Computers and Structures 181 (2017) 89–102

Contents lists available at ScienceDirect

Computers and Structures

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

A high-order stabilised ALE finite element formulation for the Euler equations on deformable domains

Ruben Sevilla *, Antonio J. Gil *, Michael Weberstadt

Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Swansea, SA1 8EN Wales, UK

ARTICLE INFO

Article history:
Received 27 October 2015
Accepted 1 November 2016
Available online 22 December 2016

Keywords:
High-order
Stabilised finite elements
Deformable domain
SUPG
Euler equations

ABSTRACT

This paper presents a high-order accurate stabilised finite element formulation for the simulation of transient inviscid flow problems in deformable domains. This work represents an extension of the methodology described in Sevilla et al. (2013), where a high-order stabilised finite element formulation was used as an efficient alternative for the simulation of steady flow problems of aerodynamic interest. The proposed methodology combines the Streamline Upwind/Petrov-Galerkin method with the generalised-α method and employs an Arbitrary Lagrangian Eulerian (ALE) description to account for the motion of the underlying mesh. Two computational frameworks, based on the use of reference and spatial variables are presented, discussed and thoroughly compared. In the process, a tailor-made discrete geometric conservation law is derived in order to ensure that a uniform flow field is exactly reproduced. Several numerical examples are presented in order to illustrate the performance of the proposed methodology. The results demonstrate the optimal approximation properties of both spatial and temporal discretisations as well as the crucial benefits, in terms of accuracy, of the exact satisfaction of the discrete geometric conservation law. In addition, the behaviour of the proposed high-order formulation is analysed in terms of the chosen stabilisation parameter. Finally, the benefits of using high-order approximations for the simulation of inviscid flows in moving domains are discussed by comparing low and high-order approximations for the solution of the Euler equations on a deformable domain.

1. Introduction

The last decade has seen an increasing interest on the study of high-order finite element methods for a vast range of engineering problems. These numerical techniques have shown the potential to offer the same accuracy as low-order methods yet with a reduced computational cost. For instance, in areas such as computational electromagnetics, the use of high-order methods has shown advantages due to their lower dissipation and dispersion compared to low order methods, see for instance [2–5]. In the area of computational fluid dynamics, high-order methods have significantly attracted the attention of the research community over the last years [6]. This has been partially motivated by the ever increasing need of simulating high Reynolds number flows, where traditional finite volume methods, still predominant in industrial solvers, require an excessive number of degrees of freedom to achieve the desired level of accuracy [7].

The research in high-order methods for fluid problems during the last five years has been mainly focused on the development of accurate and efficient discontinuous Galerkin methods [8], see [9–14] to name but a few, and the generation of arbitrary order meshes suitable for high Reynolds number computations [15–21]. Despite the popularity of stabilised finite element methods such as the Streamline Upwind/Petrov-Galerkin (SUPG) method [22] and the Galerkin Least Squares method [23], both in industry and academia [24,25], their use has been traditionally restricted to linear finite elements, see for instance the review in [26]. More recently, some authors have pointed out the advantages of using stabilised finite elements in a high-order context [1,27–29]. For instance, in [1], the authors show that the SUPG method combined with quadratic and cubic approximations offers a better performance than the traditional SUPG method with linear finite elements. For a variety of steady inviscid subsonic and transonic flows and viscous laminar test cases, the advantages shown are not only in terms of a reduction of number of degrees of freedom but, more importantly, in terms of the computational time. Similar conclusions have been also presented in [30,31], where three dimensional and two dimensional turbulent test cases were

* Corresponding authors.
E-mail addresses: r.sevilla@swansea.ac.uk (R. Sevilla), a.j.gil@swansea.ac.uk (A.J. Gil).

http://dx.doi.org/10.1016/j.compstruc.2016.11.019
0045-7949/

studied and the SUPG method was compared against the discontinuous Galerkin method.

There is a wealth of fluid dynamics applications where numerical simulations can involve moving and deformable domains [32] (e.g., fluid-structure interaction problems, free-surface flows). The numerical solution of such problems is usually carried out either considering an embedded approach [33–39] or a more classical mesh boundary-fitted approach. In the case of mesh boundary-fitted approaches, two popular techniques to account for the motion of the mesh are the space-time formulation [40–42] and the semi-discrete Arbitrary Lagrangian Eulerian (ALE) formulation [43–47]. In the context of ALE high-order approximations, the main focus over the last few years has been on the development of discontinuous Galerkin formulations, see for instance [48,49]. In contrast, very little effort has been made on developing competitive high-order stabilised finite element formulations. With this in mind, this paper advocates for the use of an ALE high-order stabilised SUPG finite element formulation for the simulation of inviscid fluid flows in deformable domains.

In a different context, that of high speed multi-material problems, the use of ALE formulations involving moving and deformable domains has also been the object of intense research over the years [50–52] (refer to [53] for a recent and comprehensive review of the subject). In all the cases, an area of debate has been the so-called Geometric Conservation Law (GCL). The GCL, first introduced in [54], is a consistency condition that states that geometrical parameters must be computed in such a way that a uniform flow field is exactly represented, with no dependence on the motion of the mesh. Although several papers state that this condition is not required, see the review in [55], the work by Farhat and co-workers [56–58] has shown the relation between the GCL and the accuracy and stability of a numerical scheme. In this paper, a tailor-made discrete version of the GCL is derived where a convenient local expression (e.g. at Gauss point level) is obtained by taking advantage of the consideration of the ALE equations mapped to the reference domain. In contrast, classical ALE formulations mapped to the spatial domain [55,43–46] do not show this local feature, resulting in integral expressions which can be computationally demanding.

The present work represents an extension of the work presented in [1] and a step towards the development of a high-order accurate stabilised finite element formulation for fluid-structure interaction problems. The high-order SUPG method in [1] is extended to problems involving deformable domains by employing the ALE form of the Euler equations. In addition, the generalised-α method is considered in order to simulate transient problems with the required level of accuracy. Moreover, the present work presents a numerical study to illustrate the effect of the stabilisation parameter on the high-order solution of a non-linear hyperbolic problem.

The remainder of the paper is organised as follows. Section 2 recalls the ALE formulation of the Euler equations for an inviscid compressible fluid. Two formulations are presented differing on the selected solution variables. Section 3 presents the weak formulation of the problem. The temporal and spatial discretisations are presented in Sections 4 and 5, respectively, and the discrete GCL for the proposed scheme is derived in Section 6. In Section 7, five numerical studies are presented in order to demonstrate the applicability and performance of the proposed methodology. The first examples involve the Burgers' equations and are used to illustrate the importance of the stabilisation parameter in a stabilised finite element formulation and show the optimal properties of both the space and time approximations. The performance of the generalised-α method and the importance of the GCL is also studied using the numerical solution of Burgers' equation. The last example involves the numerical solution of the Euler equations of gas dynamics and shows the optimal properties of the approximation in a more complex setting and the advantages of using high-order polynomials in the spatial discretisation. Finally, Section 8 summarises the main conclusions of the work.

2. Arbitrary Lagrangian Eulerian formulation of the Euler equations

The simulation of fluid flow problems in the presence of a moving domain requires a specific treatment of the motion of the computational mesh. In this section, the ALE formulation of the Euler equations is recalled.

2.1. Time dependent mapping

Let us consider a time-dependent spatial domain $\Omega_t \subset \mathbb{R}^2$ with coordinates $\boldsymbol{x} = (x_1, x_2)^T$ and the reference domain $\Omega \subset \mathbb{R}^2$ with coordinates $\boldsymbol{\mathcal{X}} = (\mathcal{X}_1, \mathcal{X}_2)^T$. A diffeomorphism between the reference and the spatial configurations is denoted by

$$\boldsymbol{\Phi} : \Omega \times [0, T_f] \longrightarrow \Omega_t(\boldsymbol{\mathcal{X}}, t) \mapsto \boldsymbol{\Phi}(\boldsymbol{\mathcal{X}}, t) := \boldsymbol{x}, \tag{1}$$

where T_f denotes the final time. The deformation gradient tensor $\boldsymbol{G}$ and the mesh velocity $\boldsymbol{V}$ can be defined in terms of the time-dependent mapping $\boldsymbol{\Phi}$ as

$$\boldsymbol{G} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mathcal{X}}} \quad \text{and} \quad \boldsymbol{V} = \frac{\partial \boldsymbol{x}}{\partial t}, \tag{2}$$

respectively. The deformation gradient tensor, also known as fibre map, is used to map differential fibre elements from reference $d\boldsymbol{\mathcal{X}}$ to spatial $d\boldsymbol{x}$ configurations, as $d\boldsymbol{x} = \boldsymbol{G} d\boldsymbol{\mathcal{X}}$. In addition, the Jacobian $g = \det(\boldsymbol{G})$ of the mapping, also known as volume map, is used to map differential volume elements from reference $d\Omega$ to spatial $d\Omega_t$ configurations, as $d\Omega_t = g d\Omega$. Finally, the element area vector is mapped from initial $d\mathcal{A}$ to final $d\boldsymbol{a}$ configurations by means of the two-point tensor $\boldsymbol{H}$ (also known as co-factor of the deformation), which is related to the deformation gradient via Nanson's rule [59]

$$d\boldsymbol{a} = \boldsymbol{H} d\mathcal{A}; \qquad \boldsymbol{H} = g\boldsymbol{G}^{-T} \equiv \mathrm{cof}(\boldsymbol{G}). \tag{3}$$

2.2. Eulerian formulation

Euler equations express the conservation of mass, linear momentum and total energy of a compressible fluid. In the absence of external volume forces, the Eulerian strong form of these conservation laws in a time-dependent domain Ω_t can be expressed in conservative form as

$$\left.\frac{\partial \boldsymbol{U}}{\partial t}\right|_{\boldsymbol{x}} + \frac{\partial \boldsymbol{F}_i(\boldsymbol{U})}{\partial x_i} = \boldsymbol{0}, \tag{4}$$

where $\boldsymbol{U}$ is the vector of conservation variables, $\boldsymbol{F}_i(\boldsymbol{U})$ is the inviscid flux vector for the spatial dimension x_i, the symbol $\bullet|_{\boldsymbol{x}}$ emphasises the evaluation of the field $\bullet$ with respect to the spatial coordinate $\boldsymbol{x}$ and the Einstein notation is assumed for repeated indices. In two dimensions, these variables and fluxes are given by

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho E \end{pmatrix}, \quad \boldsymbol{F}_1 = \begin{pmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ (\rho E + p) v_1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{F}_2 = \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ (\rho E + p) v_2 \end{pmatrix}. \tag{5}$$

In the above expressions, ρ is the density, $\boldsymbol{v} = (v_1, v_2)^T$ is the velocity vector, E is the total energy and p is the pressure. An equation of state, relating energy to pressure and density, completes this system of nonlinear hyperbolic equations. For a perfect polytropic gas, the equation of state is

$$p = (\gamma - 1)\rho\left(E - \frac{1}{2}|\boldsymbol{v}|^2\right) \tag{6}$$

where γ is the ratio of the specific heat coefficients (specific heat at constant pressure over specific heat at constant volume), with value $\gamma = 1.4$ for air, see [60] for more details. As it is well-known, above Eq. (4) can be written in quasilinear format in terms of the flux Jacobian matrix $\boldsymbol{A}_i = \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{U}}$ as

$$\left.\frac{\partial \boldsymbol{U}}{\partial t}\right|_{\boldsymbol{x}} + \boldsymbol{A}_i(\boldsymbol{U})\frac{\partial \boldsymbol{U}}{\partial x_i} = \boldsymbol{0}. \tag{7}$$

2.3. ALE formulation with reference variables

Following [47], an alternative ALE conservative form of the Euler equations can be written as

$$\left.\frac{\partial \mathcal{U}}{\partial t}\right|_{\mathcal{X}} + \frac{\partial \mathcal{F}_I(\mathcal{U}/g, \boldsymbol{V}, \boldsymbol{H})}{\partial \mathcal{X}_I} = \boldsymbol{0}, \tag{8}$$

where

$$\mathcal{U} = g\boldsymbol{U}; \qquad \mathcal{F}_I(\mathcal{U}/g, \boldsymbol{V}, \boldsymbol{H}) = (\boldsymbol{F}_i(\mathcal{U}/g) - (\mathcal{U}/g)V_i)H_{iI}. \tag{9}$$

In above Eqs. (8) and (9), $\mathcal{U}$ represent the vector of 'reference' conservation variables (mapped to the reference domain), $\mathcal{F}_I(\mathcal{U}/g, \boldsymbol{V}, \boldsymbol{H})$ its associated flux vector in the reference direction $\mathcal{X}_I$ and the symbol $\bullet|_{\mathcal{X}}$ emphasises the evaluation of the field $\bullet$ with respect to the reference coordinate $\boldsymbol{\mathcal{X}}$. Notice that the flux vector is a function of the reference conservation variables $\mathcal{U}$ and, in addition, of the mapping $\boldsymbol{\Phi}$ via its fundamental kinematic measures $g, \boldsymbol{V}$ and $\boldsymbol{H}$.

