ESAIM: M2AN 51 (2017) 1987–2015
DOI: 10.1051/m2an/2017027

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

A FINITE VOLUME METHOD FOR UNDERCOMPRESSIVE SHOCK WAVES IN TWO SPACE DIMENSIONS *

Christophe Chalons[1], Christian Rohde[2] and Maria Wiebe[2]

Abstract. Undercompressive shock waves arise in many physical processes which involve multiple phases. We propose a Finite Volume method in two space dimensions to approximate weak solutions of systems of hyperbolic or hyperbolic-elliptic conservation laws that contain undercompressive shock waves. The method can be seen as a generalization of the spatially one-dimensional and scalar approach in [C. Chalons, P. Engel and C. Rohde, *SIAM J. Numer. Anal.* **52** (2014) 554–579]. It relies on a moving mesh ansatz such that the undercompressive wave is represented as a sharp interface without any artificial smearing. It is proven that the method is locally conservative and recovers planar traveling wave solutions exactly. To demonstrate the efficiency and reliability of the new scheme we test it on scalar model problems and as an application on compressible liquid-vapour flow in two space dimensions.

Mathematics Subject Classification. 35L65, 65M12, 76M25.

Received September 20, 2016. Revised March 23, 2017. Accepted April 27, 2017.

1. Introduction

We consider weak solutions for systems of conservation laws. These systems are supposed to be equipped with an entropy-entropy flux pair such that the entropy is strictly convex in a state space that is typically split into two disjoint open subsets. As a consequence one has strict hyperbolicity in the complete state space. Depending on the characteristic fields, weak solutions can contain undercompressive shock waves that connect states in the two disjoint subsets. These can be still compatible with the entropy-entropy flux pair, *i.e.*, they satisfy an appropriate entropy inequality. While undercompressive waves might be essential in many applications (see below) and consistent with the second law of thermodynamics, the entropy solution concept does not ensure anymore well-posedness of associated initial boundary value problems. Well-posedness can be restored if additional constraints are put on the undercompressive shock wave. If the constraints take the form of algebraic

Keywords and phrases. Undercompressive shock waves in 2D, hyperbolic-elliptic systems, interface tracking, Finite Volume method.

* *The first author was partially supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH. C.R. and M.W. would like to thank the German Research Foundation for financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/2) at the University of Stuttgart.*

[1] Laboratoire de Mathématiques de Versailles, UVSQ, CNRS, Université Paris-Saclay, 78035 Versailles, France.
christophe.chalons@uvsq.fr

[2] Institute for Applied Analysis and Numerical Simulation, University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany. christian.rohde|maria.wiebe@mathematik.uni-stuttgart.de

Article published by EDP Sciences

jump conditions these are denoted as kinetic relations. We refer to [24] for a comprehensive overview on well-posedness results for hyperbolic problems with undercompressive waves.

Before we turn to the main issue of this paper, that is the numerical solution of hyperbolic systems with undercompressive waves, let us mention some applications where undercompressive shock waves play a decisive role. The dynamics of multiphase materials with phase change, it may be in fluids or likewise in solids, can be modelled by hyperbolic conservation laws such that the phase boundaries are represented by undercompressive shock waves. In this context kinetic relations have been introduced for the first time, see *e.g.* [1] for phase transitions in solids and for a more general context including liquid-vapour phase [38]. Other examples can be found in the modelling of thin films [2], pedestrians' dynamics [11], sedimentary waves [21] or overshoot waves for infiltration in porous media [39].

The numerical approximation of problems with undercompressive waves is a complex issue. Exact solutions even for scalar, one-dimensional problems are typically neither monotone nor total-variation diminishing such that standard approaches like the Finite Volume or Discontinuous Galerkin schemes with monotone numerical fluxes can hardly be used. Undercompressive waves can occur as limit solutions of higher-order regularized conservation laws when the regularization parameter vanishes. If the regularization parameter is substituted by a mesh parameter a consistent discretization for problems with undercompressive shock waves can be motivated. This type of schemes has been developed in *e.g.* [18, 25]. Alternatively, schemes can use the kinetic relation directly such that the undercompressive shock wave is represented as a sharp interface and the whole problem is understood as a free boundary value problem. The Glimm-type ansatz in [23], see also [7], belongs to this class. Deterministic versions that use an extra tracking of the undercompressive waves have been introduced in [4, 5, 29, 42]. The drawback of all these schemes is the fact that they are not conservative. Conservative Finite Volume type methods have been suggested in [3]. These schemes use a special numerical flux close to an undercompressive wave (and standard monotone fluxes away from the wave).

In [6] a conservative method has been suggested that relies also on the tracking idea but uses a moving mesh. In contrast to the aforementioned methods convergence for quite general initial value problems has been shown. Moreover the approach can be extended to the *systems' case* and to *multiple space dimensions*. It is the main objective of the paper at hand to pursue these generalizations for two space dimensions in the framework of the Finite Volume method. The key ingredients are the use of moving meshes and exploiting the exact dynamics across undercompressive waves. The moving mesh approach in this paper is different from standard uses where the mesh is changed globally to reduce the error or to get aligned with appropriate transport directions (see Section 3.1 for a more detailed discussion). Here we intend to track the mesh only locally around the discrete interface and try to avoid any global changes of the mesh that effect the bulk domains. According to the choosen solution concept, only phenomena where the interface consists of closed non-intersecting curves are treated. This excludes the numerical treatment of interface - interface collision or phase extinction.

Let us finally note that there is not much numerical work for undercompressive shock waves in multiple space dimensions, which is usually also restricted to specific applications. The evolution of phase boundaries in solids is analyzed in [19,30] by a level set approach. In [13,14] the ghostfluid method is applied to compressible liquid-vapour flow. The paper [20] is devoted to the tracking of undercompressive overshoot waves in porous media where mixed phase volumes are allowed. Equally, mixed component cells appear in the moving-mesh approach of [9] where the dynamics of compressible multicomponent flow are computed.

The paper's content is structured in the following way. In Section 2 we introduce first a generic hyperbolic two-phase model and present then some specific examples. The examples include a scalar model problem as well as isothermal liquid-vapour flow. Next we present in Section 3 the geometric setting for a Finite Volume moving mesh method and the complete numerical method to track undercompressive waves. Section 4 is devoted to basic analytical statements, in particular it is shown that the new numerical method is locally conservative and consistent with planar traveling waves. In Section 5 we validate the scheme carefully and present computations for the examples introduced in Section 2. A short summary and some conclusions are given in Section 6.

2. Mathematical models

In this section we present the mathematical problems which can be solved by the Finite Volume moving mesh method which will be introduced in the subsequent Section 3. We start with a generic two-phase problem in Section 2.1. In particular the numerical scheme is formulated in terms of this general problem. To illustrate the generic problem and to prepare Section 5 on validation of the numerical method we give instances of the generic two-phase problem in Section 2.2.

2.1. A generic two-phase problem

For two convex sets $\mathcal{P}_-$, $\mathcal{P}_+ \subset \mathbb{R}^m$, $m \in \mathbb{N}$, with $\mathcal{P}_- \cap \mathcal{P}_+ = \emptyset$ let the state space $\mathcal{U} = \mathcal{P}_- \cup \mathcal{P}_+ \subset \mathbb{R}^m$ be given. In the sequel we will identify the sets $\mathcal{P}_-$, $\mathcal{P}_+$ as $\pm$-*phases*. We need also a mapping π that distinguishes different phases and is defined by

$$\pi : \mathcal{U} \to \{-,+\}, \mathbf{u} \mapsto \begin{cases} - & \text{if} \quad \mathbf{u} \in \mathcal{P}_-, \\ + & \text{if} \quad \mathbf{u} \in \mathcal{P}_+. \end{cases} \tag{2.1}$$

Let flux functions $\mathbf{f}_1, \mathbf{f}_2 \in C^1(\mathcal{U}, \mathbb{R}^m)$ be given. Then we consider for some time $T \in (0, \infty)$ and $D_T = \mathbb{R}^2 \times (0, T)$ the initial value problem for a system of conservation laws with unknown

$$\mathbf{u} = \mathbf{u}(t, \mathbf{x} = (x_1, x_2)^T) : [0, T) \times \mathbb{R}^2 \to \mathcal{P}_- \cup \mathcal{P}_+, \tag{2.2}$$

given by

$$\mathbf{u}_t + (\mathbf{f}_1(\mathbf{u}))_{x_1} + (\mathbf{f}_2(\mathbf{u}))_{x_2} = 0 \qquad \text{in} \quad D_T, \tag{2.3}$$

$$\mathbf{u}(0, \cdot) = \mathbf{u}_0 \qquad \text{in} \quad \mathbb{R}^2. \tag{2.4}$$

Thereby $\mathbf{u}_0 : \mathbb{R}^2 \to \mathcal{U}$ is the given initial function assumed to be bounded.
A function $\mathbf{u} \in L^\infty((0, T) \times \mathbb{R}^2, \mathcal{U})$ is called a *weak solution* of the initial value problem (2.3), (2.4) in D_T if

$$\int_0^T \int_{\mathbb{R}^2} \mathbf{u}\phi_t + \mathbf{f}_1(\mathbf{u})\phi_{x_1} + \mathbf{f}_2(\mathbf{u})\phi_{x_2} \, \mathrm{d}V \, \mathrm{d}t = - \int_{\mathbb{R}^2} \mathbf{u}_0 \phi(0, \mathbf{x}) \, \mathrm{d}V$$

holds for all $\phi \in \mathcal{C}_0^\infty([0, T) \times \mathbb{R}^2, \mathbb{R})$.

We furthermore assume that the system (2.3) is equipped with an entropy-entropy flux pair $(\eta, (q_1, q_2)) : \mathcal{U} \to \mathbb{R}^3$. Here the entropy function $\eta \in \mathcal{C}^2(\mathcal{U}, \mathbb{R})$ is strictly convex and the entropy fluxes $q_i \in \mathcal{C}^1(\mathcal{U}, \mathbb{R}), i = 1, 2$, satisfy the compatibility relation $\nabla\eta^T D\mathbf{f}_i = \nabla q_i^T$ in $\mathcal{U}$. The existence of an entropy pair implies

$$n_1 D\mathbf{f}_1(\mathbf{u}) + n_2 D\mathbf{f}_2(\mathbf{u}) \text{ is diagonalizable in } \mathbb{R} \text{ for all } \mathbf{u} \in \mathcal{U} \text{ and all } \mathbf{n} = (n_1, n_2)^T \in \mathcal{S}^1. \tag{2.5}$$

Note that condition (2.5) renders the system (2.3) to be hyperbolic. A weak solution $\mathbf{u} \in L^\infty((0, T) \times \mathbb{R}^2, \mathcal{U})$ is called an *entropy solution* of (2.3), (2.4) in D_T if

$$\int_0^T \int_{\mathbb{R}^2} \eta(\mathbf{u})\phi_t + q_1(\mathbf{u})\phi_{x_1} + q_2(\mathbf{u})\phi_{x_2} \, \mathrm{d}V \, \mathrm{d}t \geq - \int_{\mathbb{R}^2} \eta(\mathbf{u}_0)\phi(0, \mathbf{x}) \, \mathrm{d}V \tag{2.6}$$

holds for all $\phi \in \mathcal{C}_0^\infty([0, T) \times \mathbb{R}^2, \mathbb{R}), \phi \geq 0$.
In this paper we are interested in entropy solutions $\mathbf{u}$ that split the plane $\mathbb{R}^2$ for each time $t \in [0, T)$ in two disjunct $\pm$-phase domains $D_-(t), D_+(t)$ and a curve $\Gamma(t)$ such that for almost all $\mathbf{x} \in \mathbb{R}^2$

$$\pi\mathbf{u}(t, \mathbf{x}) = \pm \Rightarrow \mathbf{x} \in D_\pm(t) \tag{2.7}$$

and $\Gamma(t) = \overline{D_-(t)} \cap \overline{D_+(t)}$ hold. We call $D_\pm(t)$ the *$\pm$-phase domain* and $\Gamma(t)$ the *sharp interface*. For $\mathbf{x} \in \Gamma(t)$ let $\mathbf{n}(t,\mathbf{x}) \in \mathcal{S}^1$ denote the normal vector of $\Gamma(t)$ that points into $D_+(t)$. Let the function $\mathbf{u} : D_T \to \mathcal{U}$ be regular enough such that for $(t,\mathbf{x}) \in D_T$ the traces

$$\mathbf{u}_\pm(t,\mathbf{x}) := \lim_{\varepsilon \to 0, \varepsilon > 0} \mathbf{u}(t, \mathbf{x} \pm \varepsilon \mathbf{n}(t,\mathbf{x}))$$

exist. Then we define the interfacial jump

$$[\![\mathbf{u}(t,\mathbf{x})]\!] = \mathbf{u}_+(t,\mathbf{x}) - \mathbf{u}_-(t,\mathbf{x}).$$

We denote by $r(t,\mathbf{x})$ the speed of $\Gamma(t)$ in direction $\mathbf{n}(t,\mathbf{x}) = (n_1(t,\mathbf{x}), n_2(t,\mathbf{x}))^T \in \mathcal{S}^1$. Necessary conditions for the function $\mathbf{u}$ to be a weak solution of (2.3), (2.4) are the Rankine–Hugoniot conditions

$$-r(t,\cdot)\,[\![\mathbf{u}(t,\cdot)]\!] + [\![n_1 \mathbf{f}_1(\mathbf{u}(t,\cdot)) + n_2 \mathbf{f}_2(\mathbf{u}(t,\cdot))]\!] = 0. \tag{2.8}$$

Additionally we require for some function $\mathcal{K} : \mathcal{P}_- \times \mathcal{P}_+ \times \mathbb{R} \to \mathbb{R}$ that the solution $\mathbf{u}$ satisfies the kinetic relation

$$\mathcal{K}(\mathbf{u}_-(t,\cdot), \mathbf{u}_+(t,\cdot), r(t,\cdot)) = 0. \tag{2.9}$$

We give in Section 2.2 examples for physically relevant kinetic relations. Let us summarize our notion of solution for the overall free boundary problem in the following definition.

Definition 2.1 (Sharp interface solution)**.** Let a smooth curve Γ_0 and disjunct domains $D_{\pm,0}$ with $\mathbb{R}^2 = D_{+,0} \cup D_{-,0} \cup \Gamma_0$ be given. Furthermore let $\mathbf{u}_0 \in L^\infty(\mathbb{R}^2, \mathcal{U})$ be such that the range of $\mathbf{u}_0|_{D_\pm(0)}$ is in $\mathcal{P}_\pm$. Furthermore let an entropy-entropy flux pair $(\eta, (q_1, q_2))$ for (2.3) and a kinetic relation (2.9) be given.

Then a function $\mathbf{u} \in C^0([0,T); L^\infty(\mathbb{R}^2))$, $\pm$-phase domain families $\{D_\pm(t)\}_{t\in[0,T)}$ and a sharp-interface family $\{\Gamma(t)\}_{t\in[0,T)}$ are called an **entropy-compatible sharp-interface solution of (2.3), (2.4)** if the following conditions are satisfied.

(i) For $t = 0$ we have $D_\pm(0) = D_{\pm,0}$, $\Gamma(0) = \Gamma_0$ and for each $t \in (0,T)$ we have $\mathbb{R}^2 = D_-(t) \cup D_+(t) \cup \Gamma(t)$ with disjunct two-dimensional $\pm$-phase domains $D_\pm(t)$ and the curve $\Gamma(t)$.
(ii) The condition (2.7) holds for almost all $(t,\mathbf{x}) \in D_T$.
(iii) The function $\mathbf{u}$ is an entropy solution of (2.3), (2.4) in D_T.
(iv) For each $t \in (0,T)$ the function $\mathbf{u}$ satisfies the trace conditions (2.8), (2.9).

Remark 2.2.

