Journal of Computational Physics 458 (2022) 111093

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An entropy–stable p–adaptive nodal discontinuous Galerkin for the coupled Navier–Stokes/Cahn–Hilliard system

Gerasimos Ntoukas [a,*], Juan Manzanero [a,b], Gonzalo Rubio [a,b], Eusebio Valero [a,b], Esteban Ferrer [a,b]

[a] ETSIAE-UPM - School of Aeronautics, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros 3, E-28040 Madrid, Spain
[b] Center for Computational Simulation, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain

ARTICLE INFO

Article history:
Received 3 June 2021
Received in revised form 14 December 2021
Accepted 20 February 2022
Available online 25 February 2022

Keywords:
Discontinuous Galerkin
Multiphase
Cahn–Hilliard
Phase–field method
P-adaptation
Entropy–stable

ABSTRACT

We develop a novel entropy–stable discontinuous Galerkin approximation of the incompressible Navier–Stokes/Cahn–Hilliard system for p–non–conforming elements. This work constitutes an evolution of the work presented by Manzanero et al. ((2020) [10]), as it extends the discrete analysis into supporting p–adaptation (p–refinement/coarsening). The scheme is based on the summation–by–parts simultaneous–approximation term property along with Gauss–Lobatto points and suitable numerical fluxes. The p–non–conforming elements are connected through the classic mortar method, the use of central fluxes for the inviscid terms, and the BR1 scheme with additional dissipation for the viscous fluxes. The scheme is proven to retain its properties of the original conforming scheme when transitioning to p–non–conforming elements and to mimic the continuous entropy analysis of the model. We focus on dynamic polynomial adaptation as the applications of interest are unsteady multiphase flows. In this work, we introduce a heuristic adaptation criterion that depends on the location of the interface between the different phases and utilises the convection velocity to predict the movement of the interface. The scheme is verified to be total phase conserving, entropy–stable and freestream preserving for curvilinear p–non–conforming meshes. We also present the results for a rising bubble simulation and we show that for the same accuracy we get a $\times 2$ to $\times 6$ reduction in the degrees of freedom and a 41% reduction in the computational time. We compare our results for the three–dimensional dam break test case against experimental and numerical data and we show that a $\times 4.3$ to $\times 9.5$ reduction of the degrees of freedom and a 51% reduction in the computational time can be achieved compared to the p–uniform solution.

1. Introduction

Multiphase flow modelling has been a subject of intensive research for decades since it is of great interest from the scientific and industrial perspective. The high density and viscosity variations across the different components of the flow pose a significant challenge for the modelling of such systems. Thus, various numerical techniques have been developed to simulate systems of two or more immiscible fluids.

* Corresponding author.
E-mail address: gerasimos.ntoukas@upm.es (G. Ntoukas).

https://doi.org/10.1016/j.jcp.2022.111093

One–fluid models are divided into two large categories, the interface–tracking and the interface–capturing techniques. The former includes the Marker–And–Cell (MAC) and front–tracking methods whereas the latter include the widely known Volume Of Fluid (VOF), level–set and phase field methods [1]. The latter is also subdivided in two distinct categories, the sharp and diffuse interface methods. The sharp interface methods represent the interface with an infinitesimally thin interface and the VOF and level–set methods fall into this category.

In this work, we focus on a diffuse interface approach, the Cahn–Hilliard model (CH) [2]. This model is also denoted in literature as a phase–field method. Diffuse interface methods represent the transition between two immiscible fluids with an interface of finite width where the thermodynamic properties vary smoothly from the one phase to the other. For the Cahn–Hilliard equation, it can be proven through asymptotic analysis [3] that the model converges to the physical solution with a larger interface than that found in nature, which is of the order of nanometers [4]. The latter, consists an unrealistic scale to be resolved, especially for industrial applications, and thus the physical modelling can be achieved with substantially lower cost. The Cahn–Hilliard equation also entails additional benefits because of the favourable total mass conservation properties [5] and the bounded behaviour of its free–energy [6].

In [7] the authors presented a free–energy stable discretization of the Cahn–Hilliard equation which has been extended in [8] to support p–non–conforming elements. Then in [9] an entropy–stable approximation of the incompressible Navier–Stokes with artificial compressibility and variable density has been presented. Based on these works, an entropy–stable approximation for the incompressible Navier–Stokes/Cahn–Hilliard system has been constructed in [10]. The aim of this work is to extend the scheme of [10] on p–non–conforming element interfaces while retaining its entropy–stability characteristics. Through p–refinement (p–adaptation) we seek to reduce the computational cost compared to a uniform solution while retaining the accuracy and robustness of the original scheme. In addition, the applicability and effectiveness of p–adaptation for multiphase flow simulations is examined as the existing literature is focusing solely on mesh refinement (h-adaptation) so far.

The discretization scheme adopted is the nodal Discontinuous Galerkin Spectral Element Method (DGSEM) based on previous work of [10,11,8]. The DGSEM offers great flexibility as the solution is represented through an arbitrary approximation polynomial order and supports the use of unstructured meshes of curvilinear hexahedral elements to approximate complex geometries. One of its greatest attributes is the spatial locality, as the approximation order can vary across different elements.

As in [8], the choice of Gauss–Lobatto (GL) points, which through the Summation–By–Parts Simultaneous–Approximation Term (SBP–SAT) property [12,13], allows the derivation of an entropy–stable DGSEM. There is a plethora of publications focused on the construction of entropy– and energy– stable schemes for the discontinuous Galerkin method, see [12,14,10,15–18] and the references therein. Reviews of entropy–stable DGSEM schemes are given in [19,20]. Entropy–stable schemes for multiphase flows have also been studied, as in the work of [21]. DGSEM entropy–stable formulations are presented in [22,10]. These schemes correspond to conforming discretizations and additional considerations are necessary when transitioning to p–non–conforming elements, so that their stability properties are retained.

Some of the building blocks for the creation of entropy–stable operators using the mortar method are presented in [23–29]. These principles have been adopted to create stable DGSEM operators based on Gauss–Lobatto (GL) points and the SBP–SAT property for the Euler [30,31] and the Navier–Stokes equations [32–34] as well as more general operators based on different quadratures such as the Legendre–Gauss points [31,35]. There are many publications that focus on the construction of entropy–stable schemes for linear and non–linear problems. The scheme presented in this paper contains elements from both, as the construction of an entropy–stable operator for the Cahn–Hilliard equation resembles that of a linear problem whereas that for the Navier–Stokes is non–linear. However, it is the first time a complete entropy–stable p–non–conforming scheme for multiphase flow solver is presented.

The numerical simulations of multiphase flows incorporate a large variety of length scales, with one of the most important being that of the interface between the two phases. Within that region, there are steep gradients which need to be adequately resolved. On the contrary, away from the interface, the parameter of the multiphase flow model remains constant and thus there is no need for a high approximation order of the solution. It is typical, in multiphase flow simulations that the area around the interface requires a much higher resolution than the rest of the domain [10,36,37] and thus the h– and p– adaptation techniques can be applied to alleviate this peculiarity and reduce the cost of the simulation. The use of h–adaptation has been documented within the perspective of various multiphase flow models with some examples being [38,36,39–45,37].

The method of p–adaptation is a characteristic of higher order methods and has been studied as an alternative or compliment to h–adaptation within the context of Navier–Stokes equations with some indicative examples being [46–48,34, 49]. However, p–adaptation has not been considered for simulations of two or more immiscible fluids and the aim of this work is to showcase its characteristics and applicability for such flows.

In this paper we construct an entropy–stable p–non–conforming approximation for the incompressible Navier–Stokes/Cahn–Hilliard solver introduced in [10]. This has been achieved by modifying the conservative and non–conservative inviscid fluxes and the viscous fluxes to incorporate the mortar method [50]. The modified scheme is then proved analytically to be entropy–stable for the general case of non-conforming element boundaries following the methodology in [10]. In addition, we introduce a heuristic adaptation methodology that refines the region of the interface and coarsens the rest of the domain. The movement of the domain is predicted by the convection velocity and thus the user does not have to specify an adaptation interval as in [8]. Through the numerical experiments presented, we are able to show that a reduction

of the degrees of freedom and the computational cost of a simulation can be achieved compared to a uniform solution while retaining accuracy.

The rest of this work is organized as follows. First we introduce the model and its continuous entropy analysis in Sec. 2. Then, the spatial discretisation is introduced in Sec. 3, as well as the modified numerical fluxes that allow non-uniform polynomial order. The adaptation methodology is described in Sec. 4. In Sec. 5 the discrete entropy–stability proof of the scheme is presented. A verification for the freestream preservation and primary quantity conservation for a Cartesian three–dimensional p–non–conforming mesh is presented in Sec. 6. In addition, two different test cases are presented along with the results from the developed method and a comparison is made with the conforming solver version with respect to the achieved accuracy and computational cost.

2. Governing equations. Continuous entropy analysis

In this section we present the incompressible Navier–Stokes and the Cahn–Hilliard equations which will be denoted as the iNS/CH model. The adopted approach in this work is a phase field model, the Cahn–Hilliard equation [2,51]

$$c_t + \vec{\nabla}\cdot\left(c\vec{u}\right) = M_0\vec{\nabla}^2\mu, \tag{1}$$

which is an advection–diffusion equation that governs the evolution of the concentration. In the context of the iNS/CH system, the density $\rho(\vec{x},t)$ is directly computed from the concentration parameter $c(\vec{x},t)$ using a linear interpolation,

$$\rho = \rho(c) = \rho_1 c + \rho_2(1-c), \tag{2}$$

where $\rho_{1,2}$ are the densities of fluids 1 and 2, respectively, which are constant in space and time.

In (1), μ is the chemical potential,

$$\mu = \frac{\mathrm{d}f_0(c)}{\mathrm{d}c} - \frac{3}{2}\sigma\varepsilon\vec{\nabla}^2 c = f_0' - \frac{3}{2}\sigma\varepsilon\vec{\nabla}^2 c, \tag{3}$$

where $f_0(c)$ is the chemical free–energy,

$$f_0 = \frac{12\sigma}{\varepsilon}c^2(1-c)^2, \tag{4}$$

σ is the coefficient of interface tension between the fluids, ε is the interface width, and M_0 is the mobility, computed in this work with the chemical characteristic time, t_{CH}, as

$$M_0 = \frac{\varepsilon}{\sigma t_{\mathrm{CH}}}. \tag{5}$$

The parameters σ, ε, M_0, and t_{CH} are positive constants. The Cahn–Hilliard equation (1), with the chemical potential definition (3), has an associated free–energy

$$\mathcal{F}(c,\vec{\nabla}c) = f_0(c) + \frac{3}{4}\sigma\varepsilon|\vec{\nabla}c|^2. \tag{6}$$

The velocity field is calculated from the momentum equation of the Navier–Stokes equation. Following [52,53] we adopt the skew–symmetric form

$$\sqrt{\rho}\left(\sqrt{\rho}\vec{u}\right)_t + \vec{\nabla}\cdot\left(\frac{1}{2}\rho\vec{u}\vec{u}\right) + \frac{1}{2}\rho\vec{u}\cdot\vec{\nabla}\vec{u} + c\vec{\nabla}\mu = -\vec{\nabla}p + \vec{\nabla}\cdot\left(\eta\left(\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^T\right)\right) + \rho\vec{g}, \tag{7}$$

where p is an auxiliary pressure

$$p = p_s + \mathcal{F} - \mu c, \tag{8}$$

that includes the static pressure p_s and a body force approximation of the capillary pressure [10,54]. The viscosity η is computed from the concentration c and the viscocities of the fluids as

$$\eta = \eta(c) = \eta_1 c + \eta_2\,(1-c)\,. \tag{9}$$

The incompressibility constraint $\vec{\nabla}\cdot\vec{u} = 0$ is relaxed and enforced through the artificial compressibility method [55,56],

$$p_t + \rho_0 c_0^2\vec{\nabla}\cdot\vec{u} = 0, \tag{10}$$

where c_0 is the artificial sound speed, and $\rho_0 = \max(\rho_1,\rho_2)$.

Equations (1), (7) and (10) constitute the iNS/CH system of this work. It has to be noted that even in the continuous setting the skew–symmetric form (7) does not ensure conservation of momentum. However, since the capillary force term

is also non–conservative, momentum conservation could not be guaranteed with any other form of the Navier–Stokes momentum equation. Similar works on the Navier–Stokes/Cahn–Hilliard system with a skew-symmetric form are presented in [57,53].

The iNS/CH system is expressed as a general advection–diffusion equation,

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{\rho} & 0 & 0 & 0 \\ 0 & 0 & \sqrt{\rho} & 0 & 0 \\ 0 & 0 & 0 & \sqrt{\rho} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c \\ \sqrt{\rho}u \\ \sqrt{\rho}v \\ \sqrt{\rho}w \\ p \end{pmatrix}_t + \vec{\nabla}\cdot \begin{pmatrix} c\vec{u} \\ \frac{1}{2}\rho\vec{u}u + p\vec{e}_1 \\ \frac{1}{2}\rho\vec{u}v + p\vec{e}_2 \\ \frac{1}{2}\rho\vec{u}w + p\vec{e}_3 \\ 0 \end{pmatrix} + \begin{pmatrix} \vec{0} \\ \frac{1}{2}\rho\vec{u}\cdot\vec{\nabla}u + c\vec{e}_1\cdot\vec{\nabla}\mu \\ \frac{1}{2}\rho\vec{u}\cdot\vec{\nabla}v + c\vec{e}_2\cdot\vec{\nabla}\mu \\ \frac{1}{2}\rho\vec{u}\cdot\vec{\nabla}w + c\vec{e}_3\cdot\vec{\nabla}\mu \\ \rho_0 c_0^2\left(\vec{e}_1\cdot\vec{\nabla}u + \vec{e}_2\cdot\vec{\nabla}v + \vec{e}_3\cdot\vec{\nabla}w\right) \end{pmatrix} = \vec{\nabla}\cdot \begin{pmatrix} M_0\vec{\nabla}\mu \\ 2\eta\boldsymbol{S}\cdot\vec{e}_1 \\ 2\eta\boldsymbol{S}\cdot\vec{e}_2 \\ 2\eta\boldsymbol{S}\cdot\vec{e}_3 \\ \vec{0} \end{pmatrix} + \begin{pmatrix} 0 \\ \rho\vec{g}\cdot\vec{e}_1 \\ \rho\vec{g}\cdot\vec{e}_2 \\ \rho\vec{g}\cdot\vec{e}_3 \\ 0 \end{pmatrix}, \tag{11}$$

where,

$$\boldsymbol{S} = \mathrm{sym}\left(\vec{\nabla}\vec{u}\right) = \frac{1}{2}\left(\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^T\right), \tag{12}$$

is the strain tensor, and $\vec{e}_i$ are the space unit vectors. The velocity divergence of the artificial compressibility (10) is included into the non–conservative terms since it makes it easier to show stability.

We follow the notation in [58,9] to work with vectors of different nature. Space vectors (e.g. $\vec{x} = (x, y, z) \in \mathbb{R}^3$) are represented with an arrow, and state vectors (e.g. $\mathbf{q} = \left(c, \sqrt{\rho}\vec{u}, p\right) \in \mathbb{R}^5$) in bold. Moreover, we define block vectors as the result of stacking three state vectors (e.g. fluxes),

$$\overleftrightarrow{\mathbf{f}}_e = \begin{pmatrix} \mathbf{f}_{e,1} \\ \mathbf{f}_{e,2} \\ \mathbf{f}_{e,3} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_e \\ \mathbf{g}_e \\ \mathbf{h}_e \end{pmatrix}, \quad \overleftrightarrow{\mathbf{f}}_v = \begin{pmatrix} \mathbf{f}_{v,1} \\ \mathbf{f}_{v,2} \\ \mathbf{f}_{v,3} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_v \\ \mathbf{g}_v \\ \mathbf{h}_v \end{pmatrix}, \tag{13}$$

and define the operator Υ that transforms a 5×3 (state–space) matrix into a 15×1 block vector,

$$\overleftrightarrow{\mathbf{f}} = \Upsilon \begin{pmatrix} f_1 & g_1 & h_1 \\ f_2 & g_2 & h_2 \\ f_3 & g_3 & h_3 \\ f_4 & g_4 & h_4 \\ f_5 & g_5 & h_5 \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \\ \mathbf{h} \end{pmatrix}. \tag{14}$$

The products of state, space and block vectors are defined as

$$\overleftrightarrow{\mathbf{f}}\cdot\overleftrightarrow{\mathbf{g}} = \sum_{i=1}^{3}\mathbf{f}_i^T\mathbf{g}_i, \quad \vec{g}\cdot\overleftrightarrow{\mathbf{f}} = \sum_{i=1}^{3} g_i\mathbf{f}_i, \quad \vec{g}\mathbf{f} = \begin{pmatrix} g_1\mathbf{f} \\ g_2\mathbf{f} \\ g_3\mathbf{f} \end{pmatrix}. \tag{15}$$

The divergence and gradient operators, using (15), take the form

$$\vec{\nabla}\cdot\overleftrightarrow{\mathbf{f}} = \sum_{i=1}^{3}\frac{\partial\mathbf{f}_i}{\partial x_i}, \quad \vec{\nabla}\mathbf{q} = \begin{pmatrix} \mathbf{q}_x \\ \mathbf{q}_y \\ \mathbf{q}_z \end{pmatrix}. \tag{16}$$

State matrices (i.e. 5×5 matrices) are denoted with an underline, e.g. $\underline{\mathbf{B}}$. State matrices can be combined to construct a block matrix,

$$\mathcal{B} = \begin{pmatrix} \underline{\mathbf{B}}_{11} & \underline{\mathbf{B}}_{12} & \underline{\mathbf{B}}_{13} \\ \underline{\mathbf{B}}_{21} & \underline{\mathbf{B}}_{22} & \underline{\mathbf{B}}_{23} \\ \underline{\mathbf{B}}_{31} & \underline{\mathbf{B}}_{32} & \underline{\mathbf{B}}_{33} \end{pmatrix}. \tag{17}$$

Block matrices can be multiplied with a block vector to obtain a new block vector. A matrix multiplication in space (e.g. a rotation) can be performed in the following way

$$\vec{g} = \boldsymbol{M}\overleftrightarrow{\mathbf{f}}. \tag{18}$$

For each of the variables in the state vector, we construct the block matrix version of $\boldsymbol{M}$ ($\underline{\mathbf{I}}_5$ is the 5×5 identity matrix),

$$\mathcal{M} = \begin{pmatrix} \boldsymbol{M}_{11}\underline{\mathbf{I}}_5 & \boldsymbol{M}_{12}\underline{\mathbf{I}}_5 & \boldsymbol{M}_{13}\underline{\mathbf{I}}_5 \\ \boldsymbol{M}_{21}\underline{\mathbf{I}}_5 & \boldsymbol{M}_{22}\underline{\mathbf{I}}_5 & \boldsymbol{M}_{23}\underline{\mathbf{I}}_5 \\ \boldsymbol{M}_{31}\underline{\mathbf{I}}_5 & \boldsymbol{M}_{32}\underline{\mathbf{I}}_5 & \boldsymbol{M}_{33}\underline{\mathbf{I}}_5 \end{pmatrix}, \tag{19}$$

so that we can compactly write

$$\overleftrightarrow{\mathbf{g}} = \mathcal{M}\overleftrightarrow{\mathbf{f}}. \tag{20}$$

More details are given in [58].

Using the notation introduced above, the iNS/CH system (11) can be written in the form of a general advection–diffusion equation,

$$\underline{\mathbf{m}}\mathbf{q}_t + \vec{\nabla}\cdot\overleftrightarrow{\mathbf{f}}_e(\mathbf{q}) + \sum_{m=1}^{5}\overleftrightarrow{\boldsymbol{\phi}}_m(\mathbf{q})\cdot\vec{\nabla}w_m = \vec{\nabla}\cdot\overleftrightarrow{\mathbf{f}}_v\left(\mathbf{q},\vec{\nabla}\mathbf{w}\right) + \mathbf{s}(\mathbf{q}), \tag{21}$$

with state vector $\mathbf{q} = \left(c, \sqrt{\rho}\vec{u}, p\right)$, gradient variables vector $\mathbf{w} = (w_1, w_2, ..., w_5) = \left(\mu, \vec{u}, p\right)$, mass matrix $\underline{\mathbf{m}}$,

$$\underline{\mathbf{m}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sqrt{\rho}\boldsymbol{I}_3 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{22}$$

inviscid fluxes $\overleftrightarrow{\mathbf{f}}_e(\mathbf{q})$,

$$\mathbf{f}_{e,1} = \mathbf{f}_e = \begin{pmatrix} cu \\ \frac{1}{2}\rho u^2 + p \\ \frac{1}{2}\rho uv \\ \frac{1}{2}\rho uw \\ 0 \end{pmatrix}, \quad \mathbf{f}_{e,2} = \mathbf{g}_e = \begin{pmatrix} cv \\ \frac{1}{2}\rho uv \\ \frac{1}{2}\rho v^2 + p \\ \frac{1}{2}\rho vw \\ 0 \end{pmatrix}, \quad \mathbf{f}_{e,3} = \mathbf{h}_e = \begin{pmatrix} cw \\ \frac{1}{2}\rho uw \\ \frac{1}{2}\rho vw \\ \frac{1}{2}\rho w^2 + p \\ 0 \end{pmatrix}, \tag{23}$$

non–conservative term coefficients $\overleftrightarrow{\boldsymbol{\phi}}_m(\mathbf{q})$,

$$\begin{aligned} \overleftrightarrow{\boldsymbol{\phi}}_1 &= \Upsilon\begin{pmatrix} 0 \\ c\vec{e}_1 \\ c\vec{e}_2 \\ c\vec{e}_3 \\ 0 \end{pmatrix}, \quad \overleftrightarrow{\boldsymbol{\phi}}_2 = \Upsilon\begin{pmatrix} 0 \\ \frac{1}{2}\rho\vec{u} \\ 0 \\ 0 \\ \rho_0 c_0^2\vec{e}_1 \end{pmatrix}, \quad \overleftrightarrow{\boldsymbol{\phi}}_3 = \Upsilon\begin{pmatrix} 0 \\ 0 \\ \frac{1}{2}\rho\vec{u} \\ 0 \\ \rho_0 c_0^2\vec{e}_2 \end{pmatrix}, \quad \overleftrightarrow{\boldsymbol{\phi}}_4 = \Upsilon\begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{2}\rho\vec{u} \\ \rho_0 c_0^2\vec{e}_3 \end{pmatrix}, \\ \overleftrightarrow{\boldsymbol{\phi}}_5 &= \Upsilon\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \end{aligned} \tag{24}$$

viscous fluxes $\overleftrightarrow{\mathbf{f}}_v\left(\vec{\nabla}\mathbf{w}\right)$,

$$\mathbf{f}_{v,1} = \mathbf{f}_v = \begin{pmatrix} M_0\mu_x \\ 2\eta\boldsymbol{S}_{11} \\ 2\eta\boldsymbol{S}_{21} \\ 2\eta\boldsymbol{S}_{31} \\ 0 \end{pmatrix}, \quad \mathbf{f}_{v,2} = \mathbf{g}_v = \begin{pmatrix} M_0\mu_y \\ 2\eta\boldsymbol{S}_{12} \\ 2\eta\boldsymbol{S}_{22} \\ 2\eta\boldsymbol{S}_{32} \\ 0 \end{pmatrix}, \quad \mathbf{f}_{v,3} = \mathbf{h}_v = \begin{pmatrix} M_0\mu_z \\ 2\eta\boldsymbol{S}_{13} \\ 2\eta\boldsymbol{S}_{23} \\ 2\eta\boldsymbol{S}_{33} \\ 0 \end{pmatrix}, \tag{25}$$

and source term $\mathbf{s}(\mathbf{q}) = \left(0, \rho\vec{g}, 0\right)$.

