfying relation (29) are deleted from the mesh, see Fig. 1-c. A proper choice of the parameter $\bar{\alpha}$ plays a key role in the definition of the correct fluid domain. When $\bar{\alpha}$ assumes a high value, a larger number of elements is preserved. This can lead to an inaccurate fluid profile and in the extreme case, it restores the Delaunay mesh. Conversely, if $\bar{\alpha}$ is excessively small, only a few elements are retained, resulting in a loss of smoothness in the fluid profile or in the appearance of unphysical voids. For a more flexible identification of the fluid domain, different threshold values may be defined in different mesh regions, e.g. $\bar{\alpha}_{int}$ for inner portions and $\bar{\alpha}_{surf}$ for free surfaces. The former is employed to control distortion within the fluid bulk and it is typically assigned a higher value to prevent the formation of non-physical voids inside the domain. The latter is used instead to eliminate unrealistic elements at the free surface and it is usually assigned a smaller value to ensure a more accurate fluid profile. In the numerical examples analysed in Section 6, the two $\bar{\alpha}$ parameters are chosen as $\bar{\alpha}_{int} = 2$ and $\bar{\alpha}_{surf} = 1.2$.

4. Particle Virtual Element Method (PVEM) for weakly compressible fluids

The Virtual Element Method (VEM) is a variant of the finite element method designed for the solution of differential problems on polytopal meshes (polygons in 2D and polyhedra in 3D). The main idea of the VEM is the implicit definition (hence the term *virtual*) of the space of the local shape functions in each element. In the present Lagrangian weakly compressible fluid context, a first-order VEM is considered. The obtained method will be referred to as a Particle Virtual Element Method (PVEM).

Following [19,20], the key idea of the VEM is to replace the functions $\boldsymbol{v}$ and p with their projections $\boldsymbol{v}_\pi$ and p_π, onto a polynomial space. However, the projected parts lead to a rank-deficient problem and a stabilization has to be included in the formulation.

4.1. Projection step

In the PVEM, the variables $\boldsymbol{v}$ and p are collected in a vector $\boldsymbol{\Phi}$ defined as:

$$\boldsymbol{\Phi} = \begin{Bmatrix} \boldsymbol{v} \\ p \end{Bmatrix} = \begin{Bmatrix} v_x \\ v_y \\ v_z \\ p \end{Bmatrix} \tag{30}$$

The variables in $\boldsymbol{\Phi}$ are split into the projected part $\boldsymbol{\Phi}_{\boldsymbol{\pi}}$ and a remainder (the error between the discrete approximate solution and its polynomial approximation):

$$\boldsymbol{\Phi} = \boldsymbol{\Phi}_\pi + (\boldsymbol{\Phi} - \boldsymbol{\Phi}_{\boldsymbol{\pi}}), \qquad \boldsymbol{\Phi}_\pi = \{\boldsymbol{v}_\pi, p_\pi\} \tag{31}$$

Within the projection step, the variables are mapped onto a polynomial space of degree k. In the present work, linear ansatz functions are used, so that only $k = 1$ is considered. At the elemental level, the $\boldsymbol{\Phi}_\pi$ assumes the form:

$$\boldsymbol{\Phi}_\pi = \begin{Bmatrix} v_{x\pi} \\ v_{y\pi} \\ v_{z\pi} \\ p_\pi \end{Bmatrix} = \boldsymbol{a} \begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} \tag{32}$$

where x, y, z are element local coordinates. Let $\hat{\boldsymbol{\Phi}}$ be the vector of nodal degrees of freedom

$$\hat{\boldsymbol{\Phi}} = \begin{Bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{Bmatrix} \tag{33}$$

The projection $\boldsymbol{\Phi}_\pi$ is fully defined once the matrix $\boldsymbol{a}$, containing the virtual parameters, is known in terms of the nodal degrees of freedom $\hat{\boldsymbol{\Phi}}$. To determine $\boldsymbol{a}$, the following orthogonality condition between the gradient of an arbitrary first-order polynomial $\tilde{\boldsymbol{\Phi}}$, and the gradient of the ansatz remainder [26] is imposed

$$\int_{\Omega_e} \nabla\tilde{\boldsymbol{\Phi}}(\nabla\boldsymbol{\Phi}_\pi - \nabla\boldsymbol{\Phi})\, d\Omega_e = 0 \tag{34}$$

When employing linear shape functions, both $\nabla\tilde{\boldsymbol{\Phi}}$ and $\nabla\boldsymbol{\Phi}_\pi$ are constant at the elemental level, leading to

$$\nabla\boldsymbol{\Phi}_\pi = \frac{1}{\Omega_e} \int_{\Omega_e} \nabla\boldsymbol{\Phi}\, d\Omega_e \tag{35}$$

Since $\boldsymbol{\Phi}$ is not known inside the element e, the integral on the right-hand side is not computable and, hence, it is transformed into a boundary integral as:

$$\nabla\boldsymbol{\Phi}_\pi = \frac{1}{\Omega_e} \int_{\partial\Omega_e} \boldsymbol{\Phi} \otimes \boldsymbol{n}\, d\Gamma_e \tag{36}$$

where $\boldsymbol{n}$ represents the outward normal to the boundary $\Gamma_e = \partial\Omega_e$ of VE e. Since a linear polynomial space is considered, the gradients of the projected velocity and pressure fields are constant within each element and, with a minor abuse of notation, $\nabla\boldsymbol{\Phi}_\pi$ becomes:

$$\nabla\boldsymbol{\Phi}_\pi = \begin{bmatrix} a_{12} & a_{13} & a_{14} \\ a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} \end{bmatrix} \tag{37}$$

Eq. (36) allows for the computation of the virtual parameters related to the gradient of the projected variables, i.e. of the coefficients of the linear part only of $\boldsymbol{\Phi}_\pi$. To ensure the uniqueness of $\boldsymbol{\Phi}_\pi$, the coefficients $a_{11}, a_{21}, a_{31}, a_{41}$ of the constant part have also to be defined. A possibility is to impose that the average of the vertex values of the projection $\boldsymbol{\Phi}_\pi$ is equal to the average of the vertex values of the virtual function $\hat{\boldsymbol{\Phi}}$:

$$\frac{1}{n_V} \sum_{I=1}^{n_V} \boldsymbol{\Phi}_\pi(\boldsymbol{X}_I) = \frac{1}{n_V} \sum_{I=1}^{n_V} \hat{\boldsymbol{\Phi}}_I \tag{38}$$

where n_V is the number of element vertices, $\boldsymbol{X}_I$ are the coordinates of node I and $\hat{\boldsymbol{\Phi}}_I$ is the vector of velocity and pressure at node I. By substituting Eq. (32) into Eq. (38) and noting that one can write

$$\boldsymbol{a} = \left[\begin{array}{c|c} \begin{matrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{matrix} & \nabla\boldsymbol{\Phi}_\pi \end{array}\right] \Rightarrow \boldsymbol{\Phi}_\pi(\boldsymbol{X}_I) = \boldsymbol{a} \begin{Bmatrix} 1 \\ \hline \boldsymbol{X}_I \end{Bmatrix} = \begin{Bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{Bmatrix} + \nabla\boldsymbol{\Phi}_\pi \boldsymbol{X}_I \tag{39}$$

the remaining parameters can be evaluated as:

$$\{a_{11}, a_{21}, a_{31}, a_{41}\}^T = \frac{1}{n_V} \sum_{I=1}^{n_V} (\hat{\boldsymbol{\Phi}}_I - \nabla\boldsymbol{\Phi}_\pi\, \boldsymbol{X}_I) \tag{40}$$

Through Eqs. (36) and (40), the projected velocity $\boldsymbol{v}_\pi$ and pressure p_π of the VE e are completely defined in terms of nodal values $\hat{\boldsymbol{\Phi}}$ and $\boldsymbol{\Phi}_\pi$ can be expressed as

$$\boldsymbol{\Phi}_\pi(x, y, z) = \begin{Bmatrix} v_\pi \\ p_\pi \end{Bmatrix} = \left[\begin{array}{c|c} \Pi_v & 0 \\ \hline 0 & \Pi_p \end{array}\right] \begin{Bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{Bmatrix} = \Pi(x, y, z)\hat{\boldsymbol{\Phi}} \tag{41}$$

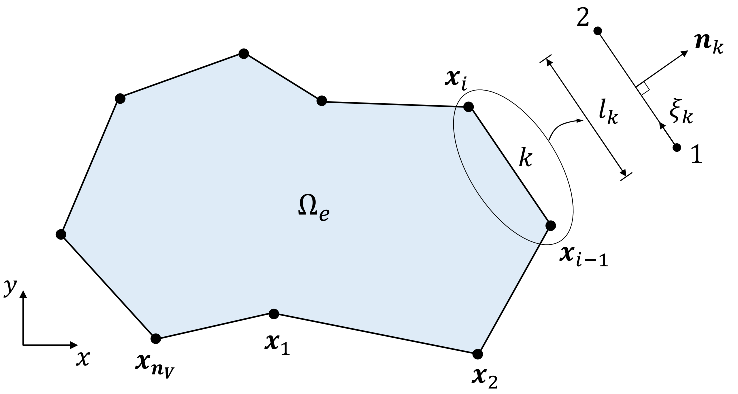


Fig. 2. Velocity modelling on boundary of 2D VE with n_V nodes.

4.2. Computation of the boundary integral

In 2D problems, the right-hand side of Eq. (36) can be evaluated easily since the element boundaries are straight segments. Using linear shape functions, velocities and pressures along edges of the element are explicitly computable through nodal values $u_{x_1}, u_{y_1}, u_{x_2}, u_{y_2}, p_1, p_2$:

$$\boldsymbol{\Phi}_k = \begin{Bmatrix} v_x \\ v_y \\ p \end{Bmatrix}_k = \begin{Bmatrix} N_{k_1} v_{x_1} + N_{k_2} v_{x_2} \\ N_{k_1} v_{y_1} + N_{k_2} v_{y_2} \\ N_{k_1} p_1 + N_{k_2} p_2 \end{Bmatrix} \tag{42}$$

Shape functions N_{k_1} and N_{k_2} related to node 1 and 2 of edge k (see Fig. 2) are defined in function of local dimensionless abscissa ξ_k as:

$$N_{k_1} = 1 - \xi_k, \qquad N_{k_2} = \xi_k, \qquad \xi_k = \frac{|\boldsymbol{x}_{k_2} - \boldsymbol{x}_{k_1}|}{l_k} \quad \text{with} \quad 0 \le \xi_k \le 1 \tag{43}$$

being l_k and $\boldsymbol{x}_k$ the length and coordinates of edge k, respectively.