- We have restricted ourselves in Definition 2.1 to smooth interface motion. However, the solution concept allows for discontinuous solutions (shock waves) in the $\pm$-phase domains. Note that condition (iv) gives an additional implicit regularity requirement on the solution, *i.e.* the traces are supposed to exist.
- Definition 2.1 excludes several phenomena, for example the interface-interface interaction. In the case where two interfaces interact we get at collision time t two phase domains, such that the interface curve defined by the set $\Gamma(t) = \overline{D_-(t)} \cap \overline{D_+(t)}$ is not smooth and in particular conditions (2.8), (2.9) cannot be defined in the classical sense (without having a properly defined normal). The same problem appears for phase extinctions, where the interface curve shrinks to a singular point.
  Entropy-compatible sharp-interface solutions are only defined on $\mathbb{R}^2$. For bounded domains, the interface - boundary interaction has to be prescribed by an additional condition like in the fluid mechanical case Young's law for contact angles in equilibria.

For our numerical approach the Riemann problem for the one-dimensional version of (2.3) will be important. Fix some $\mathbf{n} \in \mathcal{S}^1$ and define

$$\mathbf{f}(\mathbf{u}) = n_1 \mathbf{f}_1(\mathbf{u}) + n_2 \mathbf{f}_2(\mathbf{u}), \quad \mathbf{u} \in \mathcal{U}. \tag{2.10}$$

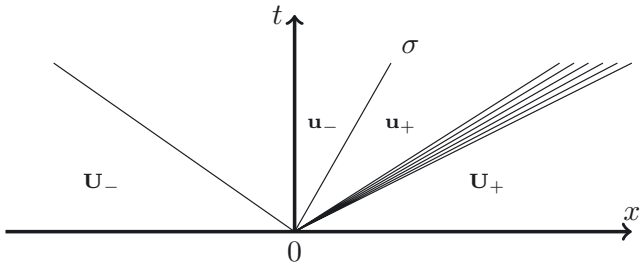


FIGURE 1. Sketch of a Riemann pattern with phase transition wave with speed σ. The adjacent states are $\mathbf{u}_{\mp} \in \mathcal{P}_{\mp}$.

Then for states $\mathbf{U}_{\pm} \in \mathcal{P}_{\pm}$, the Riemann problem is the special initial value problem

$$\mathbf{w}_t + (\mathbf{f}(\mathbf{w}))_x = 0 \qquad \text{in} \qquad (0,\infty)\times\mathbb{R}, \qquad \mathbf{w}(0,x) = \begin{cases} \mathbf{U}_- & \text{if} \quad x<0, \\ \mathbf{U}_+ & \text{if} \quad x>0, \end{cases} \tag{2.11}$$

with unknown $\mathbf{w} = \mathbf{w}(t,x) \in \mathcal{U}$. It is a reasonable assumption that the exact entropy solution $\mathbf{w}$ of (2.11) is a self-similar function that connects the left state $\mathbf{U}_-$ and the right state $\mathbf{U}_+$ by at most m elementary waves (*i.e.*, shock waves, contact waves, rarefaction waves, attached shock-rarefaction waves, each of them within either $\mathcal{P}_-$ or $\mathcal{P}_+$) and exactly one phase transition wave. The phase transition wave is a shock wave that connects a state $\mathbf{u}_-$ in $\mathcal{P}_-$ with a state $\mathbf{u}_+$ in $\mathcal{P}_+$. Across this wave the conditions (2.8), (2.9) have to hold (see [24] for a general theory and [8,10,16,26,29] for specific cases). The range of the function $\mathbf{w}$ is in $\mathcal{P}_-$ left to the phase transition and in $\mathcal{P}_+$ otherwise (see Fig. 1 for some illustration). In the following, we do not need to know the exact Riemann problem solution but only the speed of the phase transition, as well as the two adjacent values. This might be even given by an approximative solver [5,32]. To combine both cases we introduce the notation of an interface solver.

Definition 2.3 (Interface solver)**.** Let $\mathbf{U}_{\pm} \in \mathcal{P}_{\pm}$ and $\mathbf{f}$ from (2.10) be given. For some kinetic relation (2.9) the mapping $\mathcal{R}_{\mathbf{f}} : \mathcal{P}_- \times \mathcal{P}_+ \to \mathbb{R} \times \mathcal{P}_- \times \mathcal{P}_+$ with

$$\mathcal{R}_{\mathbf{f}}(\mathbf{U}_-, \mathbf{U}_+) := (\sigma, \mathbf{u}_-, \mathbf{u}_+),$$

is called an *interface solver for* (2.3) if the following conditions hold.

(i) $\mathcal{R}_{\mathbf{f}}$ is a continuous mapping.
(ii) The states $\mathbf{u}_{\pm} \in \mathcal{P}_{\pm}$ and σ satisfy

$$-\sigma(\mathbf{u}_- - \mathbf{u}_+) + \mathbf{f}(\mathbf{u}_-) - \mathbf{f}(\mathbf{u}_+) = 0, \quad \mathcal{K}(\mathbf{u}_-, \mathbf{u}_+, \sigma) = 0. \tag{2.12}$$

(iii) If there is a $r \in \mathbb{R}$ such that $\mathbf{U}_-$, $\mathbf{U}_+$ fulfill (2.12) with $\sigma = r$ then $\mathcal{R}_{\mathbf{f}}(\mathbf{U}_-, \mathbf{U}_+) = (r, \mathbf{U}_-, \mathbf{U}_+)$.

We call σ the *speed of the interface*.

2.2. Applications

In this section we present two examples for the generic two-phase problem from Section 2.1. These examples are a scalar model problem as well as the isothermal Euler equations for compressible liquid-vapour flow. Both models will be used for testing the new algorithm in Section 5.

2.2.1. A scalar model problem

As an introductory example we will start with a simple scalar model problem, which consists of the cubic flux $f_1(u) = u^3$ in x_1-direction and zero flux $f_2(u) = 0$ in x_2-direction, *i.e.*

$$u_t + (u^3)_{x_1} + (0)_{x_2} = 0, \qquad u(0,\cdot) = u_0. \tag{2.13}$$

The model problem can be equipped with *e.g.* the entropy-entropy flux pair

$$\eta(u) = \frac{1}{2}u^2, \quad (q_1(u), q_2(u)) = \left(\frac{3}{4}u^4, 0\right).$$

The $\pm$-phases are identified with the regions of concavity/convexity of the cubic flux:

$$\mathcal{P}_- = (-\infty, 0), \ \mathcal{P}_+ = (0, \infty).$$

Shock waves that connect states in different phases can be undercompressive. The one-dimensional problem (2.11) that will appear at the phase boundary has always a flux of the shape $n_1 u^3$ for $n_1 \in [-1, 1]$ which is either convex-concave or concave-convex. A suitable kinetic relation following [26] is given as

$$\mathcal{K}(u_-, u_+, s) = u_- - \varphi(u_+) \text{ with } \varphi(u) = -0.75u. \tag{2.14}$$

For this kinetic relation and $\mathbf{n} \in \mathcal{S}^1$ an interface solver $\mathcal{R}_{n_1 f_1 + n_2 f_2}$ can be computed as in [24] according to

$$\mathcal{R}_{n_1 f_1 + n_2 f_2}(U_-, U_+) =$$

$$\begin{cases} \Big(a(\varphi(U_+), U_+, \mathbf{n}), \varphi(U_+), U_+\Big) & \text{if} \quad n_1 < 0 \text{ and } U_- < \varphi^{\#}(U_+), \\ \Big(a(U_-, \varphi^{-1}(U_-), \mathbf{n}), U_-, \varphi^{-1}(U_-)\Big) & \text{if} \quad n_1 \geq 0 \text{ and } a(U_-, \varphi^{-1}(U_-), \mathbf{n}) < a(U_-, U_+, \mathbf{n}), \\ \Big(a(U_-, U_+, \mathbf{n}), U_-, U_+\Big) & \text{otherwise,} \end{cases}$$

with

$$\varphi^{\#}(u) = -0.25u \qquad \text{and} \qquad a(u_-, u_+, \mathbf{n}) = \begin{cases} n_1 3(u_-)^2 & \text{if} \quad u_- = u_+ \\ n_1 \frac{(u_-)^3 - (u_+)^3}{u_- - u_+} & \text{otherwise.} \end{cases} \tag{2.15}$$

2.2.2. Compressible liquid-vapour flow

The isothermal Euler equations with non-monotone pressure function govern the dynamics of ideal liquid-vapour flow. The corresponding initial value problem for initial density function ρ_0 and momentum function $\mathbf{m}_0$ reads as

$$\begin{pmatrix} \rho \\ m_1 \\ m_2 \end{pmatrix}_t + \begin{pmatrix} m_1 \\ \frac{m_1^2}{\rho} + p(\rho) \\ \frac{m_1 m_2}{\rho} \end{pmatrix}_{x_1} + \begin{pmatrix} m_2 \\ \frac{m_1 m_2}{\rho} \\ \frac{m_2^2}{\rho} + p(\rho) \end{pmatrix}_{x_2} = 0 \qquad \text{in } D_T,$$
$$(\rho(0,\cdot), \mathbf{m}(0,\cdot)) = (\rho_0, \mathbf{m}_0) \ \text{ in } \mathbb{R}^2. \tag{2.16}$$

The unknowns are the density $\rho = \rho(t, \mathbf{x})$ and the momentum $\mathbf{m} = \mathbf{m}(t, \mathbf{x}) = (m_1(t, \mathbf{x}), m_2(t, \mathbf{x}))^T$. The given pressure function p is chosen in Van-der-Waals form

$$p : (0, B^{-1}) \to \mathbb{R}^+, \ \rho \mapsto R\theta \frac{\rho}{1 - B\rho} - A\rho^2, \tag{2.17}$$

with positive constants $A, B, \theta, R > 0$, where the fixed temperature θ is chosen in the subcritical regime such that p is non-monotone (see Fig. 2). We denote by $\rho_{\text{liq}}^{\text{spinod}} < \rho_{\text{vap}}^{\text{spinod}}$ the extreme values of the interval where

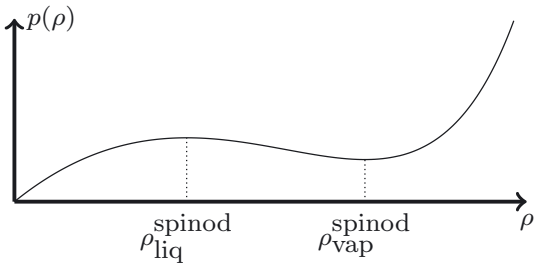


FIGURE 2. Pressure function $p = p(\rho)$.

the pressure function p is decreasing. The density (as well as approximations of it) must not take values in the interval $(\rho_{\text{liq}}^{\text{spinod}}, \rho_{\text{vap}}^{\text{spinod}})$, *i.e.*

$$\rho : [0,T) \times \mathbb{R}^2 \to \left(0, \rho_{\text{liq}}^{\text{spinod}}\right] \cup \left[\rho_{\text{vap}}^{\text{spinod}}, B^{-1}\right).$$

Therefore we define liquid and vapour phases according to

$$\mathcal{P}_- = \left(0, \rho_{\text{liq}}^{\text{spinod}}\right] \times \mathbb{R}^2, \ \mathcal{P}_+ = \left[\rho_{\text{vap}}^{\text{spinod}}, B^{-1}\right) \times \mathbb{R}^2. \tag{2.18}$$

The canonical entropy-entropy flux pair for the isothermal Euler equations is given as

$$\eta(\rho, \mathbf{m}) = \rho\Psi(\rho) + \frac{|\mathbf{m}|^2}{2\rho}, \quad \mathbf{q}(\rho, \mathbf{m}) = (q_1(\rho, \mathbf{m}), q_2(\rho, \mathbf{m})) = \left(\frac{m_1}{\rho}(\eta + p(\rho)), \frac{m_2}{\rho}(\eta + p(\rho))\right),$$

with (the Helmholtz free energy Ψ defined from) $\Psi'(\rho) = \frac{p(\rho)}{\rho^2}$. A suitable kinetic relation (see [38]) that will also be used in the numerical computations is

$$\mathcal{K}((\rho_-, \mathbf{m}_-), (\rho_+, \mathbf{m}_+), r) = \mu(\rho_-) + 0.5\left(\frac{\mathbf{m}_-}{\rho_-} \cdot \mathbf{n} - r\right)^2 - \mu(\rho_+) - 0.5\left(\frac{\mathbf{m}_+}{\rho_+} \cdot \mathbf{n} - r\right)^2 + k_* j,$$

with the Gibb's free energy μ given through $\mu' = p'/\rho$, relative mass flux $j = \rho_-(\frac{\mathbf{m}_-}{\rho_-} \cdot \mathbf{n} - r)$ and mobility $k_* > 0$. Actually it can be shown (see *e.g.* [32]) that this choice is accordance with the second law of thermodynamics and fixes the amount of entropy condition across phase boundaries to be $-k_* j^2$, namely

$$-r(\eta(\rho_-, \mathbf{m}_-) - \eta(\rho_+, \mathbf{m}_+)) + (\mathbf{q}(\rho_-, \mathbf{m}_-) \cdot \mathbf{n} - \mathbf{q}(\rho_+, \mathbf{m}_+) \cdot \mathbf{n}) = -k_* j^2.$$

For the numerical tests we choose $k_* = 3$.

The construction of an interface solver that allows for entropy compatible sharp-interface solutions has just recently been established in full generality (see [10,41] for the framework). Since the details of the construction are not important in the sequel we skip them refering to [41]. Note that we restrict ourselves in this presentation to cases where surface tension is negligible (*e.g.* for temperature θ close to the critical temperature). However, Riemann solvers that account for surface tension are available [32] and could also be used without altering the concept of the algorithm.

2.2.3. Compressible multifluid flow

We consider the dynamics of an isothermal mixture of several immiscible fluids. For two fluids in one space dimension the Euler like equations are

$$\begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho v \end{pmatrix}_t + \begin{pmatrix} \rho_1 v \\ \rho_2 v \\ \rho v^2 + p \end{pmatrix}_x = 0 \qquad \text{in} \quad D_T. \tag{2.19}$$

The unknowns are the densities $\rho_1 = \rho_1(t,x)$, $\rho_2 = \rho_2(t,x)$ of the two fluids and the momentum $\rho v = \rho(t,x)v(t,x)$. The density ρ of the mixture is given as $\rho = \rho_1 + \rho_2$. For the mixture of two perfect gases, the pressure function is described by the ideal gas law

$$p = (R_1\rho_1 + R_2\rho_2)T,$$

where T is the temperature and R_1, R_2 two species dependent constants. The model at hand does in fact not fit into the generic two-phase model, since there is no partitioning of the state space $\mathcal{U}$ into two phase domains. Nevertheless, a typical Riemann wave pattern contains a contact discontinuity separating the two fluids [27]. This contact discontinuity can be understood as a limiting case of an undercompressive shock wave. The new algorithm, that will be presented in Section 3, can be applied to (2.19) by replacing the interface solver from Definition 2.3 by a solver, that returns velocity $\sigma(= v)$ as well as left and right hand state $\mathbf{U}_\pm$ at the contact discontinuity. The exact tracking of the contact discontinuity then ensures, that no pressure oscillation appear at the contact line.

3. A Finite Volume moving mesh method

The numerical method we will present in the following for solving two-phase problems (2.3) will be a combination of suitable Finite Volume methods on moving meshes motivated in Section 3.1.1 and an interface tracking approach described in Section 3.1.2. Proceeding in this way dynamical remeshing of the mesh is required, that will be introduced in Section 3.1.3. Finally the new Finite Volume moving mesh method for generic two-phase problems is summarized in Section 3.2.

3.1. Moving meshes, tracking and remeshing

The basis of the new algorithm will be the Finite Volume moving mesh method for hyperbolic conservation laws, since it enables us to keep track of the interface curve between the $\pm$-phases if the motion of the mesh edges is chosen appropriately.