It has to be noted that the same gradient variables $\mathbf{w}$ are used in both the non–conservative terms and the viscous fluxes. Furthermore, in Section 2.1 it is proven that they are also the entropy variables related to the mathematical entropy.

2.1. Entropy analysis of the iNS/CH system

The entropy analysis for the continuous system is presented in [10]. It has been shown in [10] that there exists a pair of mathematical entropy function $\mathcal{E}(\mathbf{q})$ (mathematical entropy) and $\mathbf{w}$ (entropy variables) that contract the system of equations (21) into a conservation law [59],

$$\mathbf{w}^T\left(\underline{\mathbf{m}}\mathbf{q}_t + \vec{\nabla}\cdot\overleftrightarrow{\mathbf{f}}_e(\mathbf{q}) + \sum_{m=1}^{5}\overleftrightarrow{\boldsymbol{\phi}}_m(\mathbf{q})\cdot\vec{\nabla}w_m - \vec{\nabla}\cdot\overleftrightarrow{\mathbf{f}}_v\left(\mathbf{q},\vec{\nabla}\mathbf{w}\right)\right) = \mathcal{E}_t + \nabla\cdot\vec{f}^{\mathcal{E}} + \nabla\mathbf{w}^T\cdot\overleftrightarrow{\mathbf{f}}_v = 0. \tag{26}$$

The contraction of the inviscid fluxes and the non–conservative terms is

$$\mathbf{w}^T \left(\vec{\nabla} \cdot \overleftrightarrow{\mathbf{f}}_e (\mathbf{q}) + \sum_{m=1}^{5} \overleftrightarrow{\boldsymbol{\phi}}_m (\mathbf{q}) \cdot \vec{\nabla} w_m \right) = \vec{\nabla} \cdot \vec{f}_e^{\mathcal{E}}. \tag{27}$$

Then it is proven that the viscous fluxes are dissipative,

$$\vec{\nabla}\mathbf{w}^T \cdot \overleftrightarrow{\mathbf{f}}_v = M_0 |\vec{\nabla}\mu|^2 + 2\eta \boldsymbol{S} : \boldsymbol{S} \geqslant 0. \tag{28}$$

The contraction of the inviscid fluxes (27) from the entropy variables happens automatically, if the conservative and non–conservative fluxes satisfy

$$\mathbf{e}_m^T \overleftrightarrow{\mathbf{f}}_e = \mathbf{w}^T \overleftrightarrow{\boldsymbol{\phi}}_m, \tag{29}$$

by construction [10], with $\mathbf{e}_m^T$ being a state unit vector along the m-th state variable. Its associated entropy flux is

$$\vec{f}_e^{\mathcal{E}} = \mathbf{w}^T \overleftrightarrow{\mathbf{f}}_e. \tag{30}$$

For the iNS/CH system, the mathematical entropy is,

$$\mathcal{E} = \mathcal{F}(c, \vec{\nabla} c) + \mathcal{K}\left(\sqrt{\rho}\vec{u}\right) + \mathcal{E}_{\text{AC}}(p) = f_0(c) + \frac{3}{4}\sigma\varepsilon|\vec{\nabla} c|^2 + \frac{1}{2}\rho v_{tot}^2 + \frac{p^2}{2\rho_0 c_0^2} \geqslant 0, \tag{31}$$

where $\mathcal{F}$ is the Cahn–Hilliard free–energy, $\mathcal{K}$ is the traditional kinetic energy and $\mathcal{E}_{\text{AC}}$ is the artificial compressibility energy term that was introduced in [9]. In addition, $v_{tot}^2 = u^2 + v^2 + w^2$ is the square of the total speed.

The density ρ in (31) should be positive, which is ensured by imposing a limit in the maximum and minimum density values as

$$\rho\left(\hat{c}\right) = \rho_1 \hat{c} + \rho_2 \left(1 - \hat{c}\right), \quad \hat{c} = \min\left(\max\left(c, 0\right), 1\right). \tag{32}$$

The clipping function is used only to compute the density, which is used only as an input in the momentum equation as in [52]. The concentration parameter of the Cahn–Hilliard equation is not clipped and is left to contain values outside the $c \in [0, 1]$ interval. Thus, the proposed methodology is phase–conservative and the free–energy is not altered.

In [10], it is proven that the entropy variables $\mathbf{w}$

$$\mathbf{w} = \left(\mu, u, v, w, \frac{p}{\rho_0 c_0^2}\right), \tag{33}$$

contract the time derivative of the state vector, $\mathbf{q}_t$, into the time derivative of the entropy $\mathcal{E}_t$, plus an additional divergence term of a time derivative flux $\vec{f}_t^{\mathcal{E}}$,

$$\vec{f}_t^{\mathcal{E}} = -\frac{3}{2}\sigma\varepsilon c_t \vec{\nabla} c, \tag{34}$$

such that

$$\mathbf{w}^T \underline{\mathbf{m}} \mathbf{q}_t = \mathcal{E}_t + \vec{\nabla} \cdot \vec{f}_t^{\mathcal{E}}. \tag{35}$$

With these results, we confirm that the iNS/CH system, (11), with entropy, (31), satisfies the conservation law

$$\mathcal{E}_t + \vec{\nabla} \cdot \vec{f}^{\mathcal{E}} = -\vec{\nabla}\mathbf{w}^T \cdot \overleftrightarrow{\mathbf{f}}_v, \tag{36}$$

with entropy flux $\vec{f}^{\mathcal{E}}$,

$$\vec{f}^{\mathcal{E}} = \vec{f}_t^{\mathcal{E}} + \vec{f}_e^{\mathcal{E}} + \vec{f}_v^{\mathcal{E}}. \tag{37}$$

The entropy equation can be written as a global equation by integrating over the domain Ω,

$$\frac{\mathrm{d}\bar{\mathcal{E}}}{\mathrm{d}t} + \int_{\partial\Omega} \vec{f}^{\mathcal{E}} \cdot \vec{n}\, \mathrm{d}S = -\int_{\Omega} \left(M_0 |\vec{\nabla}\mu|^2 + 2\eta \boldsymbol{S} : \boldsymbol{S} \right) \mathrm{d}\vec{x} \leqslant 0, \tag{38}$$

where $\bar{\mathcal{E}}$ is the total entropy,

$$\bar{\mathcal{E}} = \int_{\Omega} \mathcal{E}\, \mathrm{d}\vec{x}. \tag{39}$$

Eq. (38) shows that the total entropy (39) is always dissipated in the interior of the domain, and it can only increase due to boundary exchanges [10].

The discrete approximation will be constructed so that it mimics (38), which will guarantee that the discrete entropy of the approximation will remain bounded by the boundary and initial data.

2.1.1. Boundary conditions

In order for the entropy to remain bounded, it has to be combined with appropriate boundary conditions. Two different types are considered in this work, as we examine the effect of free– and no–slip boundary conditions. Specifically for the Cahn–Hilliard equation, a non–homogeneous Neumann condition for the concentration [60], and a homogeneous Neumann condition for the chemical potential are considered [10]

$$-\frac{3}{2}\sigma\varepsilon\frac{\partial c}{\partial\vec{n}}\bigg|_{\partial\Omega}=f_w'(c),\quad \frac{\partial\mu}{\partial\vec{n}}\bigg|_{\partial\Omega}=0, \tag{40}$$

where $f_w(c)$ is the boundary free–energy function that controls the wall contact angle. One choice is to use the function [60]

$$f_w(c)=\frac{1}{2}\sigma\cos(\theta_w)(2c-1)(1+2c-2c^2),\quad f_w'(c)=6\sigma\cos(\theta_w)c(1-c), \tag{41}$$

where θ_w is the imposed contact angle with the wall. In most simulations performed in this work we use a 90° angle, which simplifies $f_w(c)=0$.

The entropy balance (38) with both wall boundary conditions takes the same form and is defined as

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\bar{\mathcal{E}}+\int_{\partial\Omega}f_w(c)\,\mathrm{d}S\right)=-\int_{\Omega}\left(M_0|\vec{\nabla}\mu|^2+2\eta\boldsymbol{S}:\boldsymbol{S}\right)\mathrm{d}\vec{x}\leqslant 0, \tag{42}$$

where the volume entropy is augmented with the surface free–energy as in [60,7].

3. Space and time discretization

In this section we provide the details for the construction of an entropy–stable DGSEM approximation. We focus on a tensor product DGSEM with Gauss–Lobatto (GL) points as it satisfies the Summation–By–Parts Simultaneous–Approximation–Term (SBP–SAT) property [61]. This choice allows us to mimic the continuous stability of the system and devise a discrete entropy law.

3.1. Differential geometry and curvilinear elements

The computational domain Ω is tessellated into non–overlapping hexahedral elements, which are then geometrically transformed from a reference element $e=[-1,1]^3$ by means of a transfinite mapping. This mapping relates the physical $(\vec{x}=(x^1,x^2,x^3)=(x,y,z))$ and the local $(\vec{\xi}=(\xi^1,\xi^2,\xi^3)=(\xi,\eta,\zeta))$ coordinates,

$$\vec{x}=\vec{X}(\vec{\xi})=\vec{X}(\xi,\eta,\zeta). \tag{43}$$

From the transformation (43) three covariant basis vectors can be defined,

$$\vec{a}_i=\frac{\partial\vec{X}}{\partial\xi^i},\quad i=1,2,3, \tag{44}$$

and three contravariant basis vectors,

$$\vec{a}^i=\vec{\nabla}\xi^i=\frac{1}{J}\left(\vec{a}_j\times\vec{a}_k\right),\quad (i,j,k)\ \text{cyclic}, \tag{45}$$

where

$$J=\vec{a}_1\cdot\left(\vec{a}_2\times\vec{a}_3\right) \tag{46}$$

is the Jacobian of the mapping $\vec{X}$. The contravariant coordinate vectors satisfy the metric identities [62],

$$\sum_{i=1}^{3}\frac{\partial\left(Ja_n^i\right)}{\partial\xi^i}=0,\quad n=1,2,3, \tag{47}$$

where a_n^i is the n–th Cartesian component of the contravariant vector $\vec{a}^i$.

The volume weighted contravariant basis $J\vec{a}^i$ is used to transform differential operators from physical $(\vec{\nabla})$ to reference $(\vec{\nabla}_\xi)$ space. The divergence of a vector is [58]

$$\vec{\nabla}\cdot\vec{f}=\frac{1}{J}\vec{\nabla}_\xi\cdot\left(\boldsymbol{M}^T\vec{f}\right), \tag{48}$$

where $\boldsymbol{M} = \left(J\vec{a}^{\xi}, J\vec{a}^{\eta}, J\vec{a}^{\zeta}\right)$. We use (19) to write the divergence of an entire block vector compactly. Thus, the block matrix $\mathcal{M}$ is expressed as,

$$\mathcal{M} = \begin{pmatrix} Ja_1^1\underline{\mathbf{I}}_5 & Ja_1^2\underline{\mathbf{I}}_5 & Ja_1^3\underline{\mathbf{I}}_5 \\ Ja_2^1\underline{\mathbf{I}}_5 & Ja_2^2\underline{\mathbf{I}}_5 & Ja_2^3\underline{\mathbf{I}}_5 \\ Ja_3^1\underline{\mathbf{I}}_5 & Ja_3^2\underline{\mathbf{I}}_5 & Ja_3^3\underline{\mathbf{I}}_5 \end{pmatrix}, \tag{49}$$

which allows us to write (48) for all the state variables,

$$\vec{\nabla}\cdot\overleftrightarrow{\mathbf{f}} = \frac{1}{J}\vec{\nabla}_{\xi}\cdot\left(\mathcal{M}^T\overleftrightarrow{\mathbf{f}}\right) = \frac{1}{J}\vec{\nabla}_{\xi}\cdot\overleftrightarrow{\tilde{\mathbf{f}}}, \tag{50}$$

with $\overleftrightarrow{\tilde{\mathbf{f}}}$ being the block vector of the contravariant fluxes,

$$\overleftrightarrow{\tilde{\mathbf{f}}} = \mathcal{M}^T\overleftrightarrow{\mathbf{f}}, \quad \tilde{\mathbf{f}}^i = J\vec{a}^i\cdot\overleftrightarrow{\mathbf{f}}. \tag{51}$$

The gradient of a scalar is [58]

$$\vec{\nabla}w = \frac{1}{J}\boldsymbol{M}\vec{\nabla}_{\xi}w, \tag{52}$$

which we can also extend to all entropy variables using (49),

$$\overleftrightarrow{\mathbf{g}} = \vec{\nabla}\mathbf{w} = \frac{1}{J}\mathcal{M}\vec{\nabla}_{\xi}\mathbf{w}, \tag{53}$$

and to non–conservative terms,

$$\overleftrightarrow{\boldsymbol{\phi}}_m\cdot\vec{\nabla}w_m = \frac{1}{\mathcal{J}}\overleftrightarrow{\tilde{\boldsymbol{\phi}}}_m\cdot\vec{\nabla}_{\xi}w_m. \tag{54}$$

The iNS/CH equation, (11), is transformed into a system of four first order equations with the use of the auxiliary variables $\overleftrightarrow{\mathbf{g}} = \vec{\nabla}\mathbf{w}$ and $\vec{g}_c = \vec{\nabla}c$ such that

$$\begin{aligned} &\underline{\mathbf{m}}\mathbf{q}_t + \vec{\nabla}\cdot\overleftrightarrow{\mathbf{f}}_e(\mathbf{q}) + \sum_{m=1}^{5}\left(\overleftrightarrow{\boldsymbol{\phi}}_m(\mathbf{q})\cdot\vec{\nabla}w_m\right) = \vec{\nabla}\cdot\overleftrightarrow{\mathbf{f}}_v\left(\overleftrightarrow{\mathbf{g}}\right) + \mathbf{s}(\mathbf{q}), \\ &\overleftrightarrow{\mathbf{g}} = \vec{\nabla}\mathbf{w}, \\ &\mu = f_0'(c) - \frac{3}{2}\sigma\varepsilon\vec{\nabla}\cdot\vec{g}_c, \\ &\vec{g}_c = \vec{\nabla}c. \end{aligned} \tag{55}$$

The transformation of the operators to the reference space is accomplished using the mapped derivative operators (50), (53), and (54) properties,

$$J\underline{\mathbf{m}}\mathbf{q}_t + \vec{\nabla}_{\xi}\cdot\overleftrightarrow{\tilde{\mathbf{f}}}_e(\mathbf{q}) + \sum_{m=1}^{5}\left(\overleftrightarrow{\tilde{\boldsymbol{\phi}}}_m(\mathbf{q})\cdot\vec{\nabla}_{\xi}w_m\right) = \vec{\nabla}_{\xi}\cdot\overleftrightarrow{\tilde{\mathbf{f}}}_v\left(\overleftrightarrow{\mathbf{g}}\right) + J\mathbf{s}(\mathbf{q}), \tag{56a}$$

$$J\overleftrightarrow{\mathbf{g}} = \mathcal{M}\vec{\nabla}_{\xi}\mathbf{w}, \tag{56b}$$

$$J\mu = Jf_0'(c) - \frac{3}{2}\sigma\varepsilon\vec{\nabla}_{\xi}\cdot\vec{\tilde{g}}_c, \tag{56c}$$

$$J\vec{g}_c = \boldsymbol{M}\vec{\nabla}_{\xi}c. \tag{56d}$$

The DG approximation is obtained from weak forms of (56). The inner products in the reference element, E, for state and block vectors are

$$\langle\mathbf{f},\mathbf{g}\rangle_E = \int_E \mathbf{f}^T\mathbf{g}\,\mathrm{d}E, \quad \left\langle\overleftrightarrow{\mathbf{f}},\overleftrightarrow{\mathbf{g}}\right\rangle_E = \int_E \overleftrightarrow{\mathbf{f}}\cdot\overleftrightarrow{\mathbf{g}}\,\mathrm{d}E. \tag{57}$$

To construct the weak form of (56) we multiply by four test functions $\boldsymbol{\varphi}_q$, $\overleftrightarrow{\boldsymbol{\varphi}}_g$, φ_{μ}, and $\vec{\varphi}_c$ and integrate over the reference element. Using the integration–by–parts property, the system takes the form

$$
\begin{aligned}
&\left\langle J\underline{\mathbf{m}}\mathbf{q}_t, \boldsymbol{\varphi}_q\right\rangle_E + \int_{\partial E} \boldsymbol{\varphi}_q^T \left(\overleftrightarrow{\tilde{\mathbf{f}}}_e + \sum_{m=1}^{5} \overleftrightarrow{\tilde{\boldsymbol{\phi}}}_m w_m - \overleftrightarrow{\tilde{\mathbf{f}}}_v\right)\cdot \hat{n}\,\mathrm{d}S_\xi - \left\langle \overleftrightarrow{\tilde{\mathbf{f}}}_e, \vec{\nabla}_\xi \boldsymbol{\varphi}_q\right\rangle_E \\
&\qquad - \sum_{m=1}^{5}\left\langle w_m, \vec{\nabla}_\xi \cdot \left(\boldsymbol{\varphi}_q^T \overleftrightarrow{\tilde{\boldsymbol{\phi}}}_m\right)\right\rangle_E = -\left\langle \overleftrightarrow{\tilde{\mathbf{f}}}_v, \vec{\nabla}_\xi \boldsymbol{\varphi}_q\right\rangle_E + \left\langle J\mathbf{s}, \boldsymbol{\varphi}_q\right\rangle_E, \\
&\left\langle J\overleftrightarrow{\mathbf{g}}, \overleftrightarrow{\boldsymbol{\varphi}}_g\right\rangle_E = \int_{\partial E} \mathbf{w}^T \left(\overleftrightarrow{\tilde{\boldsymbol{\varphi}}}_g \cdot \hat{n}\right)\mathrm{d}S_\xi - \left\langle \mathbf{w}, \vec{\nabla}_\xi \cdot \overleftrightarrow{\tilde{\boldsymbol{\varphi}}}_g\right\rangle_E, \\
&\left\langle J\mu, \varphi_\mu\right\rangle_E = \left\langle Jf_0', \varphi_\mu\right\rangle_E - \int_{\partial E} \frac{3}{2}\sigma\varepsilon\varphi_\mu \vec{\tilde{\mathbf{g}}}_c \cdot \hat{n}\,\mathrm{d}S_\xi + \left\langle \frac{3}{2}\sigma\varepsilon \vec{\tilde{\mathbf{g}}}_c, \vec{\nabla}_\xi \varphi_\mu\right\rangle_E, \\
&\left\langle J\vec{\mathbf{g}}_c, \vec{\varphi}_c\right\rangle_E = \int_{\partial E} c\vec{\tilde{\varphi}}_c \cdot \hat{n}\,\mathrm{d}S_\xi - \left\langle c, \vec{\nabla}_\xi \cdot \vec{\tilde{\varphi}}_c\right\rangle_E,
\end{aligned}
\tag{58}
$$

where $\hat{n}$ is the unit outward vector and $\mathrm{d}S_\xi$ the surface differential of each face of the reference element E. The surface integrals extend to all six faces of an element,

$$
\int_{\partial E} \vec{\tilde{f}}\cdot\hat{n}\,\mathrm{d}S_\xi = \int_{[-1,1]^2} \tilde{f}^1\,\mathrm{d}\eta\,\mathrm{d}\zeta\Big|_{\xi=-1}^{\xi=1} + \int_{[-1,1]^2} \tilde{f}^2\,\mathrm{d}\xi\,\mathrm{d}\zeta\Big|_{\eta=-1}^{\eta=1} + \int_{[-1,1]^2} \tilde{f}^3\,\mathrm{d}\xi\,\mathrm{d}\eta\Big|_{\zeta=-1}^{\zeta=1}. \tag{59}
$$

The relation connecting the physical and reference surface differentials is given by,

$$
\mathrm{d}S^i = \left|J\vec{a}^i\right|\mathrm{d}\xi^j\,\mathrm{d}\xi^k = \mathcal{J}_f^i\,\mathrm{d}S_\xi^i, \tag{60}
$$

where we have defined the face Jacobian $\mathcal{J}_f^i = \left|\mathcal{J}\vec{a}^i\right|$. We can write the surface flux in either reference element, $\vec{\tilde{f}}\cdot\hat{n}$, or physical space, $\vec{f}\cdot\vec{n}$, through the following relations

$$
\vec{\tilde{f}}\cdot\hat{n}^i\,\mathrm{d}S_\xi = \left(\boldsymbol{\mathcal{M}}^T\overleftrightarrow{\mathbf{f}}\right)\cdot\hat{n}^i\,\mathrm{d}S_\xi = \vec{f}\cdot\left(\boldsymbol{\mathcal{M}}\hat{n}^i\right)\mathrm{d}S_\xi = \vec{f}\cdot\vec{n}\left|J\vec{a}^i\right|\mathrm{d}S_\xi = \vec{f}\cdot\vec{n}^i\,\mathrm{d}S. \tag{61}
$$

Therefore, the surface integrals can be written in both physical and reference spaces,

$$
\int_{\partial E} \vec{\tilde{f}}\cdot\hat{\mathbf{n}}\,\mathrm{d}S_\xi = \int_{\partial e} \vec{f}\cdot\vec{n}\,\mathrm{d}S. \tag{62}
$$

3.2. Polynomial approximation and the DGSEM

In this part of the paper we introduce the discrete version of (58). The solutions and functions are approximated by order N polynomials in each element E,

$$
\mathbf{q} \approx \mathbf{Q}\left(\vec{\xi}\right) = \sum_{i,j,k=0}^{N} \mathbf{Q}_{ijk}(t) l_i(\xi) l_j(\eta) l_k(\zeta), \tag{63}
$$

where the approximation order N can vary from element to element. In (63), $l_i(\xi)$ are the Lagrange interpolating polynomials whose nodes are a set of Gauss–Lobatto points $\{\xi_i\}_{i=0}^N$ in the reference element E and $\mathbf{Q}_{ijk}(t) = \mathbf{Q}(\xi_i, \eta_j, \zeta_k, t)$ are the (time dependent) nodal values of an arbitrary function $\mathbf{Q}$. We use $\mathcal{I}^N$ to denote the polynomial interpolation operator [63]. The notation is as follows: we use lower cases for the exact functions, whereas upper cases represent their polynomial approximation.