In general, the evaluation of Eq. (36) in 3D problems is not straightforward since the boundaries of the element are polygonal surfaces on which the displacement model is unknown. In the special case of VEs resulting from the agglomeration of tetrahedra, the element faces are always triangular and the standard shape functions of linear triangles completely define the displacement model over the face, so that the boundary integrals can be easily computed. The position vector $\boldsymbol{x}^\tau$ in each triangle τ is formally defined by application of the mapping

$$\boldsymbol{N}^\tau = (\xi, \eta, 1 - \xi - \eta) \tag{44}$$

to the vectors $\boldsymbol{X}_I^\tau$ of nodal coordinates of vertex I of triangle τ

$$\boldsymbol{x}^\tau = \boldsymbol{N}^\tau \boldsymbol{X}_I^\tau \qquad \forall I \in \tau \tag{45}$$

being ξ, η local dimensionless coordinates. The velocity and pressure fields $\boldsymbol{\Phi}^\tau$ in each triangular face τ are defined as:

$$\boldsymbol{\Phi}^\tau = \boldsymbol{N}^\tau \hat{\boldsymbol{\Phi}}_I^\tau \qquad \forall I \in \tau \tag{46}$$

where $\hat{\boldsymbol{\Phi}}_I^\tau$ are nodal degrees of freedom of triangle τ at node I.

Finally, introducing Eq. (46) into Eq. (36), the following expression is obtained:

$$\nabla \boldsymbol{\Phi}_\pi = \frac{1}{\Omega_e} \int_{\partial \Omega_e} \boldsymbol{\Phi} \otimes \boldsymbol{n} \, d\Gamma = \frac{1}{\Omega_e} \sum_{\tau=1}^{n_f} \int_{\Gamma_\tau} \boldsymbol{\Phi}_\tau \otimes \boldsymbol{n}_\tau \, d\Gamma = \frac{1}{\Omega_e} \sum_{\tau=1}^{n_f} \sum_{g=1}^{n_g} \omega_g N_\zeta^\tau \boldsymbol{\Phi}_g^\tau \otimes \boldsymbol{n}_g^\tau \tag{47}$$

where n_f and n_g denote the number of triangular faces and the number of Gauss points per triangle, respectively, and N_ζ^τ is the constant Jacobian of triangle τ. Since linear triangles are considered, a one-point quadrature rule is sufficient. Therefore, the number of Gauss points is assumed to be $n_g = 1$ with weight $\omega_g = 1/2$. The subscript g stands for quantities evaluated at a Gauss point of the triangle τ, with local coordinates $\xi = 1/3$ and $\eta = 1/3$. The Jacobian of the isoparametric mapping N_ζ^τ and the normal vector $\boldsymbol{n}_g^\tau$ (constant in each triangle) are evaluated as follows:

$$N_\zeta^\tau = \|\boldsymbol{g}_\zeta^\tau\| = \left\| \frac{\partial \boldsymbol{x}^\tau}{\partial \xi} \times \frac{\partial \boldsymbol{x}^\tau}{\partial \eta} \right\| \tag{48}$$

$$\boldsymbol{n}_g^\tau = \frac{\boldsymbol{g}_\zeta^\tau}{\|\boldsymbol{g}_\zeta^\tau\|} \tag{49}$$

4.3. Virtual element stabilization

The expression (41) of $\boldsymbol{\Phi}_\pi = \{\boldsymbol{v}_\pi, p_\pi\}$ and (47) of $\nabla\boldsymbol{\Phi}_\pi$ obtained in the previous paragraph can be used directly in the weak forms (10)–(11). However, the formulation of VEs based only on these consistent terms would lead to a rank-deficient element if the number of vertices is larger than three in 2D or four in 3D. When more vertices are considered, VEs have to be stabilized. It is worth noticing that the VEM for 3-node triangles and 4-node tetrahedra coincides with the FEM with triangular and tetrahedral finite elements with linear shape functions, respectively, and hence the entire formulation of Section 2 could be presented within the VEM framework. However, employing a VE mesh is in general computationally more expensive than a finite element one of the same order and the use of the VEM is recommended only as a local countermeasure to improve element quality in specific regions where it is needed (see Section 5). For this reason, we prefer to present the approach as a standard FEM, with VEs introduced selectively only in localized parts of the domain.

In the literature, two different stabilization techniques are commonly used for VEs. The first stabilization is based on the degrees of freedom and it introduces a pointwise error measure between nodal values $\boldsymbol{\Phi}_I$ and the projected variables $\boldsymbol{\Phi}_\pi$ computed at vertices $\boldsymbol{x}_I$ [21–23]. The second stabilization method, developed by Wriggers et al. [37], consists of adding a new, positive definite strain energy into the VEM framework. In the present work, the stabilization based on degrees of freedom has been adopted. Note that the equal order interpolation violates the LBB condition allowing spurious oscillations in the pressure field. Therefore, not only does the VE formulation has to be properly stabilized, but also unphysical pressure modes have to be removed by means of an appropriate stabilization. For the former, following [38] the stabilizing term can be written as the difference between nodal velocities $\boldsymbol{v}_I$ and the projected ones at each node as (see, e.g., [21–23] for a detailed derivation):

$$\Delta\boldsymbol{v}_I = \boldsymbol{V}_I - \boldsymbol{V}_\pi(\boldsymbol{X}_I) = \left(\boldsymbol{I} - \Pi_v(\boldsymbol{X}_I)\right)\boldsymbol{V}_I \tag{50}$$

leading to the definition of a stabilization stiffness matrix $\boldsymbol{K}_e^s$ to be added to the consistent one $\boldsymbol{K}_{\mu,e}^c$

$$\boldsymbol{K}_e^s = \frac{\gamma}{n_V}\sum_{I=1}^{n_V}\left(\boldsymbol{I} - \Pi_v(\boldsymbol{X}_I)\right)\left(\boldsymbol{I} - \Pi_v(\boldsymbol{X}_I)\right) \tag{51}$$

being γ a scaling factor defined as follows:

$$\gamma = \begin{cases} \gamma_0\,\rho\,c & \text{2D} \\ \gamma_0\,\rho\,c\,h_e & \text{3D} \end{cases} \tag{52}$$

where γ_0 denotes a tuning parameter introduced to match the magnitude of the consistent part (e.g. $\gamma_0 = 0.001$), whereas h_e represents the characteristic length of the VE. Note that to guarantee the correct scaling, in two-dimensional cases the factor γ is set to be constant, whereas it depends on the element characteristic length in 3D problems.

The semi-discretized momentum conservation assumes the same form of Eq. (14) where the final stiffness matrix for each VE is evaluated by summing the contribution of the consistent and stabilizing parts, that is $\boldsymbol{K}_e = \boldsymbol{K}_{\mu,e}^c + \boldsymbol{K}_e^s$

However, an additional stabilizing term is required since the LBB condition is not satisfied. In order to avoid the introduction of further stabilization techniques, the DPS (already introduced in Section 2.1 for standard PFEM, see Eq. (19)) has also been adopted for the present VE framework using the projector $\tilde{\boldsymbol{\Pi}}_p$ defined in (18) with reference to triangular and tetrahedral elements. In the more general case of polygonal or polyhedral VEs, $\tilde{\boldsymbol{\Pi}}_p$ collects the averages of element nodal pressure values, defined at the elemental level as:

$$\tilde{\boldsymbol{\Pi}}_p = \tilde{\boldsymbol{N}}^p\boldsymbol{P}_e \tag{53}$$

being $\tilde{\boldsymbol{N}}^p$ the vector having as dimension the number of vertices n_V and defined as $\tilde{\boldsymbol{N}}^p = \frac{1}{n_V}[1,\dots,1]$.

4.4. Mass matrix

In explicit methods, the mass matrix has to be diagonalized. Among the several available diagonalization techniques, the simplest row-summing technique is here adopted:

$$M_{e,ii} = \sum_{j=1} M_{e,ij} \tag{54}$$

As stated in [20], the stabilization of the mass matrix is required only when the problem is dominated by the reaction term. Moreover, thanks to the diagonal structure and the fact that all entries are non-zero, the mass matrix is never singular. Therefore, it is sufficient to consider the projection part for the evaluation of the mass matrix and no stabilization is needed. The central difference scheme is also adopted as the time integration of the PVEM. As already pointed out, the conditional stability of the method poses a restriction on the time-step size, requiring the respect of the CFL condition (27). In the present work, the characteristic length for a VE is defined as

$$h_e^{3D} = \frac{6V_e}{A_{ext}} \qquad h_e^{2D} = \frac{4A}{Per} \tag{55}$$

where for the 3D case V_e represents the VE volume and A_{ext} denotes its total external surface area, while in 2D A_e is the element area and Per its perimeter. The coefficient 6 in Eq. (55) defines the diameter of the sphere inscribed in a tetrahedron. In the case of a general polyhedron, it is not rigorous but it provides a reasonable estimate of the characteristic length. A similar consideration holds also for the 2D case.

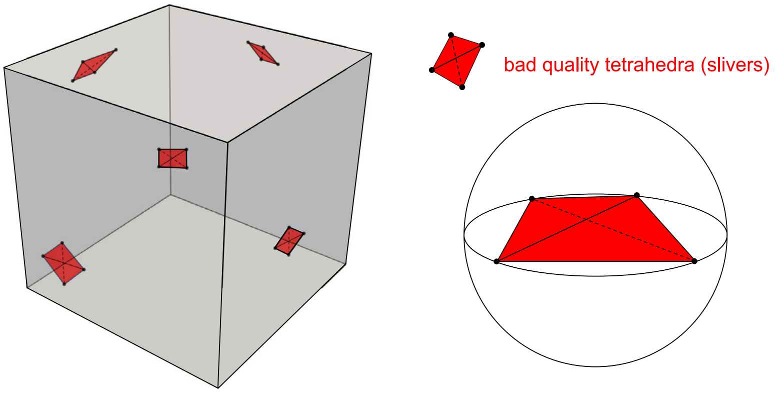


Fig. 3. Badly-shaped tetrahedra (slivers).