One of the first works on Finite Volume moving mesh methods for conservation laws in one space dimension was presented by Harten and Hyman [17]. They have chosen the motion of the mesh points in a way such that the numerical diffusion (caused by averaging in the Godunov method) is minimized which results in a tracking of the discontinuity for isolated shock waves or contact discontiuities. In [34] moving mesh methods were also used, but the motion of the mesh was chosen with the aid of a so called monitor function which is a measure for the numerical error. Points of the mesh obey a partial differential equation including the monitor function and accumulate in regions with a high error. Analytical results including proofs of convergence for one-dimensional moving mesh methods are presented in [33].

The extension of the method to the two-dimensional case was done in [36] and includes several numerical examples. Some extensions and applications of the two-dimensional moving mesh method were studied in *e.g.* [12, 28, 31, 35].

The application of moving mesh schemes for the tracking of undercompressive shock waves has been done in [42] for phase transitions in solid materials as well as in [6] for scalar conservation laws with convex-concave flux functions. Both publications present a conservative scheme in (only) one space dimension.

In the following section we shortly review the moving mesh method in two space dimensions which we will use later in order to resolve undercompressive shock waves.

3.1.1. Finite Volume schemes on moving meshes

Let for $N \in \mathbb{N}$ a time partition $t^0 = 0 < t^1 < \ldots < t^N = T$ of $[0,T]$ be given. We fix some $n \in \{0,\ldots,N-1\}$. In order to motivate a Finite Volume scheme on moving meshes for conservation laws (2.3) we consider a prism-like space-time cell $[0,T]\times\Omega \supset K_{st} = \{(t,\mathbf{x}) \,|\, t^n \le t \le t^{n+1}, \mathbf{x} \in K(t)\}$, see Figure 3, with $0 \le t^n < t^{n+1} \le T$,

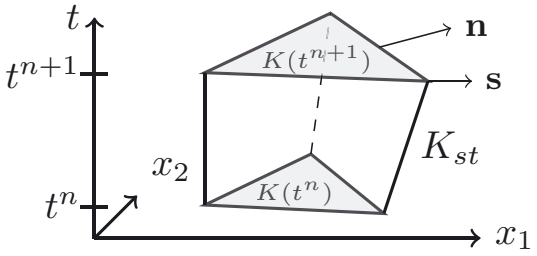


FIGURE 3. Space-time cell K_{st} in $[0,T]\times\mathbb{R}^2$.

$K(t) := \mathrm{conv}(\mathbf{p}_0(t),\mathbf{p}_1(t),\mathbf{p}_2(t))$ for $t\in[t^n,t^{n+1}]$ and time dependent points $\mathbf{p}_0=\mathbf{p}_0(t)$, $\mathbf{p}_1=\mathbf{p}_1(t)$, $\mathbf{p}_2=\mathbf{p}_2(t)\in\mathbb{R}^2$ which evolve linearly in time with speed $\mathbf{s}_0$, $\mathbf{s}_1$, $\mathbf{s}_2\in\mathbb{R}^2$ according to

$$\mathbf{p}_l(t)=\mathbf{p}_l(t^n)+(t-t^n)\mathbf{s}_l \qquad (l=1,2,3). \tag{3.1}$$

Suppose that we have a function $\mathbf{u}\in\mathcal{C}^1([0,T)\times\mathbb{R}^2,\mathcal{U})$, that satisfies (2.3) in the classical sense. Then, we integrate equation (2.3) over the space-time cell K_{st} and get with $\mathbf{f}(\mathbf{u}):=(\mathbf{f}_1(\mathbf{u}),\mathbf{f}_2(\mathbf{u}))$ the relation

$$\int_{t^n}^{t^{n+1}}\int_{K(t)}\mathbf{u}_t\,\mathrm{d}V\,\mathrm{d}t+\int_{t^n}^{t^{n+1}}\int_{\partial K(t)}\mathbf{f}(\mathbf{u})\cdot\mathbf{n}\,\mathrm{d}A\,\mathrm{d}t=0.$$

We apply Reynolds' transport theorem and get for the first term

$$\int_{t^n}^{t^{n+1}}\int_{K(t)}\mathbf{u}_t\,\mathrm{d}V\,\mathrm{d}t=\int_{K(t^{n+1})}\mathbf{u}(t^{n+1},\cdot)\,\mathrm{d}V-\int_{K(t^n)}\mathbf{u}(t^n,\cdot)\,\mathrm{d}V-\int_{t^n}^{t^{n+1}}\int_{\partial K(t)}(\mathbf{s}\cdot\mathbf{n})\mathbf{u}\,\mathrm{d}A\,\mathrm{d}t. \tag{3.2}$$

Here $\mathbf{s}:\partial K(t)\to\mathbb{R}^2$ is the speed of a point $\mathbf{x}(t)=\lambda\mathbf{p}_i(t)+(1-\lambda)\mathbf{p}_j(t)\in\partial K(t)$ with $i\neq j\in\{0,1,2\}$ and $\lambda\in[0,1]$, *i.e.* $\mathbf{s}(\mathbf{x})=\lambda\mathbf{s}_i+(1-\lambda)\mathbf{s}_j$, and $\mathbf{n}(t):\partial K(t)\to\mathcal{S}^1$ is the outer unit normal at the boundary of $K(t)$. This gives the conservation form of (2.3) on K_{st}, *i.e.*

$$\int_{K(t^{n+1})}\mathbf{u}(t^{n+1},\cdot)\,\mathrm{d}V-\int_{K(t^n)}\mathbf{u}(t^n,\cdot)\,\mathrm{d}V+\int_{t^n}^{t^{n+1}}\int_{\partial K(t)}(\mathbf{f}(u)-\mathbf{u}\mathbf{s}^T)\cdot\mathbf{n}\,\mathrm{d}A\,\mathrm{d}t=0. \tag{3.3}$$

Based on the conservation form (3.3) on a space-time cell K_{st} we will derive a Finite Volume scheme on moving meshes. Before doing that we state the definitions for a (fixed) mesh and a moving mesh.

Definition 3.1 (Mesh in $\mathbb{R}^2$)**.** Assume P_3 being the set all 3-polygons with positive two-dimensional Hausdorff measure $|\cdot|_2$. Let $I\subset\mathbb{Z}$ be an index set.

Then we call

$$\tau=\{K_i\,|\,K_i\in P_3, i\in I\}$$

a *mesh of* $\mathbb{R}^2$ *with index set* I if the following conditions hold.

- $\mathbb{R}^2=\bigcup_{i\in I}K_i$
- $\mathring{K}_i\cap\mathring{K}_j=\emptyset\ \forall\, i,j\in I, i\neq j$
- Either $\left|\overline{K_i}\cap\overline{K_j}\right|_1=0$ or $\left|\overline{K_i}\cap\overline{K_j}\right|_1>0\Rightarrow\overline{K_i}\cap\overline{K_j}$ is a line segment between two common vertices of K_i and K_j. Here $|\cdot|_1$ denotes the one-dimensional Hausdorff measure.

We define $S_{i,j} = \overline{K_i} \cap \overline{K_j}$ and call it *edge*, if $|S_{i,j}|_1 > 0$, $i \neq j$. The index set of all edges is defined as

$$E = \{(i,j) \in I \times I \,|\, |S_{i,j}|_1 > 0\}.$$

For $i \in I$, the index set of all neighbors of K_i is given as $N(i) = \{j \in I \,|\, |S_{i,j}|_1 > 0\}$. For each edge $S_{i,j}$ we define $\mathbf{n}_{i,j} \in \mathcal{S}^1$ as the outer unit vector at $S_{i,j}$ w.r.t. K_i. The mesh width h and the smallest incircle diameter k are defined as

$$h := \max_{(i,j)\in E} \{|S_{i,j}|_1\} \text{ and } k := \min_{i \in I} \left\{ \frac{2\,|K_i|_2}{|\partial K_i|_1} \right\}.$$

In the sequel we will simply write $|\cdot|$ for the one- and two-dimensional Hausdorff measures $|\cdot|_1$ and $|\cdot|_2$, if the meaning is clear from the context.

Definition 3.2 (Moving mesh)**.** Let a mesh $\tau = \{K_i \,|\, K_i \in P_3, i \in I\}$ on $\mathbb{R}^2$ with an index set I and some interval $[t_1, t_2]$ be given. Assume that for each $i \in I$ there is a continuous function

$$\Phi_i : [t_1, t_2] \to \mathcal{L}_a(K_i, \mathbb{R}^2), t \mapsto \Phi_i^t,$$

with $\Phi_i^{t_1} = \mathrm{id}$, such that the global function

$$\Phi^t : \mathbb{R}^2 \to \mathbb{R}^2, x \mapsto \Phi_i^t(x) \text{ for } x \in K_i$$

is well-defined and onto. Here $\mathcal{L}_a(K_i, \mathbb{R}^2)$ denotes the space of affine mappings from K_i to $\mathbb{R}^2$.

Then we call $\mathcal{T} = (\tau, \{\Phi_i\}_{i\in I})$ a *moving mesh* for $[t_1, t_2]$. For all $t \in [t_1, t_2]$ the set

$$\tau(t) := \{\Phi_i^t(K_i)\}_{i \in I}$$

is the moving mesh at time t with index set I. We define the time dependent elements $K_i(t)$ and the time dependent edges $S_{i,j}(t)$ of the moving mesh $\mathcal{T} = (\tau, \{\Phi_i\}_{i\in I})$ by

$$K_i(t) := \Phi_i^t(K_i) \text{ and } S_{i,j}(t) := \Phi_i^t(S_{i,j}) = \Phi_j^t(S_{i,j}).$$

The equality $\Phi_i^t(S_{i,j}) = \Phi_j^t(S_{i,j})$ follows from the construction of Φ^t as a globally well-defined function.

Note that Definition 3.2 implies that the index set I remains invariant in time.

Suppose we have a moving mesh $(\tau, \{\Phi_i\}_{i\in I})$ of $\mathbb{R}^2$ with time dependent elements $K_i(t)$, $i \in I$. Then we can insert the set $\{(\mathbf{x}, t) \in \mathbb{R}^2 \times [t^n, t^{n+1}] \,|\, \mathbf{x} \in K_i(t)\}$ as a space-time cell K_{st} into equation (3.3). We split the boundary $\partial K_i(t)$ for fixed t into edges $S_{i,j}(t)$, $j \in N(i)$ and get the nth time step of a Finite Volume scheme on moving meshes of the form

$$\left|K_i(t^{n+1})\right| \mathbf{u}_i^{n+1} = |K_i(t^n)|\, \mathbf{u}_i^n - \Delta t^n \sum_{j \in N(i)} \left|S_{i,j}\left(t^{n+1/2}\right)\right| \left(\mathbf{g}_{i,j}^n(\mathbf{u}_i^n, \mathbf{u}_j^n) + \mathbf{h}_{i,j}^n(\mathbf{u}_i^n, \mathbf{u}_j^n) \right), \tag{3.4}$$

where $t^{n+1/2} = t^n + 0.5\Delta t^n$. The time step in the scheme (3.4) is given as $\Delta t^n := t^{n+1} - t^n$, and the terms $\mathbf{g}_{i,j}^n$ and $\mathbf{h}_{i,j}^n$ denote approximations of the average boundary fluxes

$$\frac{1}{\Delta t^n} \frac{1}{\left|S_{i,j}(t^{n+1/2})\right|} \int_{t^n}^{t^{n+1}} \int_{S_{i,j}(t)} \mathbf{f}(\mathbf{u}) \cdot \mathbf{n}_{i,j} \, \mathrm{d}A \, \mathrm{d}t \text{ and } -\frac{1}{\Delta t^n} \frac{1}{\left|S_{i,j}(t^{n+1/2})\right|} \int_{t^n}^{t^{n+1}} \int_{S_{i,j}(t)} \mathbf{s} \cdot \mathbf{n}_{i,j} \mathbf{u} \, \mathrm{d}A \, \mathrm{d}t,$$

respectively. The exact requirements to the flux terms $\mathbf{g}_{i,j}^n$ and $\mathbf{h}_{i,j}^n$ are stated in the following definition.

Definition 3.3 (Numerical and geometrical flux functions)**.** Let $(\tau, \{\Phi_i\}_{i\in I})$ be a moving mesh for the time interval $[t^n, t^{n+1}]$ and denote by $\mathbf{s}^{i,j}$ the speed of the midpoint of the edge $S_{i,j}$.

Then the functions $\mathbf{g}_{i,j}^n = \mathbf{g}_{i,j}^n(\mathbf{u}, \mathbf{v})$ and $\mathbf{h}_{i,j}^n = \mathbf{h}_{i,j}^n(\mathbf{u}, \mathbf{v})$ are called *numerical flux* and *geometrical flux function* for the system (2.3) if they are Lipschitz continuous in both arguments and fulfill the following properties.

- Consistency: $\forall\, \mathbf{u} \in \mathcal{U}\ \forall\, i \in I\ \forall\, j \in N(i)$:

$$\mathbf{g}^n_{i,j}(\mathbf{u},\mathbf{u}) = \mathbf{f}(\mathbf{u}) \cdot \mathbf{n}_{i,j}(t^{n+1/2}) \text{ and } \mathbf{h}^n_{i,j}(\mathbf{u},\mathbf{u}) = -\mathbf{n}_{i,j}(t^{n+1/2}) \cdot \mathbf{s}^{i,j}\,\mathbf{u}, \tag{3.5}$$

- Conservation: $\forall\, \mathbf{u},\mathbf{v} \in \mathcal{U}\ \forall\, i \in I\ \forall\, j \in N(i)$:

$$\mathbf{g}^n_{i,j}(\mathbf{u},\mathbf{v}) + \mathbf{h}^n_{i,j}(\mathbf{u},\mathbf{v}) = -\Big(\mathbf{g}^n_{j,i}(\mathbf{v},\mathbf{u}) + \mathbf{h}^n_{j,i}(\mathbf{v},\mathbf{u})\Big). \tag{3.6}$$

Common choices for the numerical flux are for example the Lax−Friedrichs flux, the Godunov-(type) flux or the Roe flux, see *e.g.* [22]. The geometrical flux $\mathbf{h}^n_{i,j}$ can be treated exactly the same as the numerical flux with the corresponding flux function $\mathbf{f}(\mathbf{u}) = -\mathbf{u}(\mathbf{s}^{i,j})^T$.

Formula (3.4) together with a numerical and geometrical flux function will define an iteration step of the Finite Volume scheme.

Definition 3.4 (Finite Volume step on moving meshes)**.** Let $\mathcal{T} = (\tau, \{\Phi_i\})$ be a moving mesh with index set I for the time interval $[t^n, t^{n+1}]$ and let $\{\mathbf{u}^n_i \in \mathcal{U}\}_{i\in I}$ be a set with elements in $\mathcal{U}$. The mapping

$$\text{FVS} : (\{\mathbf{u}^n_i\}_{i\in I}, \mathcal{T}) \mapsto \{\mathbf{u}^{n+1}_i\}_{i\in I}$$

is called *Finite Volume step* for (2.3), if the values $\mathbf{u}^{n+1}_i$ are computed from formula (3.4) with a numerical flux function $\mathbf{g}^n_{i,j}$ and a geometrical flux function $\mathbf{h}^n_{i,j}$. Furthermore we have for $n = 0$

$$\mathbf{u}^0_i = \frac{1}{|K_i(0)|}\int_{K_i(0)} \mathbf{u}_0 \,\mathrm{d}V. \tag{3.7}$$

The resulting Finite Volume scheme on moving meshes is summarized in the following algorithm.