In the discrete setting, we must ensure that the approximation of the metrics is freestream preserving. Having a *watertight* mesh is a necessary condition but not sufficient. For freestream preservation, the mesh has to satisfy two additional conditions [64],

1. **Condition (F):** the projection of the discrete volume weighted contravariant bases at the faces has to be also continuous $\mathcal{J}\vec{a}^i\big|_{face}^{e_L} = \mathcal{J}\vec{a}^i\big|_{face}^{e_R}$. The symbol $\mathcal{J}$ represents the polynomial approximation of the Jacobian.
2. **Condition (V):** the approximation of the contravariant bases has to satisfy a discrete version of the metric identities (47).

The mapping function is approximated with the interpolation operator (63). However, although the mapping is represented by order N polynomials, the genuine order of the mapping is given by the approximation order of the faces and the edges. We follow the rules for the construction of a watertight mesh of [8,65]. We highlight the construction of a *watertight* mesh that satisfies the two conditions:

- Edges: the approximation order of the edges in the mesh has to be unique, and the face functions that share an edge must reduce to the same curvilinear function at the edge. The order of an edge shared by various faces has to be $N_{\text{edge}} \leq \min\left(N_{f_1}, N_{f_2}, \ldots\right)$ for general three-dimensional non–conforming elements, two dimensional, two dimensional extruded and conforming problems, at most.
- Faces: the approximation order of the faces in the mesh has to be unique and it must be set to $N_f \leq \min\left(N_{e_L}, N_{e_R}\right)$ for two–dimensional, two–dimensional extruded, and conforming problems and to $N_f \leq \min\left(N_{e_L}, N_{e_R}\right)/2$ for general three–dimensional non–conforming elements, at most.
- Volume: the contravariant basis have to be computed in a curl form [66]

$$\mathcal{J}a_n^i = -\hat{\vec{x}}^i \cdot \nabla_\xi \times \mathcal{I}^N\left(\mathcal{X}_l \nabla_\xi \mathcal{X}_m\right), \quad i,n = 1,2,3, \quad (n,m,l) \text{ cyclic}. \tag{64}$$

Next, the integrals of the weak formulation are approximated using Gauss quadratures,

$$\int_{-1}^{1} F \,\mathrm{d}\xi \approx \int_N F \,\mathrm{d}\xi = \sum_{m=0}^{N} w_m F_m, \tag{65}$$

where $\{w_i\}_{i=0}^N$ are the quadrature weights that correspond to the Gauss–Lobatto nodes $\{\xi_i\}_{i=0}^N$. For Gauss–Lobatto points, the approximation is exact if for an $f(\xi)$ whose order is at most $2N-1$. The approximation of surface integrals is performed similarly, replacing the exact function with the polynomial approximation and the exact integrals by Gauss quadratures in (59),

$$\begin{aligned}\int_{\partial E} \vec{\tilde{f}} \cdot \hat{n} \,\mathrm{d}S_\xi \approx \int_{\partial E,N} \vec{\tilde{F}} \cdot \hat{n} \,\mathrm{d}S_\xi &= \int_N \tilde{F}^1 \,\mathrm{d}\eta \,\mathrm{d}\zeta \Big|_{\xi=-1}^{\xi=1} + \int_N \tilde{F}^2 \,\mathrm{d}\xi \,\mathrm{d}\zeta \Big|_{\eta=-1}^{\eta=1} + \int_N \tilde{F}^3 \,\mathrm{d}\xi \,\mathrm{d}\eta \Big|_{\zeta=-1}^{\zeta=1} \\ &= \sum_{j,k=0}^{N} w_{jk}\left(\tilde{F}^1_{Njk} - \tilde{F}^1_{0jk}\right) + \sum_{i,k=0}^{N} w_{ik}\left(\tilde{F}^2_{iNk} - \tilde{F}^2_{i0k}\right) + \sum_{i,j=0}^{N} w_{ij}\left(\tilde{F}^3_{ijN} - \tilde{F}^3_{ij0}\right).\end{aligned} \tag{66}$$

The simultaneous–approximation–term (SAT) property [67] and the exactness of the numerical quadrature and yield the discrete Gauss law [61,68]: for any polynomials $\vec{\tilde{\mathbf{F}}}$ and $\mathbf{V}$ in $\mathbb{P}^N$,

$$\left\langle \vec{\nabla}_\xi \cdot \vec{\tilde{\mathbf{F}}}, \mathbf{V} \right\rangle_{E,N} = \int_{\partial E,N} \left(\vec{\tilde{\mathbf{F}}} \cdot \hat{n}\right) \mathbf{V} \,\mathrm{d}S_\xi - \left\langle \vec{\tilde{\mathbf{F}}}, \vec{\nabla}_\xi \mathbf{V} \right\rangle_{E,N}, \tag{67}$$

and thus they are used to construct entropy–stable schemes using split–forms [61].

We use the properties (63), (65), and (66) to discretize the continuous weak forms (58),

$$\begin{aligned}&\left\langle \mathcal{J}\underline{\mathbf{M}}\mathbf{Q}_t, \boldsymbol{\varphi}_q \right\rangle_{E,N} + \int_{\partial E,N} \boldsymbol{\varphi}_q^T \left(\vec{\tilde{\mathbf{F}}}_e + \sum_{m=1}^{5} \vec{\tilde{\boldsymbol{\Phi}}}_m W_m - \vec{\tilde{\mathbf{F}}}_v\right) \cdot \hat{n} \,\mathrm{d}S_\xi - \left\langle \vec{\tilde{\mathbf{F}}}_e, \vec{\nabla}_\xi \boldsymbol{\varphi}_q \right\rangle_{E,N} \\ &\qquad - \sum_{m=1}^{5} \left\langle W_m, \vec{\nabla}_\xi \cdot \left(\boldsymbol{\varphi}_q^T \vec{\tilde{\boldsymbol{\Phi}}}_m\right) \right\rangle_{E,N} = -\left\langle \vec{\tilde{\mathbf{F}}}_v, \vec{\nabla}_\xi \boldsymbol{\varphi}_q \right\rangle_{E,N} + \left\langle \mathcal{J}\mathbf{S}, \boldsymbol{\varphi}_q \right\rangle_{E,N}, \\ &\left\langle \mathcal{J}\vec{\mathbf{G}}, \vec{\boldsymbol{\varphi}}_g \right\rangle_{E,N} = \int_{\partial E,N} \mathbf{W}^T \left(\vec{\tilde{\boldsymbol{\varphi}}}_g \cdot \hat{n}\right) \mathrm{d}S_\xi - \left\langle \mathbf{W}, \vec{\nabla}_\xi \cdot \vec{\tilde{\boldsymbol{\varphi}}}_g \right\rangle_{E,N}, \\ &\left\langle \mathcal{J}\mu, \varphi_\mu \right\rangle_{E,N} = \left\langle \mathcal{J}F_0', \varphi_\mu \right\rangle_{E,N} - \int_{\partial E,N} \frac{3}{2}\sigma\varepsilon\varphi_\mu \vec{\tilde{G}}_c \cdot \hat{n} \,\mathrm{d}S_\xi + \left\langle \frac{3}{2}\sigma\varepsilon \vec{\tilde{G}}_c, \vec{\nabla}_\xi \varphi_\mu \right\rangle_{E,N}, \\ &\left\langle \mathcal{J}\vec{G}_c, \vec{\varphi}_c \right\rangle_{E,N} = \int_{\partial E,N} C\vec{\tilde{\varphi}}_c \cdot \hat{n} \,\mathrm{d}S_\xi - \left\langle C, \vec{\nabla}_\xi \cdot \vec{\tilde{\varphi}}_c \right\rangle_{E,N}.\end{aligned} \tag{68}$$

In (68), as part of the discretization, we restrict the test functions to polynomial spaces.

The inter–element coupling of the Euler conservative fluxes and the viscous fluxes is achieved through the suitable choice of numerical fluxes. Boundary conditions are also enforced by numerical fluxes in the element boundary quadratures of (68),

$$\begin{aligned}&\overleftrightarrow{\mathbf{F}}_e \approx \overleftrightarrow{\mathbf{F}}_e^{\star}\left(\mathbf{Q}_L,\mathbf{Q}_R\right),\quad \overleftrightarrow{\mathbf{F}}_v \approx \overleftrightarrow{\mathbf{F}}_v^{\star}\left(\overleftrightarrow{\mathbf{G}}_L,\overleftrightarrow{\mathbf{G}}_R\right),\quad \mathbf{W}\approx \mathbf{W}^{\star}\left(\mathbf{Q}_L,\mathbf{Q}_R\right),\\ &\vec{G}_c \approx \vec{G}_c^{\star}\left(\vec{G}_{cL},\vec{G}_{cR}\right),\quad C\approx C^{\star}\left(C_L,C_R\right).\end{aligned} \tag{69}$$

Whereas for non–conservative terms, we follow [69,9,10] and use *diamond fluxes* at the boundaries,

$$\overleftrightarrow{\mathbf{\Phi}}_m W_m \approx \left(\overleftrightarrow{\mathbf{\Phi}}_m W_m\right)^{\diamond}\left(\mathbf{Q}_L,\mathbf{Q}_R\right). \tag{70}$$

A detailed definition of the fluxes as well as the connectivity between adjacent L and R p–non–conforming faces is presented below in Sec. 3.3. The non–conservative fluxes are not singled valued and can admit jumps in the values from one face to the other.

Inserting the numerical (69) and diamond (70) fluxes into (68) completes the discretization in space,

$$\begin{aligned}&\left\langle \mathcal{J}\underline{\mathbf{M}}\mathbf{Q}_t,\boldsymbol{\varphi}_q\right\rangle_{E,N} + \int_{\partial E,N}\boldsymbol{\varphi}_q^T\left(\overleftrightarrow{\tilde{\mathbf{F}}}_e^{\star}+\sum_{m=1}^{5}\left(\overleftrightarrow{\tilde{\mathbf{\Phi}}}_m W_m\right)^{\diamond}-\overleftrightarrow{\tilde{\mathbf{F}}}_v^{\star}\right)\cdot\hat{n}\,\mathrm{d}S_\xi - \left\langle \overleftrightarrow{\tilde{\mathbf{F}}}_e,\vec{\nabla}_\xi\boldsymbol{\varphi}_q\right\rangle_{E,N}\\ &\qquad -\sum_{m=1}^{5}\left\langle W_m,\vec{\nabla}_\xi\cdot\left(\boldsymbol{\varphi}_q^T\overleftrightarrow{\tilde{\mathbf{\Phi}}}_m\right)\right\rangle_{E,N} = -\left\langle \overleftrightarrow{\tilde{\mathbf{F}}}_v,\vec{\nabla}_\xi\boldsymbol{\varphi}_q\right\rangle_{E,N}+\left\langle \mathcal{J}\mathbf{S},\boldsymbol{\varphi}_q\right\rangle_{E,N},\end{aligned} \tag{71a}$$

$$\left\langle \mathcal{J}\overleftrightarrow{\mathbf{G}},\overleftrightarrow{\boldsymbol{\varphi}}_g\right\rangle_{E,N} = \int_{\partial E,N}\mathbf{W}^{\star,T}\left(\overleftrightarrow{\tilde{\boldsymbol{\varphi}}}_g\cdot\hat{n}\right)\mathrm{d}S_\xi - \left\langle \mathbf{W},\vec{\nabla}_\xi\cdot\overleftrightarrow{\tilde{\boldsymbol{\varphi}}}_g\right\rangle_{E,N}, \tag{71b}$$

$$\left\langle \mathcal{J}\mu,\varphi_\mu\right\rangle_{E,N} = \left\langle \mathcal{J}F_0',\varphi_\mu\right\rangle_{E,N} - \int_{\partial E,N}\frac{3}{2}\sigma\varepsilon\varphi_\mu\vec{\tilde{G}}_c^{\star}\cdot\hat{n}\,\mathrm{d}S_\xi + \left\langle \frac{3}{2}\sigma\varepsilon\vec{\tilde{G}}_c,\vec{\nabla}_\xi\varphi_\mu\right\rangle_{E,N}, \tag{71c}$$

$$\left\langle \mathcal{J}\vec{G}_c,\vec{\varphi}_c\right\rangle_{E,N} = \int_{\partial E,N}C^{\star}\vec{\tilde{\varphi}}_c\cdot\hat{n}\,\mathrm{d}S_\xi - \left\langle C,\vec{\nabla}_\xi\cdot\vec{\tilde{\varphi}}_c\right\rangle_{E,N}. \tag{71d}$$

The final transformation is to apply the discrete Gauss law (67) to the non–conservative terms and inviscid fluxes of (71a), in (71b), and in (71d), and use (62) to write surface integrals in physical variables. The final form of the semi–discretized iNS/CH system is

$$\begin{aligned}&\left\langle \mathcal{J}\underline{\mathbf{M}}\mathbf{Q}_t,\boldsymbol{\varphi}_q\right\rangle_{E,N} + \int_{\partial e,N}\boldsymbol{\varphi}_q^T\left(\overleftrightarrow{\tilde{\mathbf{F}}}_e^{\star}-\overleftrightarrow{\mathbf{F}}_e+\sum_{m=1}^{5}\left(\left(\overleftrightarrow{\mathbf{\Phi}}_m W_m\right)^{\diamond}-\overleftrightarrow{\mathbf{\Phi}}_m W_m\right)\right)\cdot\vec{n}\,\mathrm{d}S\\ &\qquad +\left\langle \boldsymbol{\varphi}_q,\vec{\nabla}_\xi\cdot\overleftrightarrow{\tilde{\mathbf{F}}}_e\right\rangle_{E,N}+\sum_{m=1}^{5}\left\langle \boldsymbol{\varphi}_q,\overleftrightarrow{\tilde{\mathbf{\Phi}}}_m\cdot\vec{\nabla}_\xi W_m\right\rangle_{E,N}\\ &\qquad = \int_{\partial e,N}\boldsymbol{\varphi}_q^T\overleftrightarrow{\mathbf{F}}_v^{\star}\cdot\vec{n}\,\mathrm{d}S - \left\langle \overleftrightarrow{\tilde{\mathbf{F}}}_v,\vec{\nabla}_\xi\boldsymbol{\varphi}_q\right\rangle_{E,N}+\left\langle \mathcal{J}\mathbf{S},\boldsymbol{\varphi}_q\right\rangle_{E,N},\end{aligned} \tag{72a}$$

$$\left\langle \mathcal{J}\overleftrightarrow{\mathbf{G}},\overleftrightarrow{\boldsymbol{\varphi}}_g\right\rangle_{E,N} = \int_{\partial e,N}\left(\mathbf{W}^{\star,T}-\mathbf{W}^T\right)\left(\overleftrightarrow{\boldsymbol{\varphi}}_g\cdot\vec{n}\right)\mathrm{d}S + \left\langle \overleftrightarrow{\tilde{\boldsymbol{\varphi}}}_g,\vec{\nabla}_\xi\mathbf{W}\right\rangle_{E,N}, \tag{72b}$$

$$\left\langle \mathcal{J}\mu,\varphi_\mu\right\rangle_{E,N} = \left\langle \mathcal{J}F_0',\varphi_\mu\right\rangle_{E,N} - \int_{\partial e,N}\frac{3}{2}\sigma\varepsilon\varphi_\mu\vec{\tilde{G}}_c^{\star}\cdot\vec{n}\,\mathrm{d}S + \left\langle \frac{3}{2}\sigma\varepsilon\vec{\tilde{G}}_c,\vec{\nabla}_\xi\varphi_\mu\right\rangle_{E,N}, \tag{72c}$$

$$\left\langle \mathcal{J}\vec{G}_c,\vec{\varphi}_c\right\rangle_{E,N} = \int_{\partial e,N}\left(C^{\star}-C\right)\vec{\varphi}_c\cdot\vec{n}\,\mathrm{d}S + \left\langle \vec{\tilde{\varphi}}_c,\vec{\nabla}_\xi C\right\rangle_{E,N}. \tag{72d}$$

The efficiency of the algorithm is improved with the use of single calculation of the local gradient of the entropy variables, for both non–conservative terms in (72a) and gradients in (72b) [10].

3.3. Numerical fluxes

In this section we introduce the specification of the numerical fluxes $\mathbf{F}_e^\star$, $\mathbf{F}_v^\star$, $\mathbf{W}^\star$, $\vec{G}_c^\star$, $C^\star$, and diamond fluxes $(\overleftrightarrow{\mathbf{\Phi}}_m W_m)^\diamond$ for p–non–conforming element interfaces. We also present how the mortar operations are embedded within the numerical fluxes.

In the previous work of [10] two numerical fluxes were proposed for the inviscid fluxes, an entropy conserving central flux and an entropy–stable exact Riemann solver derived in [70]. In the context of p–non–conforming approximations, the exact Riemann solver is not provably entropy–stable when combined with the mortar method. We have adopted the entropy–conserving central fluxes combined with the mortar method as they provide results with similar accuracy as the original scheme in [10]. For the viscous fluxes and concentration gradient, we use the Bassi–Rebay 1 (BR1) scheme [71] along with the mortar method and we introduce additional dissipation to make them entropy–stable by penalising the jump of the chemical potential across neighbouring elements.

The inviscid fluxes and non–conservative terms are rotational invariant as the rotational invariance of the flux [72,9] enables us to write the normal flux $\overleftrightarrow{\mathbf{F}}_e \cdot \vec{n}$ from a rotated version of the inviscid flux x–component $\mathbf{F}_e$,

$$\overleftrightarrow{\mathbf{F}}_e \cdot \vec{n} = \underline{\mathbf{T}}^T \mathbf{F}_e\left(\underline{\mathbf{T}}\mathbf{Q}\right) = \underline{\mathbf{T}}^T \mathbf{F}_e\left(\mathbf{Q}_n\right), \quad \underline{\mathbf{T}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & n_x & n_y & n_z & 0 \\ 0 & t_{1,x} & t_{1,y} & t_{1,z} & 0 \\ 0 & t_{2,x} & t_{2,y} & t_{2,z} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{73}$$

where $\underline{\mathbf{T}}$ is a rotation matrix that only affects velocities, $\vec{n} = \left(n_x, n_y, n_z\right)$ is the normal unit vector to the face, and $\vec{t}_1$ and $\vec{t}_2$ are two tangent unit vectors to the face. When the rotation matrix $\underline{\mathbf{T}}$ multiplies the state vector $\mathbf{Q}$, we retrieve the face normal state vector $\mathbf{Q}_n$,

$$\mathbf{Q}_n = \underline{\mathbf{T}}\mathbf{Q} = \left(C, \sqrt{\rho}U_n, \sqrt{\rho}V_{t1}, \sqrt{\rho}V_{t2}, P\right), \tag{74}$$

where $U_n = \vec{U} \cdot \vec{n}$ the normal velocity, and $V_{ti} = \vec{U} \cdot \vec{t}_i$ $(i = 1, 2)$ are the two tangent velocities. Note that the reference system rotation does not affect the total speed

$$V_{tot}^2 = U^2 + V^2 + W^2 = U_n^2 + V_{t1}^2 + V_{t2}^2. \tag{75}$$

The conservative inviscid flux terms take the form

$$\overleftrightarrow{\mathbf{F}}_e \cdot \vec{n} = \underline{\mathbf{T}}^T \begin{pmatrix} CU_n \\ \frac{1}{2}\rho U_n^2 + p \\ \frac{1}{2}\rho U_n V_{t1} \\ \frac{1}{2}\rho U_n V_{t2} \\ 0 \end{pmatrix}. \tag{76}$$

The non–conservative terms,

$$\sum_{m=1}^{5} \overleftrightarrow{\mathbf{\Phi}}_m W_m = \Upsilon \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2}\rho U^2 + \mu C & \frac{1}{2}\rho UV & \frac{1}{2}\rho UW \\ \frac{1}{2}\rho UV & \frac{1}{2}\rho V^2 + \mu C & \frac{1}{2}\rho VW \\ \frac{1}{2}\rho UW & \frac{1}{2}\rho VW & \frac{1}{2}\rho W^2 + \mu C \\ \rho_0 c_0^2 U & \rho_0 c_0^2 V & \rho_0 c_0^2 W \end{pmatrix}, \tag{77}$$

are also rotationally invariant,

$$\sum_{m=1}^{5} \left(\overleftrightarrow{\mathbf{\Phi}}_m W_m\right) \cdot \vec{n} = \underline{\mathbf{T}}^T \begin{pmatrix} 0 \\ \frac{1}{2}\rho U_n^2 + \mu C \\ \frac{1}{2}\rho U_n V_{t1} \\ \frac{1}{2}\rho U_n V_{t2} \\ \rho_0 c_0^2 U_n \end{pmatrix} = \underline{\mathbf{T}}^T \mathbf{F}_{\Phi W}\left(\mathbf{Q}_n\right). \tag{78}$$

3.3.1. The mortar element method

Following [50], we introduce the projection operators $\mathcal{P}_{lh}$ and $\mathcal{P}_{hl}$ from the order N_l (low order) space to its $N_h > N_l$ (high order) counterpart and vice versa known as the *mortar method* [73]. Let F, G be two polynomial functions that belong in $F \in \mathbb{P}^{N_l}$ and $G \in \mathbb{P}^{N_h}$ spaces. Then following the work of [23,28] for the derivation of SBP–preserving operators which satisfy the M–compatibility condition, the projection operator is designed to satisfy

$$\langle F, \mathcal{P}_{hl} G \rangle_{E,N_l} = \langle \mathcal{P}_{lh} F, G \rangle_{E,N_h} . \tag{79}$$

When using the operator (79), the augmentation of polynomial order is done through interpolation, whereas the opposite operation is a projection known as *restriction*.

The interpolation operator is defined as

$$(\mathcal{P}_{lh})_{ij} = l_j^{N_l}\left(\xi_i^{N_h}\right), \tag{80}$$

that is, we evaluate the Lagrange interpolating polynomials of the low–order N_l space at the GL nodes of the high–order N_h space. The choice of (80) for the projection, (79) gives the restriction (backward) projection

$$\mathcal{P}_{hl} = M_l^{-1}\mathcal{P}_{lh}^T M_h, \quad M_e = \mathrm{diag}\left(w_0^{N_e}, w_1^{N_e}, ..., w_{N_e}^{N_e},\right), \tag{81}$$

where $w_j^{N_e}$ are the quadrature weights at the GL quadrature points for an element E with polynomial order N_e. The mortar operator keeps the integral value, which means that it is designed to hold the following property

$$\langle \mathcal{P}_{hl}\left(\mathcal{P}_{lh}\left(A\right)\right), 1\rangle_{E,N_l} = \langle \mathcal{P}_{lh}\left(A\right), \mathcal{P}_{lh}\left(1\right)\rangle_{E,N_h} = \langle A, \mathcal{P}_{hl}\left(\mathcal{P}_{lh}\left(1\right)\right)\rangle_{E,N_l} = \langle A, 1\rangle_{E,N_l}, \tag{82}$$

which is useful for proving that the scheme is conservative [28]. However, (80) and (81) are not invertible. That is $\mathcal{P}_{hl}\mathcal{P}_{lh} \neq \mathcal{I}_h$ and $\mathcal{P}_{lh}\mathcal{P}_{hl} \neq \mathcal{I}_l$. This means that when a polynomial $A \in \mathbb{P}^{N_l}$ from the lower space is projected to the mortar, it is not recovered by using the restriction operator,

$$\mathcal{P}_{hl}\left(\mathcal{P}_{lh}\left(A\right)\right) \neq A. \tag{83}$$

The only exception to (83) is for any constant function, $\mathcal{P}_{hl}\left(\mathcal{P}_{lh}\left(k\right)\right) = k$.