5. Agglomeration in PVEM

Being the PFEM based on a continuous remeshing process, a fast and efficient mesh generator constitutes the core of the method. To this purpose, the Delaunay tessellation represents the tool of primary choice. In the 2D version, the algorithm generates triangular meshes with optimal geometrical properties, such as the minimization of the maximum radius of a triangle circumcircle and the maximization of the minimum angle among all triangles. However, these optimal properties are not guaranteed in the 3D tessellation and badly shaped tetrahedra, called *slivers*, can be generated (see Fig. 3). Slivers are tetrahedra whose four vertices almost lie on the same plane and their presence impacts the computational performance of the numerical solution. While implicit methods benefit from their unconditional stability, enabling the use of larger time-step sizes despite the presence of poorly shaped elements, in explicit time integration schemes the time-step size is limited by the CFL condition (see Eq. (27)). In principle, just one overly distorted element is sufficient to drastically reduce the stable time-step size.

Mesh optimization techniques have been introduced to circumvent the problems created by badly shaped elements. They can be classified either as static or dynamic mesh improvements [48]. Static methods are intended to be applied once, at the beginning of the analysis, to optimize the quality of the initial mesh [48,49]. Dynamic techniques are conceived to maintain a good mesh quality during the entire analysis through local improvements and mesh regenerations [50]. Among them, the smoothing technique proposed by Meduri et al. [18] is specifically designed for the PFEM. It consists of the application of two algorithms: the first one is based on the solution of a fictitious elastic problem with fixed boundary and imposed stress distribution, properly designed to produce nodal movements that improve the geometric quality of the mesh; the second one consists of geometry-based operations to solve particular cases in which the first algorithm is ineffective. This technique is very efficient in dealing with slivers inside the fluid bulk. However, it may still encounter some difficulties in confined regions with fixed geometry, where badly shaped elements are defined by boundary or interface nodes.

Very recently, Sukumar et al. [42] proposed an alternative technique to remove slivers from the computational mesh. The idea exploits the possibility of VEM of using elements of arbitrary polyhedral shapes. It consists of merging slivers (or in general badly-shaped elements) with their neighbouring elements to form a VE characterized by a larger characteristic length. The agglomeration of badly shaped finite elements represents an efficient technique to increase the critical time-step size in explicit methods. Agglomeration techniques for VEs have already been used in other contexts, see e.g. [51,52].

5.1. Agglomeration algorithm

Following the idea proposed by Sukumar et al. [42], the present work investigates the agglomeration of distorted linear triangles (2D, see Fig. 4) and tetrahedra (3D, see Fig. 5) in the PVEM framework to form arbitrary-shaped VEs. The agglomeration process is performed only in a limited portion of the fluid domain, i.e., only in regions characterized by the presence of poorly shaped elements. The remaining part of the mesh is formed by regular linear tetrahedra, for which the virtual and finite element formulations coincide.

The fluid domain Ω^t is discretized, at each time instant $t \in [0, T]$, by a mesh $\mathcal{T}^t = \mathcal{T}(t)$ composed of linear triangles in 2D or linear tetrahedral elements in 3D. Within a Lagrangian framework, the mesh $\mathcal{T}^t$ evolves in time following the domain Ω^t. In the spirit of the PFEM, the connectivities of the mesh $\mathcal{T}^t$ change only when the mesh is globally too distorted. At every time step, for each element, the local stable time step is computed as:

$$\Delta t_e^{n+1} = \left(\frac{h_e^{n+1}}{c} \right) \tag{56}$$

All the elements e with a local stable time-step size lower than a reference value $\Delta \bar{t}$ are grouped in a subset $\bar{\mathcal{T}}^t$:

$$\bar{\mathcal{T}}^t = \left\{ \forall e \in \mathcal{T}^t \, : \, \Delta t_e < \Delta \bar{t} \right\} \tag{57}$$

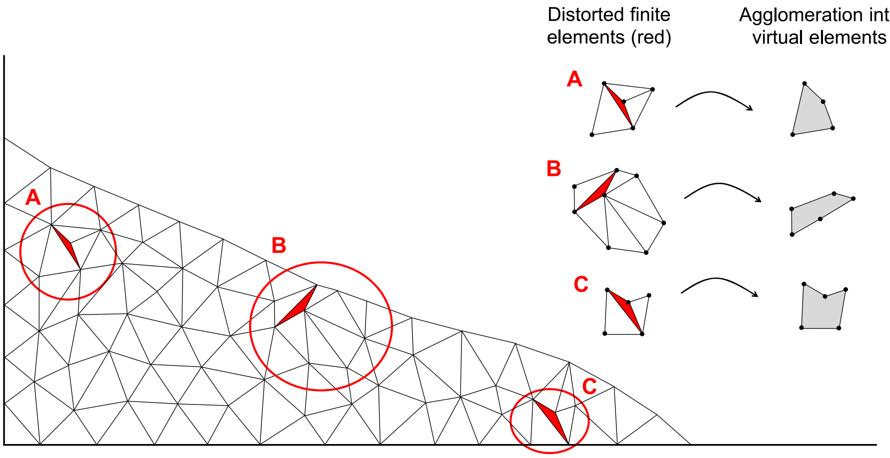

Distorted finite elements (red)
Agglomeration into virtual elements
A
B
C
A
B
C

Fig. 4. 2D agglomeration. Distorted 2D elements are agglomerated with their neighbours to form a VE.

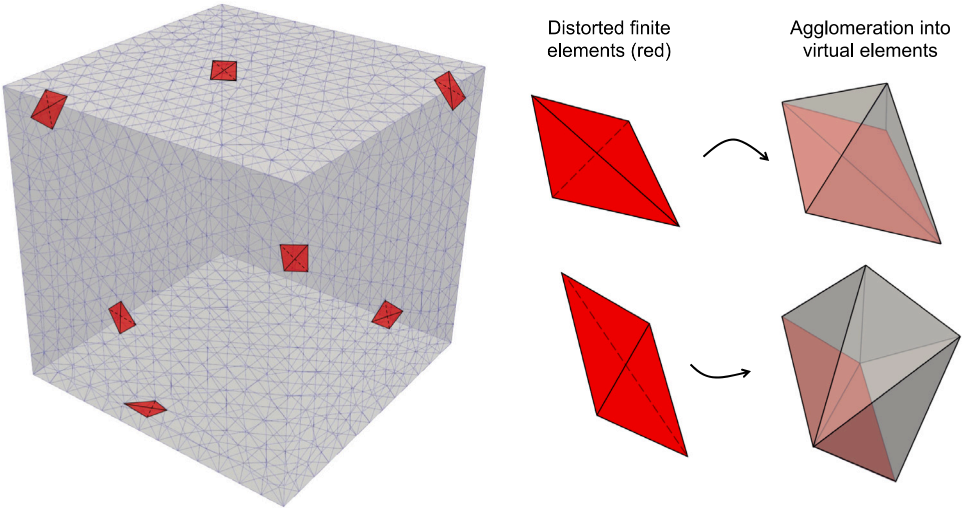

Distorted finite elements (red)
Agglomeration into virtual elements

Fig. 5. 3D agglomeration. Distorted 3D elements are agglomerated with their neighbours to form a VE.

where $\Delta\bar{t} = \beta \Delta t_m$ and $\beta \leq 1$ is a user defined parameter (e.g. $\beta = 0.6$), which scales the mean stable time-step size Δt_m. Elements belonging to the subset $\bar{\mathcal{T}}^t$ are considered excessively distorted, hence they have to be agglomerated with one or more neighbours to form a VE characterized by a larger characteristic length.

Every element e_i of $\bar{\mathcal{T}}^t$ is agglomerated with the neighbour element $e_{\bar{j}}$ leading to the agglomerated element $e_{i\bar{j}}$ endowed with the larger local stable time step. If, after the first agglomeration, the local stable time step is still smaller than $\Delta\bar{t}$, the operation is repeated, and a new element is agglomerated to the VE $e_{i\bar{j}}$. This procedure is repeated until the obtained agglomerated VE guarantees a stable time step larger than $\Delta\bar{t}$. This element agglomeration allows for an effective and straightforward removal of slivers from the computational mesh, leading to a significant increase in the stable time-step size. The proposed agglomeration technique is illustrated in Algorithm 1.

It is worth noting that the parameter β assumes a pivotal role in the agglomeration process, determining whether a standard finite element is considered too distorted and, hence, to be agglomerated. As a consequence, a larger number of VEs are generated if a higher value for β is selected. In contrast, when a small value of β is chosen, only a few highly distorted elements are involved in the agglomeration procedure.

The proposed method is particularly efficient in addressing the mesh distortion problem, especially when traditional smoothing algorithms yield limited improvements. Its main advantages are:

- it does not move the nodes, and therefore it is suitable also for handling slivers with nodes constrained on the boundary;
- it does not require the convection of nodal data from the old to the new mesh since nodes are not moved;
- it does not require the solution of additional equations (as it is the case in standard smoothing methods), thus the increase in computational cost is minimal;

Algorithm 1 PVEM agglomeration

```
Define the reference time-step size $\Delta\bar{t} = \beta\Delta t_m$
Identify the subset $\bar{\mathcal{T}}^t$ of overly distorted elements with $\Delta t_e < \Delta\bar{t}$
for every $e_i \in \bar{\mathcal{T}}^t$ do
    Evaluate $\Delta t_i = \frac{h_{e_i}}{c}$
    while $\Delta t_i < \Delta\bar{t}$ do
        find the neighbour elements of $e_i$ and list them in the vector $Q_i$
        for every $e_j \in Q_i$ do
            create a VE $e_{ij} = e_i \cup e_j$
            compute $\Delta t_{ij}$ of the agglomerated VE $e_{ij}$
        end for
        find $e_{\bar{j}}$ so that $\Delta t_{i\bar{j}} = \max_j \Delta t_{ij}$
        the VE $e_{i\bar{j}}$ replace $e_i$ and $e_{\bar{j}}$
        if $\Delta t_{i\bar{j}} \geq \Delta\bar{t}$ then
            exit
        else
            $e_i \leftarrow e_{i\bar{j}}$
        end if
    end while
end for
```

- the agglomeration is applied only where needed, hence, the created VEs represent only a small portion of the computational mesh;
- the PFEM is based on continuous remeshing, hence, when a new mesh is created, all the VEs are removed and a mesh of triangles (or tetrahedra) only is formed again.