It is instructive to re-write the above ALE Euler Eq. (8) in quasi-linear form.[1] Application of the chain rule to (8) yields

$$\left.\frac{\partial \mathcal{U}}{\partial t}\right|_{\mathcal{X}} + \frac{\partial \mathcal{F}_I}{\partial(\mathcal{U}/g)}\frac{\partial(\mathcal{U}/g)}{\partial \mathcal{X}_I} + \frac{\partial \mathcal{F}_I}{\partial \boldsymbol{V}}\frac{\partial \boldsymbol{V}}{\partial \mathcal{X}_I} + \frac{\partial \mathcal{F}_I}{\partial \boldsymbol{H}} : \frac{\partial \boldsymbol{H}}{\partial \mathcal{X}_I} = \boldsymbol{0}. \tag{10}$$

Above Eq. (10) can be re-written as

$$\begin{aligned}&\left.\frac{\partial \mathcal{U}}{\partial t}\right|_{\mathcal{X}} + \mathcal{A}_I(\mathcal{U}/g, \boldsymbol{V}, \boldsymbol{H})\frac{\partial \mathcal{U}}{\partial \mathcal{X}_I} - \frac{\mathcal{U}}{g}((\nabla_{\mathcal{X}}\boldsymbol{V}) : \boldsymbol{H}) \\ &- \frac{\mathcal{U}}{g}\left(\boldsymbol{V} \cdot \underbrace{(\nabla_{\mathcal{X}} \cdot \boldsymbol{H})}_{\boldsymbol{0}}\right) \\ &= \boldsymbol{0},\end{aligned} \tag{11}$$

where $\nabla_{\mathcal{X}} := \frac{\partial}{\partial \mathcal{X}}$ represents the reference gradient operator and $\nabla_{\mathcal{X}}\cdot$ the reference divergence operator. Notice that the last term on the left hand side is zero (as $\boldsymbol{H}$ is a divergence free field [61]) and the flux Jacobian matrix $\mathcal{A}_I(\mathcal{U}/g, \boldsymbol{V}, \boldsymbol{H})$ is given by

$$\begin{aligned}\mathcal{A}_I(\mathcal{U}/g, \boldsymbol{V}, \boldsymbol{H}) &= \frac{\partial \mathcal{F}_I}{\partial \mathcal{U}} = \frac{\partial \mathcal{F}_I}{\partial(\mathcal{U}/g)}\frac{\partial(\mathcal{U}/g)}{\partial \mathcal{U}} \\ &= \frac{1}{g}(\boldsymbol{A}_i(\mathcal{U}/g) - \boldsymbol{I}V_i)H_{iI},\end{aligned} \tag{12}$$

where $\boldsymbol{I}$ denotes the identity tensor. The flux Jacobian matrix $\mathcal{A}_I(\mathcal{U}/g, \boldsymbol{V}, \boldsymbol{H})$ depends not only on the reference conservation spatial variables $\mathcal{U}$, but also on the kinematic measures $g, \boldsymbol{V}$ and $\boldsymbol{H}$.

[1] The quasi-linear form will be used later on when obtaining the SUPG stabilisation of the ALE equations in reference variables.

2.4. ALE formulation with spatial variables

Alternatively, taking into account the relationship between spatial $\boldsymbol{U}$ and reference $\mathcal{U}$ variables ($\mathcal{U} = g\boldsymbol{U}$), Eq. (8) can be written in terms of the spatial variables $\boldsymbol{U}$ as

$$g\left.\frac{\partial \boldsymbol{U}}{\partial t}\right|_{\mathcal{X}} + \boldsymbol{U}\left.\frac{\partial g}{\partial t}\right|_{\mathcal{X}} + \frac{\partial \mathcal{F}_I(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{H})}{\partial \mathcal{X}_I} = \boldsymbol{0}, \tag{13}$$

or, in quasi-linear form[2] (refer to Eq. (11)), as

$$g\left.\frac{\partial \boldsymbol{U}}{\partial t}\right|_{\mathcal{X}} + \boldsymbol{U}\left.\frac{\partial g}{\partial t}\right|_{\mathcal{X}} + \widehat{\mathcal{A}}_I(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{H})\frac{\partial \boldsymbol{U}}{\partial \mathcal{X}_I} - \boldsymbol{U}((\nabla_{\mathcal{X}}\boldsymbol{V}) : \boldsymbol{H}) = \boldsymbol{0}, \tag{14}$$

with the flux Jacobian matrix $\widehat{\mathcal{A}}_I(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{H})$ defined as

$$\widehat{\mathcal{A}}_I(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{H}) = (\boldsymbol{A}_i(\boldsymbol{U}) - \boldsymbol{I}V_i)H_{iI}. \tag{15}$$

For completeness, the relationship between the different flux Jacobian matrices defined thus far (refer to Eqs. (7), (12), (15)) is written as

$$\widehat{\mathcal{A}}_I(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{H}) = g\mathcal{A}_I(\mathcal{U}/g, \boldsymbol{V}, \boldsymbol{H}) = (\boldsymbol{A}_i(\boldsymbol{U}) - \boldsymbol{I}V_i)H_{iI}. \tag{16}$$

Remark 1. It is worth emphasising that the time derivative terms appearing in Eqs. (4) and (7) denote the so-called spatial (local) time derivative, whereas the time derivative terms in Eqs. (8), (10), (11), (13) and (14) denote the referential time derivative. The relationship between these two time derivatives is simply

$$\left.\frac{\partial \square}{\partial t}\right|_{\mathcal{X}} = \left.\frac{\partial \square}{\partial t}\right|_{\boldsymbol{x}} + \frac{\partial \square}{\partial \boldsymbol{x}} \cdot \boldsymbol{V}, \tag{17}$$

where the last term on the right hand side denotes the convective term. In the remainder of the paper, and unless otherwise stated, the subindex of the referential time derivative will be dropped to simplify the notation, as

$$\dot{\square} \equiv \left.\frac{\partial \square}{\partial t}\right|_{\mathcal{X}}. \tag{18}$$

3. Weak formulation

This section presents the stabilised finite element weak formulation of the ALE form of the Euler equations using the SUPG method [22,26]. A regular partition of the reference domain Ω into non-overlapping elements is assumed, i.e. $\Omega = \bigcup_e \Omega_e$, such that $\Omega_i \bigcap \Omega_j = \varnothing$, for $i \neq j$. The weighting functions and the solution are approximated in terms of piecewise continuous polynomials of order p in space, i.e. they belong to the finite dimensional space

$$\mathcal{V}_h := \left\{\boldsymbol{V}_h \mid \boldsymbol{V}_h \in \left(\mathcal{C}^0(\Omega)\right)^m, \boldsymbol{V}_h|_{\Omega_e} \in \left(\mathcal{P}_{\mathrm{p}}(\Omega_e)\right)^m, \forall \Omega_e \in \Omega\right\}, \tag{19}$$

where m is the number of components of the solution vector ($m = 4$ in two dimensions).

3.1. Weak formulation with reference variables

Using the framework with reference variables presented in Section 2.3, the weak formulation is: find $\mathcal{U} \in \mathcal{V}_h$ such that

$$\mathcal{A}_{\mathtt{Gal}}(\mathcal{U}, \boldsymbol{W}) + \mathcal{A}_{\mathtt{SUPG}}(\mathcal{U}, \boldsymbol{W}) + \mathcal{A}_{\mathtt{BC}}(\mathcal{U}, \boldsymbol{W}) = 0 \tag{20}$$

for all $\boldsymbol{W} \in \mathcal{V}_h$, where

[2] The quasi-linear form will also be used later on to obtain the SUPG stabilisation of the ALE equations in spatial variables.

$$\mathcal{A}_{\texttt{Gal}}(\mathcal{U},\boldsymbol{W}) = \int_\Omega \left(\boldsymbol{W}\cdot\dot{\mathcal{U}} - \frac{\partial \boldsymbol{W}}{\partial \mathcal{X}_I}\cdot \boldsymbol{\mathcal{F}}_I(\mathcal{U}/g,\boldsymbol{V},\boldsymbol{H}) \right) d\Omega, \tag{21}$$

$$\mathcal{A}_{\texttt{BC}}(\mathcal{U},\boldsymbol{W}) = \int_{\partial\Omega} \boldsymbol{W}\cdot\boldsymbol{\mathcal{F}}_I(\mathcal{U}/g,\boldsymbol{V},\boldsymbol{H})\,\mathcal{N}_I\,d\Gamma \tag{22}$$

and

$$\mathcal{A}_{\texttt{SUPG}}(\mathcal{U},\boldsymbol{W}) = \sum_e \int_{\Omega_e} \boldsymbol{W}^{st}(\boldsymbol{W})\cdot\left(\dot{\mathcal{U}} + \frac{\partial \boldsymbol{\mathcal{F}}_I(\mathcal{U}/g,\boldsymbol{V},\boldsymbol{H})}{\partial \mathcal{X}_I}\right) d\Omega, \tag{23}$$

with

$$\boldsymbol{W}^{st}(\boldsymbol{W}) = \boldsymbol{\tau}(\mathcal{U}/g,\boldsymbol{V},\boldsymbol{H})\,\boldsymbol{\mathcal{A}}_J^T(\mathcal{U}/g,\boldsymbol{V},\boldsymbol{H})\,\frac{\partial \boldsymbol{W}}{\partial \mathcal{X}_J}, \tag{24}$$

where $\boldsymbol{W}^{st}$ denotes the SUPG stabilised test function.

The terms $\mathcal{A}_{\texttt{Gal}}$ and $\mathcal{A}_{\texttt{BC}}$ correspond to the standard Galerkin formulation of the ALE form of the Euler equations and $\mathcal{N}_I$ denotes the Ith component of the outward unit normal vector to $\partial\Omega$. The term $\mathcal{A}_{\texttt{SUPG}}$ corresponds to the consistent SUPG stabilisation term, which is needed to counterbalance the negative diffusion introduced by the standard Galerkin formulation [43]. The SUPG term includes the so-called intrinsic time-scale matrix $\boldsymbol{\tau}$ which is key for the success of a stabilised formulation. For non-linear problems, such as the Euler equations, an optimal value of the stabilisation parameter is not known. However, a number of alternative definitions have been proposed and compared, see for instance [26]. This work adopts the definition proposed in [62] given by $\boldsymbol{\tau} = \tau\boldsymbol{I}$, where

$$\tau = \left(\frac{1}{\tau_1^2} + \frac{1}{\tau_2^2}\right)^{-\frac{1}{2}}. \tag{25}$$

Here, τ_1 represents the advective limit and τ_2 the transient limit, with these limits defined as

$$\tau_1 = \left(\sum_{a=1}^{\texttt{n}_{\texttt{en}}} \left(\frac{c}{\|\nabla_{\mathcal{X}}\rho\|} |\nabla_{\mathcal{X}}\rho\cdot\nabla_{\mathcal{X}}N_a| + |(\boldsymbol{v}-\boldsymbol{V})\cdot\nabla_{\mathcal{X}}N_a| \right)\right)^{-1} \quad \text{and}$$
$$\tau_2 = \frac{\Delta t}{2}. \tag{26}$$

Here, $c = \sqrt{\frac{\gamma p}{\rho}}$ is the speed of sound, $\texttt{n}_{\texttt{en}}$ denotes the number of nodes per element, N_a denotes the shape function associated with node a of the mesh and $\Delta t = t^{n+1} - t^n$ is the time step. It is important to highlight how, as a result of the underlying moving domain, for the evaluation of τ_1 in (26), apart from the consideration of $\mathcal{U}/g$, the difference between fluid $\boldsymbol{v}$ and domain $\boldsymbol{V}$ velocities is used in conjunction with the reference gradient $\nabla_{\mathcal{X}} \equiv \boldsymbol{H}^T\nabla_x$.

3.2. Weak formulation with spatial variables

Analogously, using the framework with spatial variables presented in Section 2.4, the weak formulation is: find $\boldsymbol{U} \in \mathcal{V}_h$ such that

$$\widehat{\mathcal{A}}_{\texttt{Gal}}(\boldsymbol{U},\boldsymbol{W}) + \widehat{\mathcal{A}}_{\texttt{SUPG}}(\boldsymbol{U},\boldsymbol{W}) + \mathcal{A}_{\texttt{BC}}(\boldsymbol{U},\boldsymbol{W}) = 0 \tag{27}$$

for all $\boldsymbol{W} \in \mathcal{V}_h$, where

$$\widehat{\mathcal{A}}_{\texttt{Gal}}(\boldsymbol{U},\boldsymbol{W}) = \int_\Omega \left((\boldsymbol{W}\cdot\dot{\boldsymbol{U}})g + (\boldsymbol{W}\cdot\boldsymbol{U})\dot{g} - \frac{\partial \boldsymbol{W}}{\partial \mathcal{X}_I}\cdot\boldsymbol{\mathcal{F}}_I(\boldsymbol{U},\boldsymbol{V},\boldsymbol{H})\right) d\Omega, \tag{28}$$

$$\mathcal{A}_{\texttt{BC}}(\boldsymbol{U},\boldsymbol{W}) = \int_{\partial\Omega} \boldsymbol{W}\cdot\boldsymbol{\mathcal{F}}_I(\boldsymbol{U},\boldsymbol{V},\boldsymbol{H})\,\mathcal{N}_I\,d\Gamma \tag{29}$$

and

$$\widehat{\mathcal{A}}_{\texttt{SUPG}}(\boldsymbol{U},\boldsymbol{W}) = \sum_e \int_{\Omega_e} \widehat{\boldsymbol{W}}^{st}(\boldsymbol{W}) \cdot\left(\dot{\boldsymbol{U}}g + \boldsymbol{U}\dot{g} + \frac{\partial \boldsymbol{\mathcal{F}}_I(\boldsymbol{U},\boldsymbol{V},\boldsymbol{H})}{\partial \mathcal{X}_I}\right) d\Omega, \tag{30}$$

with

$$\widehat{\boldsymbol{W}}^{st}(\boldsymbol{W}) = \boldsymbol{\tau}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{H})\,\widehat{\boldsymbol{\mathcal{A}}}_J^T(\boldsymbol{U},\boldsymbol{V},\boldsymbol{H})\,\frac{\partial \boldsymbol{W}}{\partial \mathcal{X}_J}, \tag{31}$$

where $\widehat{\boldsymbol{W}}^{st}$ denotes the SUPG stabilised test function.