3.3.2. Inviscid fluxes: entropy conserving central fluxes

Initially, we introduce the central numerical fluxes for the p–non–conforming interfaces which will lead to an entropy conserving approximation. Without loss of generality, we consider the inter–element face with orders $N_l < N_h$. For the inviscid conservative vector fluxes, we transform the contravariant fluxes from the lower order to the higher order element and then we take their difference (since the normal vectors are opposite) and thus (76) becomes

$$\begin{aligned}
\left(\overleftrightarrow{\mathbf{F}}_e^\star \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_h &= \left(\frac{\overleftrightarrow{\mathbf{F}}_{e,h} + \mathcal{P}_{lh}\left(\overleftrightarrow{\mathbf{F}}_{e,l}\right)}{2}\right) \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h, \\
\left(\overleftrightarrow{\mathbf{F}}_e^\star \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_l &= -\mathcal{P}_{hl}\left(\left(\overleftrightarrow{\mathbf{F}}_e^\star\right) \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_h .
\end{aligned} \tag{84}$$

The non–conservative scalar fluxes are computed on the higher order element and then transferred to the lower order element through the restriction operator,

$$\begin{aligned}
\left(\sum_{m=1}^{5}\left(\left(\overleftrightarrow{\mathbf{\Phi}}_m W_m\right)^\diamond - \overleftrightarrow{\mathbf{\Phi}}_m W_m\right) \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_h &= \sum_{m=1}^{5}\left(\overleftrightarrow{\mathbf{\Phi}}_{m,h}\left(\frac{W_{m,h} + \mathcal{P}_{lh}\left(W_{m,l}\right)}{2} - W_{m,h}\right)\right) \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h \\
&= \sum_{m=1}^{5} \overleftrightarrow{\mathbf{\Phi}}_{m,h} \frac{\left(\mathcal{P}_{lh}\left(W_{m,l}\right) - W_{m,h}\right)}{2} \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h, \\
\left(\sum_{m=1}^{5}\left(\left(\overleftrightarrow{\mathbf{\Phi}}_m W_m\right)^\diamond - \overleftrightarrow{\mathbf{\Phi}}_m W_m\right) \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_l &= \sum_{m=1}^{5} \overleftrightarrow{\mathbf{\Phi}}_{m,l} \cdot \mathcal{P}_{hl}\left(\frac{\left(\mathcal{P}_{lh}\left(W_{m,l}\right) - W_{m,h}\right)}{2} \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h\right).
\end{aligned} \tag{85}$$

In (84) and (85), $|\mathcal{J}\vec{a}|^h\vec{n}^h$ and $|\mathcal{J}\vec{a}|^l\vec{n}^l$ are the (scaled) normal vectors at the face, and they are constructed to satisfy $|\mathcal{J}\vec{a}|^h\vec{n}^h = -\left|\mathcal{J}\vec{a}\right|^l\vec{n}^l = -\mathcal{P}_{lh}\left(|\mathcal{J}\vec{a}|^l\vec{n}^l\right)$ (**Condition (F)**).

3.3.3. Viscous and chemical potential fluxes: Bassi–Rebay 1 (BR1) method

For the viscous fluxes and the chemical potential we use the Bassi–Rebay 1 (BR1) scheme [71] and we include an additional dissipation term to add interface stabilization. On a similar fashion to the inviscid fluxes, with the use of the mortar method, we get the viscous scalar fluxes,

$$\begin{aligned}
\left(\left(\mathbf{W}^\star - \mathbf{W}\right)\left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_h &= \left(\frac{\mathbf{W}_h + \mathcal{P}_{lh}\left(\mathbf{W}_l\right)}{2} - \mathbf{W}_h\right)\left|\mathcal{J}\vec{a}\right|^h \vec{n}^h = \frac{1}{2}\left(\mathcal{P}_{lh}\left(\mathbf{W}_l\right) - \mathbf{W}_h\right)\left|\mathcal{J}\vec{a}\right|^h \vec{n}^h, \\
\left(\left(\mathbf{W}^\star - \mathbf{W}\right)\left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_l &= \mathcal{P}_{hl}\left(\left(\mathbf{W}^\star - \mathbf{W}\right)\left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_h ,
\end{aligned} \tag{86}$$

and the Cahn–Hilliard scalar fluxes,

$$\begin{aligned}\left(\left(C^\star - C\right)\left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_h &= \left(\frac{C_h + \mathcal{P}_{lh}\left(C_l\right)}{2} - C_h\right)\left|\mathcal{J}\vec{a}\right|^h \vec{n}^h = \frac{1}{2}\left(\mathcal{P}_{lh}\left(C_l\right) - C_h\right)\left|\mathcal{J}\vec{a}\right|^h \vec{n}^h, \\ \left(\left(C^\star - C\right)\left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_l &= \mathcal{P}_{hl}\left(\left(C^\star - C\right)\left|\mathcal{J}\vec{a}\right|^i \vec{n}^i\right)_h.\end{aligned} \tag{87}$$

Similarly, we use the viscous vector fluxes,

$$\begin{aligned}\tilde{\overleftrightarrow{\mathbf{F}}}^\star_{v,h} \cdot \hat{n} &= \frac{1}{2}|\mathcal{J}\vec{a}|^h \vec{n}^h \cdot \left(\overleftrightarrow{\mathbf{F}}_{v,h} + \mathcal{P}_{lh}\left(\overleftrightarrow{\mathbf{F}}_{v,l}\right)\right) + \beta\left|\mathcal{J}\vec{a}\right|^h \left(\mu_h - \mathcal{P}_{lh}\left(\mu_l\right)\right), \\ \tilde{\overleftrightarrow{\mathbf{F}}}^\star_{v,l} \cdot \hat{n} &= -\mathcal{P}_{hl}\left(\tilde{\vec{F}}^\star_h \cdot \hat{n}\right),\end{aligned} \tag{88}$$

and the Cahn–Hilliard vector fluxes,

$$\begin{aligned}\tilde{\vec{G}}^\star_{c,h} \cdot \hat{n} &= \frac{1}{2}|\mathcal{J}\vec{a}|^h \vec{n}^h \cdot \left(\vec{G}_{c,h} + \mathcal{P}_{lh}\left(\vec{G}_{c,l}\right)\right) + \beta\left|\mathcal{J}\vec{a}\right|^h \left(C_h - \mathcal{P}_{lh}\left(C_l\right)\right), \\ \tilde{\vec{G}}^\star_{c,l} \cdot \hat{n} &= -\mathcal{P}_{hl}\left(\tilde{\vec{G}}^\star_{c,h} \cdot \hat{n}\right),\end{aligned} \tag{89}$$

where β is the penalty parameter computed in this work following [74,7],

$$\beta = \kappa_\beta \frac{N_h(N_h+1)}{2}|\mathcal{J}_f|\{\!\{\mathcal{J}^{-1}\}\!\}, \tag{90}$$

with κ_β a dimensionless free parameter (in this work we only use $\kappa_\beta = 0$ to disable interface stabilization, and $\kappa_\beta = 1$ to enable interface stabilization) and $\vec{n}_L$ is the outward normal vector to the left element. In (90) N_h is the polynomial order of the higher order element, $|\mathcal{J}_f|$ is the surface Jacobian of the face, and $\{\!\{\mathcal{J}^{-1}\}\!\}$ is the average of the inverse of the Jacobians of left and right elements.

3.4. Boundary conditions

The final step to complete the scheme is the approximation of the boundary conditions. In this section, we describe the boundary conditions for the inviscid conservative and non–conservative fluxes as well as the viscous fluxes. Throughout this work, we consider free– and no–slip walls.

3.4.1. Inviscid flux

The inviscid numerical flux controls the normal velocity $\vec{u}\cdot\vec{n} = 0$ for both free– and no–slip boundary conditions. The application of the boundary conditions is done by creating a mirrored ghost state $\mathbf{Q}^e_n$ and applying a p–conforming version of the fluxes (84) and (85),

$$\mathbf{Q}^i_n = \begin{pmatrix} C \\ \sqrt{\rho}U_n \\ \sqrt{\rho}V_{t1} \\ \sqrt{\rho}V_{t2} \\ P \end{pmatrix}, \quad \mathbf{Q}^e_n = \begin{pmatrix} C \\ -\sqrt{\rho}U_n \\ \sqrt{\rho}V_{t1} \\ \sqrt{\rho}V_{t2} \\ P \end{pmatrix}. \tag{91}$$

The numerical (76) and diamond (78) central fluxes for (91) are,

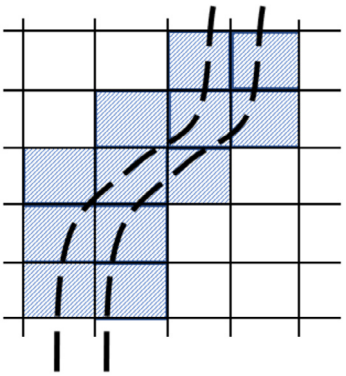
$$\overleftrightarrow{\mathbf{F}}^\star_e \cdot \vec{n} = \underline{\mathbf{T}}^T \begin{pmatrix} 0 \\ \frac{1}{2}\rho U_n^2 + P \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \sum_{m=1}^{5}\left(\overleftrightarrow{\mathbf{\Phi}}_m W_m\right)^\diamond \cdot \vec{n} = \underline{\mathbf{T}}^T \begin{pmatrix} 0 \\ C\mu \\ \frac{1}{2}\rho U_n V_{t1} \\ \frac{1}{2}\rho U_n V_{t2} \\ 0 \end{pmatrix}. \tag{92}$$

3.4.2. Viscous and Cahn–Hilliard fluxes

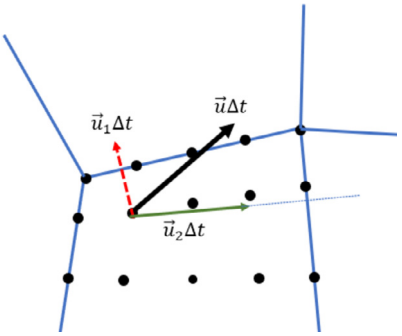
For the free–slip wall boundary condition, we impose zero normal stress $\boldsymbol{S}\cdot\vec{n} = 0$ (Neumann), whereas for the no–slip wall boundary condition, we impose zero velocity $\vec{u} = 0$ (Dirichlet). The concentration, concentration gradient, entropy variables, and viscous fluxes at the boundaries are:

1. *Free–slip wall boundary condition.*

$$C^\star = C, \quad \vec{G}^\star_c = g'(C), \quad \mathbf{W}^\star = \mathbf{W}, \quad \overleftrightarrow{\mathbf{F}}^\star_v \cdot \vec{n} = 0. \tag{93}$$



(a) Visualisation of the heuristic adaptation process. The marked elements contain the interface and are those which will be refined, whereas the rest of the elements are coarsened.



(b) Representation of the interface movement prediction method. The red vector, which exceeds the elements boundaries, will cause the refinement of the neighbouring element on top. The green vector does not exceeds the elements boundaries and thus the element on the right will not be refined.

Fig. 1. Visualisation of the methods used to perform p–adaptation for multiphase flow applications in this paper.

2. *No–slip wall boundary condition.*

$$C^\star = C,\ \ \vec{G}_c^\star = g'(C),\ \ \mathbf{W}^\star = (\mu, 0, 0, 0, P)\,,\ \ \overleftrightarrow{\mathbf{F}}_v^\star \cdot \vec{n} = \begin{pmatrix} 0 \\ 2\eta\mathcal{S} \\ 0 \end{pmatrix} \cdot \vec{n}. \tag{94}$$

4. Adaptation methodology

In this section we introduce the polynomial order adaptation methodology for the iNS/CH system. When performing multiphase flow simulations, an important requirement is to resolve properly the interface region between the two phases. Through this work, we seek to exploit the flexibility of the DGSEM and perform the local refinement through p–adaptation as an alternative to h–adaptation which has been covered in previous works [36,43,37]. In many occasions, such as in the numerical cases presented in Section 6, the resolution required to resolve the interface region and the resolution required for the rest of the domain vary. Therefore, to make the simulations more efficient and achieve savings in computational time, we reduce the total degrees of freedom by locally refining the interface region.

The indicator used to perform p–adaptation is based on the knowledge of the location of the interface as the phase field parameter ϕ ranges from $0.1 < \phi < 0.9$ within this region, a choice which is usual in diffuse interface literature [75]. A visualization of how the elements are refined according to the position of the interface is given in Fig. 1a. If an element is marked for refinement, then it is refined to a user–specified polynomial order in all directions, whereas elements that do not contain any part of the interface are coarsened also to a user–specified order. An additional step, which extends the work presented in [8], is that the convection term of the Cahn–Hilliard equation is used to predict the movement of the interface across elements. Specifically, we calculate the normal velocity to each element's face for each nodal point and by multiplying with the timestep we derive an estimation for the movement of the interface before marching in time. Then, when the interface crosses to a non–refined element, the polynomial order across the domain is adapted. This is an improvement from the work presented in [8], where the adaptation interval was a user–specified parameter. An example of this extension is given in Fig. 1b.

In this work, we use the operator $\mathcal{P}_{PN}$ (79) to perform an L^2 projection from order P to order N as presented in [63],

$$\left(\Phi^N\right) = \mathcal{P}_{PN}\left(\Phi^P\right). \tag{95}$$

For interpolation we use the operator (80) and for restriction the operator (81).

This method is effective since the interface of the Cahn-Hilliard equation involves a single constant scalar. Higher order methods such as discontinuous Galerkin allow to change the resolution to an arbitrary order without the need to alter the mesh. In addition, for phase field simulations, the target is to have a sufficient number of solution points within the interface to capture it appropriately. For other discretization methods, this number is defined to be at least 5 points within this region [76], while some researchers prefer a number close to 10 [77,78]. Therefore, since we have a priori knowledge of the element size and the interface width parameter, the user can specify the polynomial order to achieve the desired

value of points within the interface. The advantage over similar marker methods used for AMR approaches is that it does not require a rigorous tuning of parameters such as in [42,79,80]. An additional refinement indicator for the Navier–Stokes equations could be combined with the aforementioned criterion to refine the flow locally to other features outside the interface, but this is beyond the scope of this work.

5. Semi–discrete stability analysis

The steps for the semi–discrete stability are given in [9,10]. The difference between the p–conforming and the p–non–conforming versions of the scheme arises in the interior boundary terms and their special treatment with the inclusion of the mortar method. The rest of the volume and boundary terms follow the same methodology as they are not affected by the inter–element connectivity. The analysis is semi–discrete since we assume exactness of the approximation in time.

Through this analysis, we prove that the spatial discretization (72) mimics the continuous entropy conservation law (38). We also omit the effect of lower order source terms ($\mathbf{S}=0$).

The detailed analysis of [10] is also summarised here in Appendix A.1. The analysis begins with the test functions, which are chosen to be $\boldsymbol{\varphi}_q=\mathbf{W}$, $\vec{\boldsymbol{\varphi}}_g=\overleftrightarrow{\mathbf{F}}_v$, $\varphi_\mu=C_t$, $\vec{\varphi}_c=\frac{3}{2}\sigma\varepsilon\vec{G}_c$ and the combination of (72a) and (72b), and that of (72c) with (72d), to obtain (96a) and (96b) accordingly,

$$\begin{aligned}\left\langle \mathcal{J}\underline{\mathbf{MQ}}_t,\mathbf{W}\right\rangle_{E,N}+\int_{\partial e,N}\mathbf{W}^T\left(\overleftrightarrow{\mathbf{F}}_e^\star-\overleftrightarrow{\mathbf{F}}_e+\sum_{m=1}^{5}\left(\overleftrightarrow{\mathbf{\Phi}}_m W_m\right)^\diamond-\overleftrightarrow{\mathbf{\Phi}}_m W_m\right)\cdot\vec{n}\,\mathrm{d}S\\ +\left\langle\mathbf{W},\vec{\nabla}_\xi\cdot\overleftrightarrow{\tilde{\mathbf{F}}}_e\right\rangle_{E,N}+\sum_{m=1}^{5}\left\langle\mathbf{W},\overleftrightarrow{\tilde{\mathbf{\Phi}}}_m\cdot\vec{\nabla}_\xi W_m\right\rangle_{E,N}\\ =\int_{\partial e,N}\left(\mathbf{W}^T\overleftrightarrow{\mathbf{F}}_v^\star+\mathbf{W}^{\star,T}\overleftrightarrow{\mathbf{F}}_v-\mathbf{W}^T\overleftrightarrow{\mathbf{F}}_v\right)\cdot\vec{n}\,\mathrm{d}S-\left\langle\mathcal{J}\overleftrightarrow{\mathbf{G}},\overleftrightarrow{\mathbf{F}}_v\right\rangle_{E,N},\end{aligned}\tag{96a}$$

$$\begin{aligned}\left\langle\mathcal{J}\mu,C_t\right\rangle_{E,N}=&\left\langle\mathcal{J}F_0',C_t\right\rangle_{E,N}+\frac{3}{2}\sigma\varepsilon\left\langle\mathcal{J}\vec{G}_{c,t},\vec{G}_c\right\rangle_{E,N}\\ &-\int_{\partial e,N}\frac{3}{2}\sigma\varepsilon\left(C_t\vec{G}_c^\star+C_t^\star\vec{G}_c-C_t\vec{G}_c\right)\cdot\vec{n}\,\mathrm{d}S.\end{aligned}\tag{96b}$$

This merges four equations into two.

We first study the time derivative terms of (96a) and (96b) which are transformed to [10]

$$\left\langle\mathcal{J}\underline{\mathbf{MQ}}_t,\mathbf{W}\right\rangle_{E,N}=\left\langle\mathcal{J}\mathcal{E}_t,1\right\rangle_{E,N}-\int_{\partial e,N}\frac{3}{2}\sigma\varepsilon\left(C_t\vec{G}_c^\star+C_t^\star\vec{G}_c-C_t\vec{G}_c\right)\cdot\vec{n}\,\mathrm{d}S.\tag{97}$$

Eq. (97) is the discrete version of (35). As a result, we get the discrete version of (35) for the entropy function,

$$\mathcal{E}=\mathcal{F}+\mathcal{K}+\mathcal{E}_{\mathrm{AC}}=F_0+\frac{3}{4}\sigma\varepsilon|\vec{G}_c|^2+\frac{1}{2}\rho V_{tot}^2+\frac{P^2}{2\rho_0c_0^2},\tag{98}$$

and for the time entropy flux $\vec{F}_t^{\mathcal{E}}$, which appears as the argument of the surface integral in (97).

The contraction of the inviscid volume terms into a boundary entropy flux holds discretely as in the continuous analysis (27), as shown in [10]. Using properties (29) and (30) we show that

$$\left\langle\mathbf{W},\vec{\nabla}_\xi\cdot\overleftrightarrow{\tilde{\mathbf{F}}}_e\right\rangle_{E,N}+\sum_{m=1}^{5}\left\langle\mathbf{W},\overleftrightarrow{\tilde{\mathbf{\Phi}}}_m\cdot\vec{\nabla}_\xi W_m\right\rangle_{E,N}=\int_{\partial e,N}\vec{F}^{\mathcal{E}}\cdot\vec{n}\,\mathrm{d}S.\tag{99}$$

Analogously to (28), viscous volume terms are dissipative if

$$\left\langle\mathcal{J}\overleftrightarrow{\mathbf{G}},\overleftrightarrow{\mathbf{F}}_v\left(\overleftrightarrow{\mathbf{G}}\right)\right\rangle_{E,N}\geqslant 0.\tag{100}$$

So we have shown that the discrete volume terms mimic the continuous entropy analysis for the volume terms. The rest of the proof focuses on the boundary terms and the incorporation of the mortar method to show that the proposed scheme maintains the entropy–stability characteristics when transitioning to p–non–conforming grids.