6. Numerical examples

In this section, the proposed VEM approach for weakly compressible fluids is tested against an analytical solution. Then, the PVEM agglomeration technique is applied to both 2D and 3D problems. In all the examples, a Newtonian fluid is considered under the hypothesis of non-slip conditions at all boundaries.

6.1. Test with analytical solution

To validate the VEM (without remeshing) for weakly compressible fluids formulated in the considered mixed velocity–pressure framework, the 2D test presented in [53] is here performed. This example consists of a 1 × 1 m square domain containing a Newtonian fluid ($\rho = 1$ kg/m^3, $\mu = 1$ Pa s, $c = 3.16 \cdot 10^4$ m/s) subjected to the following prescribed body forces per unit volume:

$$\begin{aligned} b_x =& (12-24y)x^4 + (-24+48y)x^3 + (12-48y+72y^2-48y^3)x^2 + (-2+24y-72y^2+48y^3)x + 1 - 4y + 12y^2 - 8y^3 \\ b_y =& (8-48y+48y^2)x^3 + (-12+72y-72y^2)x^2 + (4-24y+48y^2-48y^3+24y^4)x - 12y^2 + 24y^3 - 12y^4 \end{aligned} \tag{58}$$

Under the given body forces, the solutions for horizontal, and vertical velocities, as well as pressure (v_x, v_y, p, respectively) at the steady state can be analytically obtained:

$$\begin{aligned} v_x(x,y) &= x^2(1-x)^2(2y-6y^2+4y^3) \\ v_y(x,y) &= -y^2(1-y)^2(2x-6x^2+4x^3) \\ p(x,y) &= x(1-x) \end{aligned} \tag{59}$$

Three VE meshes are considered: (i) a mesh of quadrilateral elements, (ii) a mesh of hexagonal elements and (iii) a mesh of Voronoi elements. Velocity magnitude and pressure contour plots at the steady state (reached after approximately 0.1 s) are illustrated in Figs. 6 and 7 with the computational meshes. The steady-state numerical solutions along the domain midlines obtained with the three different meshes are compared to the analytical values in Fig. 8, showing a perfect agreement for both pressure and velocity fields.

Finally, a convergence analysis has been performed for five different quadrilateral meshes with mean sizes of 0.2, 0.1, 0.05, 0.025 and 0.01 m, leading to 36, 121, 441, 1681 and 10 201 nodes, respectively. The velocity error has been measured with a normalized L^2 norm:

$$\frac{\|v - v_h\|}{\|v\|} = \frac{\sqrt{\int_\Omega (v - v_h)^2 d\Omega}}{\sqrt{\int_\Omega v^2 d\Omega}}$$

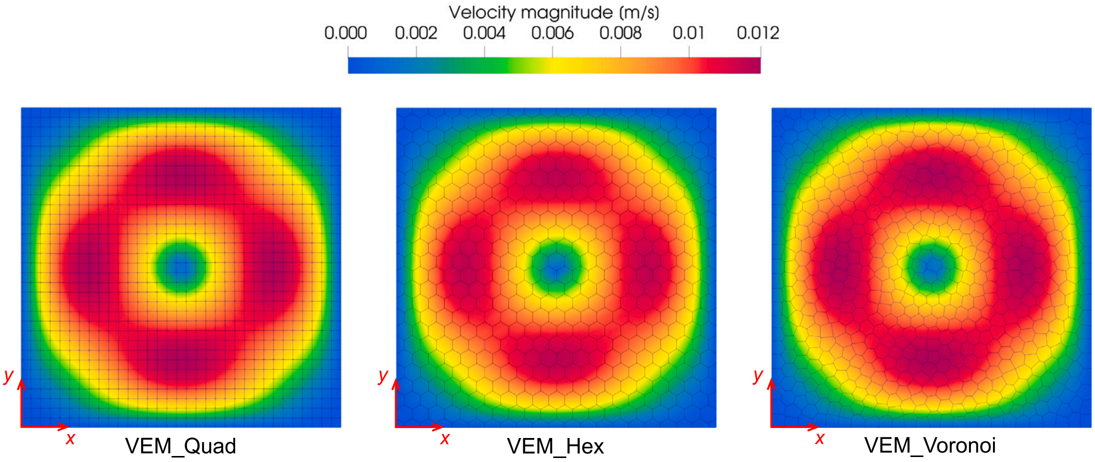


Fig. 6. Test with analytical solutions. Velocity contour plots at steady state for three different meshes.

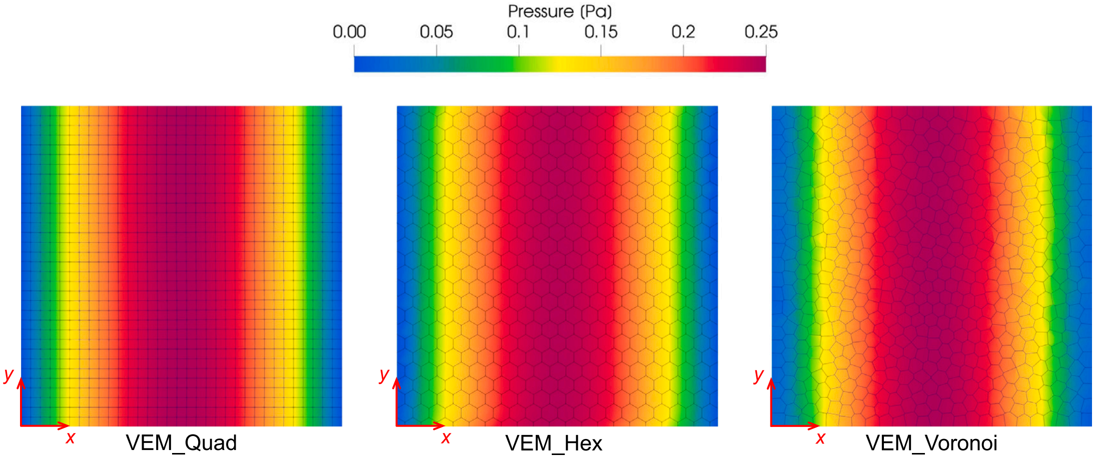


Fig. 7. Test with analytical solutions. Pressure contour plots at steady state for three different meshes.

where v and v_h are the exact and numerical solutions of the considered variable. The convergence analysis is reported in Fig. 9, where the error measures at $t = 0.1$ s are depicted for horizontal and vertical velocities (denoted as v_x, v_y, respectively).

6.2. 2D sloshing

The two-dimensional sloshing test proposed in [54] is used to test the PVEM agglomeration technique. The example consists of a fixed rigid tank containing an inviscid Newtonian fluid characterized by the density $\rho = 1000$ kg/m^3, wave speed $c = 350$ m/s and bulk modulus $K_f = 1.23 \cdot 10^8$ Pa. The fluid configuration at $t = 0$ s is illustrated in Fig. 10 where the initial profile of the free surface, denoted as $H(x)$, is defined according to the following expression:

$$H(x, t = 0) = A - a\ cos[k(x + \lambda/2)] \tag{60}$$

where the constant A is set to $A = 1$ m, the wave amplitude a is taken as $a = 0.1$ A, the wave length is set to $\lambda = 2$ m and $k = 2\pi/\lambda$.

The computational domain has been discretized with nearly 6000 nodes and approximately 11 000 elements. A comparison between the exact solution proposed in [54] and the numerical ones for the vertical position in time at the midpoint P (see Fig. 10) is shown in Fig. 11a. The problem has been analysed with both the PVEM agglomeration and the standard Lagrangian PFEM (denoted as PVEM and PFEM 2D in Fig. 11, respectively). The two approaches show good agreement with the analytical and reference solutions in [54].

Fig. 11b shows the stable time-step size obtained with standard PFEM and with the proposed agglomeration technique with the scaling factor $\beta = 0.6$. A significant increment of stable time-step size can be observed, proving the effectiveness of the proposed technique also in a 2D setting. Moreover, an almost constant stable time-step size, independent of the mesh evolution, can be observed in Fig. 11b. The minimum and average time-step values for both methods are reported in Table 1. It can be observed that

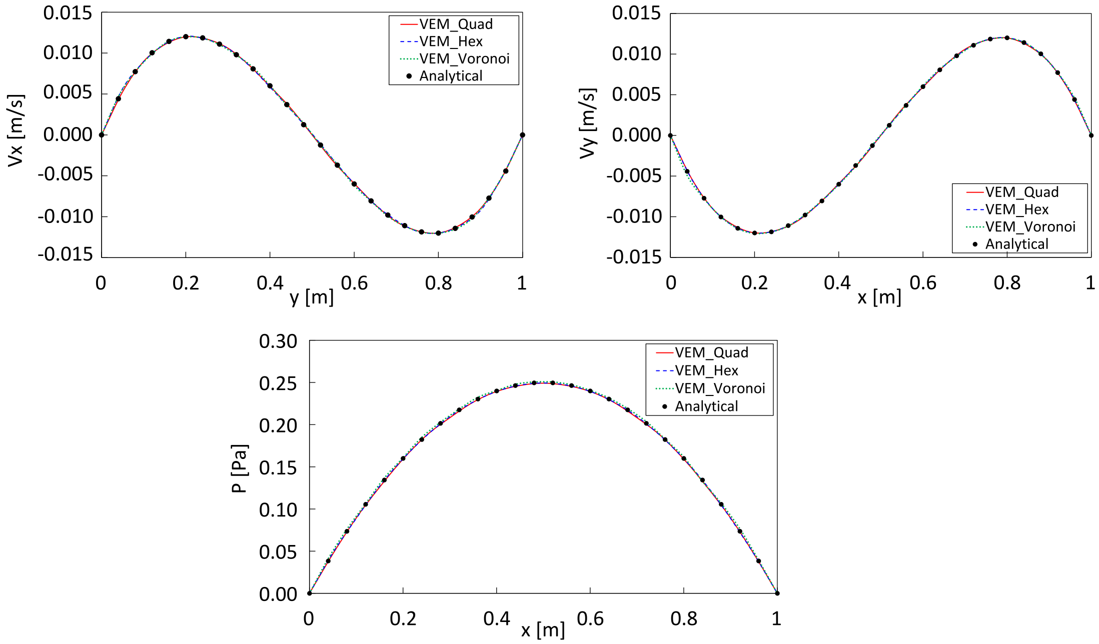


Fig. 8. Test with analytical solutions. Horizontal, vertical velocities and pressure at steady state along the domain midlines $x = 0.5$ m and $y = 0.5$ m.