4. Temporal discretisation

This section presents the semi-discrete system of equations after temporal discretisation is carried out by using the generalised-α method [63]. Both ALE formulations, namely reference and spatial variables, are presented in what follows.

The generalised-α method was originally proposed in [64] for second-order problems in the context of structural dynamics to be later extended to first-order systems arising in fluid dynamics applications [63]. The method is particularly well suited for the integration of stiff systems due to its inherent dissipation properties. Its efficiency and accuracy have been studied in [65], showing an excellent performance in comparison with other time integration algorithms.

The generalised-α method expresses a solution field $\mathcal{U}$ and its time derivative $\dot{\mathcal{U}}$ at time steps $t^{n+\alpha_f}$ and $t^{n+\alpha_m}$, respectively, in terms of the solution and its time derivative at time steps t^n and t^{n+1}, as

$$\mathcal{U}^{n+\alpha_f} = \alpha_f\left(\mathcal{U}^{n+1} - \mathcal{U}^n\right) + \mathcal{U}^n, \tag{32}$$

$$\dot{\mathcal{U}}^{n+\alpha_m} = \frac{\alpha_m}{\vartheta\Delta t}\left(\mathcal{U}^{n+1} - \mathcal{U}^n\right) + \left(1 - \frac{\alpha_m}{\vartheta}\right)\dot{\mathcal{U}}^n, \tag{33}$$

and

$$\dot{\mathcal{U}}^{n+1} = \frac{1}{\vartheta\Delta t}\left(\mathcal{U}^{n+1} - \mathcal{U}^n\right) - \left(\frac{1-\vartheta}{\vartheta}\right)\dot{\mathcal{U}}^n. \tag{34}$$

Notice that the solution fields $\mathcal{U}$ ($\dot{\mathcal{U}}$) represent indistinctly either $\mathcal{U}$ ($\dot{\mathcal{U}}$) when the formulation in terms of reference variables is considered or $\boldsymbol{U}$ ($\dot{\boldsymbol{U}}$) when the formulation in terms of spatial variables is employed.

The method is known to be second-order accurate for the solution $\mathcal{U}$ and first-order accurate for its time derivative $\dot{\mathcal{U}}$ if $\vartheta = 1/2 + \alpha_m - \alpha_f$, see [63] for more details. The parameters α_m and α_f are commonly expressed in terms of a single parameter β_∞ that enables to easily control the high frequency dissipation introduced by the algorithm. The expressions

$$\alpha_m = \frac{3-\beta_\infty}{2(1+\beta_\infty)} \quad \text{and} \quad \alpha_f = \frac{1}{1+\beta_\infty}, \tag{35}$$

are typically used, see [64,63,65] for more details. It is worth noting that different choices of the parameter β_∞ allow to recover well-known methods, such as the trapezoidal rule ($\beta_\infty = 1$) or the two-step backward difference formula ($\beta_\infty = 0$) [66].

4.1. Semi-discrete system with reference variables

Following the application of the generalised α-method, it is now possible to replace $\dot{\mathcal{U}}$ by $\dot{\mathcal{U}}^{n+\alpha_m}$ and $\mathcal{U}$ by $\mathcal{U}^{n+\alpha_f}$ in the weak form defined by Eqs. (20)–(24), to yield the semi-discrete system

$$\int_\Omega \boldsymbol{W}\cdot\dot{\boldsymbol{\mathcal{U}}}^{n+\alpha_m}d\Omega-\int_\Omega\frac{\partial\boldsymbol{W}}{\partial\mathcal{X}_I}\cdot\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{\mathcal{U}}/g,\boldsymbol{V},\boldsymbol{H})\,d\Omega+\int_{\partial\Omega}\boldsymbol{W}\cdot\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{\mathcal{U}}/g,\boldsymbol{V},\boldsymbol{H})\,\mathcal{N}_I\,d\Gamma+\sum_e\int_{\Omega_e}\boldsymbol{W}^{st,n+\alpha_f}(\boldsymbol{W})\cdot\left(\dot{\boldsymbol{\mathcal{U}}}^{n+\alpha_m}+\frac{\partial\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{\mathcal{U}}/g,\boldsymbol{V},\boldsymbol{H})}{\partial\mathcal{X}_I}\right)d\Omega=0, \tag{36}$$

where the flux vector $\mathcal{F}_I$ and the SUPG stabilised test function $\boldsymbol{W}^{st}$ are evaluated at time $t^{n+\alpha_f}$.

4.2. Semi-discrete system with spatial variables

Analogously, replacing $\dot{\boldsymbol{U}}$ by $\dot{\boldsymbol{U}}^{n+\alpha_m}$ and $\boldsymbol{U}$ by $\boldsymbol{U}^{n+\alpha_f}$ in the weak form defined by Eqs. (27)–(31) leads to the semi-discrete system

$$\int_\Omega\left(\boldsymbol{W}\cdot\dot{\boldsymbol{U}}^{n+\alpha_m}\right)g d\Omega+\int_\Omega(\boldsymbol{W}\cdot\boldsymbol{U}^{n+\alpha_f})\dot{g}d\Omega-\int_\Omega\frac{\partial\boldsymbol{W}}{\partial\mathcal{X}_I}\cdot\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{H})\,d\Omega+\int_{\partial\Omega}\boldsymbol{W}\cdot\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{H})\,\mathcal{N}_I\,d\Gamma+\sum_e\int_{\Omega_e}\widehat{\boldsymbol{W}}^{st,n+\alpha_f}(\boldsymbol{W})\cdot\left(\dot{\boldsymbol{U}}^{n+\alpha_m}g+\boldsymbol{U}^{n+\alpha_f}\dot{g}+\frac{\partial\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{H})}{\partial\mathcal{X}_I}\right)d\Omega=0, \tag{37}$$

where the flux vector $\mathcal{F}_I$ and the SUPG stabilised test function $\widehat{\boldsymbol{W}}^{st}$ are evaluated at time $t^{n+\alpha_f}$.

5. Spatial discretisation

This section presents the spatial discretisation of the semi-discrete system of equations derived in the previous section. The solution and its time derivative are approximated on a reference element, with local coordinates $\boldsymbol{\xi}=(\xi,\eta)$, using Lagrange polynomials of order p, as

$$\mathcal{U}_h(\xi,t)=\sum_{b=1}^{\mathrm{n_{en}}}N_b(\xi)\mathcal{U}_b(t),\qquad\dot{\mathcal{U}}_h(\xi,t)=\sum_{b=1}^{\mathrm{n_{en}}}N_b(\xi)\dot{\mathcal{U}}_b(t) \tag{38}$$

where $\{\mathcal{U}_b\}_{b=1,\dots,\mathrm{n_{en}}}$ and $\{\dot{\mathcal{U}}_b\}_{b=1,\dots,\mathrm{n_{en}}}$ are the time-dependent nodal values of the solution and its time derivative respectively and N_b are the shape functions, Lagrange polynomials of degree p in this work. The standard isoparametric mapping is considered between the local coordinates $\boldsymbol{\xi}$ and the reference coordinates $\boldsymbol{\mathcal{X}}$, namely

$$\boldsymbol{\mathcal{X}}(\boldsymbol{\xi})=\sum_{b=1}^{\mathrm{n_{en}}}N_b(\boldsymbol{\xi})\boldsymbol{\mathcal{X}}_b, \tag{39}$$

where $\{\boldsymbol{\mathcal{X}}_b\}_{b=1,\dots,\mathrm{n_{en}}}$ denote the nodal coordinates of a generic element Ω_e, see for instance [67].

5.1. Discrete system with reference variables

Introducing the approximations for $\mathcal{U}_h$ and $\dot{\mathcal{U}}_h$ defined in Eq. (38) within Eq. (36) and selecting the weighting function $\boldsymbol{W}$ in the space spanned by the shape functions, the non-linear system of equations

$$\boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{U}}^{n+1},\boldsymbol{\mathcal{U}}^{n},\dot{\boldsymbol{\mathcal{U}}}^{n})=\boldsymbol{0} \tag{40}$$

is obtained, where the elemental contribution to the global residual vector $\boldsymbol{\mathcal{R}}$ is computed as

$$\boldsymbol{\mathcal{R}}_a^e=\int_{\Omega_e}N_a\dot{\boldsymbol{\mathcal{U}}}_h^{n+\alpha_m}d\Omega-\int_{\Omega_e}\frac{\partial N_a}{\partial\mathcal{X}_I}\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{\mathcal{U}}_h/g,\boldsymbol{V},\boldsymbol{H})d\Omega+\int_{\partial\Omega_e\cap\partial\Omega}N_a\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{\mathcal{U}}_h/g,\boldsymbol{V},\boldsymbol{H})\,\mathcal{N}_I\,d\Gamma+\int_{\Omega_e}\boldsymbol{\mathcal{Q}}_a^{st,n+\alpha_f}(N_a)\left(\dot{\boldsymbol{\mathcal{U}}}_h^{n+\alpha_m}+\frac{\partial\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{\mathcal{U}}_h/g,\boldsymbol{V},\boldsymbol{H})}{\partial\mathcal{X}_I}\right)d\Omega, \tag{41}$$

where

$$\boldsymbol{\mathcal{Q}}_a^{st,n+\alpha_f}(N_a)=\boldsymbol{\tau}^{n+\alpha_f}(\boldsymbol{\mathcal{U}}_h/g,\boldsymbol{V},\boldsymbol{H})\,\boldsymbol{\mathcal{A}}_J^{n+\alpha_f}(\boldsymbol{\mathcal{U}}_h/g,\boldsymbol{V},\boldsymbol{H})\,\frac{\partial N_a}{\partial\mathcal{X}_J}, \tag{42}$$

and the time discretisation of the kinematic measures $g,\boldsymbol{V}$ and $\boldsymbol{H}$ will be presented later in Section 6. A Newton-Raphson algorithm is then applied by linearising the non-linear residual and truncating the Taylor expansion of $\boldsymbol{\mathcal{R}}_{(k)}$ at first order, leading to the sparse non-symmetric linear system of equations

$$\frac{\partial\boldsymbol{\mathcal{R}}_{(k)}\left(\boldsymbol{\mathcal{U}}_{(k)}^{n+1},\boldsymbol{\mathcal{U}}^{n},\dot{\boldsymbol{\mathcal{U}}}^{n}\right)}{\partial\boldsymbol{\mathcal{U}}_{(k)}^{n+1}}\Delta\boldsymbol{\mathcal{U}}_{(k)}^{n+1}=-\boldsymbol{\mathcal{R}}_{(k)}\left(\boldsymbol{\mathcal{U}}_{(k)}^{n+1},\boldsymbol{\mathcal{U}}^{n},\dot{\boldsymbol{\mathcal{U}}}^{n}\right) \tag{43}$$

to be solved at each iteration of the Newton algorithm, where the kth iterative approximation of the solution is denoted by $\boldsymbol{\mathcal{U}}_{(k)}^{n+1}$ and $\Delta\boldsymbol{\mathcal{U}}_{(k)}^{n+1}=\boldsymbol{\mathcal{U}}_{(k+1)}^{n+1}-\boldsymbol{\mathcal{U}}_{(k)}^{n+1}$.

The linear systems are solved using the generalised minimum residual (GMRES) method proposed in [68] combined with an incomplete LU (ILU) factorisation as a pre-conditioner [69]. A detailed analysis of different pre-conditioners that can be used for the simulation of compressible Euler flows by using stabilised high-order methods can be found in [70].