Algorithm 1 (Finite Volume scheme on moving meshes for (2.3), (2.4))**.**

1: ***procedure*** FINITE VOLUME SCHEME$(\mathbf{u}_0, T, \mathcal{T})$
2: $t^0 = 0,\ n = 0$
3: $\{\mathbf{u}^0_i\}_{i\in I} = \left\{\frac{1}{|K_i(0)|}\int_{K_i(0)} \mathbf{u}_0 \,\mathrm{d}V\right\}_{i\in I}$ ▷ *Initial values*
4: ***while*** $t^n < T$ ***do***
5: $\Delta t^n = \min\limits_{i\in I}\left(|K_i(t^n)| \cdot |\partial K_i(t^n)|^{-1} \cdot \lambda_\infty(\mathbf{u}^n_i)^{-1}\right)$ ▷ *CFL condition*
6: $\{\mathbf{u}^{n+1}_i\}_{i\in I} = FVS(\{\mathbf{u}^n_i\}_{i\in I}, \mathcal{T})$ ▷ *Finite Volume Step*
7: $t^{n+1} = t^n + \Delta t^n,\ n = n+1$

By $\lambda_\infty(\mathbf{u})$ we denote the absolute value of the biggest eigenvalue of $D\mathbf{f}_1$ and $D\mathbf{f}_2$. The choice of the time step Δt^n in line 6 ensures that the CFL condition holds. With that scheme, we can define the approximate solution computed within Algorithm 1 as a piecewise constant function.

Definition 3.5. Let bounded initial data $\mathbf{u}_0$, and a moving mesh $\mathcal{T} = (\tau, \{\Phi_i\}_{i\in I})$ on $[0, T]$ be given.

Then, we define the *numerical approximation* $\mathbf{u}_h$ given by Algorithm 1 as piecewise constant function on the time dependent elements $K_i(t)$, that is

$$\mathbf{u}_h(t,x) = \mathbf{u}^n_i \qquad \text{if } t \in [t^n, t^{n+1}) \text{ and } \mathbf{x} \in K_i(t).$$

The following Lemmas 3.6, 3.8 and 3.9 state conservation and invariance properties of the Finite Volume scheme 1. Since the new Algorithm 2 for undercompressive shock waves presented within this paper will consist of a Finite Volume scheme for a certain choice of a moving mesh, we can reuse these Lemmata later on in order to prove properties of the new Algorithm 2.

Lemma 3.6 (Conservation property). *Consider Algorithm 1 for the initial value problem (2.3), (2.4). Assume that* $(\mathbf{u}_h(t^n,\cdot)-\mathbf{u}_0)\in L^1(\mathbb{R}^2)$ *holds for each* $n\in\{0,\dots,N\}$.

Then, Algorithm 1 is conservative for each $n\in\{0,\dots,N\}$, *i.e.*

$$\int_{\mathbb{R}^2}(\mathbf{u}_h(t^n,\cdot)-\mathbf{u}_0)\,\mathrm{d}V=0.$$

Proof. Substracting the local averages $\mathbf{u}_i^0$ from (3.4) and summing up with respect to $i\in I$, the conservation properties (3.6) give the statement. □

The next lemma is independent of the numerical scheme and a preparation for the proof of Lemma 3.8 below. It states how the change of cell measures of a moving mesh can be computed via the velocity of the boundary points.

Lemma 3.7 (Discrete Reynolds' transport theorem). *Let a moving mesh* $(\tau,\{\Phi_i\})$ *be given and* $K_i\in\tau$, $K_i(t)=\Phi_i^t(K_i)$.

Then we have for each $n\in\{0,\dots,N\}$ *and* $i\in I$

$$\big|K_i(t^{n+1})\big|-|K_i(t^n)|=\int_{t^n}^{t^{n+1}}\int_{\partial K_i(t)}\mathbf{s}\cdot\mathbf{n}\,\mathrm{d}A\,\mathrm{d}t,\tag{3.8}$$

where $\mathbf{n}$ *is the outer unit normal vector of* $K_i(t)$ *and* $\mathbf{s}=\mathbf{s}(t,\mathbf{x})=\frac{\mathrm{d}}{\mathrm{d}t}\Phi_i^t(\mathbf{x})$ *is the speed of a boundary point* $\mathbf{x}\in\partial K_i$.

Proof. The statement can be obtained by setting $\mathbf{u}$ as a constant function in (3.2). □

Lemma 3.8 (Invariance of constant initial datum). *Consider the Finite Volume Algorithm 1 on a moving mesh* $(\tau,\{\Phi_i\}_{i\in I})$ *for the system of hyperbolic conservation laws (2.3). Suppose that the initial data* $\mathbf{u}_0\in\mathcal{U}$ *is constant.*

Then, we have $\mathbf{u}_h(t,\mathbf{x})\equiv\mathbf{u}_0$ *for all times* $t\in[0,T]$ *and* $\mathbf{x}\in\mathbb{R}^2$.

Proof. This property can be shown by induction. Since the initial values $\mathbf{u}_0^n$ are computed as in (3.7) the desired property is fulfilled for $n=0$. For all $n>0$ we use the scheme (3.4) to compute $\mathbf{u}_i^{n+1}$, where we assume $\mathbf{u}_i^n\equiv\mathbf{u}_0$ for all $K_i(t^n)$, $i\in I$. Then the consistency properties (3.5) imply

$$\begin{aligned}\big|K_i(t^{n+1})\big|\,\mathbf{u}_i^{n+1}=&|K_i(t^n)|\,\mathbf{u}_0-\Delta t^n\sum_{j\in N(i)}\Big|S_{i,j}(t^{n+1/2})\Big|\,\mathbf{f}(\mathbf{u}_0)\cdot\mathbf{n}_{i,j}(t^{n+1/2})\\&+\Delta t^n\sum_{j\in N(i)}\Big|S_{i,j}(t^{n+1/2})\Big|\,\mathbf{n}_{i,j}(t^{n+1/2})\cdot\mathbf{s}^{i,j}\,\mathbf{u}_0.\end{aligned}\tag{3.9}$$

We know that the speed of the boundary points is constant in time and linear in space while the normal vector is constant in space and linear in time (restricted on $S_{i,j}$). Using an appropriate rotation matrix $\mathbf{R}\in\mathbb{R}^{2\times2}$ the normal vector $\mathbf{n}_{i,j}$ can be represented by

$$\mathbf{n}_{i,j}(t)=\frac{1}{|S_{i,j}(t)|}\mathbf{R}(\mathbf{p}_1^{i,j}(t)-\mathbf{p}_2^{i,j}(t)).$$

Then we compute

$$\begin{aligned}\Delta t^n \left|S_{i,j}(t^{n+1/2})\right| \mathbf{n}_{i,j}(t^{n+1/2}) \cdot \mathbf{s}^{i,j} &= \mathbf{R}\left(\Delta t^n \mathbf{p}_1^{i,j}(t^{n+1/2}) - \Delta t^n \mathbf{p}_2^{i,j}(t^{n+1/2})\right) \cdot \mathbf{s}^{i,j} \\ &= \mathbf{R}\int_{t^n}^{t^{n+1}} \mathbf{p}_1^{i,j}(t) - \mathbf{p}_2^{i,j}(t)\,\mathrm{d}t \cdot \mathbf{s}^{i,j} \\ &= \int_{t^n}^{t^{n+1}} |S_{i,j}(t)|\,\mathbf{n}_{i,j}(t)\,\mathrm{d}t \cdot \mathbf{s}^{i,j} \\ &= \int_{t^n}^{t^{n+1}} \int_{S_{i,j}(t)} \mathbf{s}\cdot\mathbf{n}_{i,j}\,\mathrm{d}A\,\mathrm{d}t.\end{aligned} \tag{3.10}$$

Using equality (3.10) and $\bigcup_{j\in N(i)} S_{i,j}(t) = \partial K_i(t)$ in (3.9) gives us

$$\begin{aligned}\left|K_i(t^{n+1})\right| \mathbf{u}_i^{n+1} &= |K_i(t^n)|\,\mathbf{u}_0 - \Delta t^n\,\mathbf{f}(\mathbf{u}_0) \underbrace{\int_{\partial K_i(t^{n+1/2})} \mathbf{n}(t^{n+1/2})\,\mathrm{d}A}_{=0} + \mathbf{u}_0 \underbrace{\int_{t^n}^{t^{n+1}} \int_{\partial K_i(t)} \mathbf{s}\cdot\mathbf{n}\,\mathrm{d}A\,\mathrm{d}t}_{\overset{(3.8)}{=}|K_i(t^{n+1})|-|K_i(t^n)|} \\ &= \left|K_i(t^{n+1})\right| \mathbf{u}_0,\end{aligned}$$

and we get immediately the result $\mathbf{u}_i^{n+1} \equiv \mathbf{u}_0$ for all $K_i(t^{n+1})$, $i \in I$. □

The next lemma prepares the core result Theorem 4.4 on our new Finite Volume moving mesh algorithm for two-phase problems. As long as the mesh can be aligned with the interface associated to a planar sharp interface solution of (2.3), (2.4) the numerical solution will reproduce this solution exactly. In Theorem 4.4 it is shown that such a mesh will be constructed within the algorithm.

Lemma 3.9 (Planar wave consistency, No. 1)**.**
Let an entropy-compatible sharp-interface solution of (2.3), (2.4) be given as

$$\mathbf{u}(t,\mathbf{x}) = \begin{cases}\mathbf{u}_L \in \mathcal{P}_- & \text{if} \quad \mathbf{x}\cdot\boldsymbol{\nu} + ct < 0, \\ \mathbf{u}_R \in \mathcal{P}_+ & \text{if} \quad \mathbf{x}\cdot\boldsymbol{\nu} + ct > 0,\end{cases} \qquad \Gamma(t) = \{\mathbf{x}\in\mathbb{R}^2 \,|\, \mathbf{x}\cdot\boldsymbol{\nu} + ct = 0\}$$

for a fixed constant $c \in \mathbb{R}$, a unit vector $\boldsymbol{\nu} \in \mathcal{S}^1$ and the initial state $\mathbf{u}_0 = \mathbf{u}(0,\cdot)$. Let a moving mesh $(\tau, \{\Phi_i\})$ be given such that $\Gamma(t)$ is exactly represented in the sense that

$$\exists\,\mathcal{E} \subset E: \bigcup_{(i,j)\in\mathcal{E}} S_{i,j}(0) = \Gamma(0) \text{ and } \Phi_i^t|_{\Gamma(0)\cap K_i}(\mathbf{x}) = \mathbf{x} + ct\boldsymbol{\nu} \;\; \forall\, i \in I. \tag{3.11}$$

Assume that interface solvers $\mathcal{R}_{\mathbf{f}\cdot\mathbf{n}}$ with $\mathbf{n} \in \mathcal{S}^1$ of (2.3) are given and that $\mathbf{u}_h$ is the numerical approximation given by Algorithm 1 computed with the following choice of numerical fluxes

$$\begin{aligned}\mathbf{g}_{i,j}^n(\mathbf{u},\mathbf{v}) &:= \begin{cases}\mathbf{f}(\mathbf{U}(\mathbf{u},\mathbf{v}))\cdot\mathbf{n}_{i,j} & \text{if} \quad \mathbf{u}\in\mathcal{P}_-, \mathbf{v}\in\mathcal{P}_+, \\ \mathbf{f}(\mathbf{U}(\mathbf{v},\mathbf{u}))\cdot\mathbf{n}_{i,j} & \text{if} \quad \mathbf{u}\in\mathcal{P}_+, \mathbf{v}\in\mathcal{P}_-, \\ \tilde{\mathbf{g}}_{i,j}^n(\mathbf{u},\mathbf{v}) & \text{otherwise,}\end{cases} \\ \mathbf{h}_{i,j}^n(\mathbf{u},\mathbf{v}) &:= \begin{cases}-\sigma(\mathbf{u},\mathbf{v})\mathbf{U}(\mathbf{u},\mathbf{v}) & \text{if} \quad \mathbf{u}\in\mathcal{P}_-, \mathbf{v}\in\mathcal{P}_+, \\ +\sigma(\mathbf{v},\mathbf{u})\mathbf{U}(\mathbf{v},\mathbf{u}) & \text{if} \quad \mathbf{u}\in\mathcal{P}_+, \mathbf{v}\in\mathcal{P}_-, \\ \tilde{\mathbf{h}}_{i,j}^n(\mathbf{u},\mathbf{v}) & \text{otherwise,}\end{cases}\end{aligned} \tag{3.12}$$

where the values $\sigma(\mathbf{u},\mathbf{v})$, $\mathbf{U}(\mathbf{u},\mathbf{v})$ *and* $\mathbf{V}(\mathbf{u},\mathbf{v})$ *are obtained from the interface solver* $\mathcal{R}_{\mathbf{f}\cdot\mathbf{n}_{i,j}}$:

$$(\sigma(\mathbf{u},\mathbf{v}),\mathbf{U}(\mathbf{u},\mathbf{v}),\mathbf{V}(\mathbf{u},\mathbf{v})) = \mathcal{R}_{\mathbf{f}\cdot\mathbf{n}_{i,j}}(\mathbf{u},\mathbf{v})$$

and $\tilde{\mathbf{g}}_{i,j}^n$, $\tilde{\mathbf{h}}_{i,j}^n$ *are arbitrary numerical and geometrical fluxes in the sense of Definition 3.3.*

Then $\mathbf{u}$ *is resolved exactly by* $\mathbf{u}_h$ *independent of the mesh parameter, i.e.* $\mathbf{u}_h = \mathbf{u}$ *a.e.*

Proof. Again, this proof can be done by induction. First, it is clear that the initial data is resolved exactly by the approximation $\mathbf{u}_h(0,\cdot)$, since the discontinuity in $\mathbf{u}_0$ is covered by the set of mesh edges $\mathcal{E}$. This implies

$$\text{either } \mathbf{x}\cdot\nu \le 0 \;\forall\, \mathbf{x}\in K_i(0), \text{ or } \mathbf{x}\cdot\nu \ge 0 \;\forall\, \mathbf{x}\in K_i(0).$$

Therefore, the initial data $\{\mathbf{u}_i^0\}_{i\in I}$ reads

$$\mathbf{u}_i^0 = \frac{1}{|K_i(0)|}\int_{K_i(0)} \mathbf{u}_0 \,\mathrm{d}V = \begin{cases} \mathbf{u}_L & \text{if} \quad \forall\, \mathbf{x}\in K_i(0) : \mathbf{x}\cdot\nu \le 0, \\ \mathbf{u}_R & \text{if} \quad \forall\, \mathbf{x}\in K_i(0) : \mathbf{x}\cdot\nu \ge 0. \end{cases}$$

We assume that $\mathbf{u}_h(t^n,\cdot) = \mathbf{u}(t^n,\cdot)$ is valid. Before proving the induction $\mathbf{u}_h(t^{n+1},\cdot) = \mathbf{u}(t^{n+1},\cdot)$, we first have to show that (3.12) are valid choices for the numerical and geometrical fluxes, *i.e.* they fulfill the consistency properties (3.5) and the conservation property (3.6).

The consistency is valid, since

$$\mathbf{g}_{i,j}^n(\mathbf{u},\mathbf{u}) = \tilde{\mathbf{g}}_{i,j}^n(\mathbf{u},\mathbf{u}), \quad \mathbf{h}_{i,j}^n(\mathbf{u},\mathbf{u}) = \tilde{\mathbf{h}}_{i,j}^n(\mathbf{u},\mathbf{u})$$

and $\tilde{\mathbf{g}}_{i,j}^n$ and $\tilde{\mathbf{h}}_{i,j}^n$ are consistent.