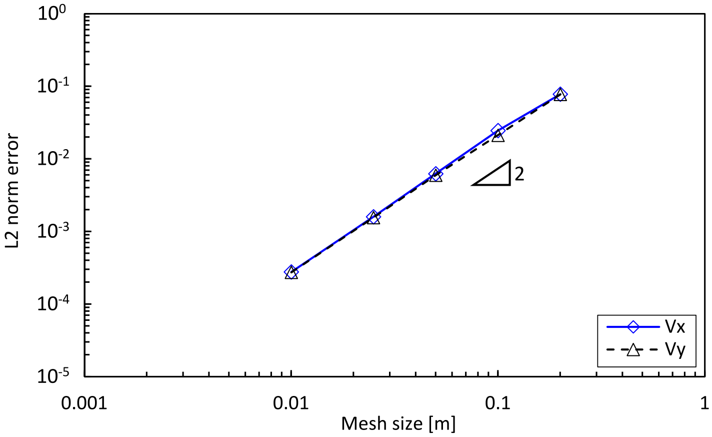


Fig. 9. Test with analytical solutions. Convergence graphs at steady state for quadrilateral mesh with different sizes.

Table 1
2D sloshing. Comparison of minimum and average time-step sizes.

Time-step size value	Minimum [s]	Average [s]
PVEM 2D	$3.04 \cdot 10^{-5}$	$3.06 \cdot 10^{-5}$
PFEM 2D	$2.55 \cdot 10^{-6}$	$1.38 \cdot 10^{-5}$

in this case, the agglomeration technique guarantees a minimum stable time step one order of magnitude larger than the standard technique. In this example, small fluid movements are expected in the lower portion of the domain. Therefore, as shown in Fig. 12, the agglomeration is performed only in correspondence with the free surface, creating only a few VEs (approximately 0.2% over the total number of elements).

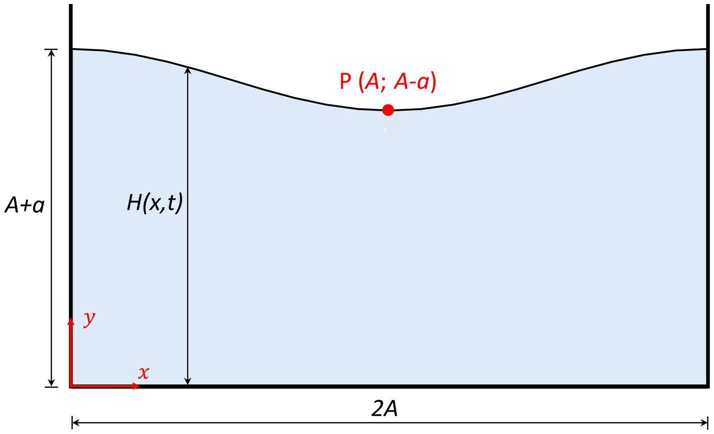


Fig. 10. 2D sloshing. Fluid configuration at $t = 0$ s.

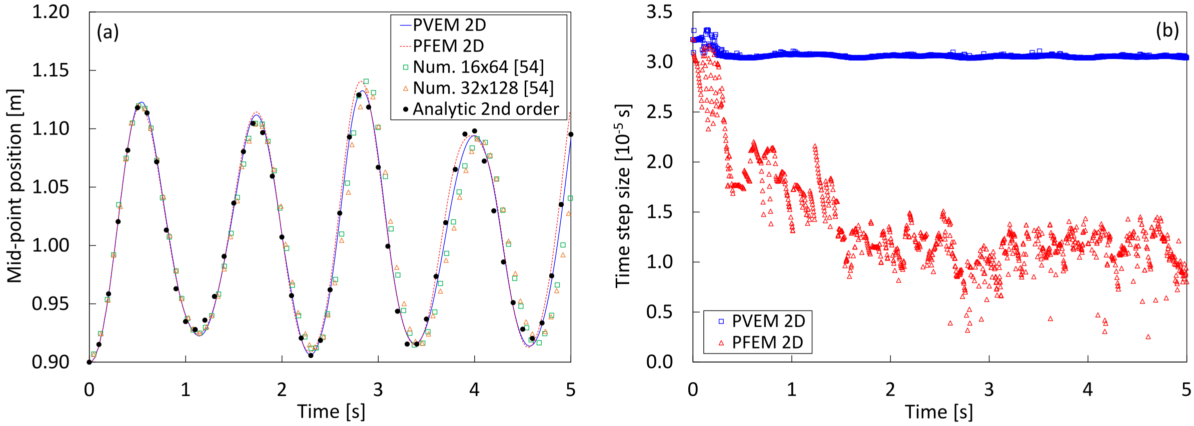


Fig. 11. 2D sloshing. Comparison of vertical position evolution at midpoint P (a) and variation of time-step size (b) obtained using different methods.

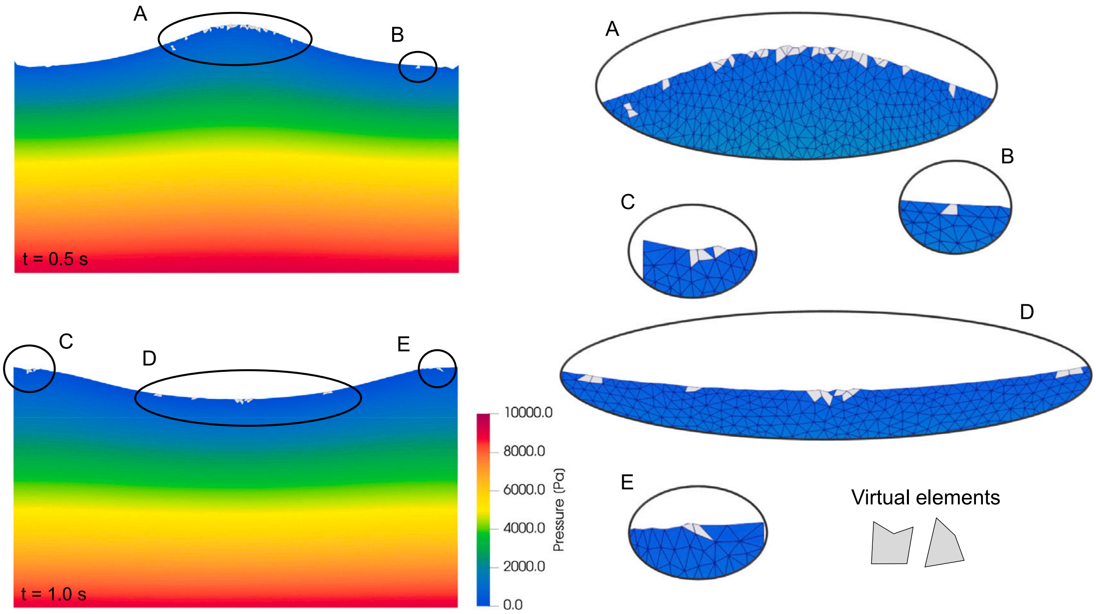
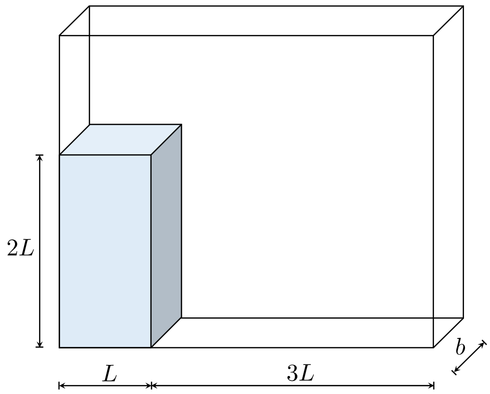


Fig. 12. 2D sloshing. Pressure contour plots and meshes at different times. Meshes composed of standard PFEM elements (triangles) and VEs (grey polygons).



Fluid properties: Newtonian fluid	
Density ρ_f	1000 kg/m^3
Viscosity μ_f	0.001 $Pa{\cdot}s$
Wave speed c_f	350 m/s
Bulk modulus K_f	$1.23 \cdot 10^8$ Pa
Geometry	
L	0.146 m
b	0.175 m
Time simulation: 1 s	

Fig. 13. 3D Dam break. Fluid and geometrical parameters of the problem.

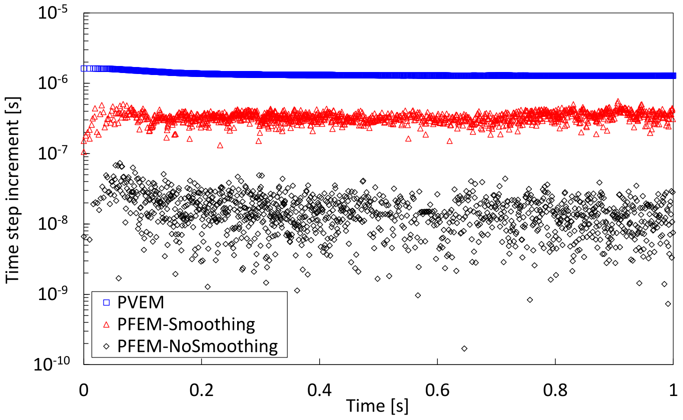


Fig. 14. 3D dam break. Variation of the time-step increment in the PVEM approach, standard PFEM with and without smoothing.

Table 2
3D dam break. Comparison of minimum and average time-step increment values.

Time-step size value	Minimum [s]	Average [s]
PVEM	$1.27 \cdot 10^{-6}$	$1.33 \cdot 10^{-6}$
PFEM-Smoothing	$1.06 \cdot 10^{-7}$	$3.34 \cdot 10^{-7}$
PFEM-NoSmoothing	$1.69 \cdot 10^{-10}$	$1.61 \cdot 10^{-8}$

6.3. 3D dam break

The PVEM agglomeration technique is tested on the classical three-dimensional dam break problem. It consists of a column of a Newtonian fluid kept on the left of a container by a vertical barrier. The geometry is depicted in Fig. 13. At the beginning of the analysis, the barrier is instantaneously removed leaving to the fluid the possibility to flow under the effect of gravity.