5.2. Discrete system with spatial variables

Similarly, introducing the approximations for $\boldsymbol{U}_h$ and $\dot{\boldsymbol{U}}_h$ defined in Eq. (38) within Eq. (37) and selecting the weighting function $\boldsymbol{W}$ in the space spanned by the shape functions, the non-linear system of equations

$$\mathbf{R}(\boldsymbol{U}^{n+1},\boldsymbol{U}^{n},\dot{\boldsymbol{U}}^{n})=\boldsymbol{0} \tag{44}$$

is obtained, where the elemental contribution to the global residual vector $\mathbf{R}$ is computed as

$$\mathbf{R}_a^e=\int_{\Omega_e}N_a\dot{\boldsymbol{U}}_h^{n+\alpha_m}g d\Omega+\int_{\Omega_e}N_a\boldsymbol{U}_h^{n+\alpha_f}\dot{g}d\Omega-\int_{\Omega_e}\frac{\partial N_a}{\partial\mathcal{X}_I}\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{U}_h,\boldsymbol{V},\boldsymbol{H})d\Omega+\int_{\partial\Omega_e\cap\partial\Omega}N_a\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{U}_h,\boldsymbol{V},\boldsymbol{H})\,\mathcal{N}_I\,d\Gamma+\int_{\Omega_e}\widehat{\mathbf{Q}}_a^{st,n+\alpha_f}(N_a)\left(\dot{\boldsymbol{U}}_h^{n+\alpha_m}g+\boldsymbol{U}_h^{n+\alpha_f}\dot{g}+\frac{\partial\boldsymbol{\mathcal{F}}_I^{n+\alpha_f}(\boldsymbol{U}_h,\boldsymbol{V},\boldsymbol{H})}{\partial\mathcal{X}_I}\right)d\Omega, \tag{45}$$

where

$$\widehat{\mathbf{Q}}_a^{st,n+\alpha_f}(N_a)=\boldsymbol{\tau}^{n+\alpha_f}(\boldsymbol{U}_h,\boldsymbol{V},\boldsymbol{H})\,\widehat{\boldsymbol{\mathcal{A}}_J}^{n+\alpha_f}(\boldsymbol{U}_h,\boldsymbol{V},\boldsymbol{H})\,\frac{\partial N_a}{\partial\mathcal{X}_J}, \tag{46}$$

and the time discretisation of the kinematic measures $g,\boldsymbol{V}$ and $\boldsymbol{H}$ will be presented later in Section 6. A Newton-Raphson algorithm is again applied, resulting in a sparse non-symmetric linear system of equations to be solved at each iteration of the Newton algorithm.

Remark 2. From the computational standpoint, it is interesting to highlight that linearisation of the element residual contributions $\boldsymbol{\mathcal{R}}_a^e$ (41) and $\mathbf{R}_a^e$ (45) involves linearisation of the stabilisation terms

$\boldsymbol{Q}_a^{st,n+\alpha_f}(N_a)$ and $\widehat{\boldsymbol{Q}}_a^{st,n+\alpha_f}(N_a)$, respectively. In this work, we follow one of the numerical approaches analysed in [71], where the above stabilisation terms are evaluated at time t^n, namely $\boldsymbol{Q}_a^{st,n}(N_a)$ and $\widehat{\boldsymbol{Q}}_a^{st,n}(N_a)$, hence avoiding the need for linearisation. The authors in [71] show that the update of the stabilisation parameter once every time step yields a faster convergence rate than that obtained when the update is carried out at every Newton-Raphson iteration (without impairing the accuracy of the overall scheme).

6. Geometric conservation law

The geometric conservation law (GCL), first introduced in [54], is a consistency requirement that guarantees that a uniform flow field can be exactly represented by the numerical algorithm. In the presence of a uniform flow field $\boldsymbol{U} := \bar{\boldsymbol{U}}_0$, both ALE formulations presented in Sections 2.3 and 2.4 reduce to

$$\dot{g} - (\nabla_{\mathcal{X}} \boldsymbol{V}) : \boldsymbol{H} = 0. \tag{47}$$

Above differential Eq. (47), which possesses the form of a conservation law, establishes a geometric relationship between the time-varying Jacobian field g and $\boldsymbol{V}$ and $\boldsymbol{H}$ [72,61], which is fulfilled at the continuum level.

However, attention must be paid when space-time discretisation is carried out. Integration by parts in (36), use of Remark 2 and consideration of a uniform flow field yields

$$\int_{\Omega} \boldsymbol{W} \cdot \bar{\boldsymbol{U}}_0 \left[\dot{g}^{n+\alpha_m} - \left(\nabla_{\mathcal{X}} \boldsymbol{V}^{n+\alpha_f}\right) : \boldsymbol{H}^{n+\alpha_f}\right] d\Omega + \sum_e \int_{\Omega_e} \boldsymbol{W}^{st,n}(\boldsymbol{W}) \cdot \bar{\boldsymbol{U}}_0 \left[\dot{g}^{n+\alpha_m} - \left(\nabla_{\mathcal{X}} \boldsymbol{V}^{n+\alpha_f}\right) : \boldsymbol{H}^{n+\alpha_f}\right] d\Omega = 0. \tag{48}$$

Analogously, integration by parts in (37), use of Remark 2 and consideration of a uniform flow field yields an identical expression to (48), except replacing $\boldsymbol{W}^{st,n}$ with $\widehat{\boldsymbol{W}}^{st,n}$. For above Eq. (48) to hold, we propose in this paper a sufficient condition to be fulfilled at every Gauss point, namely

$$\dot{g}^{n+\alpha_m} - \left(\nabla_{\mathcal{X}} \boldsymbol{V}^{n+\alpha_f}\right) : \boldsymbol{H}^{n+\alpha_f} = 0, \tag{49}$$

which is the discrete counterpart of the above GCL (47) for the stabilised formulation presented here. Above Eq. (49) is merely the time discretisation of the continuum GCL Eq. (47) when using the generalised-α method. Fulfilment of this discrete equation at every Gauss point ensures that both terms in above Eq. (48) are zero, namely the Galerkin and the stabilisation terms.

Remark 3. Note that the local character of the GCL correction (49) comes as a direct result of the consideration of the ALE equations mapped to the reference domain Ω, which enables their expression in the local form (8). In contrast, classical ALE formulations, mapped to the spatial domain Ω_t [55,43–46], do not present this local feature and they can only be expressed in an integral form, namely

$$\frac{d}{dt} \int_{\Omega_t} \boldsymbol{U} d\Omega + \int_{\partial\Omega_t} \left(\boldsymbol{F}_i(\boldsymbol{U}) - \boldsymbol{U} V_i\right) n_i \, d\Gamma = \boldsymbol{0}. \tag{50}$$

As a result, the GCL correction results in an integral equation, which can still be dealt with efficiently in the context of a Finite Volume approach [44–46], due to the local character of every control volume. However, when considering a stabilised Finite Element Method, such as the one pursued in this paper, this integral correction can be computationally very expensive, requiring the assembly of a global tangent matrix.

In addition, when only the temporal derivative is mapped to the reference domain, it is necessary to account for the time dependency of the shape functions used to approximate the solution, see for instance [73], introducing an extra cost in the computation.

The strategy followed in this paper is, provided that the time-varying fields $\boldsymbol{V}$ and $\boldsymbol{H}$ are given by the mapping $\boldsymbol{\Phi}$, to consider the Jacobian field g as an unknown variable to be solved alongside the ALE fluid equations by means of the discrete GCL Eq. (49). With that in mind, the symbol $\bar{g}$ is used to represent the discrete CGL compliant Jacobian field (instead of g). In this work, (49) is solved at every Gauss point (once per time step), where the value of the Jacobian $\bar{g}$ at time t^{n+1} is computed as

$$\bar{g}^{n+1} = \bar{g}^n + \frac{\vartheta \Delta t}{\alpha_f} \left[\nabla_{\mathcal{X}} \boldsymbol{V}^{n+\alpha_f} : \boldsymbol{H}^{n+\alpha_f} - \left(1 - \frac{\alpha_f}{\vartheta}\right) \dot{\bar{g}}^n\right]. \tag{51}$$

The following correction, similar to that presented in [48] in the context of discontinuous Galerkin methods, is then applied to the solution variable at each iteration of the Newton algorithm at every Gauss point

$$\mathcal{U}_{(k)}^{n+1} \leftarrow \mathcal{U}_{(k)}^{n+1} \frac{\bar{g}^{n+1}}{g^{n+1}}. \tag{52}$$

Remark 4. The mesh velocity can be computed analytically when a time-dependent mapping is known or directly extracted from the iterations of the generalised-α method. The first strategy can only be applied in problems where the motion is prescribed and an analytical expression of the time-dependent mapping is known, whereas the second option is the most appropriate choice in realistic problems where an analytical expression of the mapping is not available (e.g. fluid-structure interaction).

7. Numerical examples

This section presents five numerical studies to analyse the performance of the proposed methodology for a series of simple benchmark problems. First, a numerical study illustrates the influence of the stabilisation parameter on the accuracy of the simulations. Second, the optimal approximation properties of the proposed method both in space and time are verified against a problem with a well-known closed form solution. Third, the performance of the time integrator scheme, in terms of the accuracy of both the solution and its time derivative, is studied. Fourth, the importance of the GCL is illustrated and quantified. Fifth, a simple numerical example is used to illustrate the potential of the high-order proposed method for solving the Euler equations in a deformable domain.

7.1. Influence of the stabilisation parameter

The definition of the stabilisation parameter has been object of intensive research and its numerical value has an important effect on the accuracy and stability of the SUPG method. Optimal values of the stabilisation parameter are only known for the scalar linear convection-diffusion equation in one dimension using linear finite elements [43].

This section presents a numerical study to illustrate the effect of the stabilisation parameter on the accuracy of the SUPG scheme for the solution of the steady Burgers' equation. A steady problem is considered in order to eliminate the source of error introduced by the temporal discretisation in a transient problem. The source term and the boundary condition are obtained by selecting the analytical solution to be $U(x) = x^7 + 1$.

The stabilisation parameter is assumed to be dependent on x, not constant within each element, and to take the form

$$\tau_e = \hat{\tau}\frac{h_e}{\mathrm{p}^2|U|}, \tag{53}$$

where h_e is the length of the element I_e and the effect of the constant $\hat{\tau}$ is investigated. Fig. 1 shows the error of the numerical solution in the $\mathcal{L}^2(\Omega)$ norm, computed as

$$\left(\frac{\int_\Omega (u_h - u_{\text{exact}})^2 d\Omega}{\int_\Omega (u_{\text{exact}})^2 d\Omega}\right)^{1/2} \tag{54}$$

as a function of the constant $\hat{\tau}$ for an order of approximation ranging from $\mathrm{p}=1$ up to $\mathrm{p}=6$. The study is performed by using extremely coarse meshes where the number of elements varies from 2 to 7. The results in Fig. 1 are classified according to the parity of the order of approximation to aid the reader in observing the different behaviours of the error for odd and even degrees of approximation.

Low values of the stabilisation parameter τ, namely values of $\hat{\tau}$ of order 10^{-2}, result in a significant loss of accuracy in all cases. As the value of $\hat{\tau}$ is increased, more accurate results are obtained, but a different behaviour is consistently observed for odd and even degrees of approximation. For $\mathrm{p}=1$, Fig. 1(a) suggests the existence of an optimal value of the stabilisation parameter for values of $\hat{\tau}$ close to 1. If higher values of the stabilisation parameter are selected, a slight loss of accuracy is again observed, especially with finer meshes. A similar behaviour is observed for $\mathrm{p}=3$ and $\mathrm{p}=5$ in Figs. 1(b) and (c), although the effect of a large value of the stabilisation parameter is less pronounced as the degree of the approximation is increased.

For even values of the degree of approximation, the same behaviour is obtained for any degree of approximation and any number of elements. In all cases a value of $\hat{\tau} \geqslant 1$ provides the maximum accuracy. It is worth emphasising that large values of the stabilisation parameter, namely $\hat{\tau} \geqslant 10^4$ do not provide a loss of accuracy.

To further analyse the differences between odd and even values of the approximation degree p, Fig. 2 shows the difference between the analytical and the numerical solution in the domain for linear and quadratic when no stabilisation is added, i.e. $\tau = 0$.

The results illustrate that the nature of the numerical error is completely different for an odd or an even degree of approximation. These results demonstrate that the different behaviour of the solution, in terms of the parity of the degree of the approximation, is not induced by the SUPG stabilisation term introduced in the weak formulation. For an odd degree of approximation, the solution at mesh nodes shows an oscillatory character whereas for an even value of the degree of approximation, the oscillatory character is not observed at mesh nodes. It is worth noting that the different behaviour of the numerical solution in terms of the parity of the degree of approximation has been recently reported in [74] for the solution of the linear convection-diffusion equation with a standard Galerkin formulation.

7.2. Transient Burgers' equation with mesh motion

This section considers a numerical example to validate the implementation of the ALE formulation for the transient Burgers' equation with a moving mesh in one dimension. The ALE form of the transient Burgers' equation can be written in conservation form as

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}(\mathcal{U})}{\partial \mathcal{X}} = \mathcal{S} \qquad \mathcal{X} \in \Omega, \quad t > 0, \tag{55}$$

where $\mathcal{U} = gU, \mathcal{F} = (F - UV)$. The domain is $\Omega = [0,1]$ and a Dirichlet boundary condition is considered at $x = 0$. The source term and the boundary condition are obtained by selecting the analytical solution, $U(x,t)$, to satisfy the implicit equation

$$U(x,t) - \sin\left(2\pi(x - U(x,t)t)\right) = 0. \tag{56}$$

In order to test the performance of the proposed approach with a dynamic mesh, the time-dependent mapping

$$\Phi(\mathcal{X},t) = \mathcal{X} - \frac{1}{10}\sin(2\pi\mathcal{X})\sin\left(\frac{2\pi t}{T}\right) \tag{57}$$

is considered, where $T = 0.4$ denotes the period required to complete a cycle of the mapping.