The conservation for values $\mathbf{u},\mathbf{v}$ in the same phase follows again from the conservation of $\tilde{\mathbf{g}}_{i,j}^n$ and $\tilde{\mathbf{h}}_{i,j}^n$. For $\mathbf{u}\in\mathcal{P}_-$, $\mathbf{v}\in\mathcal{P}_+$ (and vice versa) the conservation is also true, since

$$\begin{aligned} \mathbf{g}_{i,j}^n(\mathbf{u},\mathbf{v}) + \mathbf{h}_{i,j}^n(\mathbf{u},\mathbf{v}) &= \mathbf{f}(\mathbf{U}(\mathbf{u},\mathbf{v}))\cdot\mathbf{n}_{i,j} - \sigma(\mathbf{u},\mathbf{v})\mathbf{U}(\mathbf{u},\mathbf{v}) \\ &= -\left(\mathbf{f}(\mathbf{U}(\mathbf{u},\mathbf{v}))\cdot\mathbf{n}_{j,i} + \sigma(\mathbf{u},\mathbf{v})\mathbf{U}(\mathbf{u},\mathbf{v})\right) \\ &= -\left(\mathbf{g}_{j,i}^n(\mathbf{v},\mathbf{u}) + \mathbf{h}_{j,i}^n(\mathbf{v},\mathbf{u})\right). \end{aligned}$$

The induction $\mathbf{u}_h(t^{n+1},\cdot) = \mathbf{u}(t^{n+1},\cdot)$ will be shown as follows: the Finite Volume step (3.4) is either

$$\begin{aligned} \left|K_i(t^{n+1})\right|\mathbf{u}_i^{n+1} = |K_i(t^n)|\,\mathbf{u}_L &- \Delta t^n \sum_{j\in N(i),(i,j)\in\mathcal{E}} \left|S_{i,j}(t^{n+1/2})\right| \left(\mathbf{g}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_R) + \mathbf{h}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_R)\right) \\ &- \Delta t^n \sum_{j\in N(i),(i,j)\notin\mathcal{E}} \left|S_{i,j}(t^{n+1/2})\right| \left(\mathbf{g}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_L) + \mathbf{h}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_L)\right) \end{aligned}$$

or

$$\begin{aligned} \left|K_i(t^{n+1})\right|\mathbf{u}_i^{n+1} = |K_i(t^n)|\,\mathbf{u}_R &- \Delta t^n \sum_{j\in N(i),(i,j)\notin\mathcal{E}} \left|S_{i,j}(t^{n+1/2})\right| \left(\mathbf{g}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_R) + \mathbf{h}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_R)\right) \\ &- \Delta t^n \sum_{j\in N(i),(i,j)\in\mathcal{E}} \left|S_{i,j}(t^{n+1/2})\right| \left(\mathbf{g}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_L) + \mathbf{h}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_L)\right), \end{aligned}$$

since $\mathbf{u}_i^n$ is either equal to $\mathbf{u}_L$ or $\mathbf{u}_R$. This means that if we show the equalities

$$\mathbf{g}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_R) + \mathbf{h}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_R) = \mathbf{g}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_L) + \mathbf{h}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_L) \tag{3.13}$$

and

$$\mathbf{g}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_L)+\mathbf{h}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_L)=\mathbf{g}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_R)+\mathbf{h}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_R) \tag{3.14}$$

for $(i,j)\in\mathcal{E}$ the scheme can be reduced for each $\mathbf{u}_i^n$ to the case of constant initial data (either $\mathbf{u}_0\equiv\mathbf{u}_L$ or $\mathbf{u}_0\equiv\mathbf{u}_R$) as in Lemma 3.8 and we get the desired property $\mathbf{u}_i^n=\mathbf{u}_{L/R}\Rightarrow\mathbf{u}_i^{n+1}=\mathbf{u}_{L/R}$.

From the assumptions, we know that $\mathbf{u}_L$ and $\mathbf{u}_R$ fulfill (2.12) with $\sigma=c$. Therefore, the interface solver $\mathcal{R}_{\mathbf{f}\cdot\mathbf{n}_{i,j}}$ gives

$$\mathcal{R}_{\mathbf{f}\cdot\mathbf{n}_{i,j}}(\mathbf{u}_L,\mathbf{u}_R)=(\sigma(\mathbf{u}_L,\mathbf{u}_R),\mathbf{U}(\mathbf{u}_L,\mathbf{u}_R),\mathbf{V}(\mathbf{u}_L,\mathbf{u}_R))=(c,\mathbf{u}_L,\mathbf{u}_R)\,,$$

and the numerical and geometrical flux functions can be rewritten as

$$\mathbf{g}_{i,j}^n(\mathbf{u}_{L/R},\mathbf{u}_{R/L})=\mathbf{f}(\mathbf{U}(\mathbf{u}_L,\mathbf{u}_R))\cdot\mathbf{n}_{i,j}=\mathbf{f}(\mathbf{u}_L)\cdot\mathbf{n}_{i,j}, \tag{3.15}$$

and

$$\mathbf{h}_{i,j}^n(\mathbf{u}_{L/R},\mathbf{u}_{R/L})=\mp\sigma(\mathbf{u}_L,\mathbf{u}_R)\mathbf{U}(\mathbf{u}_L,\mathbf{u}_R)=\mp c\mathbf{u}_L. \tag{3.16}$$

The speed of the interface points $\mathbf{x}\in S_{i,j}$, $(i,j)\in\mathcal{E}$ is

$$\mathbf{s}(\mathbf{x})=\frac{\mathrm{d}}{\mathrm{d}t}\Phi_i^t(\mathbf{x})=c\boldsymbol{\nu}, \tag{3.17}$$

and the normal vector at edges $S_{i,j}(t)$, $(i,j)\in\mathcal{E}$ is

$$\mathbf{n}_{i,j}=\begin{cases}\boldsymbol{\nu} & \text{if} \quad \mathbf{u}_i^n\in\mathcal{P}_-,\mathbf{u}_j^n\in\mathcal{P}_+,\\ -\boldsymbol{\nu} & \text{if} \quad \mathbf{u}_i^n\in\mathcal{P}_+,\mathbf{u}_j^n\in\mathcal{P}_-,\end{cases} \tag{3.18}$$

since the edges are moved along $\{\mathbf{x}\in\mathbb{R}^2\,|\,\mathbf{x}\cdot\boldsymbol{\nu}+ct=0\}$. It follows for edges $(i,j)\in\mathcal{E}$

$$\mathbf{h}_{i,j}^n(\mathbf{u}_{L/R},\mathbf{u}_{L/R})\overset{(3.5)}{=}-\mathbf{s}^{i,j}\cdot n_{i,j}(t^{n+1/2})\,\mathbf{u}_{L/R}\overset{(3.17),\,(3.18)}{=}-c\boldsymbol{\nu}\cdot(\pm\boldsymbol{\nu})\,\mathbf{u}_{L/R}=\mp c\,\mathbf{u}_{L/R}. \tag{3.19}$$

The equality (3.13) then follows as

$$\mathbf{g}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_R)+\mathbf{h}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_R)\overset{(3.15),\,(3.16)}{=}\mathbf{f}(\mathbf{u}_L)\cdot\mathbf{n}_{i,j}-c\mathbf{u}_L\overset{(3.5),\,(3.19)}{=}\mathbf{g}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_L)+\mathbf{h}_{i,j}^n(\mathbf{u}_L,\mathbf{u}_L).$$

Equation (3.14) follows similarly with the usage of the Rankine–Hugoniot conditions $(2.12)_1$

$$\begin{aligned}\mathbf{g}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_L)+\mathbf{h}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_L)&\overset{(3.15),\,(3.16)}{=}\mathbf{f}(\mathbf{u}_L)\cdot\mathbf{n}_{i,j}+c\mathbf{u}_L\\ &\overset{(3.18)}{=}\mathbf{f}(\mathbf{u}_L)\cdot\mathbf{n}_{i,j}-c\boldsymbol{\nu}\cdot\mathbf{n}_{i,j}\mathbf{u}_L\\ &\overset{(2.12)_1}{=}\mathbf{f}(\mathbf{u}_R)\cdot\mathbf{n}_{i,j}-c\boldsymbol{\nu}\cdot\mathbf{n}_{i,j}\mathbf{u}_R\\ &\overset{(3.18)}{=}\mathbf{f}(\mathbf{u}_R)\cdot\mathbf{n}_{i,j}+c\mathbf{u}_R\\ &\overset{(3.5),\,(3.19)}{=}\mathbf{g}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_R)+\mathbf{h}_{i,j}^n(\mathbf{u}_R,\mathbf{u}_R).\end{aligned}$$

□

To introduce the complete algorithm for the approximation of a sharp interface solution for (2.3), (2.4) we connect the moving mesh method with a special tracking for the phase boundary.

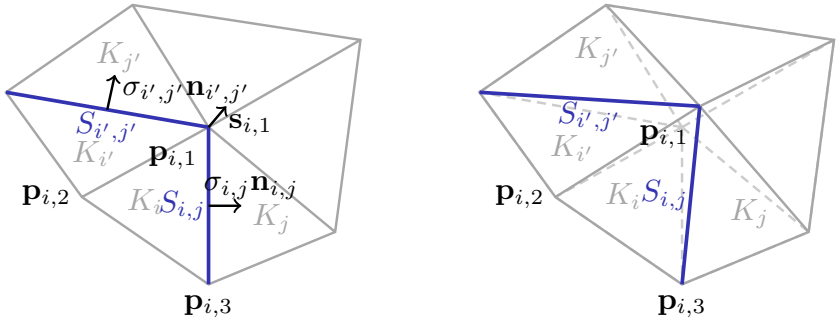


FIGURE 4. Motion of the interface curve: Mesh at time $t = t^n$ (*left figure*) and at time $t = t^{n+1}$.

3.1.2. Interface Tracking

The previous Lemma 3.9 motivates the following procedure. If we move interface edges in a way such that the position of the phase transition is tracked, we can on the one hand treat the interface separately, *e.g.* like in equations (3.12) in the case $\pi(\mathbf{u}) \neq \pi(\mathbf{v})$, and on the other hand we do not have any smearing across this curve due to averaging. The second property is crucial for computations of any generic two-phase system from Section 2 with splitted state space $\mathcal{U} = \mathcal{P}_- \cup \mathcal{P}_+$ with $\overline{\mathcal{P}_-} \cap \overline{\mathcal{P}_+} = \emptyset$, since it ensures the desired property $\mathbf{u}_h \in \mathcal{P}_- \cup \mathcal{P}_+$. The computation of the approximate location of the interface is illustrated in Figure 4 and will define the moving mesh $\mathcal{T} = (\tau, \{\Phi_i\})$ we will need as an input for the Finite Volume step in Definition 3.4.

Therefore we introduce formally, in addition to a mesh, the set of interface edges.

Definition 3.10 (Interface edges, approximate interface)**.** Let a fixed mesh τ be given with index set I and a set $\{\mathbf{u}_i \in \mathcal{U}\}_{i\in I}$. We define the *interface edge indices* as

$$\mathcal{E} = \mathcal{E}(\tau, \{\mathbf{u}_i\}_{i\in I}) = \{(i,j) \in E \,|\, \pi(\mathbf{u}_i) \neq \pi(\mathbf{u}_j)\}$$

and the *approximate interface*

$$\Gamma_h = \Gamma_h(\tau, \{\mathbf{u}_i\}_{i\in I}) = \bigcup_{(i,j)\in\mathcal{E}(\tau,\{\mathbf{u}_i\})} S_{i,j}.$$

An approximate interface is called *admissible* if it consists of one or more closed curves without any (self-)intersections.

Remark 3.11. The closedness of interfaces is only demanded in order to ease the formulation of the algorithm. Nevertheless, non-closed curves are a relevant case and appear in the numerical examples in Section 5. There we will detail the necessary algorithmical changes.

We know that if we have an admissible approximate interface $\Gamma_h(\tau, \{\mathbf{u}_i\}_{i\in I})$ each vertex that is part of the interface curve has exactly two incident edges in the mesh τ. Let an interface solver $\mathcal{R}$ for (2.3) and an admissible approximate interface at time t^n be given. The evolution of the approximate interface is computed with the following procedure. For each interface edge $S_{i,j}$, $(i,j) \in \mathcal{E}$ with $\mathbf{u}_i \in \mathcal{P}_-$ and $\mathbf{u}_j \in \mathcal{P}_+$ we consider the Riemann problem (2.11) in direction of the normal vector $\mathbf{n}_{i,j} = (n_{i,j}^{(1)}, n_{i,j}^{(2)})^T$

$$\mathbf{w}_t + \left(n_{i,j}^{(1)}\mathbf{f}_1(\mathbf{w}) + n_{i,j}^{(2)}\mathbf{f}_2(\mathbf{w})\right)_x = 0, \quad \mathbf{w}(0,x) = \begin{cases} \mathbf{u}_i & \text{if} \quad x < 0, \\ \mathbf{u}_j & \text{if} \quad x > 0, \end{cases} \tag{3.20}$$

and denote by $\sigma_{i,j}$, $\mathbf{U}_{i,j}$, $\mathbf{V}_{i,j}$ the corresponding values of the interface solver

$$(\sigma_{i,j}(\mathbf{u}_i,\mathbf{u}_j),\mathbf{U}_{i,j}(\mathbf{u}_i,\mathbf{u}_j),\mathbf{V}_{i,j}(\mathbf{u}_i,\mathbf{u}_j)) = \mathcal{R}_{n_{i,j}^{(1)}\mathbf{f}_1+n_{i,j}^{(2)}\mathbf{f}_2}(\mathbf{u}_i,\mathbf{u}_j). \tag{3.21}$$

Suppose $\mathbf{p}_{i,k}$ ($i \in I$, $k = 1,2,3$) being the kth vertex of K_i in a mesh τ and being part of the approximate interface Γ_h. Then $\mathbf{p}_{i,k}$ has exactly two incident edges in the mesh, which separate two phases, since Γ_h is a set of closed, non-intersecting curves. We call these two incident edges $S_{i,j}$ and $S_{i',j'}$, see Figure 4. We define the speed of $\mathbf{p}_{i,k}$ as the weighted average speed computed at $S_{i,j}$ and $S_{i',j'}$

$$\mathbf{s}_{i,k} := \frac{\mathbf{n}_{i,j}\sigma_{i,j}(\mathbf{u}_i,\mathbf{u}_j)\,|S_{i,j}| + \mathbf{n}_{i',j'}\sigma_{i',j'}(\mathbf{u}_{i'},\mathbf{u}_{j'})\,|S_{i',j'}|}{|S_{i,j}| + |S_{i',j'}|} \tag{3.22}$$

and the time dependent point $\mathbf{p}_{i,k}(t)$ as

$$\mathbf{p}_{i,k}(t) := \mathbf{p}_{i,k}(t^n) + (t-t^n)\mathbf{s}_{i,k}.$$

The moving mesh is then defined as $(\tau,\{\Phi_i\}_{i\in I})$ with

$$\Phi_i^t(\mathbf{x}) := \mathbf{x} + (t-t^n)\Big(\mathbf{s}_{i,3} + \lambda_{i,1}(\mathbf{x})(\mathbf{s}_{i,1}-\mathbf{s}_{i,3}) + \lambda_{i,2}(\mathbf{x})(\mathbf{s}_{i,2}-\mathbf{s}_{i,3})\Big), \tag{3.23}$$

where $\lambda_{i,1}(\mathbf{x})$, $\lambda_{i,2}(\mathbf{x})$ denote the barycentric coordinates of $\mathbf{x}$ in K_i, *i.e.*

$$(\lambda_{i,1},\lambda_{i,2}) = (\mathbf{p}_{i,1}-\mathbf{p}_{i,3},\mathbf{p}_{i,2}-\mathbf{p}_{i,3})^{-1}(\mathbf{x}-\mathbf{p}_{i,3}).$$

We want to stress that even if only points of the interface are moved we get time dependent edges that are not necessarily interface edges, see Figure 4. Therefore, numerical fluxes for the term $-\mathbf{s}\cdot\mathbf{n}_{i,j}\mathbf{u}$ are required at interface edges as well as at non-interface edges.