The problem has been analysed with a mesh of about $40k$ nodes, forming approximately $104k$ elements. The evolution of the time-step sizes obtained from the PFEM with agglomeration (i.e., PVEM), standard PFEM with the smoothing technique proposed in [18] and standard PFEM without smoothing (denoted as PVEM, PFEM-Smoothing and PFEM-NoSmoothing, respectively) are shown in Fig. 14. In this test, the scaling parameter has been set to $\beta = 0.4$. The graph shows how the PVEM agglomeration technique guarantees an almost constant and significantly higher value of the stable time-step size. Moreover, in Table 2 the minimum and average time-step values are reported. It can be observed that the minimum stable time step is one order of magnitude larger than the one obtained with the smoothing technique and four orders of magnitude larger than using Delaunay tessellation only. This substantial improvement in the stable time-step size leads to a remarkable reduction in the total computational cost: the PVEM approach simulated the test in 4 h, compared to the 14 h required by the solver using the previous smoothing method (test performed on a standard workstation with 8 computing cores).

As mentioned, the PVEM technique is applied to a limited portion of the fluid domain, where badly shaped tetrahedra are encountered. Therefore, the created VEs represent a small portion of the computational mesh. This is confirmed in Fig. 15 where

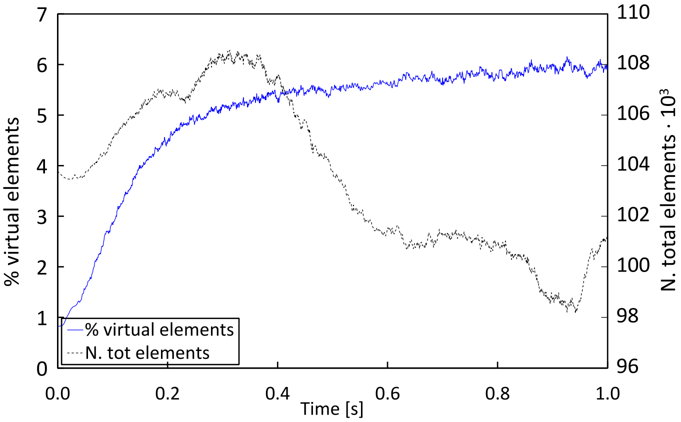


Fig. 15. 3D dam break. Evolution of number of VEs with respect to total number of elements in PVEM.

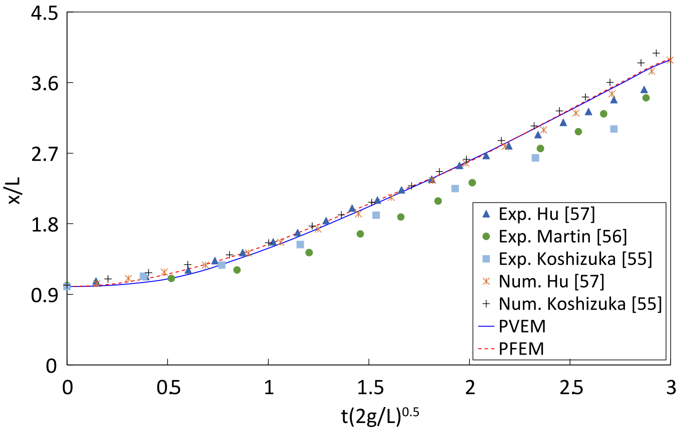


Fig. 16. 3D dam break. Time evolution of non-dimensional wave front position.

the variation of the VEs percentage and the total number of elements are reported. The number of VEs ranges from 1% to a maximum of 6%.

The numerical results, in terms of non-dimensional wavefront position, are compared to the reference values [55–57] in Fig. 16. Good agreement is achieved between the different numerical results, and in particular between the PVEM and the standard PFEM 3D. Despite the overall good accuracy, a small discrepancy between experimental and numerical results can be observed, showing a faster wavefront advancement in the latter ones. This could be caused by the instantaneous removal of the vertical wall at the beginning of the simulation, whereas a small but finite time is required for the removal in the experiment. Finally, Fig. 17 illustrates snapshots of the PVEM simulation compared to the experimental results presented in [55] at synchronized time steps, showing a good qualitative agreement.

6.4. Spherical water drop falling into a cylindrical tank

A spherical water drop falling into a cylindrical tank containing water at rest is considered (see [58]). The falling droplet collides with the tank water, merging into it after the impact. The fluid properties and geometry of the problem are reported in Fig. 18.

The fluid domain has been discretized with $104k$ nodes and approximately $535k$ elements. The time-step sizes obtained from the proposed agglomeration technique (with the scaling parameter $\beta = 0.4$), standard PFEM with and without smoothing [18] (PVEM, PFEM-Smoothing, and PFEM-NoSmoothing, respectively) are shown in Fig. 19. To prove the effectiveness of the PVEM agglomeration in the removal of overly distorted elements, Table 3 reports the minimum and average time-step values for the three cases. As can be observed, the aggregation allows for a notable increase in the time-step size, leading to a significant reduction in the overall computational burden. Also in this case, the substantial increase of the stable time-step size leads to a significant reduction in the total computational cost: the PVEM simulated the test in 6 h, compared to the 14 h required by the solver employing the previous smoothing method.

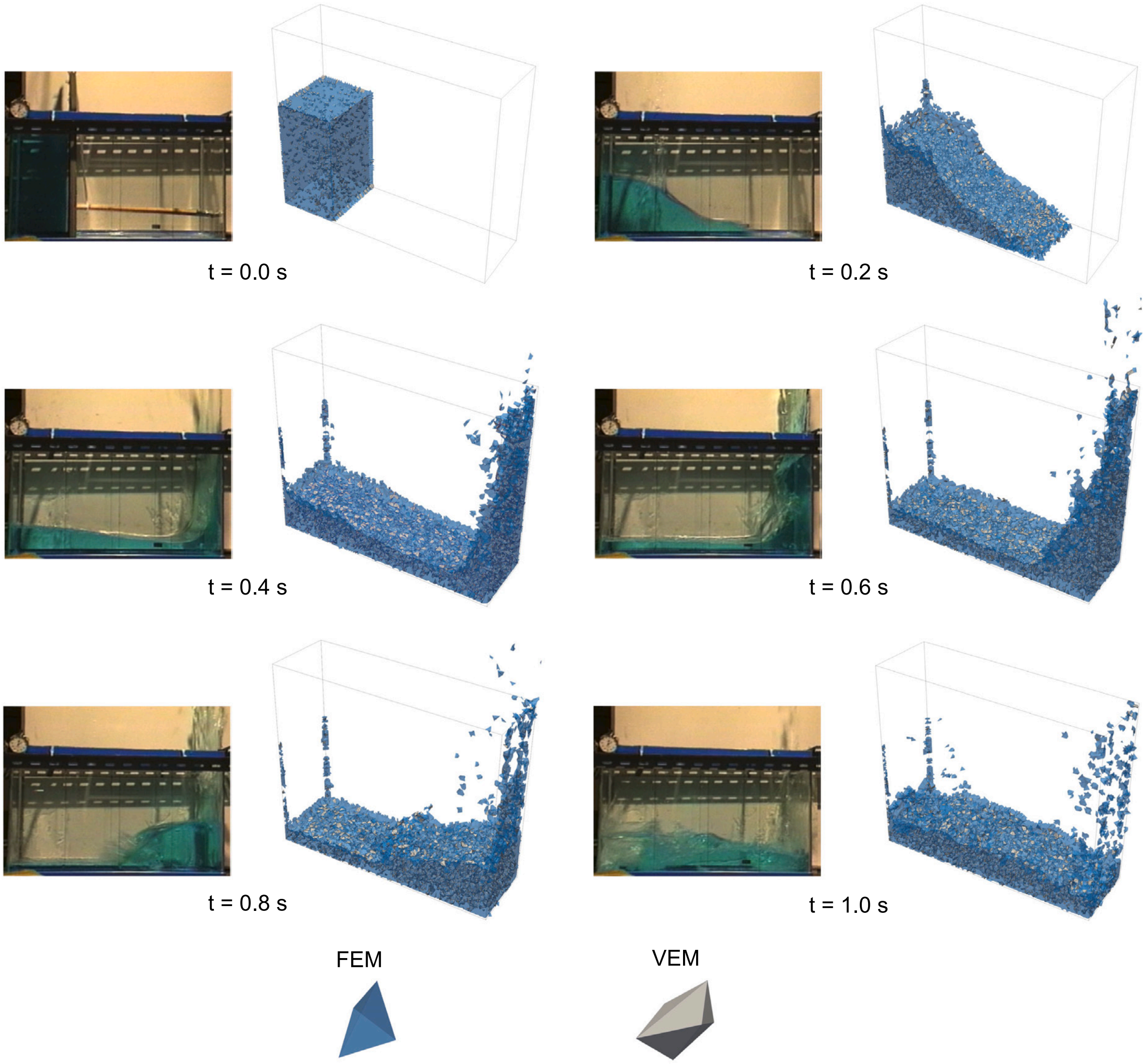
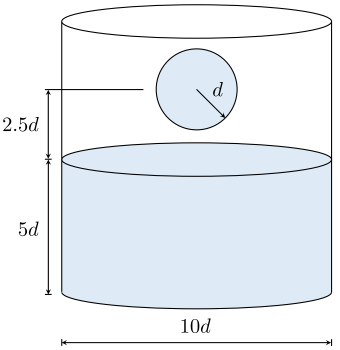


Fig. 17. 3D dam break. Comparison between snapshots of PVEM simulation at different time steps and corresponding experimental results [55]. Blue elements relate to FEM and grey elements to VEM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



Fluid properties: Newtonian fluid	
Density ρ_f	$1000\,kg/m^3$
Viscosity μ_f	$0.001\,Pa{\cdot}$s
Wave speed c_f	$350\,m/s$
Bulk modulus K_f	$1.23 \cdot 10^8\,Pa$
Geometry	
d	$0.1m$
Time simulation: $1\,s$	

Fig. 18. Water drop. Fluid and geometrical parameters of the problem.