Fig. 3 shows the initial condition ($t = 0$) and the numerical solution computed with 11 third order finite elements at time $t = 0.1$, when the displacement introduced by the time-dependent mapping is maximum.

A study of the convergence of the temporal error is performed next. The mesh is selected to be sufficiently fine so that the error induced by the spatial discretisation is always lower than the error induced by the temporal discretisation. Fig. 4 shows the convergence of the error in the $\mathcal{L}^2(\Omega)$ norm as a function of the time step Δt for the two formulations considered here (i.e., using reference and spatial variables) and for a value of $\beta_\infty = 0.5$. The error from a computation with a static mesh is also represented as a reference.

Second order convergence is observed for the error of the numerical solution and first order convergence for the error of the time derivative of the solution in all cases [64,63]. A loss of accuracy is clearly seen when the mesh is deformed according to the time-dependent mapping of Eq. (57) compared to the solution with a static mesh. This illustrates and allows us to quantify the extra error induced by the mesh motion compared to a standard computation in a static mesh.

It is important to note that the assembly of the system using spatial variables induces a marginal extra cost due to the extra terms of the residual in Eq. (45) compared to the residual with reference variables in Eq. (41). However, when the formulation with spatial variables is utilised, the computed solution is already the physical relevant quantity $\boldsymbol{U}(\boldsymbol{\mathcal{X}},t)$. In order to recover the solution at a spatial point $\boldsymbol{x}$, the time-dependent mapping is needed to obtain $\boldsymbol{U}(\boldsymbol{\Phi}(\boldsymbol{\mathcal{X}},t),t)$. On the other hand, when the formulation with reference variables is utilised, the computed solution is $\boldsymbol{\mathcal{U}}(\boldsymbol{\mathcal{X}},t)$ and the relevant physical quantity must be recovered as $\boldsymbol{U}(\boldsymbol{\mathcal{X}},t) = \boldsymbol{\mathcal{U}}(\boldsymbol{\mathcal{X}},t)g^{-1}(\boldsymbol{\mathcal{X}},t)$. It is worth noting that the mapping Jacobian g is computed at the Gauss points according to the GCL as detailed in Section 6. This means that the formulation with reference variables requires the projection of the mapping Jacobian from Gauss points to mesh nodes. For the recovery of the time derivative of the solution in the spatial configuration, the formulation with spatial variables requires the projection of the gradient of the solution to the mesh nodes in order to compute the spatial time derivative from the approximated material derivative according to the Eq. (17). The situation is slightly more involved as, first, the material time derivative of the solution must be recovered as $\dot{\boldsymbol{U}}(\boldsymbol{\mathcal{X}},t) = (\dot{\boldsymbol{\mathcal{U}}}(\boldsymbol{\mathcal{X}},t) - \boldsymbol{U}(\boldsymbol{\mathcal{X}},t)\dot{g}(\boldsymbol{\mathcal{X}},t))g^{-1}(\boldsymbol{\mathcal{X}},t)$.

The results in Fig. 4 show that the formulation with spatial variables is slightly more accurate than the formulation with reference variables, both for the solution and its time derivative. This is partially induced by the remapping required to recover the physically relevant quantities when the reference formulation is utilised. It should be emphasised that these extra steps are required because the formulation with reference variables does not approximate the physical quantity $\boldsymbol{U}$ but its product with the mapping Jacobian g.

Next, a study of the convergence of the spatial error is performed. The time step is selected to be small enough in order to guarantee that the error due to the temporal integration is lower than the spatial error and the parameter of the generalised-α

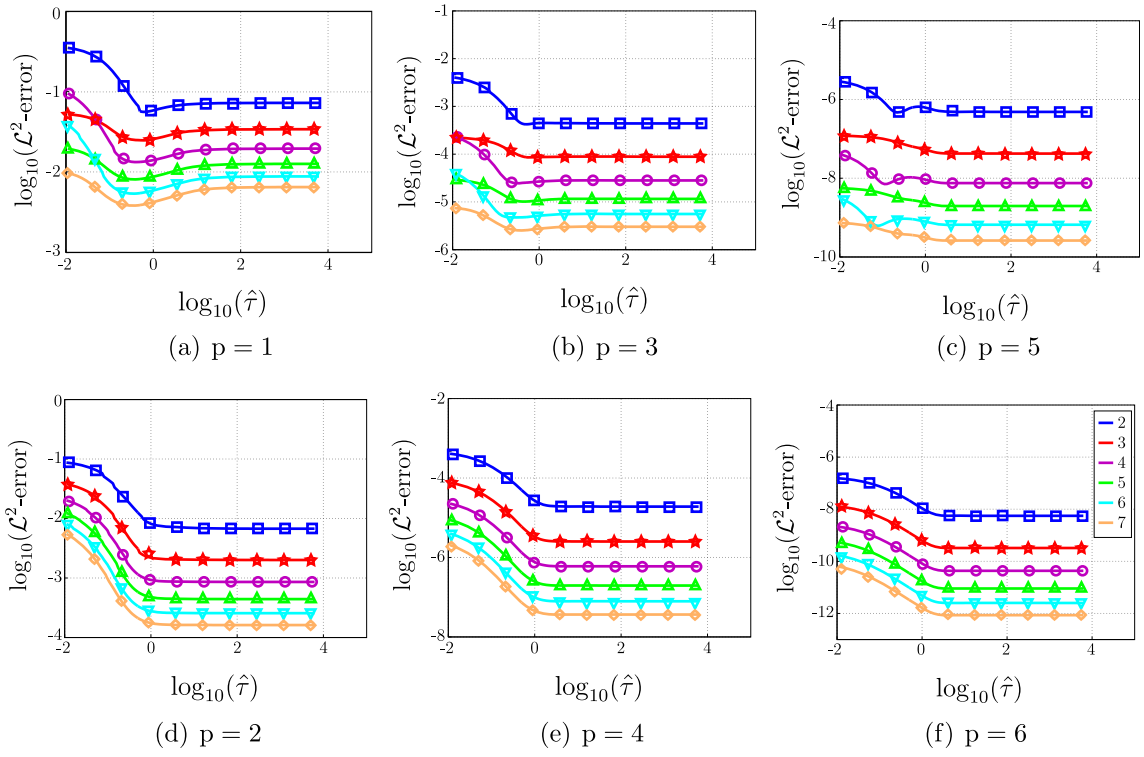


Fig. 1. Effect of the stabilisation parameter on the accuracy of the numerical solution of the steady Burgers' equation with different degrees of approximation ranging from p = 1 to p = 6. Each plot shows the error as a function of $\hat{\tau}$ appearing in Eq. (53) for coarse meshes with a number of elements ranging from 2 to 7.

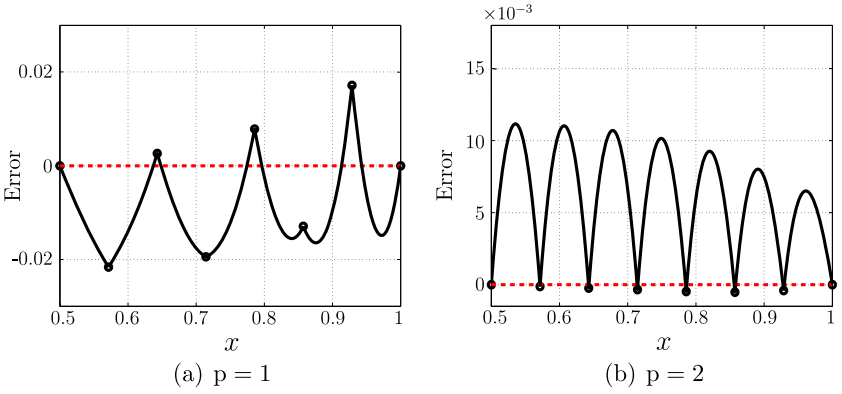


Fig. 2. Error distribution of the numerical solution of the steady Burgers' equation with $\tau = 0$ for linear and quadratic elements. The circles represent the value of the error at the mesh nodes.

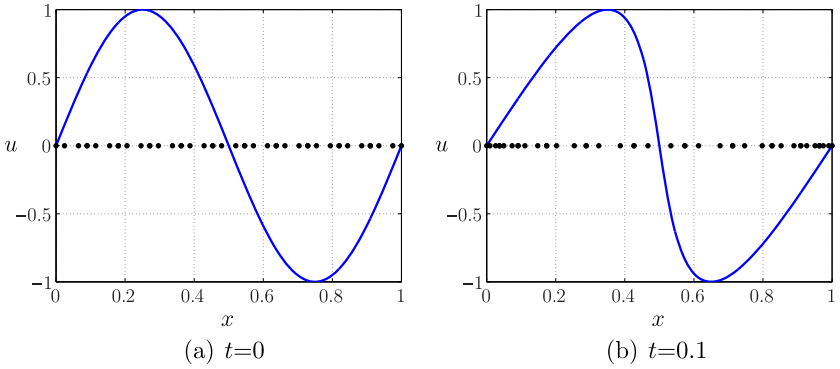


Fig. 3. Numerical solution of the transient Burgers' equation and high-order nodes for a mesh with 11 elements and p = 3.

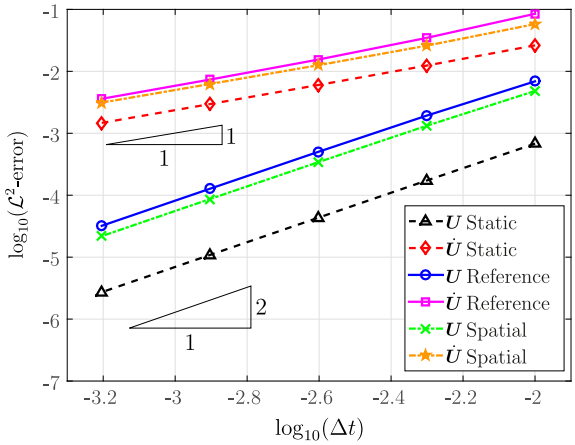


Fig. 4. Evolution of the error of the numerical solution and its time derivative in the $\mathcal{L}^2(\Omega)$ norm as a function of the time step Δt, for a static mesh computation and a dynamic mesh computation with reference and spatial variables using $p = 3$ and $\beta_\infty = 0.5$.

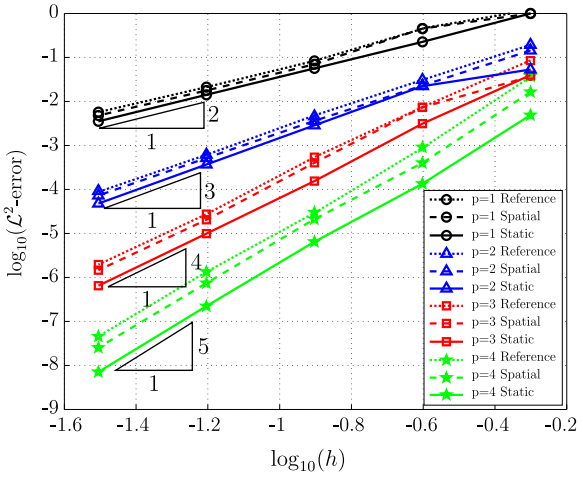


Fig. 5. Evolution of the error of the numerical solution of the transient Burgers' equation in the $\mathcal{L}^2(\Omega)$ norm as a function of the element size h, for two problems with static and dynamic mesh using both formulations with spatial and reference variables and for different degrees of approximation p by using $\beta_\infty = 1$.

method is selected to be $\beta_\infty = 1$. Fig. 5 shows the convergence of the error in the $\mathcal{L}^2(\Omega)$ norm as a function of the element size h for computations with a static mesh and a dynamic mesh using both formulations with spatial and reference variables and for different values of the degree of approximation p. The error from a computation with a static mesh is also presented for reference.

Again, the numerical solution in a moving mesh is less accurate than the numerical solution for a static mesh. This is, once more, induced by the extra terms in the ALE formulation and also due to the variation of the element size in time for a dynamic case. In all cases, the optimal rate of convergence (i.e., $p + 1$) of the error in the $\mathcal{L}^2(\Omega)$ norm is observed for a degree of approximation ranging from $p = 1$ to $p = 4$.

This numerical experiment also illustrates the benefit of using high-order polynomials for the spatial discretisation. For instance, the computation in the first mesh with a fourth order polynomial approximation provides an accuracy comparable to the computation in the finest mesh with linear polynomials. This means that the computation with fourth order elements requires approximately eight times fewer elements and half the number of degrees of freedom to provide the same accuracy as a low order computation.

7.3. Performance of the generalised-α method

The next numerical experiment section considers the same numerical example used in the previous Section to assess the accuracy of the generalised-α method when different values of the parameter β_∞ are used. This parameter is known to have a significant influence on the accuracy of the method for the simulations with both static and dynamic meshes as it enables to control the high frequency dissipation introduced by the algorithm [64,63].

Fig. 6 shows the convergence of the error of the numerical solution and its time derivative in the $\mathcal{L}^2(\Omega)$ norm as a function of the time step Δt for the two formulations presented (i.e., using reference and spatial variables) and for different values of the parameter β_∞.