Definition 3.12 (Interface motion function)**.** Let τ be a mesh with index set I and let $\{\mathbf{u}_i \in \mathcal{U}\}_{i\in I}$ be a set with elements in $\mathcal{U}$ such that the approximate interface $\Gamma_h(\tau,\{\mathbf{u}_i\})$ is admissible. Let also a time t^n and a time step Δt^n be given.

The mapping

$$\mathrm{IMF} : (t^n,\Delta t^n,\tau,\{\mathbf{u}_i^n\}_{i\in I}) \mapsto (\tau,\{\Phi_i\}_{i\in I})$$

is called *interface motion function*, if Φ_i, $i \in I$, is computed from formulas (3.21), (3.22) and (3.23).

With the interface motion function from Definition 3.12 we could now define directly the final algorithm. However, the motion of the interface described in Section 3.1.2 may cause narrow or even degenerate cells, see Figure 5. Therefore, we will define in an additional post-processing step a new mesh, where the triangles will have a better volume to perimeter ratio. This post-processing step will be called remeshing.

3.1.3. Remeshing

Definition 3.13 (Remeshing)**.** Let τ and $\hat\tau$ be meshes with index sets I and $\hat I$, respectively such that $I \triangle \hat I = I\setminus\hat I \cup \hat I\setminus I$ is finite and $K_i = \hat K_i$ for all $i \in I\cap\hat I$. Let $\{\mathbf{u}_i \in \mathcal{U}\}_{i\in I}$, $\{\hat{\mathbf{u}}_{\hat\imath} \in \mathcal{U}\}_{\hat\imath\in\hat I}$ be sets with elements from $\mathcal{U}$ such that $\Gamma_h(\tau,\{\mathbf{u}_i\})$ and $\Gamma_h(\hat\tau,\{\hat{\mathbf{u}}_{\hat\imath}\})$ are admissible approximate interfaces. An operator

$$\mathrm{RS} : (\tau,\{\mathbf{u}_i\}_{i\in I}) \mapsto (\hat\tau,\{\hat{\mathbf{u}}_{\hat\imath}\}_{\hat\imath\in\hat I})$$

is called *remeshing operator* with respect to the constants $C > c > 0$ if the following conditions hold.

- The (incircle) diameters are bounded according to

$$c \le k(\hat\tau) \le h(\hat\tau) \le C. \tag{3.24}$$

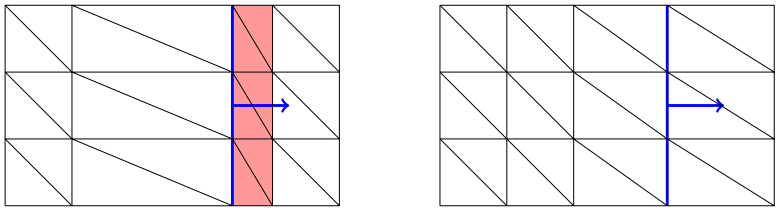


FIGURE 5. Remeshing technique: Extract of a mesh τ (with piecewise constant data $\mathbf{u}_h$) in the left figure and mesh $\hat{\tau}$ (with piecewise constant data $\hat{\mathbf{u}}_h$) in the right figure.

- The approximate interface Γ_h is invariant under remeshing, that is

$$\Gamma_h(\tau, \{\mathbf{u}_i\}_{i\in I}) = \Gamma_h(\hat{\tau}, \{\hat{\mathbf{u}}_i\}_{\hat{i}\in\hat{I}}). \tag{3.25}$$

- Conservation of mass is fullfilled according to

$$\sum_{i\in I\setminus\hat{I}} \sum_{\hat{i}\in\hat{I}\setminus I} \left|K_i \cap \hat{K}_{\hat{i}}\right| (\mathbf{u}_i - \hat{\mathbf{u}}_{\hat{i}}) = 0, \text{and } \mathbf{u}_i = \hat{\mathbf{u}}_i \; \forall\, i \in I \cap \hat{I}. \tag{3.26}$$

- The fixing criterion

$$\sum_{i\in I\setminus\hat{I}} \sum_{\hat{i}\in\hat{I}\setminus I} \left|K_i \cap \hat{K}_{\hat{i}}\right| |\mathbf{u}_i - \hat{\mathbf{u}}_{\hat{i}}|^2 \longrightarrow min. \tag{3.27}$$

Remark 3.14.

- Condition (3.24) is a condition to the new mesh $\hat{\tau}$ and ensures, that narrow triangles are avoided. Conditions (3.26) and (3.27) are conditions to the new data $\hat{\mathbf{u}}_h$ defined on the new mesh, whereas (3.25) is a compatibility condition to both mesh $\hat{\tau}$ and data $\{\hat{\mathbf{u}}_i\}_{i\in I}$ and ensures, that the position of the interface is not altered by the remeshing.
- The remeshing operator is neither unique, nor does it exist for arbitrary choices of constants C, c. We will use conditions (3.25) and (3.27) for proving the properties of the scheme in Section 4 but for the numerical computations we will omit these conditions in cases where the remeshing operator does not exist. This includes the case, where the curvature of the approximate interface is increasing (*e.g.* a shrinking circle). The cases with topology changes (interface - boundary interaction, interface - interface interaction) and phase extinction are not covered by the following Algorithm 2.
- If $c \leq k(\tau)$ and $C \geq h(\tau)$ a trivial remeshing operator is given as $\mathrm{RS}(\tau, \{\mathbf{u}_i\}) = (\tau, \{\mathbf{u}_i\})$. Contrary, if one wants to construct a new mesh $\hat{\tau}$ with improved mesh width to incircle diameter ratio $h(\hat{\tau})/k(\hat{\tau})$ one has to chose the constants c and C as $c \geq k(\tau)$ and $C \leq h(\tau)$. This means that a non-trivial remeshing is necessary.

Now we are in the position to prove one of the major results. Using the remeshing operator from Definition 3.13 we are able to guarantee that there occurs no phase smearing in our new algorithm.

Lemma 3.15. *Suppose a mesh τ is given with values $\{\mathbf{u}_i \in \mathcal{P}_- \cup \mathcal{P}_+\}_{i\in I}$.*

Then the resulting values $\{\hat{\mathbf{u}}_{\hat{i}}\}_{\hat{i}\in\hat{I}}$ of a remeshing step defined as in Definition 3.13 are all in $\mathcal{P}_-$ or $\mathcal{P}_+$, that is

$$\forall\, \hat{i} \in \hat{I} : \; \hat{\mathbf{u}}_{\hat{i}} \in \mathcal{P}_- \cup \mathcal{P}_+.$$

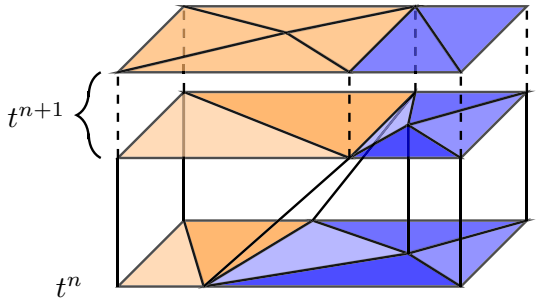


FIGURE 6. Illustration of the evolution of the moving mesh including one Finite-Volume step and Remeshing. Narrow elements are resolved by the remeshing step.

Proof. For all $i \in I \cap \hat{I}$ we have the equality $\hat{\mathbf{u}}_i = \mathbf{u}_i \in \mathcal{P}_- \cup \mathcal{P}_+$. For all other indices in $I \triangle \hat{I}$ the variational problem (3.27) under the constraint (3.26) has the Lagrangian

$$L(\hat{\mathbf{u}}, \lambda) = \sum_{\hat{\imath} \in \hat{I} \setminus I} \sum_{i \in I \setminus \hat{I}} \left| K_i \cap \hat{K}_{\hat{\imath}} \right| |\mathbf{u}_i - \hat{\mathbf{u}}_{\hat{\imath}}|^2 + \lambda \sum_{\hat{\imath} \in \hat{I} \setminus I} \sum_{i \in I \setminus \hat{I}} \left| K_i \cap \hat{K}_{\hat{\imath}} \right| (\mathbf{u}_i - \hat{\mathbf{u}}_{\hat{\imath}}),$$

with $\hat{\mathbf{u}} = (\hat{\mathbf{u}}_{\hat{\imath}})_{\hat{\imath} \in \hat{I} \setminus I}$ and the Lagrange multiplier $\lambda \in \mathbb{R}$. Taking the derivative with respect to $\hat{\mathbf{u}}_{\hat{\imath}}$ gives

$$0 = \frac{\mathrm{d}L}{\mathrm{d}\hat{\mathbf{u}}_{\hat{\imath}}}(\hat{\mathbf{u}}, \lambda) = \sum_{i \in I \setminus \hat{I}} \left| K_i \cap \hat{K}_{\hat{\imath}} \right| 2(\mathbf{u}_i - \hat{\mathbf{u}}_{\hat{\imath}}) - \lambda \sum_{i \in I \setminus \hat{I}} \left| K_i \cap \hat{K}_{\hat{\imath}} \right| = \sum_{i \in I \setminus \hat{I}} \left| K_i \cap \hat{K}_{\hat{\imath}} \right| 2\mathbf{u}_i - 2 \left| \hat{K}_{\hat{\imath}} \right| \hat{\mathbf{u}}_{\hat{\imath}} - \lambda \left| \hat{K}_{\hat{\imath}} \right|,$$

which gives the expression for $\hat{\mathbf{u}}_{\hat{\imath}}$

$$\hat{\mathbf{u}}_{\hat{\imath}} = \sum_{i \in I \setminus \hat{I}} \left(\frac{|K_i \cap \hat{K}_{\hat{\imath}}|}{|\hat{K}_{\hat{\imath}}|} \mathbf{u}_i \right) - \frac{\lambda}{2}. \tag{3.28}$$

Rewriting the constraint (3.26) and inserting (3.28) gives the equation

$$0 = \sum_{i \in I \setminus \hat{I}} \sum_{\hat{\imath} \in \hat{I} \setminus I} |K_i \cap \hat{K}_{\hat{\imath}}|(\mathbf{u}_i - \hat{\mathbf{u}}_{\hat{\imath}}) = \sum_{i \in I \setminus \hat{I}} |K_i| \mathbf{u}_i - \sum_{\hat{\imath} \in \hat{I} \setminus I} |\hat{K}_{\hat{\imath}}| \hat{\mathbf{u}}_{\hat{\imath}} = \underbrace{\sum_{i \in I \setminus \hat{I}} |K_i| \, \mathbf{u}_i - \sum_{i \in I \setminus \hat{I}} |K_i| \mathbf{u}_i}_{=0} + \underbrace{\sum_{\hat{\imath} \in \hat{I} \setminus I} |\hat{K}_{\hat{\imath}}|}_{\neq 0} \frac{\lambda}{2},$$

which implies that $\lambda = 0$. Therefore, we get

$$\hat{\mathbf{u}}_{\hat{\imath}} = \sum_{i \in I} \frac{|K_i \cap \hat{K}_{\hat{\imath}}|}{|\hat{K}_{\hat{\imath}}|} \mathbf{u}_i \quad \forall \hat{\imath} \in \hat{I}. \tag{3.29}$$

Recall that the new approximate interface $\Gamma_h(\hat{\mathcal{T}})$ coincides with the approximate interface $\Gamma_h(\mathcal{T})$ by condition (3.25). Thus the non-vanishing terms in sum (3.29) form a convex combination of values $\mathbf{u}_i$ either all in $\mathcal{P}_-$ or all in $\mathcal{P}_+$. Since $\mathcal{P}_-$ and $\mathcal{P}_+$ are both convex, we get $\hat{\mathbf{u}}_{\hat{\imath}} \in \mathcal{P}_- \cup \mathcal{P}_+$, as desired. □

3.2. The complete Finite Volume moving mesh algorithm

Now we are able to formulate the complete algorithm, that consists of an interface tracking realized with moving meshes, suitable Finite Volume schemes and a remeshing step. Due to the remeshing step we will get for each time interval $[t^n, t^{n+1})$ a new (moving) mesh. We will denote by τ^n meshes with index sets I^n and by $\mathcal{T}^n = (\tau^n, \{\Phi_i\}_{i\in I^n})$ moving meshes for time intervals $[t^n, t^{n+1}]$. By $\mathcal{T}^n(t)$ we denote the fixed mesh that results from the evolution of the moving mesh at time t, *i.e.*

$$\mathcal{T}^n(t) := \{\Phi_i^t(K_i) \,|\, i \in I^n, K_i \in \tau^n\}.$$

With this notation the new algorithm can be stated.

Algorithm 2 (Tracking-Type Algorithm)**.**

1: ***procedure*** TRACKING-TYPE ALGORITHM$(\mathbf{u}_0, T, \tau^0)$
2: $t^0 = 0$, $n = 0$
3: $\{\mathbf{u}_i^0\}_{i\in I^0} = \left\{ \frac{1}{|K_i(0)|} \int_{K_i(0)} \mathbf{u}_0 \, \mathrm{d}V \right\}_{i\in I^0}$ ▷ *Initial values*
4: ***while*** $t^n < T$ ***do***
5: $\Delta t^n = \frac{0.9}{\sigma} \min_{i\in I^n} \left(\frac{1}{2} |K_i(t^n)| \cdot |\partial K_i(t^n)|^{-1}, \sqrt{\frac{1}{6} |K_i(t^n)|} \right)$ ▷ *Time step restriction*
6: $\mathcal{T}^n = IMF(t^n, \Delta t, \tau^n, \{\mathbf{u}_i^n\}_{i\in I^n})$ ▷ *Compute Interface Motion*
7: $\{\hat{\mathbf{u}}_i^{n+1}\}_{i\in I^n} = FVS(\{\mathbf{u}_i^n\}_{i\in I^n}, \mathcal{T}^n)$ ▷ *Apply Finite Volume Scheme*
8: $(\{\mathbf{u}_i^{n+1}\}_{i\in I^{n+1}}, \tau^{n+1}) = RS(\{\hat{\mathbf{u}}_i^{n+1}\}_{i\in I^n}, \mathcal{T}(t^{n+1}))$ ▷ *Remesh*
9: $t^{n+1} = t^n + \Delta t$, $n = n + 1$

Note that the time step restriction of line 5 will be justified by the stability properties that will be derived in Lemmata 4.1, 4.2 below.

We conclude the section with a formal definition of the approximate solution as discrete counterpart of the sharp-interface solution from Definition 2.1.

Definition 3.16. Let bounded initial data $\mathbf{u}_0$, a time $T > 0$ and a (fixed) mesh τ^0 be given and assume that $\mathcal{T}^n$ are the moving meshes and $\{\mathbf{u}_i^n\}_{i\in I^n}$ the values computed within Algorithm 2.

Then, we define the *numerical approximation* $\mathbf{u}_h$ given by Algorithm 2 as piecewise constant function on the time dependent elements $K_i(t)$, that is

$$\mathbf{u}_h(t,x) = \mathbf{u}_i^n \qquad \text{if} \qquad t \in [t^n, t^{n+1}) \qquad \text{and} \qquad \mathbf{x} \in K_i(t) \text{ for } i \in I^n$$

and the *approximate interface* Γ_h as

$$\Gamma_h(t) = \Gamma_h(\mathcal{T}^n(t), \{\mathbf{u}_i^n\}_{i\in I^n}) \qquad \text{if } t \in [t^n, t^{n+1}).$$

4. Properties of the scheme

In this section basic properties of the Finite Volume moving mesh algorithm will be shown. These properties show the advantages of the moving mesh ansatz, *e.g.* the conservation property of the algorithm or the exact resolution of planar two-phase waves.