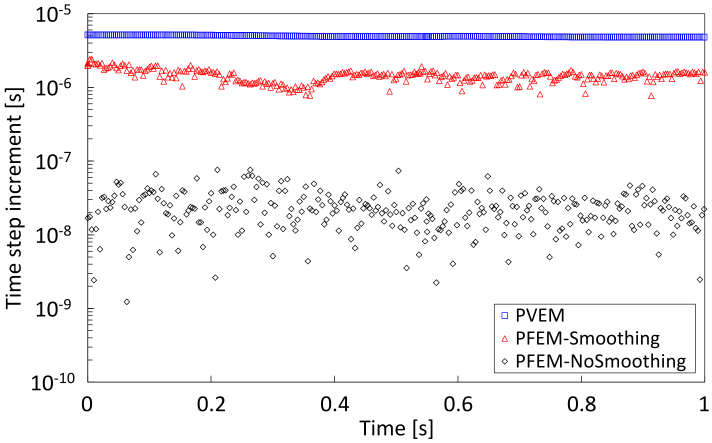


Fig. 19. Water drop. Time-step variation in PVEM and standard PFEM 3D with and without smoothing.

Table 3
Water drop. Comparison of minimum and average time-step values.

Time-step size value	Minimum [s]	Average [s]
PVEM	$4.81 \cdot 10^{-6}$	$5.17 \cdot 10^{-6}$
PFEM-Smoothing	$7.73 \cdot 10^{-7}$	$1.47 \cdot 10^{-6}$
PFEM-NoSmoothing	$1.23 \cdot 10^{-9}$	$2.41 \cdot 10^{-8}$

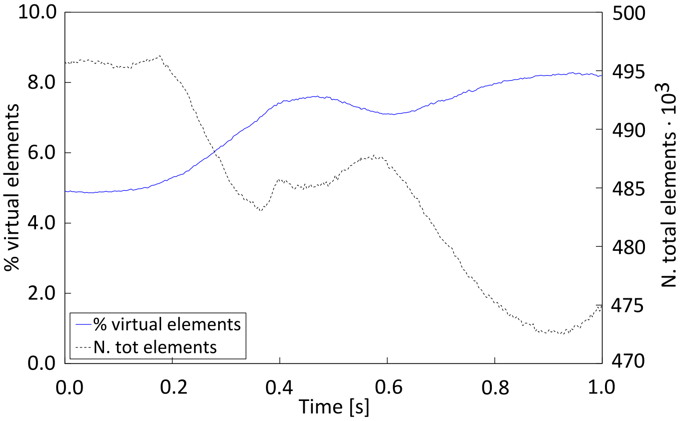


Fig. 20. Water drop. Evolution of number of VEs with respect to total number of elements in PVEM.

Fig. 20 shows the variation of the VEs percentage and the evolution of the total number of elements during the analysis. For a qualitative comparison between the numerical results, Fig. 21 depicts snapshots of the PVEM simulation at different time steps, showing good agreements with those provided in [58].

7. Conclusions

The Particle Finite Element Method (PFEM) is a mesh-based Lagrangian finite element method (FEM) particularly suited for fluid problems involving free-surface flows and breaking waves. However, a computationally effective application of the method in 3D problems using an explicit solver is challenging due to the presence of sliver elements created by the runtime remeshing process. When an explicit time integration scheme is adopted, a good mesh quality, with elements of homogeneous size, is fundamental to achieving an acceptable stable time-step size.

In this work, we demonstrated how the VEM can be conveniently utilized to significantly enhance the mesh quality in PFEM explicit dynamic analyses. Combining PFEM and VEM allows the removal of slivers by merging them with one or more neighbouring elements, thereby forming a bigger virtual element characterized by a larger critical time-step size. This technique represents a natural and efficient alternative to mesh-smoothing operations, which are in general computationally expensive and may lead to limited mesh improvements, especially when dealing with slivers that have nodes constrained on the boundary. Furthermore, the

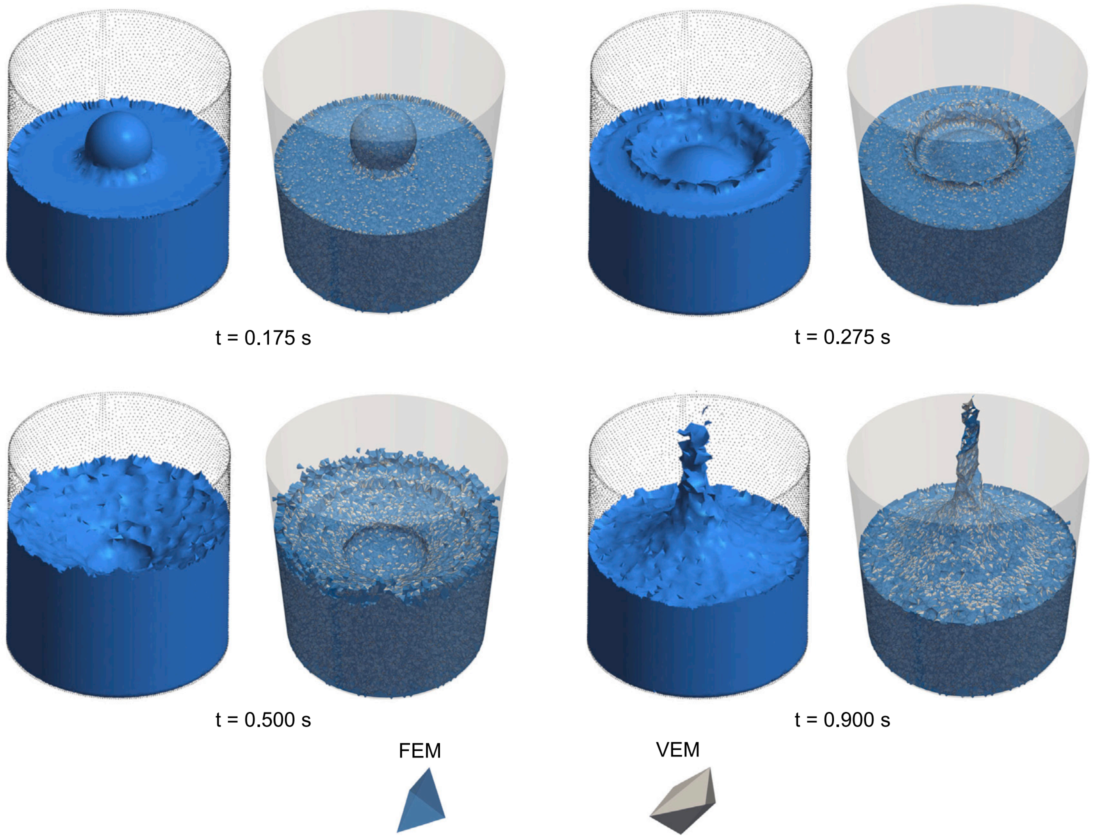


Fig. 21. Water drop. Comparison between the PVEM simulation snapshots at different time steps and the corresponding reference results [58]. Blue elements relate to FEM and grey elements to VEM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

proposed mesh improvement is executed exclusively in the localized regions of the mesh where severely distorted elements are present. Virtual elements obtained by merging a badly-shaped element with one or more neighbours are therefore introduced only locally, representing only a small portion of the total mesh. The method, developed through the combination of PFEM with the VEM agglomeration, has been called PVEM. Compared to the standard PFEM, the PVEM enables a notable reduction in the overall computational workload by allowing for substantially larger time steps.

The proposed PVEM is based on a mixed velocity–pressure VEM formulation for weakly compressible fluids. Standard techniques have been implemented to stabilize the virtual elements and to rectify the violations of the LBB condition caused by the chosen pressure model. These efforts result in a stable and robust formulation for the explicit dynamics of weakly compressible fluids within a Lagrangian setting.

The proposed approach has been validated on 2D and 3D numerical tests. In all the examples, comparisons of the numerical results with reference solutions have consistently demonstrated excellent agreement. Compared to standard PFEM without mesh smoothing, adopting the PVEM technique has enabled an increase in time-step sizes of up to three orders of magnitude.

The PVEM formulation has been designed to increase the stable time-step size in explicit analyses. However, improving mesh quality can also have a positive impact on implicit analyses. By eliminating sliver elements through the proposed agglomeration technique, results accuracy will be enhanced, as typically expected in standard FEM-based solvers, and improves the conditioning of the linear systems, making facilitating their solution and reducing the computational burden. Furthermore, in the PFEM framework, agglomerating poor-quality elements with neighbouring ones may reduce the need for frequent remeshing via Delaunay tessellation, with additional computational savings. A detailed study of the effect of the agglomeration technique in an implicit framework, represents a natural development of the present work, to be pursued in a future study.

CRediT authorship contribution statement

Cheng Fu: Writing – original draft, Validation, Software. **Massimiliano Cremonesi:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Umberto Perego:** Writing – review & editing, Supervision, Funding acquisition. **Blaž Hudobivnik:** Writing – review & editing, Supervision, Software. **Peter Wriggers:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was partially supported by the Italian Ministry of University and Research through the following projects: PRIN2022 - "DTWIX: development of Digital TWIns for multiphysics simulation of eXtreme events in civil engineering" (PRIN DTWIX - 2022AL5MSN) and PRIN2022 PNRR - "Polyhedral Galerkin methods for engineering applications to improve disaster risk forecast and management: stabilization-free operator preserving methods and optimal stabilization methods" (PRIN2022 PNRR - P2022BH5CB).

Data availability

Data will be made available on request.

References

[1] M. Cremonesi, A. Franci, S.R. Idelsohn, E. Oñate, A state of the art review of the particle finite element method (PFEM), Arch. Comput. Methods Eng. 27 (5) (2020) 1709–1735.

[2] E. Oñate, S.R. Idelsohn, F. Del Pin, R. Aubry, The particle finite element method—an overview, Int. J. Comput. Methods 1 (02) (2004) 267–307.

[3] S.R. Idelsohn, E. Oñate, F. Del Pin, The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves, Internat. J. Numer. Methods Engrg. 61 (7) (2004) 964–989.

[4] S.R. Idelsohn, E. Oñate, F. Del Pin, N. Calvo, Fluid–structure interaction using the particle finite element method, Comput. Methods Appl. Mech. Engrg. 195 (17–18) (2006) 2100–2123.

[5] M. Cremonesi, A. Frangi, U. Perego, A Lagrangian finite element approach for the analysis of fluid-structure interaction problems, Internat. J. Numer. Methods Engrg. 84 (5) (2010) 610–630.