As observed in Fig. 6(a), second order convergence is obtained for the error of the numerical solution for any choice of the parameter β_∞. The less accurate results are observed when $\beta_\infty = 0$, corresponding to the second-order backward different formula, and the accuracy improves as the value of β_∞ is increased. The most accurate results are obtained when $\beta_\infty = 1$, corresponding to the trapezoidal rule. It is worth remarking that the results with $\beta_\infty = 1$ are more than five times more accurate than the results obtained with $\beta_\infty = 0$ and the gain obtained by increasing the value of β_∞ from 0 to 0.5 is much more significant than the gain obtained by increasing the value of β_∞ from 0.5 to 1.

When the error of the time derivative of the numerical solution is studied, a different behaviour is observed depending on the value of the parameter β_∞. Again, the most accurate results are obtained for $\beta_\infty = 1$ but, in addition, the results reveal second-order convergence only for this choice of β_∞. The results show the expected first order convergence of the error of the time derivative for any other value of the parameter β_∞. Further computations, not reported here, show that second order convergence is also observed in static mesh computations only when the trapezoidal rule is considered (i.e., for $\beta_\infty = 1$).

7.4. Importance of the geometric conservation law

The next study also considers the numerical example used in the previous two Sections to assess the importance of the GCL.

First, a uniform flow field is considered and the solution is advanced in time until a final time $T_f = 2$ with a time step $\Delta t = 0.01$. The $\mathcal{L}^2(\Omega)$ norm of the error for the numerical solution is computed at each time step and represented in Fig. 7 as a function of the physical time t.

The simulation is performed with and without enforcing the discrete GCL as detailed in Section 6. It can be clearly observed that if the GCL is not enforced the scheme is not able to exactly reproduce a uniform flow field. In this example, the error oscillates in between 10^{-2} and 10^{-3} as time evolves. In contrast, when the discrete GCL is enforced the error is kept below 10^{-12}, just representing round-off errors. For these computations, the number of integration points was increased to enable the exact integration of polynomials of order 6p and the degree of the approximation was selected to be $p = 8$ in order to eliminate any source of error due to an inaccurate representation of the sinusoidal mapping of Eq. (57).

Next, the effect the discrete GCL is assessed by performing a simulation with a non-uniform field with and without enforcing the correction of the mapping Jacobian. Fig. 8 shows the convergence of the error of the numerical solution and its time derivative in the $\mathcal{L}^2(\Omega)$ norm as a function of the time step Δt with and without enforcing the GCL and for different values of the parameter β_∞.

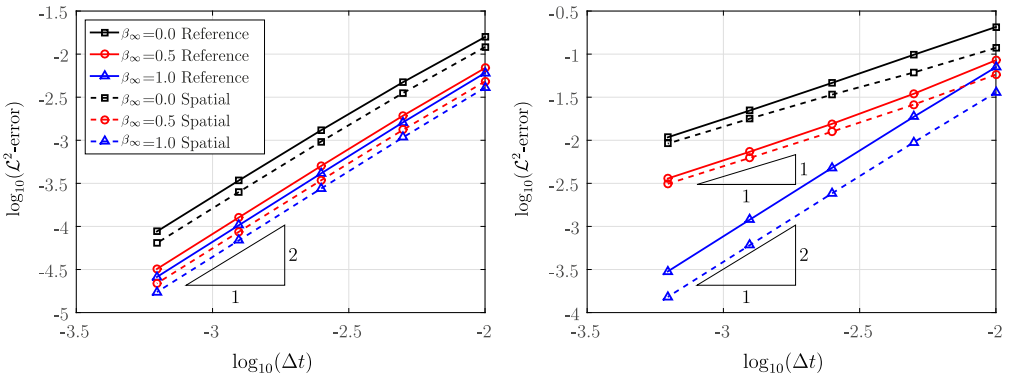


Fig. 6. Evolution of the error of the numerical solution and its time derivative in the $\mathcal{L}^2(\Omega)$ norm as a function of the time step Δt for the formulations with reference and spatial variables using different values of the parameter β_∞.

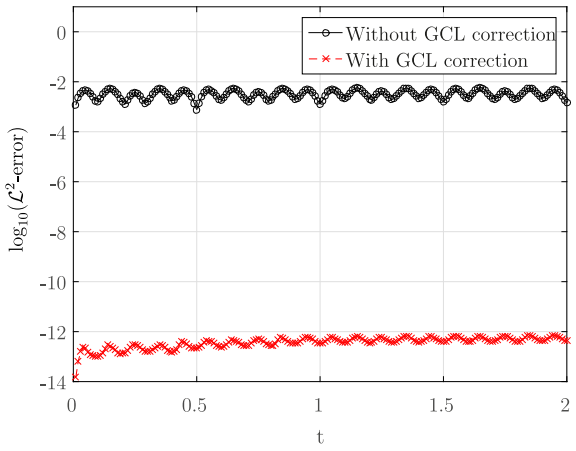


Fig. 7. Evolution of the error of the numerical solution in the $\mathcal{L}^2(\Omega)$ norm as a function of the time for the simulation of a uniform flow without the discrete GCL correction (continuous line) and with the discrete GCL correction (discontinuous line).

Despite the important differences observed when trying to exactly reproduce a uniform field, the results in Fig. 8(a) show only a marginal improvement on the solution accuracy when the discrete GCL is enforced. Similar conclusions are obtained irrespectively on the choice of the parameter β_∞ of the generalised-α method. The advantage of enforcing the GCL in terms of accuracy is more relevant when the error of the time derivative of the solution is studied. In this case, the enforcement of the GCL provides an improved approximation of the time derivative of the solution, specially when the trapezoidal rule is considered (i.e., $\beta_\infty = 1$).

7.5. Euler equations with mesh motion

The last example considers a typical test case for the simulation of the Euler equations of gas dynamics in a moving mesh, see for instance [75,76,48]. It involves the convection of a two dimensional inviscid isentropic vortex.

The computational domain is $\Omega = [-10, 10] \times [-7.5, 7.5]$ and periodic boundary conditions are considered in the whole boundary. The free-stream flow conditions correspond to a flow state $\boldsymbol{U}_\infty = (1, 0.8944, 0.4472, 3)^T$.

An analytical solution is available for this problem and it is used here in order to verify the optimality of the high-order stabilised finite element formulation proposed in this paper. The analytical solution is given by

$$\rho = \rho_\infty \left(1 - \frac{I^2(\gamma-1)M_\infty^2}{8\pi^2}\exp\left(f(x_1,x_2,t)\right)\right)^{\frac{1}{\gamma-1}} \tag{58a}$$

$$v_1 = |\boldsymbol{v}_\infty|\left(\cos\theta - \frac{I((x_2 - x_2^0) - \bar{v}_2 t)}{2\pi r_c}\exp\left(\frac{f(x_1,x_2,t)}{2}\right)\right) \tag{58b}$$

$$v_2 = |\boldsymbol{v}_\infty|\left(\sin\theta + \frac{I((x_1 - x_1^0) - \bar{v}_1 t)}{2\pi r_c}\exp\left(\frac{f(x_1,x_2,t)}{2}\right)\right) \tag{58c}$$

$$p = p_\infty \left(1 - \frac{I^2(\gamma-1)M_\infty^2}{8\pi^2}\exp\left(f(x_1,x_2,t)\right)\right)^{\frac{\gamma}{\gamma-1}} \tag{58d}$$

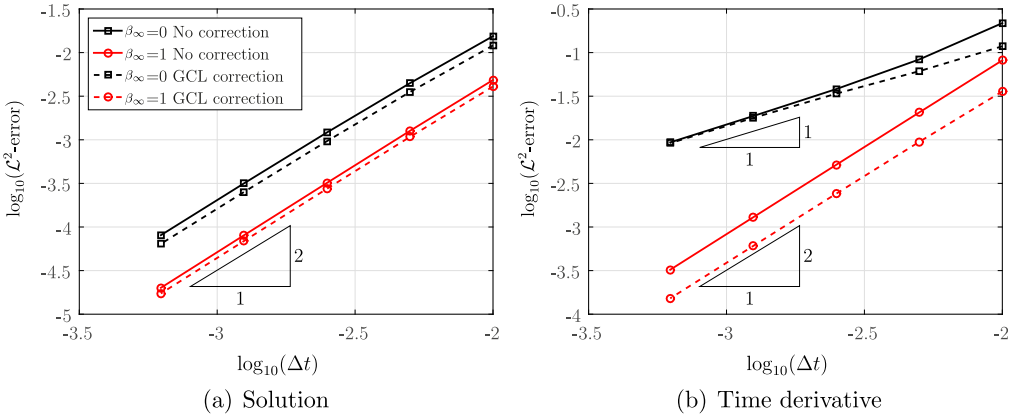
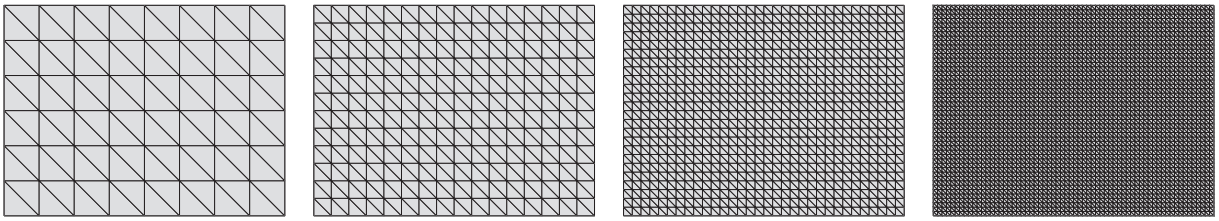


Fig. 8. Evolution of the error of the numerical solution and its time derivative in the $\mathcal{L}^2(\Omega)$ norm as a function of the time step Δt for the formulation with spatial variables using different values of the parameter β_∞ and with and without the discrete GCL correction.



(a) Mesh 1 (b) Mesh 2 (c) Mesh 3 (d) Mesh 4

Fig. 9. Four meshes used for the simulation of the convection of a two dimensional inviscid isentropic vortex.

where

$$f(x_1,x_2,t)=\frac{1-\left(x_1-x_1^0-\bar{v}_1t\right)^2-\left(x_2-x_2^0-\bar{v}_2t\right)^2}{r^2}. \tag{59}$$

Here, $M=|\boldsymbol{v}|/c$ denotes the Mach number, $I=5.0$ is the vortex intensity, $r=1.5$ defines the vortex size, $(x_1^0,x_2^0)^T=(0,0)^T$ is the initial position of the centre of the vortex, $\bar{v}_1$ and $\bar{v}_2$ are the Cartesian components of the free-stream velocity vector and $\theta=26.56^\circ$ is the angle of free stream velocity vector with respect to the x axis.

Following [48], a time-dependent mapping $\boldsymbol{\Phi}=(\Phi_1,\Phi_2)^T$ given by

$$\Phi_1(\mathcal{X}_1,\mathcal{X}_2,t)=\mathcal{X}_1+2\sin\left(\frac{\pi\mathcal{X}_1}{10}\right)\sin\left(\frac{2\pi\mathcal{X}_2}{15}\right)\sin\left(\frac{2\pi t}{T}\right) \tag{60a}$$

$$\Phi_2(\mathcal{X}_1,\mathcal{X}_2,t)=\mathcal{X}_2+\frac{3}{2}\sin\left(\frac{\pi\mathcal{X}_1}{10}\right)\sin\left(\frac{2\pi\mathcal{X}_2}{15}\right)\sin\left(\frac{4\pi t}{T}\right) \tag{60b}$$

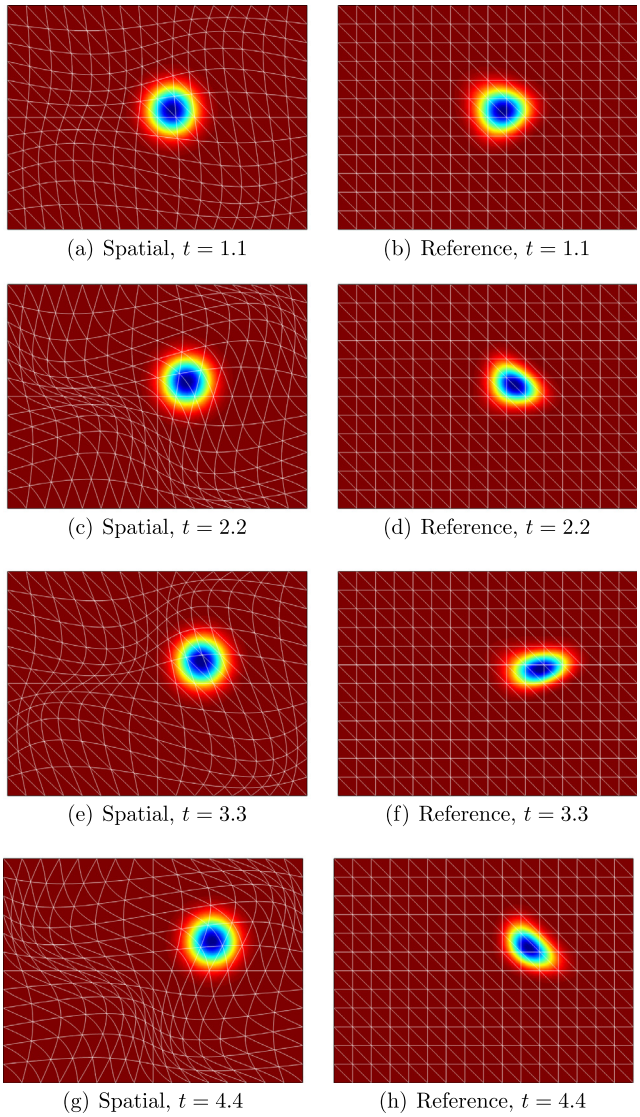
is considered in order to evaluate the performance of the proposed methodology for solving the Euler equations in a moving mesh, where $T=2$ denotes the period required to complete a cycle of the mapping.