First of all, we have to show the well-posedness of Algorithm 2. The Algorithm might fail due to two reasons. First, the motion of the interface points might lead to a degeneration of the mesh, *i.e.* the mesh defined within Algorithm 2 might not fulfill the properties given in Definition 3.2. This failure will be excluded with Lemma 4.1. Second, the time steps Δt^n might be decreasing such that the given final time T is never reached. Lemma 4.2 will show that under a certain *a priori* condition to the data computed within the Algorithm, the time step has a lower bound and thus the algorithm terminates.

Lemma 4.1 (Mesh does not degenerate)**.** *Let $(\tau, \{\Phi_i\}_{i\in I})$ be a moving mesh and assume that an upper bound to the maximum speed of the mesh points $\sigma > \sup\{\left\|\frac{\mathrm{d}}{\mathrm{d}t}\Phi_i^t(\mathbf{x})\right\| \mid i \in I, \mathbf{x} \in K_i\} \geq 0$ exists. If the time step is chosen according to*

$$0 < \Delta t^n \leq \frac{1}{\sigma} \min\left(p\frac{|K_i(t^n)|}{|\partial K_i(t^n)|}, \sqrt{\frac{1-p}{3}\,|K_i(t^n)|}\right) \tag{4.1}$$

for a constant $0 < p < 1$, then the mesh elements do not degenerate, which means

$$\forall\, t \in [t^n, t^{n+1}]\ \forall\, i \in I:\ |K_i(t)| > 0.$$

Proof. The size of a triangle $K_i(t)$ can be computed with formula (3.8) and we get

$$\begin{aligned} |K_i(t)| &= |K_i(t^n)| + \int_{t^n}^{t}\int_{\partial K_i(t)} \mathbf{s}\cdot\mathbf{n}\,\mathrm{d}A\,\mathrm{d}t \geq |K_i(t^n)| - \int_{t^n}^{t}\int_{\partial K_i(t)} \|\mathbf{s}\|_2\,\mathrm{d}A\,\mathrm{d}t \\ &\geq |K_i(t^n)| - \sigma\int_{t^n}^{t}\int_{\partial K_i(t)} \mathrm{d}A\,\mathrm{d}t = |K_i(t^n)| - \sigma\int_{t^n}^{t} |\partial K_i(t)|\,\mathrm{d}t. \end{aligned}$$

The boundary $\partial K_i(t)$ consists of three edges, that grow each at most with 2σ, thus

$$|\partial K_i(t)| \leq |\partial K_i(t^n)| + 6\sigma(t - t^n),$$

and $|K_i(t)|$ can further be estimated under the time step restriction (4.1) as

$$|K_i(t)| \geq |K_i(t^n)| - \sigma\int_{t^n}^{t} |\partial K_i(t^n)| + 6\sigma(t-t^n)\,\mathrm{d}t > |K_i(t^n)| - p\,|K_i(t^n)| - (1-p)\,|K_i(t^n)| = 0. \qquad \square$$

With an additional and natural *a priori* assumption on the data $\{\mathbf{u}_i^n\}$ we can show that not only the mesh does not degenerate but also the Algorithm terminates after a finite amount of time iterations.

Lemma 4.2 (Algorithm 2 terminates)**.** *Suppose that the conditions of Lemma 4.1 are fulfilled with equality in (4.1) and that there is a bounded set $K \subset \mathcal{P}_- \cup \mathcal{P}_+$ such that for each time step the computed values $\{\mathbf{u}_i^n\}$ stay in K. Then, Algorithm 2 terminates.*

Proof. Remeshing ensures $c \leq k = \min_{i\in I}(2\,|K_i(t^n)|\cdot|\partial K_i(t^n)|^{-1})$ with (3.24) and also $\pi c^2 < |K_i(t^n)|$. Therefore the time step can be estimated as

$$\Delta t^n = \frac{1}{\sigma}\min_{i\in I}\left(p\frac{|K_i(t^n)|}{|\partial K_i(t^n)|}, \sqrt{\frac{1-p}{3}\,|K_i(t^n)|}\right) \geq \frac{c}{\sigma}\min\left(\frac{p}{2}, \sqrt{\frac{1-p}{3}\pi}\right) =: \tilde{c}(p) > 0.$$

This ensures that the Algorithm has at most $\lceil T/\tilde{c}(p)\rceil$ iterations. $\square$

The next lemma will show the conservation property, which does not hold for *e.g.* the ghost fluid method for two-phase problems [15,30].

Lemma 4.3 (Conservation property)**.** *Consider the tracking-type Algorithm 2 for the problem (2.3), (2.4) with bounded initial datum. Assume that the numerical approximation $\mathbf{u}_h$ is such that $(\mathbf{u}_h(t^n,\cdot) - \mathbf{u}_0) \in L^1(\mathbb{R}^2)$ holds for each $n \in \{0,\dots,N\}$.*

Then we have for each $n \in \{0,\dots,N\}$ the conservation property

$$\int_{\mathbb{R}^2} (\mathbf{u}_h(t^n,\cdot) - \mathbf{u}_0)\,\mathrm{d}V = 0.$$

Proof. The conservation property of the Finite Volume scheme on moving meshes is stated in Lemma 3.6. The remeshing step RS is conservative, see (3.26), such that Algorithm 2 is conservative. □

The final theorem of this section will present a property that is valid due to the moving mesh ansatz, similar to Lemma 3.9. It ensures the exact and sharp resolution of the entropy-compatible sharp-interface solution in the case of a single planar phase transition, independent of the coarsity of the mesh. Note that, unlike in Lemma 3.9, the moving mesh and thus the motion of the interface points is not prescribed, but given through Algorithm 2.

Theorem 4.4 (Planar wave consistency, No. 2)**.** *Let an entropy-compatible sharp-interface solution of* (2.3), (2.4) *be given as*

$$\mathbf{u}(t,\mathbf{x}) = \begin{cases} \mathbf{u}_L \in \mathcal{P}_- & \text{if} \quad \mathbf{x}\cdot\boldsymbol{\nu} + ct < 0, \\ \mathbf{u}_R \in \mathcal{P}_+ & \text{if} \quad \mathbf{x}\cdot\boldsymbol{\nu} + ct > 0, \end{cases} \qquad D_\pm(t) = \{\mathbf{x}\in\mathbb{R}^2 \,|\, \mathbf{x}\cdot\boldsymbol{\nu} + ct \lessgtr 0\}$$

for a fixed constant $c\in\mathbb{R}$, a unit vector $\boldsymbol{\nu}\in\mathcal{S}^1$ and the initial state $\mathbf{u}_0 = \mathbf{u}(0,\cdot)$. Let for the initial time $t=0$ a mesh τ_0 be given, such that the interface curve $\Gamma_h(\tau_0, \{\mathbf{u}_i^0\}_{i\in I})$ coincides with the sharp interface of the problem $\Gamma(0) = \{\mathbf{x}\in\mathbb{R}^2 \,|\, \mathbf{x}\cdot\boldsymbol{\nu} = 0\}$. Assume that interface solvers $\mathcal{R}_{\mathbf{f}\cdot\mathbf{n}}$, $\mathbf{n}\in\mathcal{S}^1$, of (2.3) are given and that $\mathbf{u}_h$ is the numerical approximation given by Algorithm 2 computed with the following choice of numerical fluxes

$$\mathbf{g}_{i,j}^n(\mathbf{u},\mathbf{v}) = \begin{cases} \mathbf{f}(\mathbf{U}(\mathbf{u},\mathbf{v}))\cdot\mathbf{n}_{i,j} & \text{if} \quad \mathbf{u}\in\mathcal{P}_-, \mathbf{v}\in\mathcal{P}_+, \\ \mathbf{f}(\mathbf{U}(\mathbf{v},\mathbf{u}))\cdot\mathbf{n}_{i,j} & \text{if} \quad \mathbf{u}\in\mathcal{P}_+, \mathbf{v}\in\mathcal{P}_-, \\ \tilde{\mathbf{g}}_{i,j}^n(\mathbf{u},\mathbf{v}) & \text{otherwise,} \end{cases}$$

$$\mathbf{h}_{i,j}^n(\mathbf{u},\mathbf{v}) = \begin{cases} -\sigma(\mathbf{u},\mathbf{v})\mathbf{U}(\mathbf{u},\mathbf{v}) & \text{if} \quad \mathbf{u}\in\mathcal{P}_-, \mathbf{v}\in\mathcal{P}_+, \\ +\sigma(\mathbf{v},\mathbf{u})\mathbf{U}(\mathbf{v},\mathbf{u}) & \text{if} \quad \mathbf{u}\in\mathcal{P}_+, \mathbf{v}\in\mathcal{P}_-, \\ \tilde{\mathbf{h}}_{i,j}^n(\mathbf{u},\mathbf{v}) & \text{otherwise,} \end{cases}$$

where the values $\sigma(\mathbf{u},\mathbf{v})$, $\mathbf{U}(\mathbf{u},\mathbf{v})$ and $\mathbf{V}(\mathbf{u},\mathbf{v})$ are obtained from the interface solver

$$(\sigma(\mathbf{u},\mathbf{v}), \mathbf{U}(\mathbf{u},\mathbf{v}), \mathbf{V}(\mathbf{u},\mathbf{v})) = \mathcal{R}_{\mathbf{f}\cdot\mathbf{n}_{i,j}}(\mathbf{u},\mathbf{v})$$

and $\tilde{\mathbf{g}}_{i,j}^n$, $\tilde{\mathbf{h}}_{i,j}^n$ are arbitrary numerical and geometrical fluxes, respectively.

Then $\mathbf{u}$ is resolved exactly by $\mathbf{u}_h$ independent of the coarsity of the mesh, i.e. $\mathbf{u}_h = \mathbf{u}$ a.e.

Proof. The statement is an extension of the one in Lemma 3.9. However, the motion of the interface points is not prescribed but computed within Algorithm 2. Additionally, we have to take the remeshing step into account. Therefore, we show first that the time dependent points are computed as in Lemma 3.9, *i.e.* as $\Phi_i^t(\mathbf{x}) = \mathbf{x} + ct\boldsymbol{\nu}$. Secondly, we show that the remeshing does not change the approximate solution $\mathbf{u}_h$ in this setting but only its underlying mesh.

(a) Time dependent points. Assume that at a fixed time t^n the exact position of the interface curve is covered by edges of the mesh τ^n

$$\exists\, \mathcal{E}^n \subset E^n : \bigcup_{(i,j)\in\mathcal{E}^n} S_{i,j}^n = \{\mathbf{x}\in\mathbb{R}^2 \,|\, \mathbf{x}\cdot\boldsymbol{\nu} + ct^n = 0\}.$$

By induction we know that the interface curve at time t^n separates the values $\mathbf{u}_L$ and $\mathbf{u}_R$, *i.e.* we have for all $(i,j)\in\mathcal{E}^n$: $\mathbf{u}_i^n = \mathbf{u}_L$ and $\mathbf{u}_j^n = \mathbf{u}_R$, or vice versa. It follows that at each interface edge, the normal vector

is $\mathbf{n}_{i,j} = \boldsymbol{\nu}$. Therefore the corresponding one-dimensional Riemann problem is the one associated with the flux function $\mathbf{f}_1\nu_1 + \mathbf{f}_2\nu_2$ and initial data $\mathbf{u}_L$ and $\mathbf{u}_R$. The interface solver gives

$$\mathcal{R}_{\mathbf{f}_1\nu_1+\mathbf{f}_2\nu_2}(\mathbf{u}_L, \mathbf{u}_R) = (\sigma_{i,j}(\mathbf{u}_L, \mathbf{u}_R), \mathbf{U}_{i,j}(\mathbf{u}_L, \mathbf{u}_R), \mathbf{V}_{i,j}(\mathbf{u}_L, \mathbf{u}_R)) = (c, \mathbf{u}_L, \mathbf{u}_R), \tag{4.2}$$

since the exact solution consists out of a single phase transition with velocity c in $\boldsymbol{\nu}$-direction. The time dependent point is then computed with (3.23) as

$$\mathbf{x}(t) = \mathbf{x}(t^n) + (t - t^n)\left(\frac{c\boldsymbol{\nu}\,|S_{i,j}| + c\boldsymbol{\nu}\,|S_{i',j'}|}{|S_{i,j}| + |S_{i',j'}|}\right) = \mathbf{x}(t^n) + (t - t^n)c\boldsymbol{\nu} = \mathbf{x}(0) + tc\boldsymbol{\nu}.$$

(b) Remeshing. The remeshing gives a new triangulation $\hat{\tau}$ with new values $\{\hat{\mathbf{u}}_{\hat{i}}^n\}_{\hat{i}\in\hat{I}}$ given as in (3.29), that is $\hat{\mathbf{u}}_{\hat{i}}^n = \sum_{i\in I}(|K_i \cap \hat{K}_{\hat{i}}|)/|\hat{K}_{\hat{i}}|\mathbf{u}_i^n$. All values $\mathbf{u}_i^n$ with non-zero coefficient $|K_i \cap \hat{K}_{\hat{i}}|/|\hat{K}_{\hat{i}}|$ are either $\mathbf{u}_L$ or $\mathbf{u}_R$ and this convex combination simplifies to either $\hat{\mathbf{u}}_{\hat{i}}^n = \mathbf{u}_L$ or $\hat{\mathbf{u}}_{\hat{i}}^n = \mathbf{u}_R$.

The calculations in (a) show that the interface motion function IMF from Definition 3.12 gives a time dependent mesh that fulfills the assumption (3.11) from Lemma 3.9. Therefore, all assumptions from Lemma 3.9 are valid and the lemma is applicable. The remeshing step does not affect the numerical approximation $\mathbf{u}_h$ but only the underlying mesh as shown in (b). Thus the induction from Lemma 3.9 can be continued for each time step and we get $\mathbf{u}_h = \mathbf{u}$ a.e. □

5. Numerical results

We will conclude this work with numerical examples for the scalar cubic model problem and isothermal Euler equations. We perform the computations on the bounded domain $\Omega := (-1, 1)^2$ in all cases. Appropriate boundary conditions are given for each test case.

The realization of the interface tracking and the remeshing operator was done with the 2D Triangulation package [40] of the C++ library CGAL, Computational Geometry Algorithms Library [37]. Its Constrained_Delaunay_triangulation_2 class in combination with the hierarchy structure was extended to implement the moving mesh. This class manages a triangulation that is almost Delaunay except for a set of given constraints (in our case: prescribed edges of the interface curve) and it provides methods for the insertion, removal and motion of points. With this realization we omit condition (3.25) in cases where the remeshing is not possible, *e.g.*, when a circular interface is shrinking and we have to coarsen the approximate interface. In order to investigate the robustness of the remeshing algorithm, we check conditions (3.24) and (3.26) via stating the maximum mesh width $\max_n h(\tau^n)$, the minimal incircle diameter $\min_n k(\tau^n)$ as well as the average ratio $r = n^{-1}\sum_n \frac{h(\tau^n)}{k(\tau^n)}$. The maximum mesh width and the minimal incircle diameter correspond to the constants $C > 0$ and $c > 0$ of condition (3.24). The smaller r, the better is the triangulation (to be more precise: the sequence of triangulations) in the sense that the triangles are of similar size and small inner angles are avoided. The optimal value for r is $\frac{3}{\sqrt{3}} \approx 1.73$ which is obtained by a triangulation with equilateral triangles only.