5.1. Boundary terms

The p–non–conformity of the internal elements interfaces affects the entropy analysis for the boundary terms. In this section, we will present in detail the entropy analysis with the incorporation of the mortar method. The first step is to consider all elements of the domain by updating (96) with (97), (99), and (100), and summing over all elements deriving

$$\frac{\mathrm{d}\bar{\mathcal{E}}}{\mathrm{d}t} + \mathrm{IBT} + \mathrm{PBT} = -\left\langle \mathcal{J}\overleftrightarrow{\mathbf{G}}, \overleftrightarrow{\mathbf{F}}_v\left(\overleftrightarrow{\mathbf{G}}\right)\right\rangle_{E,N}, \tag{101}$$

where $\bar{\mathcal{E}}$ is the total entropy,

$$\bar{\mathcal{E}} = \sum_e \langle \mathcal{J}\mathcal{E}, 1\rangle_{E,N}, \tag{102}$$

IBT is the contribution from interior faces,

$$\begin{aligned}
\mathrm{IBT} =& \mathrm{IBT}_e + \mathrm{IBT}_v + \mathrm{IBT}_{ch} \\
=& -\sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_l} \left(\mathbf{W}_l^T \left(\overleftrightarrow{\mathbf{F}}_e^\star \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i \right)_l + \left(\left(\sum_{m=1}^{5} \left(\overleftrightarrow{\mathbf{\Phi}}_m W_m \right)^\diamond - \overleftrightarrow{\mathbf{\Phi}}_m W_m \right) \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i \right)_l \right) \mathrm{d}S_\xi \\
& - \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_h} \left(\mathbf{W}_h^T \left(\overleftrightarrow{\mathbf{F}}_e^\star \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i \right)_h + \left(\left(\sum_{m=1}^{5} \left(\overleftrightarrow{\mathbf{\Phi}}_m W_m \right)^\diamond - \overleftrightarrow{\mathbf{\Phi}}_m W_m \right) \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i \right)_h \right) \mathrm{d}S_\xi \\
& + \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_l} \left(\mathbf{W}_l^T \tilde{\overleftrightarrow{\mathbf{F}}}_{v,l}^\star \cdot \hat{n}_l + \left(\left(\mathbf{W}^{\star,T} - \mathbf{W}^T \right) \left|\mathcal{J}\vec{a}\right| \vec{n} \right)_l \cdot \overleftrightarrow{\mathbf{F}}_{v,l} \right) \mathrm{d}S_\xi \\
& + \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_h} \left(\mathbf{W}_l^T \tilde{\overleftrightarrow{\mathbf{F}}}_{v,h}^\star \cdot \hat{n}_h + \left(\left(\mathbf{W}^{\star,T} - \mathbf{W}^T \right) \left|\mathcal{J}\vec{a}\right| \vec{n} \right)_h \cdot \overleftrightarrow{\mathbf{F}}_{v,h} \right) \mathrm{d}S_\xi \\
& + \frac{3}{2}\sigma\varepsilon \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_l} \left(C_{t,l} \tilde{\vec{G}}_{c,l}^\star \cdot \hat{n}_l + \left(\left(C_t^\star - C_t \right) \left|\mathcal{J}\vec{a}\right| \vec{n} \right)_l \vec{G}_{c,l} \right) \mathrm{d}S_\xi \\
& + \frac{3}{2}\sigma\varepsilon \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_h} \left(C_{t,h} \tilde{\vec{G}}_{c,h}^\star \cdot \hat{n}_h + \left(\left(C_t^\star - C_t \right) \left|\mathcal{J}\vec{a}\right| \vec{n} \right)_h \vec{G}_{c,h} \right) \mathrm{d}S_\xi,
\end{aligned} \tag{103}$$

where we used the entropy flux definition (30) to reduce the number of terms. In Section 5.1.1, we prove the stability of the scheme for general interface comprised of faces with different polynomial orders $N_l \leq N_h$. Finally, PBT is the contribution from physical boundary faces and follows the same analysis of [10] as the ghost state is approximated with the same polynomial order as the external face,

$$\begin{aligned}
\mathrm{PBT} =& \mathrm{PBT}_e + \mathrm{PBT}_v + \mathrm{PBT}_{ch} \\
=& \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(\mathbf{W}^T \overleftrightarrow{\mathbf{F}}_e^\star + \sum_{m=1}^{5} \mathbf{W}^T \left(\left(\overleftrightarrow{\mathbf{\Phi}}_m W_m \right)^\diamond - \overleftrightarrow{\mathbf{\Phi}}_m W_m \right) \right) \cdot \vec{n} \mathrm{d}S \\
& - \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(\mathbf{W}^T \left(\overleftrightarrow{\mathbf{F}}_v^\star - \overleftrightarrow{\mathbf{F}}_v \right) + \mathbf{W}^{\star,T} \overleftrightarrow{\mathbf{F}}_v \right) \cdot \vec{n} \mathrm{d}S \\
& - \frac{3}{2}\sigma\varepsilon \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(C_t \vec{G}_c^\star + C_t^\star \vec{G}_c - C_t \vec{G}_c \right) \cdot \vec{n} \, \mathrm{d}S.
\end{aligned} \tag{104}$$

5.1.1. Inviscid interior boundary terms

We present the stability analysis of the inviscid interior p–non–conforming boundary terms with the central fluxes for the conservative numerical (76) and non–conservative diamond fluxes (78). We start with the first two terms of (103)

$$
\begin{aligned}
\mathrm{IBT} = &- \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_l} \left(\mathbf{W}_l^T \left(\overleftrightarrow{\mathbf{F}}_e^{\star} \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i \right)_l + \left(\sum_{m=1}^{5} \left(\left(\overleftrightarrow{\mathbf{\Phi}}_m W_m \right)^{\diamond} - \overleftrightarrow{\mathbf{\Phi}}_m W_m \right) \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i \right)_l \right) \mathrm{d}S_\xi \\
&- \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_h} \left(\mathbf{W}_h^T \left(\overleftrightarrow{\mathbf{F}}_e^{\star} \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i \right)_h + \left(\sum_{m=1}^{5} \left(\left(\overleftrightarrow{\mathbf{\Phi}}_m W_m \right)^{\diamond} - \overleftrightarrow{\mathbf{\Phi}}_m W_m \right) \cdot \left|\mathcal{J}\vec{a}\right|^i \vec{n}^i \right)_h \right) \mathrm{d}S_\xi , \\
= &T1 + T2
\end{aligned}
\tag{105}
$$

and replace with the expressions for the inter–element fluxes (84) and (85), which have been constructed using the watertight mesh condition (**Condition (F)**) to ensure that the (scaled) normal vectors at the faces satisfy $|\mathcal{J}\vec{a}|^h \vec{n}^h = -\left|\mathcal{J}\vec{a}\right|^l \vec{n}^l = -\mathcal{P}_{lh}\left(|\mathcal{J}\vec{a}|^l \vec{n}^l\right)$.

The conservative terms of (105) become

$$
\begin{aligned}
T1 = &\sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_l} \left(\mathbf{W}_l^T \frac{\mathcal{P}_{hl}\left(\left(\overleftrightarrow{\mathbf{F}}_{e,h} + \mathcal{P}_{lh}\left(\overleftrightarrow{\mathbf{F}}_{e,l} \right) \right) \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h \right)}{2} \right) \mathrm{d}S_\xi \\
&- \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_h} \left(\mathbf{W}_h^T \frac{\left(\overleftrightarrow{\mathbf{F}}_{e,h} + \mathcal{P}_{lh}\left(\overleftrightarrow{\mathbf{F}}_{e,l} \right) \right) \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h}{2} \right) \mathrm{d}S_\xi .
\end{aligned}
\tag{106}
$$

We now apply property (29) on each term of (106),

$$
\mathbf{W}_l^T \mathcal{P}_{hl}\left(\overleftrightarrow{\mathbf{F}}_{e,h} \right) = \sum_{m=1}^{5} W_{m,l} \mathcal{P}_{hl}\left(\mathbf{e}_m^T \overleftrightarrow{\mathbf{F}}_e \right) = \sum_{m=1}^{5} \mathcal{P}_{hl}\left(\mathbf{W}_h^T \overleftrightarrow{\mathbf{\Phi}}_{m,h} \right) W_{m,l}.
\tag{107}
$$

Then (106) is transformed to

$$
\begin{aligned}
T1 = &\sum_{\substack{\text{interior}\\ \text{faces}}} \sum_{m=1}^{5} \int_{f,N_l} \frac{1}{2} W_{m,l} \mathcal{P}_{hl}\left(\mathbf{W}_h^T \overleftrightarrow{\mathbf{\Phi}}_{m,h} \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h \right) \mathrm{d}S_\xi \\
&+ \sum_{\substack{\text{interior}\\ \text{faces}}} \sum_{m=1}^{5} \int_{f,N_l} \frac{1}{2} W_{m,l} \mathcal{P}_{hl}\left(\mathcal{P}_{lh}\left(\mathbf{W}_l^T \overleftrightarrow{\mathbf{\Phi}}_{m,l} \right) \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h \right) \mathrm{d}S_\xi \\
&- \sum_{\substack{\text{interior}\\ \text{faces}}} \sum_{m=1}^{5} \int_{f,N_h} \frac{1}{2} W_{m,h} \mathbf{W}_h^T \overleftrightarrow{\mathbf{\Phi}}_{m,h} \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h \mathrm{d}S_\xi \\
&- \sum_{\substack{\text{interior}\\ \text{faces}}} \sum_{m=1}^{5} \int_{f,N_h} \frac{1}{2} W_{m,h} \mathcal{P}_{lh}\left(\mathbf{W}_l^T \overleftrightarrow{\mathbf{\Phi}}_{m,l} \right) \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h \mathrm{d}S_\xi .
\end{aligned}
\tag{108}
$$

For the non–conservative terms of (105) we substitute the inter–element fluxes (85)

$$
\begin{aligned}
T2 = &- \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_l} \left(\sum_{m=1}^{5} \mathbf{W}_l^T \overleftrightarrow{\mathbf{\Phi}}_{m,l} \cdot \mathcal{P}_{hl}\left(\frac{\left(\mathcal{P}_{lh}\left(W_{m,l} \right) - W_{m,h} \right)}{2} \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h \right) \right) \mathrm{d}S_\xi \\
&- \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_h} \left(\sum_{m=1}^{5} \mathbf{W}_h^T \overleftrightarrow{\mathbf{\Phi}}_{m,h} \frac{\left(\mathcal{P}_{lh}\left(W_{m,l} \right) - W_{m,h} \right)}{2} \cdot \left|\mathcal{J}\vec{a}\right|^h \vec{n}^h \right) \mathrm{d}S_\xi .
\end{aligned}
\tag{109}
$$

Re–arranging (109) we get the following terms

$$
\begin{aligned}
T2 = & \sum_{\substack{\text{interior}\\ \text{faces}}} \frac{1}{2} \int_{f,N_l} \left(\sum_{m=1}^{5} \mathbf{W}_l^T \overleftrightarrow{\mathbf{\Phi}}_{m,l} \cdot \mathcal{P}_{hl} \left(W_{m,h} \left| \mathcal{J}\vec{a} \right|^h \vec{n}^h \right) \right) \mathrm{d}S_\xi \\
& - \sum_{\substack{\text{interior}\\ \text{faces}}} \frac{1}{2} \int_{f,N_l} \left(\sum_{m=1}^{5} \mathbf{W}_l^T \overleftrightarrow{\mathbf{\Phi}}_{m,l} \cdot \mathcal{P}_{hl} \left(\mathcal{P}_{lh} \left(W_{m,l} \right) \left| \mathcal{J}\vec{a} \right|^h \vec{n}^h \right) \right) \mathrm{d}S_\xi \\
& - \sum_{\substack{\text{interior}\\ \text{faces}}} \frac{1}{2} \int_{f,N_h} \left(\sum_{m=1}^{5} \mathbf{W}_h^T \overleftrightarrow{\mathbf{\Phi}}_{m,h} \mathcal{P}_{lh} \left(W_{m,l} \right) \cdot \left| \mathcal{J}\vec{a} \right|^h \vec{n}^h \right) \mathrm{d}S_\xi \\
& + \sum_{\substack{\text{interior}\\ \text{faces}}} \frac{1}{2} \int_{f,N_h} \left(\sum_{m=1}^{5} \mathbf{W}_h^T \overleftrightarrow{\mathbf{\Phi}}_{m,h} W_{m,h} \cdot \left| \mathcal{J}\vec{a} \right|^h \vec{n}^h \right) \mathrm{d}S_\xi .
\end{aligned}
\tag{110}
$$

We apply property (79) to the first, and third term, and we do it twice on the second term of (110) to obtain

$$
\begin{aligned}
T2 = & \sum_{\substack{\text{interior}\\ \text{faces}}} \frac{1}{2} \int_{f,N_h} \left(\sum_{m=1}^{5} \mathcal{P}_{lh} \left(\mathbf{W}_l^T \overleftrightarrow{\mathbf{\Phi}}_{m,l} \right) W_{m,h} \cdot \left| \mathcal{J}\vec{a} \right|^h \vec{n}^h \right) \mathrm{d}S_\xi \\
& - \sum_{\substack{\text{interior}\\ \text{faces}}} \frac{1}{2} \int_{f,N_l} \left(\sum_{m=1}^{5} W_{m,l} \mathcal{P}_{hl} \left(\mathcal{P}_{lh} \left(\mathbf{W}_l^T \overleftrightarrow{\mathbf{\Phi}}_{m,l} \right) \cdot \left| \mathcal{J}\vec{a} \right|^h \vec{n}^h \right) \right) \mathrm{d}S_\xi \\
& - \sum_{\substack{\text{interior}\\ \text{faces}}} \frac{1}{2} \int_{f,N_l} \left(\sum_{m=1}^{5} W_{m,l} \mathcal{P}_{hl} \left(\mathbf{W}_h^T \overleftrightarrow{\mathbf{\Phi}}_{m,h} \cdot \left| \mathcal{J}\vec{a} \right|^h \vec{n}^h \right) \right) \mathrm{d}S_\xi \\
& + \sum_{\substack{\text{interior}\\ \text{faces}}} \frac{1}{2} \int_{f,N_h} \left(\sum_{m=1}^{5} \mathbf{W}_h^T \overleftrightarrow{\mathbf{\Phi}}_{m,h} W_{m,h} \cdot \left| \mathcal{J}\vec{a} \right|^h \vec{n}^h \right) \mathrm{d}S_\xi .
\end{aligned}
\tag{111}
$$

The expressions (108) and (111) contain the same terms but with opposite signs. So when we sum them to construct the original stability equation (105) we find that inviscid interior boundaries exactly satisfy $\mathrm{IBT}_e = 0$. The inviscid interior boundary contribution to the entropy equation is identically zero, i.e. the interface approximation for the p–non–conforming entropy fluxes is entropy conserving.

5.1.2. Viscous and Cahn–Hilliard interior terms: Bassi–Rebay 1 method

In this part of the paper, we prove that the p–non–conforming treatment of the BR1 cancels the boundary integrals and the additional interface stabilization terms are dissipative. The derivation for the viscous part follows the work in [8] and the derivation for free-energy stability for the p–adaptive Cahn–Hilliard equation. The entropy analysis for the viscous terms of the Navier–Stokes equation starts by taking two neighbouring faces with polynomial orders N_h, N_l with $N_l \leq N_h$.

$$
\begin{aligned}
\mathrm{IBT}_v = & \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_l} \left(\mathbf{W}_l^T \tilde{\overleftrightarrow{\mathbf{F}}}^{\star}_{v,l} \cdot \hat{n}_l + \left(\left(\mathbf{W}^{\star,T} - \mathbf{W}^T \right) \left| \mathcal{J}\vec{a} \right| \vec{n} \right)_l \cdot \overleftrightarrow{\mathbf{F}}_{v,l} \right) \mathrm{d}S_\xi \\
& + \sum_{\substack{\text{interior}\\ \text{faces}}} \int_{f,N_h} \left(\mathbf{W}_l^T \tilde{\overleftrightarrow{\mathbf{F}}}^{\star}_{v,h} \cdot \hat{n}_h + \left(\left(\mathbf{W}^{\star,T} - \mathbf{W}^T \right) \left| \mathcal{J}\vec{a} \right| \vec{n} \right)_h \cdot \overleftrightarrow{\mathbf{F}}_{v,h} \right) \mathrm{d}S_\xi .
\end{aligned}
\tag{112}
$$

The following step is to replace the fluxes in (112) with (86) and (88),

$$
\begin{aligned}
\left(\left(\mathbf{W}^{\star,T} - \mathbf{W}^T \right) \left| \mathcal{J}\vec{a} \right| \vec{n} \right)_l \cdot \overleftrightarrow{\mathbf{F}}_l + \mathbf{W}_l^T \tilde{\overleftrightarrow{\mathbf{F}}}^{\star}_l \cdot \hat{n}_h = & \\
\frac{1}{2} \mathcal{P}_{hl} \left(\left(\mathcal{P}_{lh} \left(\mathbf{W}_l^T \right) - \mathbf{W}_h^T \right) \left| \mathcal{J}\vec{a} \right|^h \vec{n}^h \right) \cdot \overleftrightarrow{\mathbf{F}}_l & \\
- \mathbf{W}_l^T \mathcal{P}_{hl} \left(\frac{1}{2} |\mathcal{J}\vec{a}|^h \vec{n}^h \cdot \left(\overleftrightarrow{\mathbf{F}}_h + \mathcal{P}_{lh} \left(\overleftrightarrow{\mathbf{F}}_l \right) \right) + \beta \left| \mathcal{J}\vec{a} \right|^h \mathbf{e}_1 \left(\mu_h - \mathcal{P}_{lh} \left(\mu_l \right) \right) \right) . &
\end{aligned}
\tag{113}
$$

$$\begin{aligned}\left(\left(\mathbf{W}^{\star,T}-\mathbf{W}^{T}\right)\left|\mathcal{J}\vec{a}\right|\vec{n}\right)_h\cdot\vec{\tilde{\mathbf{F}}}_h+\mathbf{W}_h^T\vec{\tilde{\mathbf{F}}}_h^{\star}\cdot\hat{n}_l=&\\ \frac{1}{2}\left(\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)-\mathbf{W}_h^T\right)\left|\mathcal{J}\vec{a}\right|^h\vec{n}^h\cdot\vec{\tilde{\mathbf{F}}}_h&\\ +\mathbf{W}_h^T\frac{1}{2}|\mathcal{J}\vec{a}|^h\vec{n}^h\cdot\left(\vec{\tilde{\mathbf{F}}}_h+\mathcal{P}_{lh}\left(\vec{\tilde{\mathbf{F}}}_l\right)\right)+\beta\mathbf{W}_h^T\left|\mathcal{J}\vec{a}\right|^h\mathbf{e}_1\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right)&\\ =\frac{1}{2}\left|\mathcal{J}\vec{a}\right|^h\left(\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)\vec{\tilde{\mathbf{F}}}_h+\mathbf{W}_h^T\mathcal{P}_{lh}\left(\vec{\tilde{\mathbf{F}}}_l\right)\right)\cdot\vec{n}^h+\beta\left|\mathcal{J}\vec{a}\right|^h\mathbf{W}_h^T\mathbf{e}_1\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right).&\end{aligned}\tag{114}$$

We then apply property (79) to the second term in (113),

$$\begin{aligned}&\int_{f,N_l}\left(\frac{1}{2}\mathcal{P}_{hl}\left(\left(\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)-\mathbf{W}_h^T\right)\left|\mathcal{J}\vec{a}\right|^h\vec{n}^h\right)\cdot\vec{\tilde{\mathbf{F}}}_l\right.\\ &\left.-\mathbf{W}_l^T\mathcal{P}_{hl}\left(\frac{1}{2}|\mathcal{J}\vec{a}|^h\vec{n}^h\cdot\left(\vec{\tilde{\mathbf{F}}}_h+\mathcal{P}_{lh}\left(\vec{\tilde{\mathbf{F}}}_l\right)\right)+\beta\left|\mathcal{J}\vec{a}\right|^h\mathbf{e}_1\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right)\right)\right)\mathrm{d}S_\xi\\ &=\int_{f,N_h}\left(\frac{1}{2}\left(\left(\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)-\mathbf{W}_h^T\right)\left|\mathcal{J}\vec{a}\right|^h\vec{n}^h\right)\cdot\mathcal{P}_{lh}\left(\vec{\tilde{\mathbf{F}}}_l\right)\right.\\ &\left.-\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)\left(\frac{1}{2}|\mathcal{J}\vec{a}|^h\vec{n}^h\cdot\left(\vec{\tilde{\mathbf{F}}}_h+\mathcal{P}_{lh}\left(\vec{\tilde{\mathbf{F}}}_l\right)\right)+\beta\left|\mathcal{J}\vec{a}\right|^h\mathbf{e}_1\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right)\right)\right)\mathrm{d}S_\xi\\ &=-\int_{f,N_h}\left(\frac{1}{2}\left|\mathcal{J}\vec{a}\right|^h\left(\mathbf{W}_h^T\mathcal{P}_{lh}\left(\vec{\tilde{\mathbf{F}}}_l\right)+\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)\vec{\tilde{\mathbf{F}}}_h\right)\cdot\vec{n}^h+\beta\left|\mathcal{J}\vec{a}\right|^h\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)\mathbf{e}_1\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right)\right)\mathrm{d}S_\xi.\end{aligned}\tag{115}$$

We add the contributions from (114) and we show that the viscous terms are dissipative, with the dissipation being proportional to the square of the interface jump and the penalty parameter,

$$\begin{aligned}&+\int_{f,N_h}\left(\frac{1}{2}\left|\mathcal{J}\vec{a}\right|^h\left(\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)\vec{\tilde{\mathbf{F}}}_h+\mathbf{W}_h^T\mathcal{P}_{lh}\left(\vec{\tilde{\mathbf{F}}}_l\right)\right)\cdot\vec{n}^h+\beta\left|\mathcal{J}\vec{a}\right|^h\mathbf{W}_h^T\mathbf{e}_1\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right)\right)\mathrm{d}S_\xi\\ &-\int_{f,N_h}\left(\frac{1}{2}\left|\mathcal{J}\vec{a}\right|^h\left(\mathbf{W}_h^T\mathcal{P}_{lh}\left(\vec{\tilde{\mathbf{F}}}_l\right)+\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)\vec{\tilde{\mathbf{F}}}_h\right)\cdot\vec{n}^h+\beta\left|\mathcal{J}\vec{a}\right|^h\mathcal{P}_{lh}\left(\mathbf{W}_l^T\right)\mathbf{e}_1\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right)\right)\mathrm{d}S_\xi\\ &=\beta\int_{f,N_h}\left|\mathcal{J}\vec{a}\right|^h\left(\mu_h\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right)-\mathcal{P}_{lh}\left(\mu_l\right)\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right)\right)\mathrm{d}S_\xi\\ &=\beta\int_{f,N_h}\left|\mathcal{J}\vec{a}\right|^h\left(\mu_h-\mathcal{P}_{lh}\left(\mu_l\right)\right)^2\mathrm{d}S_\xi\geqslant 0,\end{aligned}\tag{116}$$

where we have used $\mathbf{W}_h^T\mathbf{e}_1=\mu_h$ from the definition of the entropy variables (33). An analogous procedure is followed for the Cahn–Hilliard terms,

$$\begin{aligned}\mathrm{IBT}_{ch}=&\frac{3}{2}\sigma\varepsilon\sum_{\substack{\text{interior}\\ \text{faces}}}\int_{f,N_l}\left(C_{t,l}\vec{\tilde{G}}_{c,l}^{\star}\cdot\hat{n}_l+\left(\left(C_t^{\star}-C_t\right)\left|\mathcal{J}\vec{a}\right|\vec{n}\right)_l\vec{G}_{c,l}\right)\mathrm{d}S_\xi\\ &+\frac{3}{2}\sigma\varepsilon\sum_{\substack{\text{interior}\\ \text{faces}}}\int_{f,N_h}\left(C_{t,h}\vec{\tilde{G}}_{c,h}^{\star}\cdot\hat{n}_h+\left(\left(C_t^{\star}-C_t\right)\left|\mathcal{J}\vec{a}\right|\vec{n}\right)_h\vec{G}_{c,h}\right)\mathrm{d}S_\xi\\ =&\frac{3}{4}\sigma\varepsilon\frac{\mathrm{d}}{\mathrm{d}t}\sum_{\substack{\text{interior}\\ \text{faces}}}\int_{f,N}\left|\mathcal{J}\vec{a}\right|^h\beta\left(C_h-\mathcal{P}_{lh}\left(C_l\right)\right)^2\mathrm{d}S_\xi\geqslant 0.\end{aligned}\tag{117}$$

When $\beta>0$, the integrals in IBT_v are positive (i.e. dissipative), and the integral in IBT_{ch}, which is not positive per se because of the time derivative, will be added to the entropy, since the integral argument is positive.