[6] S. Meduri, M. Cremonesi, U. Perego, O. Bettinotti, A. Kurkchubasche, V. Oancea, A partitioned fully explicit Lagrangian finite element method for highly nonlinear fluid-structure interaction problems, Internat. J. Numer. Methods Engrg. 113 (1) (2018) 43–64.

[7] M. Cremonesi, F. Ferri, U. Perego, A basal slip model for Lagrangian finite element simulations of 3D landslides, Int. J. Numer. Anal. Methods Geomech. 41 (1) (2017) 30–53.

[8] X. Zhang, K. Krabbenhoft, D.M. Pedroso, A.V. Lyamin, D. Sheng, M.V. Da Silva, D. Wang, Particle finite element analysis of large deformation and granular flow problems, Comput. Geotech. 54 (2013) 133–142.

[9] S. Idelsohn, M. Mier-Torrecilla, E. Oñate, Multi-fluid flows with the particle finite element method, Comput. Methods Appl. Mech. Engrg. 198 (33–36) (2009) 2750–2767.

[10] P. Becker, S.R. Idelsohn, E. Oñate, A unified monolithic approach for multi-fluid flows and fluid–structure interaction using the particle finite element method with fixed mesh, Comput. Mech. 55 (2015) 1091–1104.

[11] E. Oñate, A. Franci, J.M. Carbonell, A particle finite element method for analysis of industrial forming processes, Comput. Mech. 54 (1) (2014) 85–107.

[12] P.B. Ryzhakov, An axisymmetric PFEM formulation for bottle forming simulation, Comput. Mech. 4 (2017) 3–12.

[13] R. Aubry, S.R. Idelsohn, E. Oñate, Particle finite element method in fluid-mechanics including thermal convection-diffusion, Comput. Struct. 83 (17–18) (2005) 1459–1475.

[14] E. Oñate, A. Franci, J.M. Carbonell, A particle finite element method (PFEM) for coupled thermal analysis of quasi and fully incompressible flows and fluid-structure interaction problems, in: Sergio R. Idelsohn (Ed.), Numerical Simulations of Coupled Problems in Engineering, Springer, 2014, pp. 129–156.

[15] H. Edelsbrunner, E.P. Mücke, Three-dimensional alpha shapes, ACM Trans. Graph. 13 (1) (1994) 43–72.

[16] D.A. Field, Laplacian smoothing and Delaunay triangulations, Commun. Appl. Numer. Methods 4 (6) (1988) 709–712.

[17] L.A. Freitag, C. Ollivier-Gooch, Tetrahedral mesh improvement using swapping and smoothing, Internat. J. Numer. Methods Engrg. 40 (21) (1997) 3979–4002.

[18] S. Meduri, M. Cremonesi, U. Perego, An efficient runtime mesh smoothing technique for 3D explicit Lagrangian free-surface fluid flow simulations, Internat. J. Numer. Methods Engrg. 117 (4) (2019) 430–452.

[19] L. Beirao 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 (01) (2013) 199–214.

[20] L. Beirao da Veiga, F. Brezzi, L.D. Marini, A. Russo, The hitchhiker's guide to the virtual element method, Math. Models Methods Appl. Sci. 24 (08) (2014) 1541–1573.

[21] L. Beirao da Veiga, F. Brezzi, L.D. Marini, Virtual elements for linear elasticity problems, SIAM J. Numer. Anal. 51 (2) (2013) 794–812.

[22] L. Beirao da Veiga, C. Lovadina, D. Mora, A virtual element method for elastic and inelastic problems on polytope meshes, Comput. Methods Appl. Mech. Engrg. 295 (2015) 327–346.

[23] H. Chi, L. Beirao 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.

[24] P. Wriggers, B. Hudobivnik, F. Aldakheel, A virtual element formulation for general element shapes, Comput. Mech. 66 (2020) 963–977.

[25] A. Lamperti, M. Cremonesi, U. Perego, A. Russo, C. Lovadina, A Hu–Washizu variational approach to self-stabilized virtual elements: 2D linear elastostatics, Comput. Mech. 71 (5) (2023) 935–955.

[26] Mertcan Cihan, Blaž Hudobivnik, Fadi Aldakheel, Peter Wriggers, Virtual element formulation for finite strain elastodynamics, CMES - Computer Modeling in Engineering and Sciences 129 (3) (2021) 1151–1180.

[27] K. Park, H. Chi, G.H. Paulino, On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration, Comput. Methods Appl. Mech. Engrg. 356 (2019) 669–684.

[28] K. Park, H. Chi, G.H. Paulino, Numerical recipes for elastodynamic virtual element methods with explicit time integration, Internat. J. Numer. Methods Engrg. 121 (1) (2020) 1–31.

[29] P. Wriggers, W.T. Rust, B. Reddy, A virtual element method for contact, Comput. Mech. 58 (2016) 1039–1050.

[30] P. Wriggers, W.T. Rust, A virtual element method for frictional contact including large deformations, Eng. Comput. 36 (7) (2019) 2133–2161.

[31] 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, A Special Issue in Honor of the Lifetime Achievements of J. Tinsley Oden.

[32] A. Lamperti, M. Cremonesi, U. Perego, A. Russo, C. Lovadina, A Hu-Washizu variational approach to self-stabilized quadrilateral virtual elements: 2D linear elastodynamics, Comput. Mech. 74 (2024) 393–415.

[33] 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.

[34] A. Hussein, F. Aldakheel, B. Hudobivnik, P. Wriggers, P.A. Guidault, O. Allix, A computational framework for brittle crack-propagation based on efficient virtual element method, Finite Elem. Anal. Des. 159 (2019) 15–32.

[35] P. Wriggers, B. Hudobivnik, A low order virtual element formulation for finite elasto-plastic deformations, Comput. Methods Appl. Mech. Engrg. 327 (2017) 459–477.

[36] M. Cihan, B. Hudobivnik, F. Aldakheel, P. Wriggers, 3D mixed virtual element formulation for dynamic elasto-plastic analysis, Comput. Mech. 68 (2021) 1–18.

[37] P. Wriggers, B.D. Reddy, W. Rust, B. Hudobivnik, Efficient virtual element formulations for compressible and incompressible finite deformations, Comput. Mech. 60 (2017) 253–268.

[38] P. Wriggers, M.L. De Bellis, B. Hudobivnik, A Taylor–Hood type virtual element formulations for large incompressible strains, Comput. Methods Appl. Mech. Engrg. 385 (2021) 114021.

[39] K. Park, H. Chi, G.H. Paulino, B-bar virtual element method for nearly incompressible and compressible materials, Meccanica 56 (6) (2021) 1423–1439.

[40] L. Beirao 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.

[41] L. Beriao da Veiga, D. Mora, G. Vacca, The Stokes complex for virtual elements with application to Navier–Stokes flows, J. Sci. Comput. 81 (2019) 990–1018.

[42] N. Sukumar, M.R. Tupek, Virtual elements on agglomerated finite elements to increase the critical time step in elastodynamic simulations, Internat. J. Numer. Methods Engrg. 123 (19) (2022) 4702–4725.

[43] P.B. Bochev, C.R. Dohrmann, M.D. Gunzburger, Stabilization of low-order mixed finite elements for the Stokes equations, SIAM J. Numer. Anal. 44 (1) (2006) 82–101.

[44] C.R. Dohrmann, P.B. Bochev, A stabilized finite element method for the Stokes problem based on polynomial pressure projections, Internat. J. Numer. Methods Fluids 46 (2) (2004) 183–201.

[45] R. Courant, K. Friedrichs, H. Lewy, On the partial difference equations of mathematical physics, IBM J. Res. Dev. 11 (2) (1967) 215–234.

[46] A. Franci, E. Oñate, J.M. Carbonell, Unified Lagrangian formulation for solid and fluid mechanics and FSI problems, Comput. Methods Appl. Mech. Engrg. 298 (2016) 520–547.

[47] S.W. Cheng, T.K. Dey, J. Shewchuk, S. Sahni, Delaunay Mesh Generation, CRC Press Boca Raton, 2013.

[48] B.M. Klingner, Improving Tetrahedral Meshes, University of California, Berkeley, 2008.

[49] B. Joe, Construction of three-dimensional improved-quality triangulations using local transformations, SIAM J. Sci. Comput. 16 (6) (1995) 1292–1307.

[50] M. Wicke, D. Ritchie, B.M. Klingner, S. Burke, J.R. Shewchuk, J.F. O'Brien, Dynamic local remeshing for elastoplastic simulation, ACM Trans. Graph. (TOG) 29 (4) (2010) 1–11.

[51] P.F. Antonietti, S. Berrone, M. Busetto, M. Verani, Agglomeration-based geometric multigrid schemes for the virtual element method, SIAM J. Numer. Anal. 61 (1) (2023) 223–249.

[52] T. Sorgente, F. Vicini, S. Berrone, S. Biasotti, G. Manzini, M. Spagnuolo, Mesh quality agglomeration algorithm for the virtual element method applied to discrete fracture networks, Calcolo 60 (2) (2023) 27.

[53] L. Zhang, J. Ouyang, X. Zhang, On a two-level element-free Galerkin method for incompressible fluid flow, Appl. Numer. Math. 59 (8) (2009) 1894–1904.

[54] G.X. Wu, R.E. Taylor, Finite element analysis of two-dimensional non-linear transient water waves, Appl. Ocean Res. 16 (6) (1994) 363–372.

[55] S. Koshizuka, Y. Oka, Moving particle semi-implicit method: Fully Lagrangian analysis of incompressible flows, in: Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS), Barcelona, Spain, 2000, pp. 11–14.

[56] J.C. Martin, W.J. Moyce, J.C. Martin, W.J. Moyce, W.G. Penney, A.T. Price, C.K. Thornhill, Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane, Philos. Trans. R. Soc. Lond. Ser. A 244 (882) (1952) 312–324.

[57] C. Hu, M. Sueyoshi, Numerical simulation and experiment on dam break problem, J. Mar. Sci. Appl. 9 (2010) 109–114.

[58] E. Oñate, A. Franci, J.M. Carbonell, Lagrangian formulation for finite element analysis of quasi-incompressible fluids with reduced mass losses, Internat. J. Numer. Methods Fluids 74 (10) (2014) 699–731.