As in the previous examples, the time-dependent mapping is only used to compute the position of the mesh nodes. The velocities are computed from the generalised-α method and not using its exact expression that can be easily derived from the mapping (60).

Four structured triangular meshes, shown in Fig. 9, with 96, 384, 1536 and 6144 elements respectively are considered and the degree of approximation in each mesh is increased from $p=1$ to $p=3$. For a degree of approximation $p=1$ the number of nodes in each mesh is 36, 221, 825 and 3185, respectively; for $p=2$ the number of nodes in each mesh is 221, 825, 3185 and 12513, respectively and, finally, for $p=3$ the number of nodes is 475, 1813, 7081 and 27985, respectively.

Following the conclusions of the previous numerical experiments with the Burgers' equation, all the computations presented in this section are performed by using the formulation with spatial variables and by using the generalised-α method with $\beta_\infty=1$.

Fig. 10 shows the numerical solution computed in the mesh shown in Fig. 9(b) with a degree of approximation $p=3$. The left plots in Fig. 10 show the density distribution at four different instants in time. The mesh is also represented for each snapshot of the solution, illustrating the high distortion introduced by the time-dependent mapping of Eq. (60). It is worth remarking that even with a very coarse mesh, using high-order polynomial approximations ($p=3$ in this example), it is possible to propagate the vortex without any observable dissipation or distortion of the density profile. The right plots in Fig. 10 show the solution at the same instants in time but in the (undeformed) reference configuration. This Figure illustrates the change of the vortex shape induced by the time-dependent mapping when the problem is sought in the reference domain.



(a) Spatial, $t=1.1$ (b) Reference, $t=1.1$

(c) Spatial, $t=2.2$ (d) Reference, $t=2.2$

(e) Spatial, $t=3.3$ (f) Reference, $t=3.3$

(g) Spatial, $t=4.4$ (h) Reference, $t=4.4$

Fig. 10. Numerical solution of the convection of a two dimensional inviscid isentropic vortex using the mesh shown in Fig. 9(b) with a degree of approximation $p=3$. The density distribution is shown at four different instants both in the spatial domain and in the reference domain.

The simulation is performed using a constant time step and a moving mesh, resulting in a change on the Courant number at each time step. The Courant number C is defined as [1]

$$C=\frac{p\Delta t}{h}\left(|\boldsymbol{v}-\boldsymbol{V}|^2+\frac{3}{2}c^2+c\sqrt{16|\boldsymbol{v}-\boldsymbol{V}|^2+c^2}\right)^{1/2}. \tag{61}$$

The variation of the Courant number for the simulation shown in Fig. 10 is depicted in Fig. 11. For comparison purposes, the constant

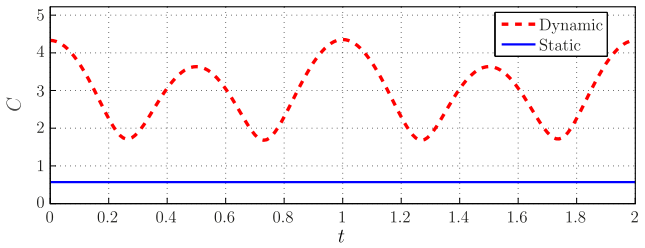


Fig. 11. Evolution of the Courant number in time for the simulation with a dynamic mesh compared to the Courant number for the static simulation.

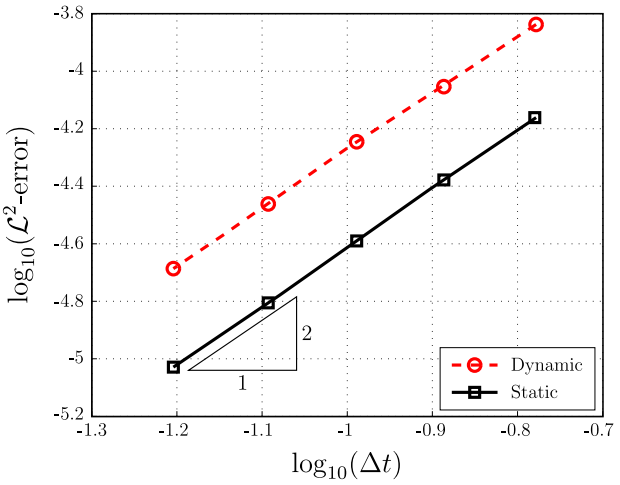


Fig. 12. Evolution of the error of the numerical solution of the Euler equations in the $\mathcal{L}^2(\Omega)$ norm as a function of the time step Δt, for two problems with a dynamic and a static mesh.

Courant number corresponding to the simulation with a static mesh and with the same degree of approximation p and time step Δt is included. The results show that depending on the velocity of the mesh relative to that of the fluid, a simulation with an underlying moving mesh can significantly increase the Courant number. Alternatively, results illustrate the variation required in the time step if the Courant number of a simulation (including mesh motion) is to be kept constant and equal to the Courant number of their corresponding static mesh.

A study of the convergence of the temporal error is performed next. The mesh is selected to be sufficiently fine so that the error induced by the spatial discretisation is always lower than the error induced by the temporal discretisation. Fig. 12 shows the convergence of the error in the $\mathcal{L}^2(\Omega)$ norm of the density as a function of the time step Δt for both static and dynamic meshes. The optimal rate of convergence is observed for both static and dynamic simulations and the results are slightly less accurate for a dynamic mesh due to the change in spatial resolution induced by the mesh motion and the extra terms of the ALE formulation. The optimal rate of convergence is also observed if the error is measured in the other conservation variables (i.e., momentum and total energy).

Finally, a study of the convergence of the spatial error is performed. In this case, the time step is selected to be small enough in order to guarantee that the error due to the temporal integration is lower than the spatial error and the parameter β_∞ of the generalised-α method is selected to be 1. Fig. 13 shows the convergence of the density error in the $\mathcal{L}^2(\Omega)$ norm as a function of the element size h for both static and dynamic meshes and for different values of the degree of approximation p. As done in previous exam-

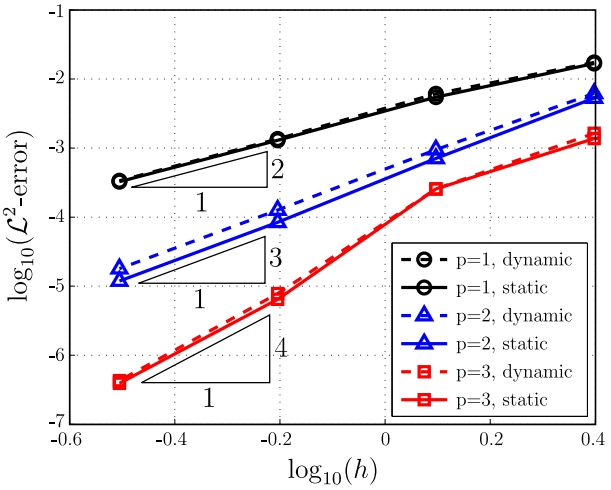


Fig. 13. Evolution of the error of the numerical solution of the Euler equations in the $\mathcal{L}^2(\Omega)$ norm as a function of the element size h, for two problems with dynamic and static mesh and for different degrees of approximation p.

ples and for comparison purposes between the static and dynamic cases, the element size for a moving mesh is considered to be the size of the initial mesh.

Again, the numerical solution in a moving mesh is slightly less accurate than the numerical solution for a static mesh but, more importantly, in all cases the optimal rate of convergence, $p+1$ of the error in the $\mathcal{L}^2(\Omega)$ norm is observed for a degree of approximation ranging from $p=1$ to $p=3$.

Fig. 13 shows that in order to obtain an accuracy of 10^{-3}, a simulation with $p=3$ can be performed on the coarsest mesh shown in Fig. 9(a) whereas a simulation with $p=1$ requires to use the third mesh shown in Fig. 9(c). This represents a save of 43% in terms of number of degrees of freedom for the high-order simulation. It can also be observed that for the same number of elements, say on the finest mesh shown in Fig. 9(d), an increase of the degree of approximation from $p=1$ to $p=2$ provides more than one order of magnitude more accurate results and, similarly, an increase of the degree of approximation from $p=2$ to $p=3$ also provides more than one order of magnitude more accurate results.

To further illustrate the benefits of using a high-order approximation, Fig. 14 shows a comparison of the density profile at time

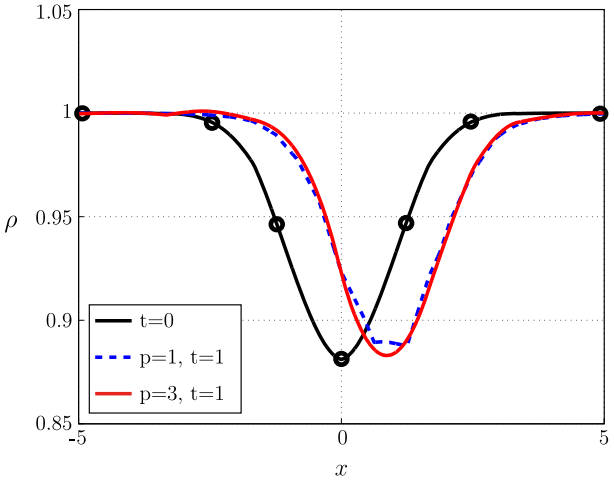


Fig. 14. Section of the density profile at time $t=0$ and at time $t=1$ computed with $p=1$ on the third mesh shown in Fig. 9(c) and with $p=3$ in the coarsest mesh shown in Fig. 9(a).

$t = 1$ computed with $p = 1$ on the third mesh shown in Fig. 9(c) and with $p = 3$ in the coarsest mesh shown in Fig. 9(a). The comparison reveals the higher dissipation introduced by the low order approach even if the mesh contains 16 elements more than the mesh used with cubic elements. The solution with linear elements also shows a slight oscillation near the minimum density value whereas with high-order elements a smooth profile is preserved.

8. Concluding remarks

The application of a high-order stabilised finite element formulation for the simulation of transient inviscid problems involving deformable domains has been considered. The proposed method is highly accurate in space and time and its optimal convergence properties have been demonstrated using numerical examples involving Burgers' and Euler equations. Two alternative formulations have been presented and tested, corresponding to the use of reference and spatial variables. The advantages of the formulation based on spatial variables have been discussed and assessed using a series of numerical examples. It has been shown that the spatial formulation provides more accurate results due to an extra remap required by the formulation with reference variables in order to recover physically relevant quantities.

The importance of the stabilisation parameter in the proposed stabilised finite element formulation has been shown for a simple problem and the numerical results clearly illustrate the different behaviour of the approximation when odd and even degrees of approximation are utilised. It has been shown that the accuracy of the computations is more sensitive to the choice of the stabilisation parameter for an odd degree of approximation.

The results also demonstrate the importance of the GCL, not only in terms of the ability to exactly reproduce a uniform flow on an underlying moving mesh but also in terms of the extra accuracy provided for a general problem. Satisfaction of the GCL reveals a slight gain in accuracy in the computed numerical solution but, crucially, a significant gain in terms of time rates of the solution. Finally, the benefits of using high-order approximations in space have been demonstrated in terms of the number of degrees of freedom required to achieve a given level of accuracy compared to traditional low order approximations.

It is in the scope of a future publication to extend the formulation presented in this paper to fluid-structure interaction problems and to further explore the performance of high-order approximations against the traditional low-order approximations. The extension to three dimensional problems, though conceptually easy, requires a very careful selection of the most appropriate preconditioner required at every nonlinear iteration. This task, along with an efficient parallelisation of our current computer platform, are the next steps of our work.

Acknowledgements

The first two authors gratefully acknowledge the financial support provided by the Sêr Cymru National Research Network for Advanced Engineering and Materials, United Kingdom. The last author acknowledges the financial support provided by EPSRC DTA grant (EP/J500318).