5.1.3. Physical boundary terms: free– and no–slip walls

We now address the stability of physical wall boundary terms (104). For the inviscid fluxes PBT_e, it remains the same as in the work of [10]. We use the entropy conserving approach with central fluxes (92),

$$\begin{aligned}\mathrm{PBT}_e &= \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(\mathbf{W}^T \overleftrightarrow{\mathbf{F}}_e^\star + \sum_{m=1}^{5} \mathbf{W}^T \left(\left(\overleftrightarrow{\mathbf{\Phi}}_m W_m \right)^\diamond - \overleftrightarrow{\mathbf{\Phi}}_m W_m \right) \right) \cdot \vec{n} \mathrm{d}S \\ &= \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(U_n \left(\frac{1}{2}\rho U_n^2 + P \right) + U_n \left(\mu C - \frac{1}{2}\rho U_n^2 - \mu C \right) - P U_n \right) \mathrm{d}S = 0. \end{aligned} \tag{118}$$

For the viscous terms, the free–slip wall boundary condition (93),

$$\begin{aligned}\mathrm{PBT}_v &= - \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(\mathbf{W}^T \left(\overleftrightarrow{\mathbf{F}}_v^\star - \overleftrightarrow{\mathbf{F}}_v \right) + \mathbf{W}^{\star,T} \overleftrightarrow{\mathbf{F}}_v \right) \cdot \vec{n} \mathrm{d}S \\ &= - \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(\mathbf{W}^T \left(0 - \overleftrightarrow{\mathbf{F}}_v \right) + \mathbf{W}^T \overleftrightarrow{\mathbf{F}}_v \right) \cdot \vec{n} \mathrm{d}S = 0, \end{aligned} \tag{119}$$

and for the no–slip wall boundary condition (94),

$$\begin{aligned}\mathrm{PBT}_v &= - \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(\mathbf{W}^T \left(\overleftrightarrow{\mathbf{F}}_v^\star - \overleftrightarrow{\mathbf{F}}_v \right) + \mathbf{W}^{\star,T} \overleftrightarrow{\mathbf{F}}_v \right) \cdot \vec{n} \mathrm{d}S \\ &= - \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(-\mu \vec{G}_\mu + \mu \vec{G}_\mu + \vec{U}\mathcal{S} - \vec{U}\mathcal{S} \right) \cdot \vec{n} \mathrm{d}S = 0, \end{aligned} \tag{120}$$

are neutrally stable. Lastly, the Cahn–Hilliard physical boundary condition for both free– and no–slip walls (93)–(94) gives,

$$\begin{aligned}\mathrm{PBT}_{ch} &= -\frac{3}{2}\sigma\varepsilon \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(C_t \vec{G}_c^\star + C_t^\star \vec{G}_c - C_t \vec{G}_c \right) \cdot \vec{n} \, \mathrm{d}S \\ &= - \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} \left(-C_t f_w'(C) + \frac{3}{2}\sigma\varepsilon \left(C_t \vec{G}_c - C_t \vec{G}_c \right) \right) \cdot \vec{n} \, \mathrm{d}S \\ &= \frac{\mathrm{d}}{\mathrm{d}t} \sum_{\substack{\text{boundary}\\ \text{faces}}} \int_{f,N} f_w(C) \, \mathrm{d}S, \end{aligned} \tag{121}$$

which represents the discrete version of the surface free–energy time derivative (42). Therefore the approximation of the physical boundary condition does not introduce any numerical dissipation.

5.2. Semi–discrete stability: summary

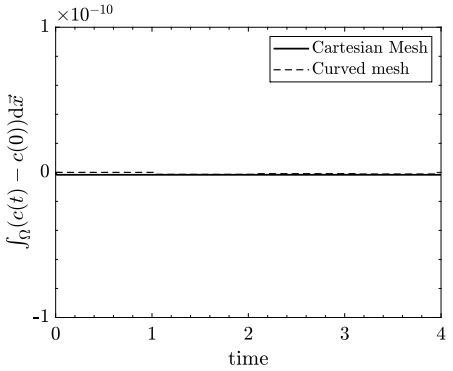
We constructed a p–non–conforming DG approximation of the iNS/CH system (72) whose properties mimic the continuous stability analysis of Sec. 2.1 for its spatial discretization. The entropy has been extended to consider the interface stabilization in the viscous terms, such that the inter–element discontinuities for the Cahn–Hilliard equation are penalized. This allows the entropy to incorporate the discontinuities and keep their size bounded without affecting the accuracy of the solution [9,10]. The form of (72) with this choice of the inviscid numerical and diamond fluxes, the Cahn–Hilliard interface stabilization and the suitable boundary conditions leads to the discrete entropy plus the surface free–energy to remain bounded by the initial condition.

6. Numerical results

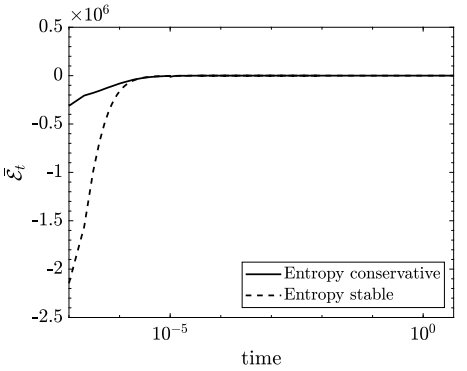
In this section we conduct three numerical experiments to showcase the capabilities of the numerical scheme. The first part comprises of a test for the entropy stability and mass conservation by marching the solution in time from a random initial condition and a test for freestream preservation. We verify that the scheme is conserving, entropy–stable and freestream preserving for Cartesian and general curvilinear meshes. The second case is that of a rising bubble. This test case is used to assess the accuracy of the scheme, establish that the interface tracking criterion is effective and quantify the benefits of the polynomial adaptation method. The third case is that of a breaking–dam, or the collapse of a column of liquid as referred in parts of the literature. For this test case, we compare the results obtained from this scheme against other numerical and experimental results.

Table 1
Physical parameters for the random initial condition test problem.

ρ_1	ρ_2 (kg/m^3)	η_1	η_2 (Pa·s)	ε (m)	t_{CH} (s)	c_0^2 (m/s^2)^2	σ (N/m)
10^3	1.0	10^{-3}	10^{-4}	$7.5 \cdot 10^{-1}$	10.0	10^2	1.0



(a) Phase conservation for the random initial condition test case.



(b) Entropy rate for the random initial condition test case.

Fig. 2. Numerical results for the random initial condition test case for a total time of $t = 4$ s.

6.1. Numerical verification

The aim of the following tests is to establish the stability, primary conservation and freestream preservation characteristics of the p–non–conforming scheme on curvilinear meshes. The baseline DGSEM has been validated in the work of [10]. We conduct a random initial condition test to verify the primary conservation and entropy–stability characteristics of the scheme and a freestream preservation test on a curvilinear mesh to verify that the conditions detailed in Section 3.2 are sufficient.

We use a random initial condition from the interval $[0, 1]$ for the concentration and the interval $[-1, 1]$ m/s for the velocities and $[-1, 1]$ Pa for the pressure. The time marching is performed using a low–storage third order Runge–Kutta scheme with $\Delta t = 10^{-7}$ s. The parameters of the simulation are given in Table 1.

We perform two series of tests, one using a Cartesian mesh and one with a mesh containing curved elements. The curved mesh has a geometrical order of approximation of $N_{geo} = 2$ and its construction is detailed in Appendix B.1. Both meshes are discretized from 4×4×4 elements defined in $\Omega = [0, 1]^3$ m. We also use periodic boundary conditions.

For the Cartesian mesh test case we use a random polynomial order for each element from the interval $N_{Cartesian} \in [2, 5]$. The conditions for having a watertight mesh are satisfied by definition [65]. For the curved mesh we use a random polynomial order from the interval $N_{curved} \in [4, 6]$ and since $N_{geo} = 2$ the geometrical conditions for three–dimensional non–conforming elements detailed in Section 3.2 are satisfied. In both test cases the polynomial order of each element is updated in a random fashion every 1000 time steps.

In Fig. 2 we present the results obtained for the random initial condition test. In Fig. 2a we show that the scheme is primary quantity preserving for Cartesian and curved meshes, as we measure the total phase using

$$\text{Total phase error} = \int_{\Omega} (c - c(0)) \, \mathrm{d}\vec{x} \tag{122}$$

The error of the total phase is of the order of 10^{-12}. Also, in Fig. 2b we show the rate of change of entropy (97) for the Cartesian mesh. Initially it is negative and as the solution is smoothed it approaches a negative value close to zero. The same trend has been observed for the test on the curvilinear mesh. We present results for an entropy–conservative and the entropy–stable version of the scheme with additional interface dissipation. These results are in accordance with the work presented in [10] for the p–conforming version of the scheme.

To assess the freestream preservation characteristics of the scheme we use the curvilinear mesh from the random initial condition test. The polynomial order of each element is a random value from the interval $N_{curved} \in [4, 6]$ which is updated every 1000 time steps in a random fashion. The initial condition is set to $\left(c, \sqrt{\rho}u, \sqrt{\rho}v, \sqrt{\rho}w, p\right) = (1, 1, 0, 0, 1)$. The solution is marched in time using the RK3 scheme with $\Delta t = 10^{-7}$ s until a final time of $t = 1$ s. To measure the error for a state variable ψ we use the $\mathbb{L}^2$ norm which is computed as

Table 2
Results for the freestream preservation test.

	$\|\psi_t(t=0)\|_{\mathbb{L}^2}$	$\|\psi(t=1)-\psi(t=0)\|_{\mathbb{L}^2}$
c	$2.029\cdot10^{-11}$	$1.201\cdot10^{-14}$
$\sqrt{\rho}u$	$3.787\cdot10^{-13}$	$1.286\cdot10^{-14}$
$\sqrt{\rho}v$	$3.793\cdot10^{-13}$	$1.261\cdot10^{-14}$
$\sqrt{\rho}w$	$3.221\cdot10^{-13}$	$1.301\cdot10^{-14}$
p	$1.098\cdot10^{-13}$	$1.274\cdot10^{-14}$

Table 3
Physical parameters of the rising bubble test problem.

ρ_1	ρ_2 (kg/m^3)	η_1	η_2 (Pa·s)	ε (m)	t_{CH} (s)	c_0^2 (m/s^2)^2	σ (N/m)	g (m/s^2)
10^3	1.0	10.0	10^{-1}	$4\cdot10^{-2}$	10^4	10^3	$19.6\cdot10^{-1}$	$9.8\cdot10^{-1}$

$$\|\psi\|_{\mathbb{L}^2}=\sqrt{\sum_e\langle\mathcal{J}\psi,\psi\rangle_{E,N}}. \tag{123}$$

The results obtained for the freestream preservation are summarized in Table 2. We present the $\mathbb{L}^2$ norm of the initial rate of change and the $\mathbb{L}^2$ norm of the error at a time instant of $t=1$ for each state variable. The results in Table 2 illustrate that the scheme is freestream preserving for a general curvilinear mesh if the conditions stated in Section 3.2 are satisfied.

6.2. Rising bubble

The rising bubble is a widespread benchmark test case used to assess the characteristics and accuracy of multiphase flow solvers. This test concerns surfacing of a bubble of light fluid that has been submerged within a heavier medium. The bubble is centred at $(0.5, 0.5)$ and a diameter of 0.5 m. The initial condition is given by

$$c(x,y;0)=1-\frac{1}{2}\left(\tanh\left(-\frac{2\,(r-0.25)}{\varepsilon}\right)+1\right),\quad r=\sqrt{(x-0.5)^2+(y-0.5)^2}. \tag{124}$$

The domain is defined in $(x,y)\in[0,1]\times[0,2]\,\mathrm{m}^2$ and we use a Cartesian mesh with an element size of $h=2^{-4}$ m. The boundary conditions in x=0 and x=1 are free–slip walls and no–slip walls in y=0 and y=2. The time marching is conducted using an explicit third order low–storage explicit Runge-Kutta scheme with a timestep of $\Delta t=7.2\cdot10^{-7}$ s. The timestep remains constant and the CFL number remains close to 0.2, similar to the analysis presented in [10].
The physical parameters of the simulation which have been derived from [81] and have been adapted for this scheme in [10] are presented in Table 3. In this work we focus only on test case 2 of [81] which is the most numerically challenging version of the rising bubble.

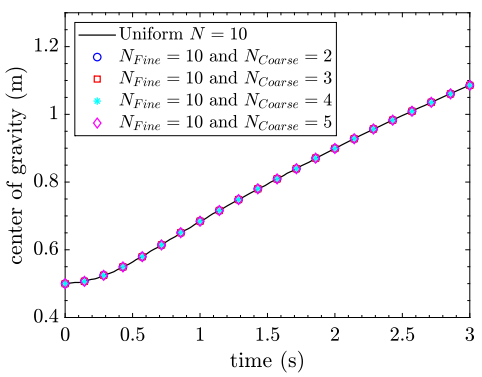
As presented in [10], with such a coarse grid it is necessary to use a high polynomial order of $N=10$ to resolve adequately the interface region and obtain the expected results. However, a polynomial order of $N=10$ is not required throughout the entire domain and thus we present results which have been derived using a polynomial order of $N_{\text{Coarse}}\in[2,5]$ for the elements situated outside the interface between the two phases.
The quantities of interest in this test case are the position of the centre of gravity X_c of the bubble (the phase with $c=0$) as it emerges and the associated rising velocity V_c of the centre of gravity. Those quantities are computed as

$$X_c=\frac{1}{A}\sum_e\langle\mathcal{J}(1-c),X\rangle_{E,N},\quad V_c=\frac{1}{A}\sum_e\langle\mathcal{J}(1-c,U\rangle_{E,N},\quad A=\sum_e\langle\mathcal{J}(1-c),1\rangle_{E,N}, \tag{125}$$

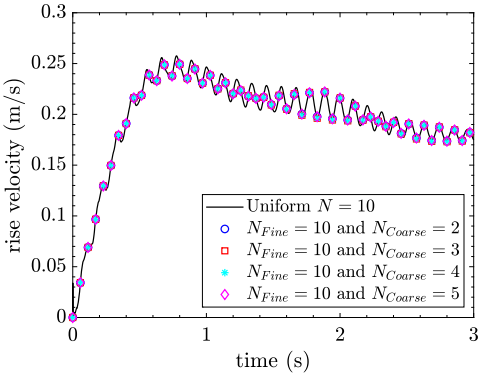
where X is the vertical position, U is the vertical velocity, $\mathcal{J}$ is the Jacobian and A is the area of the phase with $c=0$.

The results for the rising bubble test case of this scheme with uniform polynomial order have been compared and validated against other methods for multiphase flow simulation [81,82] in [10]. We compare the results obtained using the dynamically adapted p–non–conforming and p–conforming entropy–stable versions of the scheme to show that the same accuracy can be achieved with lower computational cost. In Fig. 3a we present a comparison between the uniform and adapted solution for the location of the centre of gravity. The adaptation method does not incur any additional error and the adapted solution matches the p–conforming version of the scheme. This is also the case for the rise velocity results which are presented in Fig. 3b. The presence of the oscillations in Fig. 3b has also been discussed in [10] and it could be addressed by further increasing the artificial compressibility parameter. The variation of the degrees of freedom for each case is presented in Fig. 4. The lines for the different coarse polynomial order show a similar behaviour and the distinction lies in the number of degrees of freedom. The use of $N_{\text{Coarse}}=2$ is the most advantageous as it maintains the accuracy of the solution while it reduces the degrees of freedom.

The simulations have been conducted using two Intel Xeon Gold 6230 CPUs on a single node and parallelisation though 40 OpenMP threads. The details about initial, final and average degrees of freedom are presented in Table 4. In addition we



(a) Results for the evolution of the center of gravity of the rising bubble.



(b) Results for the rising velocity of the center of gravity of the bubble.

Fig. 3. Numerical results for the rising bubble test case for different coarse level polynomial order and comparison with the uniform polynomial order solution.

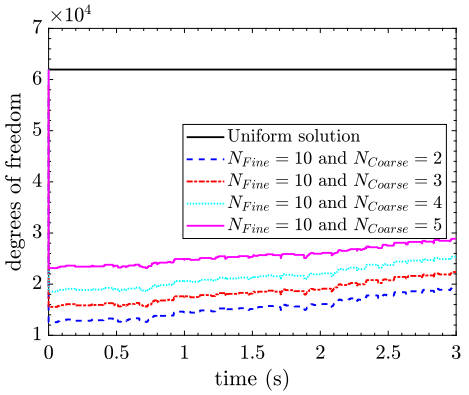


Fig. 4. Evolution of the degrees of freedom for the rising bubble test case for different coarse level polynomial order.

Table 4
Wall time and its percentage reduction for the rising bubble test case for the uniform and the adapted solutions.

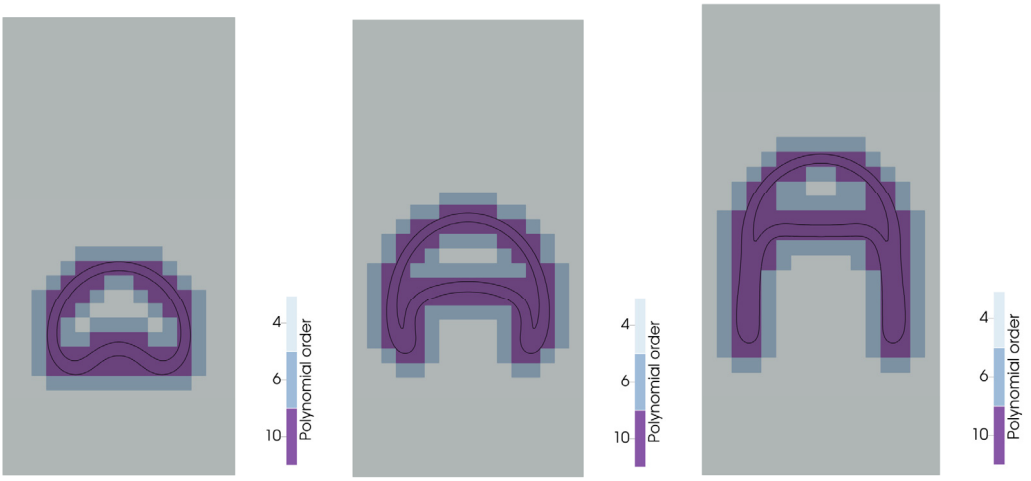
	Uniform	Adapted $N_{\text{Coarse}} = 2$	Adapted $N_{\text{Coarse}} = 5$
Elapsed time (s)	334,695	198,181 (−41%)	244,178 (−27%)
Adaptation time (s)	-	10	28

present the wall time taken to reach a final time of $t = 3$. All the numerical experiments presented (uniform, adapted) have been carried out using the same framework. As presented, the reduction in wall time does not scale as the reduction in the degrees of freedom. This can be attributed to the fact that the number of the interfaces of the elements does not change, the interpolation operation and the calculation of the mortars and due to the layout of the data. For the case of $N_{\text{Coarse}} = 2$ the reduction is approximately 41%.

A visualisation of the distribution of the polynomial order across the domain is presented in Fig. 5. The boundaries of the interface ($0.1 \leq c \leq 0.9$) are depicted with a black continuous line.

6.3. Dam break

This is a classic test case for multiphase flow solvers which has been examined experimentally [83–85] and with numerous multiphase flow modelling methodologies [86–91]. In this numerical experiment we present only the results obtained with the p–non–conforming scheme and we provide an estimation of the cost for the conforming scheme due to limited computational resources. We follow the three–dimensional test case as described in [86,90]. A column of a heavier medium with length, width, and height a is placed on the one side of a Cartesian domain with dimensions of $5a \times 1.25a \times 1a$ and the rest of the domain is occupied by the lighter fluid. Then, under the influence of gravity the column of the heavier phase

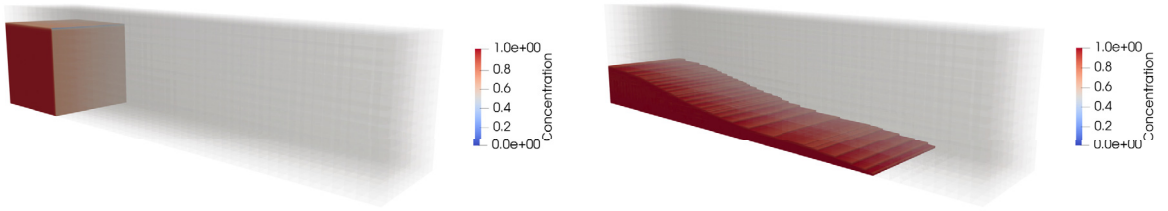


(a) $t = 1$ (b) $t = 2$ (c) $t = 3$

Fig. 5. Distribution of polynomial order across the domain for the rising bubble test case and highlights the refinement around the interface region. In solid continuous black line we denote the boundaries of the interface ($0.1 \leq c \leq 0.9$). (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

Table 5
Physical parameters for the dam breaking test case.

ρ_1	ρ_2 (kg/m^3)	η_1	η_2 (Pa·s)	ε (m)	t_{CH} (s)	c_0^2 (m/s^2)^2	σ (N/m)	g (m/s^2)
10^3	1.0	$5 \cdot 10^{-1}$	$5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	$4 \cdot 10^2$	10^2	$75.5 \cdot 10^{-3}$	9.8



(a) $t = 0$ (b) $t = 2$

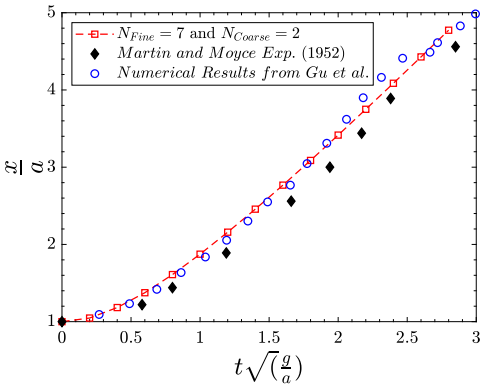
Fig. 6. Instances of the dam break test case that show the initial condition and the flow configuration at $t = 2$ s.

collapses and spreads until it reaches the other end of the domain.
The initial condition is taken by the work of [91] and it has the form

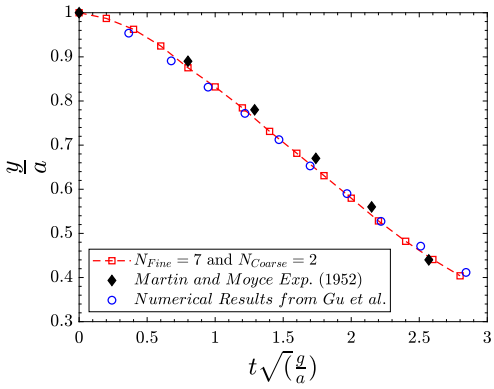
$$c(x,y,z) = \begin{cases} 0.5 - 0.5\tanh\left(2\frac{y-a}{\epsilon}\right) & \text{if } x \leq (a-r) \text{ and } y \geq (a-r) \\ 0.5 - 0.5\tanh\left(2\frac{x-a}{\epsilon}\right) & \text{if } x > (a-r) \text{ and } y < (a-r) \\ 0.5 + 0.5\tanh\left(2\frac{r-\sqrt{(x-(a-r))^2+(y-(a-r))^2}}{\epsilon}\right) & \text{if } x \geq (a-r) \text{ and } y \geq (a-r) \\ 1 & \text{otherwise.} \end{cases} \tag{126}$$

In this work, the parameter a has been chosen to be $a = 0.146$ m, $r = 0.04$ m and ϵ is the interface width parameter. The domain has been discretized with $50 \times 12 \times 10$ elements. All the boundaries are treated with free–slip boundary conditions. The polynomial order for the region around the interface in this test is $N_{\text{Fine}} = 7$ and for the rest of the domain is $N_{\text{Coarse}} = 2$. The parameters of the simulation are presented in Table 5. The artificial compressibility parameter c_0^2 has been defined to $c_0^2 = 100$ initially in a two–dimensional setting. The time marching has been performed with a low–storage third order Runge–Kutta scheme [92] with $\Delta t = 10^{-5}$ s. The initial condition and an intermediate solution for $t = 2.6$ are presented in Fig. 6.

The quantities of interest are the evolution of the tip of the liquid column as it spreads towards the rightmost boundary of the domain and the evolution of the position of the leftmost point of the heavier substance. The results presented in Fig. 7 are compared with the experimental results from [83] and the numerical results from [86]. The numerical results



(a) Results for the evolution of the tip of the wave.



(b) Results for the evolution of the highest point of the column of the heavier phase.

Fig. 7. Comparison with experimental results from Moyce et al. [83] and numerical results from Gu et al. [86] for the dam break test case.

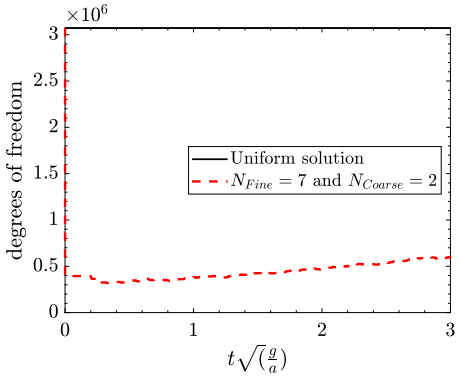


Fig. 8. Evolution of the degrees of freedom for the dam break test case.

Table 6
Wall time for the breaking dam simulation and the adaptation process and percentage reduction of the adapted solution.

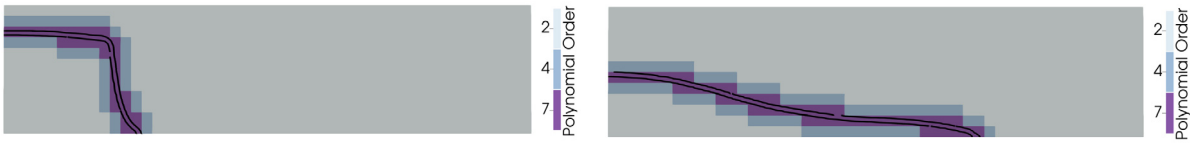
	Uniform	p–non–conforming
Elapsed time (s)	941,385	449,988 (−52%)
Adaptation time (s)	-	280

from [86] have been derived using a level–set method, which is different to the diffuse interface approach that we have adopted in this work, and thus some differences are expected.

The results presented in Fig. 7a are in a good agreement between the results from the proposed numerical scheme and the reference experimental results [83]. The small discrepancy presented for the displacement of the tip in Fig. 7a is common across significant part of the literature [88,91,86,93,94] and is attributed to the characteristics of the experiment. On the contrary the evolution of the highest point is in very good agreement between the numerical and experimental results. There is also good agreement between our work and that from Gu et al. [86], although some differences are observed in Fig. 7a for the solution after $t = 2\,s$.

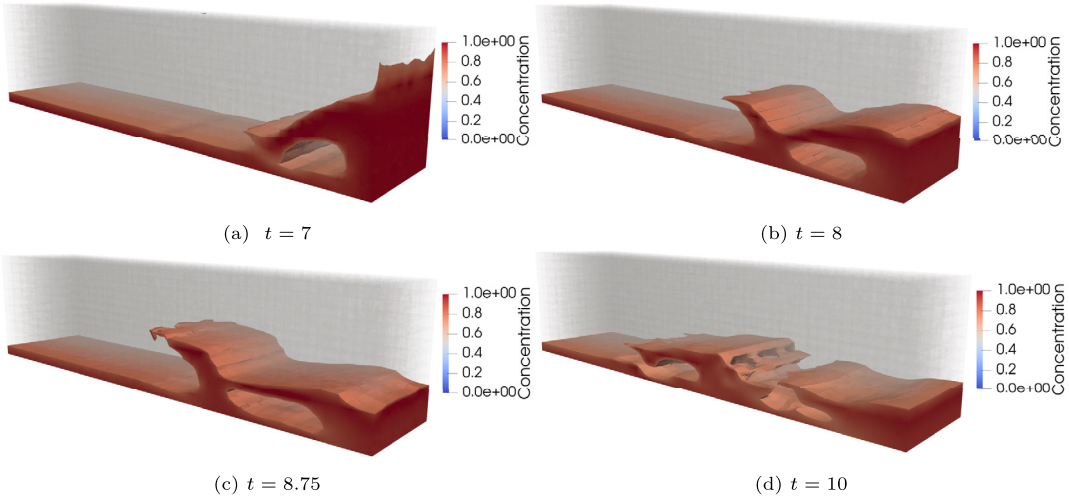
The number of degrees of freedom during the simulation is presented in Fig. 8. The reduction of degrees of freedom varies from $\times 4.3$ to $\times 9.5$ times with respect to the uniform solution with $N = 7$. The associated reduction in the computational time of the adapted solution with respect to the uniform solution is presented in Table 6. The computations have been performed on two Intel Xeon Gold 6248 CPUs using 40 OpenMP threads. The computational time for the p–conforming case has been extrapolated using the computational time for 1000 time steps. The wall–time reduction achieved for a simulation of a total time of $t = 5$ s is approximately 52%.

The distribution of the polynomial order in the domain is presented in Fig. 9. As shown, the fine polynomial order $N_{\text{Fine}} = 7$ follows the position of the interface. The visualization of the flow after the impact with the rightmost boundary is presented in Fig. 10. The structure is typical and similar to the one presented in [37], as the wave folds and impacts the water bed.



(a) $t = 0.5$ (b) $t = 2$

Fig. 9. Distribution of the polynomial order across the domain that shows the refinement near the interface region. In solid continuous like we denote the boundaries of the interface ($0.1 \leq c \leq 0.9$).



(a) $t = 7$ (b) $t = 8$

(c) $t = 8.75$ (d) $t = 10$

Fig. 10. Visualisation of the flow structure after the heavy fluid has reached and impacted the rightmost boundary of the domain.

7. Conclusions

In this paper we have extended the discontinuous Galerkin scheme of [10] to be entropy–stable when transitioning to p–non–conforming meshes. The incompressible Navier–Stokes system with Gauss–Lobatto points, the summation–by–parts simultaneous–approximation–term property, suitable numerical fluxes and the mortar method mimics the properties of the continuous analysis. Since we focus on multiphase flow problems of unsteady nature, we have developed and presented a dynamic adaptation methodology that exploits the characteristics of the DGSEM and adapts the polynomial order automatically without the need to predetermine an adaptation interval.

We have verified the total phase conservation characteristics of the scheme through a demanding random initial condition test case. Furthermore, two numerical experiments have been conducted to validate the accuracy and the performance of the p–non–conforming solver. The first one, is the typical test of a rising bubble with a high density ratio ($\rho_2/\rho_1 = 1000$), which shows that savings of approximately 40% in the computational time can be achieved for the same accuracy. The second test, is that of the collapse of a liquid column, which showcases the potential of p–adaptation for multiphase flow simulations as we attain a reduction of 51% in the computational time. In general, we have explored the potential of p–adaptation for multiphase flows and have shown that it is effective at reducing the cost of multiphase flow simulations while retaining the desired accuracy.

CRediT authorship contribution statement

Gerasimos Ntoukas: Conceptualization, Formal analysis, Methodology, Writing – review & editing. **Juan Manzanero:** Conceptualization, Formal analysis, Methodology, Writing – review & editing. **Gonzalo Rubio:** Conceptualization, Formal analysis, Methodology, Writing – review & editing. **Eusebio Valero:** Conceptualization, Formal analysis, Methodology, Resources, Writing – review & editing. **Esteban Ferrer:** Conceptualization, Formal analysis, Methodology, Resources, Writing – review & editing.

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

Gerasimos Ntoukas and Esteban Ferrer would like to thank the European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie grant agreement No. 813605 for the ASIMIA ITN-EID project. Gonzalo Rubio and Eusebio Valero acknowledge the funding received by the grant Ministry of Innovation of Spain, project SIMOPAIR (Project No. REF:RTI2018-097075-B-100) funded by MCIN/AEI/ 10.13039/501100011033 and by ERDF A way of making Europe.

Appendix A. Stability analysis

A.1. Semi–discrete stability analysis

In this part we provide the details for the entropy stability proof of the volume terms of the spatial discretization (72) which has been introduced in [10]. Even for p–non–conforming approximations, these steps are similar to the original analysis. The entropy analysis reproduces the continuous analysis steps discretely. This is done by making the appropriate choice for the test functions which contract the system into a single equation.

First, we follow [7] by taking the time derivative of (72d),

$$\left\langle \mathcal{J}\vec{G}_{c,t}, \vec{\varphi}_c \right\rangle_{E,N} = \int_{\partial e,N} \left(C_t^\star - C_t\right) \vec{\varphi}_c \cdot \vec{n}\, \mathrm{d}S + \left\langle \vec{\tilde{\varphi}}_c, \vec{\nabla}_\xi C_t \right\rangle_{E,N}, \tag{A.1}$$

and then substitute the test functions, which are chosen to be, $\boldsymbol{\varphi}_q = \mathbf{W}$, and $\overleftrightarrow{\boldsymbol{\varphi}}_g = \overleftrightarrow{\mathbf{F}}_v$, $\varphi_\mu = C_t$, and $\vec{\varphi}_c = \frac{3}{2}\sigma\varepsilon\vec{G}_c$. The system of equations takes the form

$$\begin{aligned}
\left\langle \mathcal{J}\underline{\mathbf{M}}\mathbf{Q}_t, \mathbf{W} \right\rangle_{E,N} &+ \int_{\partial e,N} \mathbf{W}^T \left(\overleftrightarrow{\mathbf{F}}_e^\star - \overleftrightarrow{\mathbf{F}}_e + \sum_{m=1}^{5} \left(\overleftrightarrow{\boldsymbol{\Phi}}_m W_m \right)^\diamond - \overleftrightarrow{\boldsymbol{\Phi}}_m W_m \right) \cdot \vec{n}\, \mathrm{d}S \\
&+ \left\langle \mathbf{W}, \vec{\nabla}_\xi \cdot \overleftrightarrow{\tilde{\mathbf{F}}}_e \right\rangle_{E,N} + \sum_{m=1}^{5} \left\langle \mathbf{W}, \overleftrightarrow{\tilde{\boldsymbol{\Phi}}}_m \cdot \vec{\nabla}_\xi W_m \right\rangle_{E,N} \\
&= \int_{\partial e,N} \mathbf{W}^T \overleftrightarrow{\mathbf{F}}_v^\star \cdot \vec{n}\, \mathrm{d}S - \left\langle \overleftrightarrow{\tilde{\mathbf{F}}}_v, \vec{\nabla}_\xi \mathbf{W} \right\rangle_{E,N},
\end{aligned} \tag{A.2a}$$

$$\left\langle \mathcal{J}\overleftrightarrow{\mathbf{G}}, \overleftrightarrow{\mathbf{F}}_v \right\rangle_{E,N} = \int_{\partial e,N} \left(\mathbf{W}^{\star,T} - \mathbf{W}^T \right) \left(\overleftrightarrow{\mathbf{F}}_v \cdot \vec{n} \right) \mathrm{d}S + \left\langle \overleftrightarrow{\tilde{\mathbf{F}}}_v, \vec{\nabla}_\xi \mathbf{W} \right\rangle_{E,N}, \tag{A.2b}$$

$$\left\langle \mathcal{J}\mu, C_t \right\rangle_{E,N} = \left\langle \mathcal{J}F_0', C_t \right\rangle_{E,N} - \int_{\partial e,N} \frac{3}{2}\sigma\varepsilon C_t \vec{G}_c^\star \cdot \vec{n}\, \mathrm{d}S + \left\langle \frac{3}{2}\sigma\varepsilon\vec{\tilde{G}}_c, \vec{\nabla}_\xi C_t \right\rangle_{E,N}, \tag{A.2c}$$

$$\frac{3}{2}\sigma\varepsilon \left\langle \mathcal{J}\vec{G}_{c,t}, \vec{G}_c \right\rangle_{E,N} = \int_{\partial e,N} \frac{3}{2}\sigma\varepsilon \left(C_t^\star - C_t\right) \vec{G}_c \cdot \vec{n}\, \mathrm{d}S + \left\langle \frac{3}{2}\sigma\varepsilon\vec{\tilde{G}}_c, \vec{\nabla}_\xi C_t \right\rangle_{E,N}. \tag{A.2d}$$

Next, the last term in (A.2a) is replaced by the last term in (A.2b), and the last term in (A.2c) by the last term in (A.2d),

$$\begin{aligned}
\left\langle \mathcal{J}\underline{\mathbf{M}}\mathbf{Q}_t, \mathbf{W} \right\rangle_{E,N} &+ \int_{\partial e,N} \mathbf{W}^T \left(\overleftrightarrow{\mathbf{F}}_e^\star - \overleftrightarrow{\mathbf{F}}_e + \sum_{m=1}^{5} \left(\overleftrightarrow{\boldsymbol{\Phi}}_m W_m \right)^\diamond - \overleftrightarrow{\boldsymbol{\Phi}}_m W_m \right) \cdot \vec{n}\, \mathrm{d}S \\
&+ \left\langle \mathbf{W}, \vec{\nabla}_\xi \cdot \overleftrightarrow{\tilde{\mathbf{F}}}_e \right\rangle_{E,N} + \sum_{m=1}^{5} \left\langle \mathbf{W}, \overleftrightarrow{\tilde{\boldsymbol{\Phi}}}_m \cdot \vec{\nabla}_\xi W_m \right\rangle_{E,N} \\
&= \int_{\partial e,N} \left(\mathbf{W}^T \overleftrightarrow{\mathbf{F}}_v^\star + \mathbf{W}^{\star,T} \overleftrightarrow{\mathbf{F}}_v - \mathbf{W}^T \overleftrightarrow{\mathbf{F}}_v \right) \cdot \vec{n}\, \mathrm{d}S - \left\langle \mathcal{J}\overleftrightarrow{\mathbf{G}}, \overleftrightarrow{\mathbf{F}}_v \right\rangle_{E,N},
\end{aligned} \tag{A.3a}$$

$$\begin{aligned}
\left\langle \mathcal{J}\mu, C_t \right\rangle_{E,N} &= \left\langle \mathcal{J}F_0', C_t \right\rangle_{E,N} + \frac{3}{2}\sigma\varepsilon \left\langle \mathcal{J}\vec{G}_{c,t}, \vec{G}_c \right\rangle_{E,N} \\
&- \int_{\partial e,N} \frac{3}{2}\sigma\varepsilon \left(C_t \vec{G}_c^\star + C_t^\star \vec{G}_c - C_t \vec{G}_c \right) \cdot \vec{n}\, \mathrm{d}S.
\end{aligned} \tag{A.3b}$$

Throughout this process, we assume that time integration is exact and thus we can use the chain rule on (A.3b),

$$\begin{aligned}\left\langle \mathcal{J}F_0', C_t\right\rangle_{E,N} + \frac{3}{2}\sigma\varepsilon\left\langle \mathcal{J}\vec{G}_{c,t}, \vec{G}_c\right\rangle_{E,N} &= \left\langle \mathcal{J}\left(F_0 + \frac{3}{4}\sigma\varepsilon|\vec{G}_c|^2\right)_t, 1\right\rangle_{E,N} \\ &= \left\langle \mathcal{J}\mathcal{F}_t, 1\right\rangle_{E,N},\end{aligned} \tag{A.4}$$

to obtain the time derivative of the discrete approximation of the free–energy (6).

The next step is to examine each term in (A.3). More specifically the time derivative coefficients in Appendix A.1.1, the inviscid volume terms in Appendix A.1.2 and the viscous volume terms in Appendix A.1.3.

A.1.1. Time derivative coefficients

The discrete inner product of the time derivatives found in (96a) is transformed to

$$\left\langle \mathcal{J}\underline{\mathbf{M}}\mathbf{Q}_t, \mathbf{W}\right\rangle_{E,N} = \left\langle \mathcal{J}C_t, \mu\right\rangle_{E,N} + \left\langle \mathcal{J}\left(\sqrt{\rho}\vec{U}\right)_t, \sqrt{\rho}\vec{U}\right\rangle_{E,N} + \frac{1}{\rho_0 c_0^2}\left\langle \mathcal{J}P_t, P\right\rangle_{E,N}, \tag{A.5}$$

into which we insert the discrete approximation of $\langle \mathcal{J}C_t, \mu\rangle_{E,N}$ found in (A.3b), and apply the chain rule in time for the kinetic and artificial compressibility terms,

$$\begin{aligned}\left\langle \mathcal{J}\underline{\mathbf{M}}\mathbf{Q}_t, \mathbf{W}\right\rangle_{E,N} &= \left\langle \mathcal{J}\mathcal{F}_t, 1\right\rangle_{E,N} - \int_{\partial e,N}\frac{3}{2}\sigma\varepsilon\left(C_t\vec{G}_c^{\star} + C_t^{\star}\vec{G}_c - C_t\vec{G}_c\right)\cdot\vec{n}\,\mathrm{d}S \\ &\quad + \left\langle \mathcal{J}\left(\frac{1}{2}\rho V_{tot}^2\right)_t, 1\right\rangle_{E,N} + \left\langle \mathcal{J}\left(\frac{P^2}{2\rho_0 c_0^2}\right)_t, 1\right\rangle_{E,N} \\ &= \left\langle \mathcal{J}\left(\mathcal{F} + \mathcal{K} + \mathcal{E}_{\mathrm{AC}}\right)_t, 1\right\rangle_{E,N} \\ &\quad - \int_{\partial e,N}\frac{3}{2}\sigma\varepsilon\left(C_t\vec{G}_c^{\star} + C_t^{\star}\vec{G}_c - C_t\vec{G}_c\right)\cdot\vec{n}\,\mathrm{d}S \\ &= \left\langle \mathcal{J}\mathcal{E}_t, 1\right\rangle_{E,N} - \int_{\partial e,N}\frac{3}{2}\sigma\varepsilon\left(C_t\vec{G}_c^{\star} + C_t^{\star}\vec{G}_c - C_t\vec{G}_c\right)\cdot\vec{n}\,\mathrm{d}S.\end{aligned} \tag{A.6}$$

Eq. (A.6) is the discrete version of (35). As a result, we get the discrete version of (35) for the entropy,

$$\mathcal{E} = \mathcal{F} + \mathcal{K} + \mathcal{E}_{\mathrm{AC}} = F_0 + \frac{3}{4}\sigma\varepsilon|\vec{G}_c|^2 + \frac{1}{2}\rho V_{tot}^2 + \frac{P^2}{2\rho_0 c_0^2}, \tag{A.7}$$

and for the time entropy flux $\vec{F}_t^{\mathcal{E}}$, which appears as the argument of the surface integral in (97).

A.1.2. Inviscid volume terms

The next step is to show that the inviscid volume terms are contracted into a boundary entropy flux as in the continuous analysis (27),

$$\left\langle \mathbf{W}, \vec{\nabla}_\xi\cdot\overleftrightarrow{\tilde{\mathbf{F}}}_e\right\rangle_{E,N} + \sum_{m=1}^{5}\left\langle \mathbf{W}, \overleftrightarrow{\tilde{\mathbf{\Phi}}}_m\cdot\vec{\nabla}_\xi W_m\right\rangle_{E,N} = \int_{\partial e,N}\vec{F}^{\mathcal{E}}\cdot\vec{n}\,\mathrm{d}S. \tag{A.8}$$

Two properties were used in the continuous analysis: the integration by parts (which holds discretely by the discrete Gauss law (67)), and the relationship between inviscid fluxes and non–conservative terms (29), which also holds for contravariant components,

$$\mathbf{e}_m^T\overleftrightarrow{\tilde{\mathbf{F}}}_e = \mathbf{e}_m^T\left(\mathcal{M}^T\overleftrightarrow{\mathbf{F}}_e\right) = \boldsymbol{M}^T\left(\mathbf{e}_m^T\overleftrightarrow{\mathbf{F}}_e\right) = \boldsymbol{M}^T\left(\mathbf{W}^T\overleftrightarrow{\mathbf{\Phi}}_m\right) = \mathbf{W}^T\left(\mathcal{M}^T\overleftrightarrow{\mathbf{\Phi}}_m\right) = \mathbf{W}^T\overleftrightarrow{\tilde{\mathbf{\Phi}}}_m. \tag{A.9}$$

Eq. (A.9) together with the discrete Gauss law (67) is sufficient to prove (A.8). Initially, we apply the discrete Gauss law (67) in the first term of (A.8) and then express the scalar product of the resulting integral as the sum of its components

$$\left\langle \overleftrightarrow{\tilde{\mathbf{F}}}_e, \vec{\nabla}_\xi\mathbf{W}\right\rangle_{E,N} = \sum_{m=1}^{5}\left\langle \mathbf{e}_m^T\overleftrightarrow{\tilde{\mathbf{F}}}_e, \vec{\nabla}_\xi W_m\right\rangle_{E,N}. \tag{A.10}$$

Thus, (A.8) becomes

$$\begin{aligned}\left\langle \mathbf{W}, \vec{\nabla}_\xi \cdot \overleftrightarrow{\tilde{\mathbf{F}}}_e \right\rangle_{E,N} &= \int_{\partial e,N} \mathbf{W}^T \overleftrightarrow{\mathbf{F}}_e \cdot \vec{n}\, \mathrm{d}S - \left\langle \overleftrightarrow{\tilde{\mathbf{F}}}_e, \vec{\nabla}_\xi \mathbf{W} \right\rangle_{E,N} \\ &= \int_{\partial e,N} \mathbf{W}^T \overleftrightarrow{\mathbf{F}}_e \cdot \vec{n}\, \mathrm{d}S - \sum_{m=1}^{5} \left\langle \mathbf{e}_m^T \overleftrightarrow{\tilde{\mathbf{F}}}_e, \vec{\nabla}_\xi W_m \right\rangle_{E,N} \\ &= \int_{\partial e,N} \vec{F}^{\mathcal{E}} \cdot \vec{n}\, \mathrm{d}S - \sum_{m=1}^{5} \left\langle \mathbf{W}^T \overleftrightarrow{\tilde{\mathbf{\Phi}}}_m, \vec{\nabla}_\xi W_m \right\rangle_{E,N}.\end{aligned} \tag{A.11}$$

A.1.3. Viscous volume terms

Analogously to (28), viscous volume terms are dissipative if

$$\left\langle \mathcal{J}\overleftrightarrow{\mathbf{G}}, \overleftrightarrow{\mathbf{F}}_v\left(\overleftrightarrow{\mathbf{G}}\right) \right\rangle_{E,N} \geqslant 0. \tag{A.12}$$

Thus, we closely follow the continuous steps,

$$\begin{aligned}\left\langle \mathcal{J}\overleftrightarrow{\mathbf{G}}, \overleftrightarrow{\mathbf{F}}_v \right\rangle_{E,N} &= M_0 \left\langle \mathcal{J}\vec{G}_\mu, \vec{G}_\mu \right\rangle_{E,N} + \left\langle \mathcal{J}\vec{G}_{\vec{u}}, 2\eta\mathcal{S} \right\rangle_{E,N} \\ &= M_0 \left\langle \mathcal{J}\vec{G}_\mu, \vec{G}_\mu \right\rangle_{E,N} + \left\langle \mathcal{J}\left(\frac{\vec{G}_{\vec{u}} + \vec{G}_{\vec{u}}^T}{2}\right), 2\eta\mathcal{S} \right\rangle_{E,N} \\ &= \left\langle \mathcal{J}\left(M_0 |\vec{G}_\mu|^2 + 2\eta\mathcal{S} : \mathcal{S}\right), 1 \right\rangle_{E,N} \geqslant 0,\end{aligned} \tag{A.13}$$

where $\mathcal{S} = \boldsymbol{\mathcal{S}}\left(\vec{G}_{\vec{u}}\right)$ is the strain tensor (12) evaluated with the discrete gradients $\vec{G}_{\vec{u}}$. Therefore, the discrete viscous volume terms are dissipative.