References

[1] Sevilla R, Hassan O, Morgan K. An analysis of the performance of a high-order stabilised finite element method for simulating compressible flows. Comput Methods Appl Mech Eng 2013;253:15–27.
[2] Ainsworth M. Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. J Comput Phys 2004;198(1):106–30.
[3] Ledger PD, Morgan K, Hassan O. Frequency and time domain electromagnetic scattering simulations employing higher order edge elements. Comput Methods Appl Mech Eng 2005;194(2–5):105–25.
[4] Sevilla R, Hassan O, Morgan K. The use of hybrid meshes to improve the efficiency of a discontinuous Galerkin method for the solution of Maxwell's equations. Comput Struct 2014;137:2–13.
[5] Sevilla R, Fernández-Méndez S, Huerta A. Comparison of high-order curved finite elements. Int J Numer Methods Eng 2011;87(8):719–34.
[6] Wang ZJ, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, Deconinck H, Hartmann R, Hillewaert K, Huynh HT, Kroll N, May G, Persson P-O, van Leer B, Visbal M. High-order CFD methods: current status and perspective. Int J Numer Methods Fluids 2013;72(8):811–45.
[7] Kroll N. The ADIGMA project. In: Kroll N, Bieler H, Deconinck H, Couaillier V, van der Ven H, Sørensen K, editors. ADIGMA – a European initiative on the development of adaptive higher-order variational methods for aerospace applications. Notes on numerical fluid mechanics and multidisciplinary design, vol. 113. Springer; 2010. p. 1–9 [chapter 1].
[8] Cockburn B. Discontinuous Galerkin methods for computational fluid dynamics. In: Stein E, de Borst R, Hughes T, editors. Encyclopedia of computational mechanics, vol. 3. New York: Wiley; 2004. p. 91–128 [chapter 4].
[9] Peraire J, Nguyen N, Cockburn B. A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations. AIAA paper 363; 2010. p. 2010.
[10] Sevilla R, Fernández-Méndez S, Huerta A. NURBS-enhanced finite element method (NEFEM): a seamless bridge between CAD and FEM. Arch Comput Methods Eng 2011;18(4):441–84.
[11] Bassi F, Botti L, Colombo A, Rebay S. Agglomeration based discontinuous Galerkin discretization of the Euler and Navier-Stokes equations. Comput Fluids 2012;61:77–85.
[12] Hartmann R. Higher-order and adaptive discontinuous Galerkin methods with shock-capturing applied to transonic turbulent delta wing flow. Int J Numer Methods Fluids 2013;72(8):883–94.
[13] Giorgiani G, Fernández-Méndez S, Huerta A. Hybridizable discontinuous Galerkin with degree adaptivity for the incompressible Navier-Stokes equations. Comput Fluids 2014;98:196–208.
[14] Kirby AC, Mavriplis DJ, Wissink AM. An adaptive explicit 3D discontinuous Galerkin solver for unsteady problems. In: 22nd AIAA computational fluid dynamics conference. p. 3046.
[15] Shephard MS, Flaherty JE, Jansen KE, Li X, Luo X, Chevaugeon N, Remacle J-F, Beall MW, O'Bara RM. Adaptive mesh generation for curved domains. Appl Numer Math 2005;52(2–3):251–71.
[16] Persson P-O, Peraire J. Curved mesh generation and mesh refinement using lagrangian solid mechanics. In: Proceedings of the 47th AIAA aerospace sciences meeting and exhibit. AIAA; 2009.
[17] Xie ZQ, Sevilla R, Hassan O, Morgan K. The generation of arbitrary order curved meshes for 3D finite element analysis. Comput Mech 2013;51(3):361–74.
[18] Poya R, Sevilla R, Gil AJ. A unified approach for a posteriori high-order curved mesh generation using solid mechanics. Comput Mech 2016;58(3):457–90.
[19] Toulorge T, Geuzaine C, Remacle J-F, Lambrechts J. Robust untangling of curvilinear meshes. J Comput Phys 2013;254:8–26.
[20] Gargallo-Peiró A, Roca X, Peraire J, Sarrate J. Distortion and quality measures for validating and generating high-order tetrahedral meshes. Eng Comput 2014:1–15.
[21] Moxey D, Ekelschot D, Keskin U, Sherwin S, Peiró J. A thermo-elastic analogy for high-order curvilinear meshing with control of mesh validity and quality. Proc Eng 2014;82(0):127–35.
[22] Brooks AN, Hughes TJR. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comput Methods Appl Mech Eng 1982;32(1–3):199–259.
[23] Hughes T, Franca L, Hulbert G. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. Comput Methods Appl Mech Eng 1989;73:173–89.
[24] Chalot FL. Industrial aerodynamics. In: Stein E, de Borst R, Hughes T, editors. Encyclopedia of computational mechanics, vol. 3. New York: Wiley; 2004. p. 407–58 [chapter 12].
[25] Corson D, Jaiman R, Shakib F. Industrial application of RANS modelling: capabilities and needs. Int J Comput Fluid Dynam 2009;23(4):337–47.
[26] Hughes T, Scovazzi G, Tezduyar TE. Stabilized methods for compressible flows. J Sci Comput 2010;43:343–68.
[27] Chalot F, Normand P-E. Higher-order stabilized finite elements in an industrial Navier-Stokes code. In: Kroll N, Bieler H, Deconinck H, Couaillier V, van der Ven H, Sørensen K, editors. ADIGMA – a European initiative on the development of adaptive higher-order variational methods for aerospace applications. Notes on numerical fluid mechanics and multidisciplinary design, vol. 113. Springer; 2010. p. 145–65 [chapter 11].
[28] Erwin JT, Anderson WK, Kapadia S, Wang L. Three-dimensional stabilized finite elements for compressible Navier-Stokes. AIAA J 2013;51(6):1404–19.
[29] Erwin JT, Anderson WK, Wang L, Kapadia S. High-order finite element method for three-dimensional turbulent Navier-Stokes. AIAA paper 2571; 2013.
[30] Glasby RS, Burgess N, Anderson K, Wang L, Allmaras S, Mavriplis D. Comparison of SU/PG and DG finite-element techniques for the compressible navier-stokes equations on anisotropic unstructured meshes. In: Proceedings

of the 51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition.
[31] Wang L, Anderson WK, Erwin JT, Kapadia S. Discontinuous Galerkin and Petrov Galerkin methods for compressible viscous flows. Comput Fluids 2014;100:13–29.
[32] Tezduyar TE. Finite element methods for flow problems with moving boundaries and interfaces. Arch Comput Methods Eng 2001;8(2):83–130.
[33] Zhang L, Gerstenberger A, Wang X, Liu WK. Immersed finite element method. Comput Methods Appl Mech Eng 2004;193(21):2051–67.
[34] Gerstenberger A, Wall WA. An extended finite element method/Lagrange multiplier based approach for fluid-structure interaction. Comput Methods Appl Mech Eng 2008;197(19):1699–714.
[35] Gil AJ, Arranz Carreño A, Bonet J, Hassan O. The immersed structural potential method for haemodynamic applications. J Comput Phys 2010;22:8613–41.
[36] Hesch C, Gil AJ, Arranz Carreño A, Bonet J. The immersed structural potential method for haemodynamic applications. Comput Methods Appl Mech Eng 2012;247–248:51–64.
[37] Gil AJ, Arranz Carreño A, Bonet J, Hassan O. An enhanced immersed structural potential method for fluid-structure interaction. J Comput Phys 2013;250:178–205.
[38] Hesch C, Gil AJ, Arranz Carreño A, Bonet J, Bestch P. A mortar approach for fluid structure interaction problems: immersed strategies for deformable and rigid bodies. Comput Methods Appl Mech Eng 2014;278:853–82.
[39] Marco O, Sevilla R, Zhang Y, Ródenas JJ, Tur M. Exact 3D boundary representation in finite element analysis based on Cartesian grids independent of the geometry. Int J Numer Methods Eng 2015;103(6):445–68.
[40] Masud A, Hughes T. A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems. Comput Methods Appl Mech Eng 1997;146:91–126.
[41] van der Vegt JJW, van der Ven H. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. General formulation. J Comput Phys 2002;182(2):546–85.
[42] Tezduyar TE, Sathe S, Keedy R, Stein K. Space-time finite element techniques for computation of fluid-structure interactions. Comput Methods Appl Mech Eng 2006;195(17):2002–27.
[43] Donea J, Huerta A. Finite element methods for flow problems. Wiley; 2005.
[44] Zhang Z, Gil A, Hassan O, Morgan K. The simulation of 3D unsteady incompressible flows with moving boundaries on unstructured meshes. Comput Fluids 2008;37(5):620–31.
[45] Wood C, Gil AJ, Hassan O, Bonet J. A partitioned coupling approach for dynamic fluid-structure interaction with applications to biological membranes. Int J Numer Methods Fluids 2008;57:555–81.
[46] Wood C, Gil AJ, Hassan O, Bonet J. Partitioned block Gauss-Seidel coupling for dynamic fluid-structure interaction. Comput Struct 2010;88:1367–82.
[47] Gil A, Bonet J, Silla J, Hassan O. A discrete geometric conservation law (DGCL) for a cell vertex finite-volume algorithm on moving domains. Int J Numer Methods Biomed Eng 2010;26(6):770–9.
[48] Persson P-O, Bonet J, Peraire J. Discontinuous Galerkin solution of the Navier-Stokes equation on deformable domains. Comput Methods Appl Mech Eng 2009;198:1585–95.
[49] Česenek J, Feistauer M, Kosik A. DGFEM for the analysis of airfoil vibrations induced by compressible flow. ZAMM-J Appl Math Mech/Zeitschrift Angew Math Mech 2013;93(6–7):387–402.
[50] Scovazzi G. A discourse on Galilean invariance, SUPG stabilization, and the variational multiscale framework. Comput Methods Appl Mech Eng 2007;196 (4–6):1108–32.
[51] Dobrev V, Ellis T, Kolev T, Rieben R. Curvilinear finite elements for Lagrangian hydrodynamics. Int J Numer Methods Fluids 2011;65(11–12):1295–310.
[52] Barlow AJ. A compatible finite element multi-material ALE hydrodynamics algorithm. Int J Numer Methods Fluids 2008;56(8):953–64.
[53] Barlow A, Maire P-H, Rider W, Rieben R, Shashkov M. Arbitrary Lagrangian Eulerian methods for modeling high-speed compressible multimaterial flows. J Comput Phys 2016;322:603–65.
[54] Thomas P, Lombard C. Geometric conservation law and its application to flow computations on moving grids. AIAA J 1979;17(10):1030–7.
[55] Donea J, Huerta A, Ponthot JP, Rodríguez-Ferran A. Arbitrary Lagrangian-Eulerian methods. In: Stein E, de Borst R, Hughes T, editors. Encyclopedia of computational mechanics, vol. 1. Wiley; 2004. p. 413–37 [chapter 14].
[56] Lesoinne M, Farhat C. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. Comput Methods Appl Mech Eng 1996;134(1):71–90.
[57] Guillard H, Farhat C. On the significance of the geometric conservation law for flow computations on moving meshes. Comput Methods Appl Mech Eng 2000;190(11):1467–82.
[58] Geuzaine P, Grandmont C, Farhat C. Design and analysis of ALE schemes with provable second-order time-accuracy for inviscid and viscous flow simulations. J Comput Phys 2003;191(1):206–27.
[59] Bonet J, Gil AJ, Wood RD. Nonlinear solid mechanics for finite element analysis: statics. Cambridge University Press; 2016.
[60] Laney CB. Computational gasdynamics. Cambridge: Cambridge University Press; 1998.
[61] Bonet J, Gil AJ, Lee CH, Aguirre M, Ortigosa R. A first order hyperbolic framework for large strain computational solid dynamics. Part I: Total Lagrangian isothermal elasticity. Comput Methods Appl Mech Eng 2015;283:689–732.
[62] Tezduyar T, Senga M. Stabilization and shock-capturing parameters in SUPG formulation of compressible flows. Comput Methods Appl Mech Eng 2006;195:1621–32.
[63] Jansen KE, Whiting CH, Hulbert GM. A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. Comput Methods Appl Mech Eng 2000;190:305–19.
[64] Chung J, Hulbert G. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized α-method. J Appl Mech 1993;60(2):371–5.
[65] Dettmer W, Perić D. An analysis of the time integration algorithms for the finite element solutions of incompressible Navier-Stokes equations based on a stabilised formulation. Comput Methods Appl Mech Eng 2003;192 (9):1177–226.
[66] Gear CW. Numerical initial value problems in ordinary differential equations. Prentice-Hall; 1971.
[67] Solin P, Segeth K, Dolezel I. Higher-order finite element methods. New York: Chapman and Hall/CRC; 2003.
[68] Saad Y, Schultz M. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Comput 1986;7:856–69.
[69] Saad Y. Iterative methods for sparse linear systems. New York: PWS Publishing; 1996.
[70] Nejat A, Ollivier-Gooch C. Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations. J Comput Phys 2008;227(4):2366–86.
[71] Catabriga L, Coutinho A, Tezduyar T. Compressible flow SUPG parameters computed from element matrices. Commun Numer Methods Eng 2005;21 (9):465–76.
[72] Gil AJ, Lee J, Bonet CH. A stabilised Petrov-Galerkin formulation for linear tetrahedral elements in compressible, nearly incompressible and truly incompressible fast dynamics. Comput Methods Appl Mech Eng 2014;276:659–90.
[73] Formaggia L, Nobile F. Stability analysis of second-order time accurate schemes for ALE–FEM. Comput Methods Appl Mech Eng 2004;193 (39):4097–116.
[74] Cai Q, Kollmannsberger S, Sala-Lardies E, Huerta A, Rank E. On the natural stabilization of convection dominated problems using high order Bubnov-Galerkin finite elements. Comput Math Appl 2014;66:2545–58.
[75] Xia Y, Liu X, Luo H, Nourgaliev R. A third-order implicit discontinuous Galerkin method based on a Hermite WENO reconstruction for time-accurate solution of the compressible Navier-Stokes equations. Int J Numer Methods Fluids 2015.
[76] Wang L, Mavriplis DJ. Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations. J Comput Phys 2007;225(2):1994–2015.