Appendix B. Curved mesh

B.1. Curved mesh definition

The mesh used to verify that the scheme is total phase conservative, entropy–stable and freestream preserving for curvilinear elements is presented in Section 6.1 here. We make use of the mesh presented in [69,8]. The mesh is a $4 \times 4 \times 4$ cube defined in $\Omega = [0, 1]^3$ with all boundaries being treated with periodic boundary conditions. Initially we construct a

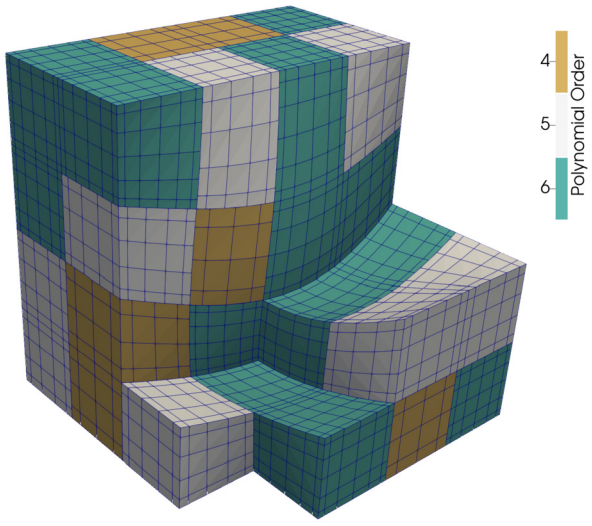


Fig. B.11. Curvilinear mesh used for the random initial condition and freestream preservation error. We also present an instance of the random distribution of the polynomial order.

Cartesian mesh and apply a transformation to the space variables $\vec{\chi} = (\chi_1, \chi_2, \chi_3)^T$ to create the curved mesh in physical space $\vec{x}$. The transformation function used is

$$x_l = \chi_l + 0.1 \sin(\pi \chi_1) \sin(\pi \chi_2) \sin(\pi \chi_3) \quad \text{for } x_l = 1, 2, 3. \tag{B.1}$$

The mesh has been generated using the HOPR package [95] with a geometrical order of approximation of $N_{geo} = 2$. A cutout of the mesh is presented in Fig. B.11 along with an example of the random polynomial order distribution used for the tests in Section 6.1.

References

[1] S. Mirjalili, S. Jain, M. Dodd, Interface-Capturing Methods for Two-Phase Flows: An Overview and Recent Developments, Center for Turbulence Research - Annual Research Brief, vol. 12, 2017, pp. 117–135.
[2] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, J. Chem. Phys. 28 (2) (1958) 258–267.
[3] G. Caginalp, X. Chen, Convergence of the phase field model to its sharp interface limits, Eur. J. Appl. Math. 9 (4) (1998) 417–445.
[4] P. Yue, C. Zhou, J.J. Feng, Sharp-interface limit of the Cahn-Hilliard model for moving contact lines, J. Fluid Mech. 645 (2010) 279.
[5] J. Kim, S. Lee, Y. Choi, S.M. Lee, D. Jeong, Basic principles and practical applications of the Cahn–Hilliard equation, Math. Probl. Eng. 2016 (2016).
[6] J. Shen, X. Yang, A phase-field model and its numerical approximation for two-phase incompressible flows with different densities and viscosities, SIAM J. Sci. Comput. 32 (3) (2010) 1159–1179.
[7] J. Manzanero, G. Rubio, D.A. Kopriva, E. Ferrer, E. Valero, A free–energy stable nodal discontinuous Galerkin approximation with summation–by–parts property for the Cahn–Hilliard equation, J. Comput. Phys. 403 (2020) 109072.
[8] G. Ntoukas, J. Manzanero, G. Rubio, E. Valero, E. Ferrer, A free–energy stable p–adaptive nodal discontinuous Galerkin for the Cahn–Hilliard equation, J. Comput. Phys. (2021) 110409.
[9] J. Manzanero, G. Rubio, D.A. Kopriva, E. Ferrer, E. Valero, An entropy–stable discontinuous Galerkin approximation for the incompressible Navier–Stokes equations with variable density and artificial compressibility, J. Comput. Phys. 408 (2020) 109241.
[10] J. Manzanero, G. Rubio, D.A. Kopriva, E. Ferrer, E. Valero, Entropy–stable discontinuous Galerkin approximation with summation–by–parts property for the incompressible Navier–Stokes/Cahn–Hilliard system, J. Comput. Phys. 408 (2020) 109363.
[11] E. Ferrer, R. Willden, A high order discontinuous Galerkin finite element solver for the incompressible Navier–Stokes equations, Comput. Fluids 46 (1) (2011) 224–230.
[12] T.C. Fisher, M.H. Carpenter, High-order entropy stable finite difference schemes for nonlinear conservation laws: finite domains, J. Comput. Phys. 252 (2013) 518–557.
[13] M.H. Carpenter, T.C. Fisher, E.J. Nielsen, S.H. Frankel, Entropy stable spectral collocation schemes for the Navier–Stokes equations: discontinuous interfaces, SIAM J. Sci. Comput. 36 (5) (2014) B835–B867.
[14] G.J. Gassner, A.R. Winters, D.A. Kopriva, Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations, J. Comput. Phys. 327 (2016) 39–66.
[15] T. Chen, C.W. Shu, Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws, J. Comput. Phys. 345 (2017) 427–461.
[16] A.R. Winters, R.C. Moura, G. Mengaldo, G.J. Gassner, S. Walch, J. Peiro, et al., A comparative study on polynomial dealiasing and split form discontinuous Galerkin schemes for under-resolved turbulence computations, J. Comput. Phys. 372 (2018) 1–21.
[17] G.J. Gassner, A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods, SIAM J. Sci. Comput. 35 (3) (2013) A1233–A1253.
[18] B. Biswas, H. Kumar, A. Yadav, Entropy stable discontinuous Galerkin methods for ten-moment Gaussian closure equations, J. Comput. Phys. 431 (2021) 110148.
[19] T. Chen, C.W. Shu, Review of Entropy Stable Discontinuous Galerkin Methods for Systems of Conservation Laws on Unstructured Simplex Meshes, 2020.
[20] E. Tadmor, Entropy Stable Schemes, Handbook of Numerical Analysis., vol. 17, Elsevier, 2016, pp. 467–493.
[21] J. Kou, S. Sun, Entropy stable modeling of non-isothermal multi-component diffuse-interface two-phase flows with realistic equations of state, Comput. Methods Appl. Mech. Eng. 341 (2018) 221–248.
[22] F. Coquel, C. Marmignon, P. Rai, F. Renac, An entropy stable high-order discontinuous Galerkin spectral element method for the Baer-Nunziato two-phase flow model, J. Comput. Phys. 431 (2021) 110135.
[23] K. Mattsson, M.H. Carpenter, Stable and accurate interpolation operators for high-order multiblock finite difference methods, SIAM J. Sci. Comput. 32 (4) (2010) 2298–2320.
[24] J.E. Kozdon, L.C. Wilcox, Stable coupling of nonconforming, high-order finite difference methods, SIAM J. Sci. Comput. 38 (2) (2016) A923–A952.
[25] A. Nissen, K. Kormann, M. Grandin, K. Virta, Stable difference methods for block-oriented adaptive grids, J. Sci. Comput. 65 (2) (2015) 486–511.
[26] M. Almquist, S. Wang, J. Werpers, Order-preserving interpolation for summation-by-parts operators at nonconforming grid interfaces, SIAM J. Sci. Comput. 41 (2) (2019) A1201–A1227.
[27] J.E. Kozdon, L.C. Wilcox, An energy stable approach for discretizing hyperbolic equations with nonconforming discontinuous Galerkin methods, J. Sci. Comput. 76 (3) (2018) 1742–1784.
[28] L. Friedrich, A.R. Winters, D.CD.R. Fernández, G.J. Gassner, M. Parsani, M.H. Carpenter, An entropy stable h/p non-conforming discontinuous Galerkin method with the summation-by-parts property, J. Sci. Comput. 77 (2) (2018) 689–725.
[29] M. Parsani, M.H. Carpenter, E.J. Nielsen, Entropy stable discontinuous interfaces coupling for the three-dimensional compressible Navier-Stokes equations, J. Comput. Phys. 290 (1) (2015) 132–138.
[30] M.H. Carpenter, M. Parsani, E.J. Nielsen, T.C. Fisher, Towards an entropy stable spectral element framework for computational fluid dynamics, in: 54th AIAA Aerospace Sciences Meeting, 2016, p. 1058.
[31] S. Shadpey, D.W. Zingg, Entropy-stable multidimensional summation-by-parts discretizations on hp-adaptive curvilinear grids for hyperbolic conservation laws, J. Sci. Comput. 82 (3) (2020) 1–46.
[32] M. Parsani, M.H. Carpenter, T.C. Fisher, E.J. Nielsen, Entropy stable staggered grid discontinuous spectral collocation methods of any order for the compressible Navier–Stokes equations, SIAM J. Sci. Comput. 38 (5) (2016) A3129–A3162.
[33] D.C.D.R. Fernández, M.H. Carpenter, L. Dalcin, S. Zampini, M. Parsani, Entropy stable h/p-nonconforming discretization with the summation-by-parts property for the compressible Euler and Navier–Stokes equations, Partial Differ. Equ. Appl. 1 (2) (2020) 1–54.
[34] D.C.D.R. Fernández, M.H. Carpenter, L. Dalcin, L. Fredrich, A.R. Winters, G.J. Gassner, et al., Entropy-stable p-nonconforming discretizations with the summation-by-parts property for the compressible Navier–Stokes equations, Comput. Fluids (2020) 104631.
[35] J. Chan, M. Bencomo, D.C. Fernández, Mortar-based entropy-stable discontinuous Galerkin methods on non-conforming quadrilateral and hexahedral meshes, arXiv preprint, arXiv:2005.03237, 2020.

[36] Z. Xie, D. Pavlidis, J.R. Percival, J.L. Gomes, C.C. Pain, O.K. Matar, Adaptive unstructured mesh modelling of multiphase flows, Int. J. Multiph. Flow 67 (2014) 104–110.
[37] T. Fondelli, A. Andreini, B. Facchini, Numerical simulation of dam-break problem using an adaptive meshing approach, Energy Proc. 82 (2015) 309–315.
[38] L.C. Ngo, H.G. Choi, A multi-level adaptive mesh refinement for an integrated finite element/level set formulation to simulate multiphase flows with surface tension, Comput. Math. Appl. 79 (4) (2020) 908–933.
[39] M. Sussman, A parallelized, adaptive algorithm for multiphase flows in general geometries, Comput. Struct. 83 (6–7) (2005) 435–444.
[40] N. Nangia, B.E. Griffith, N.A. Patankar, A.P.S. Bhalla, A robust incompressible Navier-Stokes solver for high density ratio multiphase flows, J. Comput. Phys. 390 (2019) 548–594.
[41] D. Fuster, A. Bagué, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, et al., Simulation of primary atomization with an octree adaptive mesh refinement and VOF method, Int. J. Multiph. Flow 35 (6) (2009) 550–565.
[42] P. Yue, C. Zhou, J.J. Feng, C.F. Ollivier-Gooch, H.H. Hu, Phase-field simulations of interfacial dynamics in viscoelastic fluids using finite elements with adaptive meshing, J. Comput. Phys. 219 (1) (2006) 47–67.
[43] H.D. Ceniceros, A.M. Roma, A nonstiff, adaptive mesh refinement-based method for the Cahn–Hilliard equation, J. Comput. Phys. 225 (2) (2007) 1849–1862.
[44] J. Wackers, G. Deng, E. Guilmineau, A. Leroyer, P. Queutey, M. Visonneau, Combined refinement criteria for anisotropic grid refinement in free-surface flow simulation, Comput. Fluids 92 (2014) 209–222.
[45] I. Ginzburg, G. Wittum, Two-phase flows on interface refined grids modeled with VOF., staggered finite volumes, and spline interpolants, J. Comput. Phys. 166 (2) (2001) 302–335.
[46] A.M. Rueda-Ramírez, G. Rubio, E. Ferrer, E. Valero, An anisotropic p-adaptation multigrid scheme for discontinuous Galerkin methods, in: Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018, Springer, Cham, 2020, pp. 549–560.
[47] M. Kompenhans, G. Rubio, E. Ferrer, E. Valero, Comparisons of p-adaptation strategies based on truncation-and discretisation-errors for high order discontinuous Galerkin methods, Comput. Fluids 139 (2016) 36–46.
[48] D. Ekelschot, D. Moxey, S. Sherwin, J. Peiró, A p-adaptation method for compressible flow problems using a goal-based error indicator, Comput. Struct. 181 (2017) 55–69.
[49] N. Chalmers, G. Agbaglah, M. Chrust, C. Mavriplis, A parallel hp-adaptive high order discontinuous Galerkin method for the incompressible Navier-Stokes equations, J. Comput. Phys. X 2 (2019) 100023.
[50] D.A. Kopriva, S.L. Woodruff, M.Y. Hussaini, Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method, Int. J. Numer. Methods Eng. 53 (1) (2002) 105–122.
[51] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system. III. Nucleation in a two-component incompressible fluid, J. Chem. Phys. 31 (3) (1959) 688–699.
[52] J.L. Guermond, L. Quartapelle, A projection FEM for variable density incompressible flows, J. Comput. Phys. 165 (1) (2000) 167–188.
[53] J. Shen, X. Yang, Energy stable schemes for Cahn–Hilliard phase-field model of two-phase incompressible flows, Chin. Ann. Math., Ser. B 31 (5) (2010) 743–758.
[54] J. Lowengrub, L. Truskinovsky, Quasi–incompressible Cahn–Hilliard fluids and topological transitions, Proc. R. Soc. Lond., Ser. A, Math. Phys. Eng. Sci. 454 (1978) (1998) 2617–2654.
[55] J. Shen, On a new pseudocompressibility method for the incompressible Navier-Stokes equations, Appl. Numer. Math. 21 (1) (1996) 71–90.
[56] X. Feng, J. Kou, S. Sun, A novel energy stable numerical scheme for Navier-Stokes-Cahn-Hilliard two-phase flow model with variable densities and viscosities, in: International Conference on Computational Science, Springer, 2018, pp. 113–128.
[57] F. Boyer, C. Lapuerta, S. Minjeaud, B. Piar, M. Quintard, Cahn–Hilliard/Navier–Stokes model for the simulation of three-phase flows, Transp. Porous Media 82 (3) (2010) 463–483.
[58] G.J. Gassner, A. Winters Andrew, F. Hindenlang, D.A. Kopriva, The BR1 scheme is stable for the compressible Navier-Stokes equations, J. Sci. Comput. (2017) 04.
[59] E. Tadmor, Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems, Acta Numer. 12 (2003) 451–512.
[60] S. Dong, On imposing dynamic contact-angle boundary conditions for wall-bounded liquid–gas flows, Comput. Methods Appl. Mech. Eng. 247 (2012) 179–200.
[61] M.H. Carpenter, T.C. Fisher, E.J. Nielsen, S.H. Frankel, Entropy stable spectral collocation schemes for the Navier-Stokes equations: discontinuous interfaces, SIAM J. Sci. Comput. 36 (5) (2014) B835–B867.
[62] D.A. Kopriva, Metric identities and the discontinuous spectral element method on curvilinear meshes, J. Sci. Comput. 26 (3) (2006) 301.
[63] D.A. Kopriva, Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers, Springer Science & Business Media, 2009.
[64] D.A. Kopriva, F.J. Hindenlang, T. Bolemann, G.J. Gassner, Free-stream preservation for curved geometrically non-conforming discontinuous Galerkin spectral elements, J. Sci. Comput. 79 (3) (2019) 1389–1408.
[65] D.A. Kopriva, F.J. Hindenlang, T. Bolemann, G.J. Gassner, Free-stream preservation for curved geometrically non-conforming discontinuous Galerkin spectral elements, J. Sci. Comput. 79 (3) (2019) 1389–1408.
[66] D.A. Kopriva, Metric identities and the discontinuous spectral element method on curvilinear meshes, J. Sci. Comput. 26 (3) (2006) 301.
[67] M.H. Carpenter, D. Gottlieb, S. Abarbanel, The stability of numerical boundary treatments for compact high-order finite-difference schemes, J. Comput. Phys. 108 (2) (1993) 272–295.
[68] D.A. Kopriva, A polynomial spectral calculus for analysis of DG spectral element methods, in: Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016, Springer, 2017, pp. 21–40.
[69] M. Bohm, A.R. Winters, G.J. Gassner, D. Derigs, F. Hindenlang, J. Saur, An entropy stable nodal discontinuous Galerkin method for the resistive MHD equations. Part I: theory and numerical verification, J. Comput. Phys. 422 (2020) 108076.
[70] F. Bassi, F. Massa, L. Botti, A. Colombo, Artificial compressibility Godunov fluxes for variable density incompressible flows, Comput. Fluids 169 (2018) 186–200.
[71] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, J. Comput. Phys. 131 (2) (1997) 267–279.
[72] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer, 2009.
[73] C. Mavriplis, A posteriori error estimators for adaptive spectral element techniques, in: Proceedings of the Eighth GAMM-Conference on Numerical Methods in Fluid Mechanics, Springer, 1990, pp. 333–342.
[74] J. Manzanero, A.M. Rueda–Ramírez, G. Rubio, E. Ferrer, The Bassi Rebay 1 scheme is a special case of the Symmetric Interior Penalty formulation for discontinuous Galerkin discretisations with Gauss–Lobatto points, J. Comput. Phys. 363 (2018) 1–10.
[75] D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling, J. Comput. Phys. 155 (1) (1999) 96–127.
[76] P. Boyanova, M. Neytcheva, Efficient numerical solution of discrete multi-component Cahn–Hilliard systems, Comput. Math. Appl. 67 (1) (2014) 106–121.

[77] P. Yue, J.J. Feng, C. Liu, J. Shen, A diffuse-interface method for simulating two-phase flows of complex fluids, J. Fluid Mech. 515 (2004) 293.
[78] K.E. Teigen, P. Song, J. Lowengrub, A. Voigt, A diffuse-interface method for two-phase flows with soluble surfactants, J. Comput. Phys. 230 (2) (2011) 375–393.
[79] I. Barosan, P. Anderson, H. Meijer, Application of mortar elements to diffuse-interface methods, Comput. Fluids 35 (10) (2006) 1384–1399.
[80] M. Hintermüller, M. Hinze, C. Kahle, An adaptive finite element Moreau–Yosida-based solver for a coupled Cahn–Hilliard/Navier–Stokes system, J. Comput. Phys. 235 (2013) 810–827.
[81] S.R. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, et al., Quantitative benchmark computations of two-dimensional bubble dynamics, Int. J. Numer. Methods Fluids 60 (11) (2009) 1259–1288.
[82] B.S. Hosseini, S. Turek, M. Möller, C. Palmes, Isogeometric analysis of the Navier–Stokes–Cahn–Hilliard equations with application to incompressible two-phase flows, J. Comput. Phys. 348 (2017) 171–194.
[83] J.C. Martin, W.J. Moyce, J. Martin, W. Moyce, W.G. Penney, A. Price, et al., Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane, Philos. Trans. R. Soc. Lond. Ser. A, Math. Phys. Sci. 244 (882) (1952) 312–324.
[84] S. Koshizuka, Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, Nucl. Sci. Eng. 123 (3) (1996) 421–434, Available from: https://doi.org/10.13182/NSE96-A24205.
[85] B. Ramaswamy, M. Kawahara, Lagrangian finite element analysis applied to viscous free surface fluid flow, Int. J. Numer. Methods Fluids 7 (9) (1987) 953–984.
[86] Z. Gu, H. Wen, C.H. Yu, T.W. Sheu, Interface-preserving level set method for simulating dam-break flows, J. Comput. Phys. 374 (2018) 249–280.
[87] W. Price, Y. Chen, A simulation of free surface waves for incompressible two-phase flows using a curvilinear level set formulation, Int. J. Numer. Methods Fluids 51 (3) (2006) 305–330.
[88] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1) (1981) 201–225, Available from: https://www.sciencedirect.com/science/article/pii/0021999181901455.
[89] A. Muta, P. Ramachandran, P. Negi, An efficient, open source, iterative ISPH scheme, Comput. Phys. Commun. 255 (2020) 107283.
[90] P.H. Chiu, Y.T. Lin, A conservative phase field method for solving incompressible two-phase flows, J. Comput. Phys. 230 (1) (2011) 185–204.
[91] V. Joshi, R.K. Jaiman, A positivity preserving and conservative variational scheme for phase-field modeling of two-phase flows, J. Comput. Phys. 360 (2018) 137–166.
[92] J. Williamson, Low-storage Runge-Kutta schemes, J. Comput. Phys. 35 (1) (1980) 48–56.
[93] K. Yang, T. Aoki, Weakly compressible Navier-Stokes solver based on evolving pressure projection method for two-phase flow simulations, J. Comput. Phys. 431 (2021) 110113.
[94] D. Sun, W. Tao, A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows, Int. J. Heat Mass Transf. 53 (4) (2010) 645–655.
[95] F. Hindenlang, T. Bolemann, C-D. Munz, Mesh curving techniques for high order discontinuous Galerkin simulations, in: IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach, Springer, 2015, pp. 133–152.