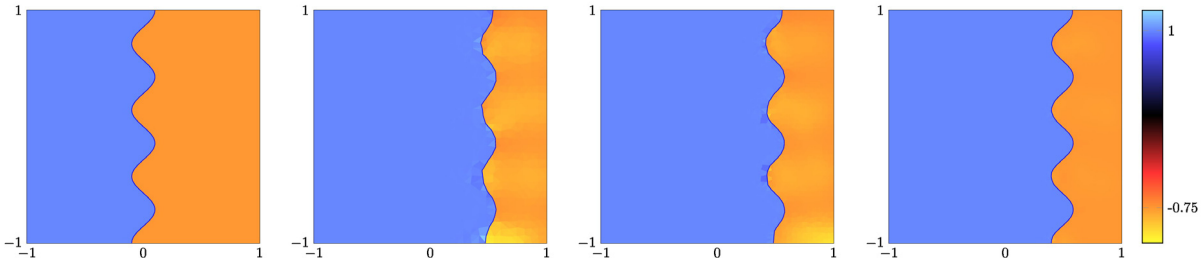
5.1. Conservation laws with a cubic flux function

We will start with the model problem from Section 2.2.1 using all notations from this section. The initial data u_0 is chosen as a jump function with non-planar phase interface: $u_0(\mathbf{x}) = u_L = 1 \in \mathcal{P}_+$ if $0.1\sin(11x_2) + x_1 > 0$, and $u_R = -0.75 \in \mathcal{P}_-$ otherwise. It can be checked that the Rankine–Hugoniot condition (2.8) and the kinetic relation from (2.14) are satisfied with (see (2.15))

$$r = 0.8125 = a(u_R, u_L, (-1, 0)^T).$$

Therefore the exact entropy-compatible sharp-interface solution u of (2.13) will propagate the initial data in x_1-direction with velocity r:

$$u(t, \mathbf{x}) = \begin{cases} u_L & \text{if} \quad 0.1\sin(11x_2) + x_1 > r\,t, \\ u_R & \text{else.} \end{cases}$$



(a) Initial state. **(b)** $t = 0.6$, 1916 elements. **(c)** $t = 0.6$, 4044 elements. **(d)** $t = 0.6$, 16254 elements.

FIGURE 7. Plot of exact initial state and approximate sharp interface solution u for different mesh sizes. The numbers of elements refer to the mesh at final time $t = 0.6$. For the last example (d) we get $\min_n k(\tau^n) = 0.00439$, $\max_n h(\tau^n) = 0.0435$ and $r = 7.324$.

TABLE 1. Experimental orders of convergence (EOC) for the cubic flux problem ($t = 0.1$).

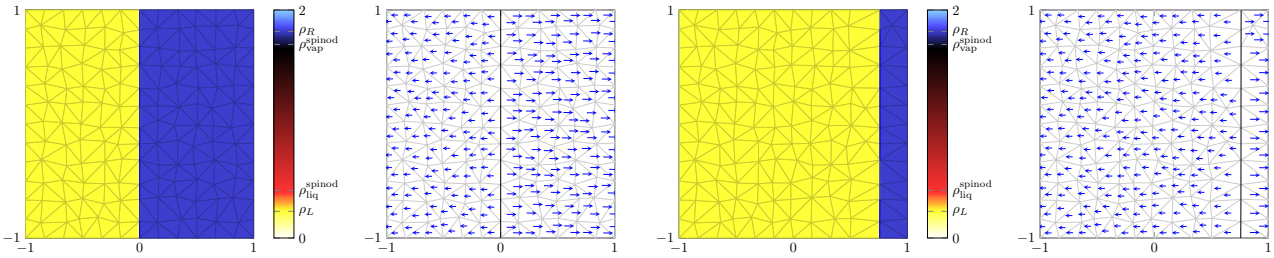
$\lvert\mathcal{T}(t)\rvert$	$h(\mathcal{T}(t))$	$t^{-1}\int_0^t h(\mathcal{T}(s))\,\mathrm{d}s$	$k(\mathcal{T}(t))$	$\lVert(u_h - u)(t,\cdot)\rVert_{L^1(\Omega)}$	EOC(t)
774	0.2005	0.2076	0.0215	2.3628×10^{-2}	1.85
1764	0.1322	0.1341	0.0185	1.0919×10^{-2}	1.89
3988	0.0887	0.0906	0.0075	5.1381×10^{-3}	1.89
9166	0.0626	0.0606	0.0097	2.6562×10^{-3}	1.88
20 272	0.0390	0.0400	0.0070	1.0918×10^{-3}	1.80
46 024	0.0270	0.0270	0.0045	5.6278×10^{-4}	1.26
102 790	0.0192	0.0183	0.0029	3.6766×10^{-4}	

The boundary conditions are chosen as absorbing boundary conditions, *i.e.* no wave is emitted from the boundary inside of the domain Ω. Since the interface curve is not closed, we need a special treatment of the interface vertices located at the boundary of the domain. The velocity $\mathbf{s}_{i,k}$ of an interface vertex $\mathbf{p}_{i,k}$ is then prescribed by the velocity computed at the *only* incident interface edge $S_{i,j}$ by $\mathbf{s}_{i,k} := \mathbf{n}_{i,j}\sigma_{i,j}(\mathbf{u}_i, \mathbf{u}_j)$ replacing (3.22). If the new position $\mathbf{p}_{i,k} + \Delta t\mathbf{s}_{i,k}$ departs from $\partial\Omega$ we subsequently project it onto the boundary in tangential direction of the interface curve.

The numerical results are depicted in Figure 7, where we observe the expected transport behaviour. With u we compute L^1-errors and the (experimental) orders of convergence, see Table 1. The errors and the orders listed in Table 1 show that we obtain convergence with order around 1.8. This appears to be better than the expected value 1.0. The reason is probably that the classical computation of a convergence order is not appropriate here, as one has to assume convergence with decreasing mesh width, which is time dependent and discontinuous due to the interface tracking and the remeshing steps. Finally, note that the exact solution is not planar such that Theorem 4.4 does not apply and we do not get $u = u_h$.

5.2. Isothermal Euler equations in two space dimensions

In this section we will present numerical results for the two-dimensional isothermal Euler equations (2.16) and Van-der-Waals pressure (2.17) with constants $A = 3$, $B = \frac{1}{3}$, $\theta = 0.85$ and $R = \frac{8}{3}$. Contrary to the cubic flux problem we have now a state space that is separated by an interval, the unstable spinodal region, into two sets, *cf.* (2.18). The one-dimensional problem (3.20) in normal direction $\mathbf{n} = (n_1, n_2)^T$ resulting out of (2.16)



(a) Initial states ρ_0 and $\mathbf{m}_0 = (m_{1,0}, m_{2,0})$. **(b)** Approximate solutions ρ_h and $\mathbf{m}_h$ at time $t = 4$.

FIGURE 8. Numerical solution for an isolated liquid-vapour phase transition with $264 - 288$ elements, $\min_n k(\tau^n) = 0.0519$, $\max_n h(\tau^n) = 0.349$ and $r = 4.365$.

is given by

$$\begin{pmatrix} \rho \\ m_1 \\ m_2 \end{pmatrix}_t + \begin{pmatrix} m_1 n_1 + m_2 n_2 \\ \dfrac{m_1^2}{\rho} n_1 + p(\rho) n_1 + \dfrac{m_1 m_2}{\rho} n_2 \\ \dfrac{m_2^2}{\rho} n_2 + p(\rho) n_2 + \dfrac{m_1 m_2}{\rho} n_1 \end{pmatrix}_x = 0. \tag{5.1}$$

A combination of the second and third line of (5.1) leads to a system with two-dimensional state space using the projected momentum $m_p = \mathbf{m} \cdot \mathbf{n} = m_1 n_1 + m_2 n_2$ as unknown. In fact, the used interface solver from [41] applies to this system which readily can be used to design an interface solver for (5.1).

Example 5.1 (Planar shock wave)**.**

We start with a validation example and take initial conditions that drive a planar moving phase boundary as entropy-compatible sharp-interface solution. At the boundary we apply absorbing boundary conditions. Precisely, the initial conditions are

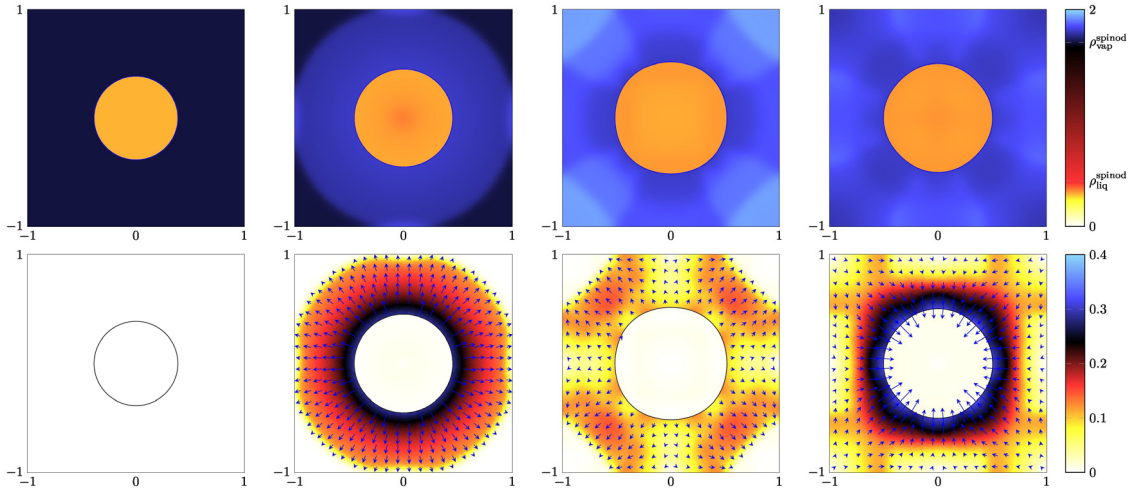
$$(\rho_0, m_{1,0}, m_{2,0})(\mathbf{x}) = \begin{cases} (0.234, -0.127, -0.002) & x_1 < 0, \\ (1.812, 0.173, -0.002) & x_1 > 0, \end{cases}$$

such that the entropy-compatible sharp-interface solution is $(\rho, m_1, m_2)(t, \mathbf{x}) = \mathbf{u}_0(x_1 - 0.19\, t, x_2)$. For a single phase boundary, we know due to Theorem 4.4 that the Finite Volume moving mesh algorithm 2 will return the exact solution. Theorem 4.4 does not account for inevitable rounding errors. To check the influence of roundoffs the numerical approximations can be seen in Figure 8. They show the minor effects of roundoffs and illustrate the statement of Theorem 4.4. Most notably, we observe that the algorithm does not introduce any values in the spinodal region $(\rho_{\text{liq}}^{\text{spinod}}, \rho_{\text{vap}}^{\text{spinod}})$.

Example 5.2 (Single bubble)**.**

Next, we continue with an example which involves more intricate geometric dynamics of the interface. This time, the initial condition consists of a vapour bubble surrounded by the fluid in liquid phase. We make the choice

$$(\rho_0, m_{1,0}, m_{2,0})(\mathbf{x})(\mathbf{x}) = \begin{cases} (1.7, 0, 0) & \|\mathbf{x}\|_2^2 < 0.15, \\ (0.3, 0, 0) & \text{else} \end{cases}$$



(a) Initial state. **(b)** t=0.4 **(c)** t=0.8 **(d)** t=1.2

FIGURE 9. Numerical solution for a bubble in non-equilibrium state. The first line displays density, the second line momentum (refering to the arrows) and absolute value of momentum (refering to the color bar). The initial mesh has 45 138 elements. The number of triangles of the meshes during the runtime of the algorithm was 45 092 to 45 188 triangles which is a remarkably small range. The remeshing routine gives $\min_n k(\tau^n) = 0.00142$, $\max_n h(\tau^n) = 0.0278$ and $r = 7.876$.

for the initial state and use reflecting boundary conditions. The result of the Finite Volume moving mesh algorithm 2 for that problem is depicted in Figure 9. One observes an oscillating bubble with decreasing amplitude of the oscillation. The oscillation appears due to the reflecting boundary condition and not due to surface tension effects which are not considered in the mathematical model.

Example 5.3 (Shock wave hitting a phase transition)**.**

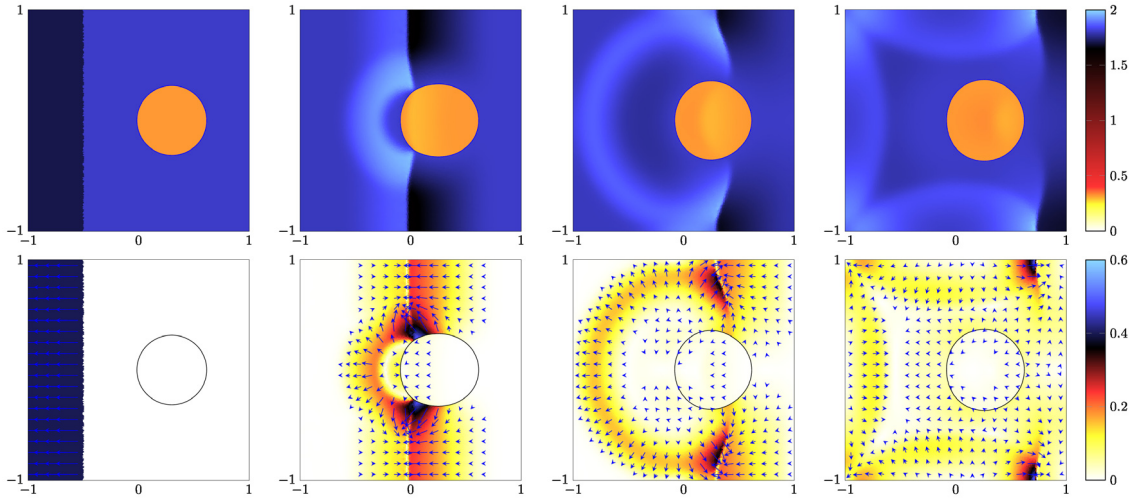
As a last example we want to show the interaction of a phase transition and classical waves. We take as initial condition a vapour bubble surrounded by liquid in Maxwell equilibrium together with a discontinuity at the left, which will result in a classical wave propagating to the right, *i.e.* in direction of the bubble.

$$(\rho_0, m_{1,0}, m_{2,0})(\mathbf{x}) = \begin{cases} (\rho_{\text{vap}}^{\text{Maxw}} = 0.3207, 0, 0) & \left\|\mathbf{x} - (0.3, 0)^T\right\|_2^2 < 0.1, \\ (\rho_{\text{liq}}^{\text{Maxw}} = 1.8071, 0, 0) & \left\|\mathbf{x} - (0.3, 0)^T\right\|_2^2 > 0.1 \text{ and } x_1 > -0.5, \\ (1.7010, -0.4, 0) & \text{else.} \end{cases}$$

Maxwell equilibria $\rho_{\text{vap}}^{\text{Maxw}}$ and $\rho_{\text{liq}}^{\text{Maxw}}$ are characterized by equal pressure and Gibbs free energy. They are supposed to provide stable static solutions. At the boundary we choose reflecting boundary conditions.

The results of the computation in Figure 10 show the effect of the classical wave on the phase transition. After it has reached the approximate interface, the phase transition states are not Maxwell equilibria anymore and the interface starts moving. Also, after the collision one can see the reflected waves. We know from (2.6) that for entropy-compatible sharp-interface solutions $\mathbf{u}$ the total entropy may not increase in time. Taking into account the boundary condition this means here that the quantity

$$S[\mathbf{u}(t,\cdot)] = \int_\Omega \eta(\mathbf{u}(t,\cdot))\, \mathrm{d}V + \int_0^t \int_{\partial\Omega} n_1 q_1(\mathbf{u}) + n_2 q_2(\mathbf{u})\, \mathrm{d}A\, \mathrm{d}s \tag{5.2}$$



(a) Initial state. **(b)** t=0.6 **(c)** t=0.8 **(d)** t=1.0

FIGURE 10. Numerical solution for a shock moving towards a phase transition in Maxwell equilibrium. The initial mesh has 45 162 elements. The number of elements during runtime was 45 100 to 45 162. This is again a small range which underlines the effectivity of the remeshing routine. The sizes of the elements is characterized by $\min_n k(\tau^n) = 0.00180$, $\max_n h(\tau^n) = 0.0261$ and $r = 7.111$.

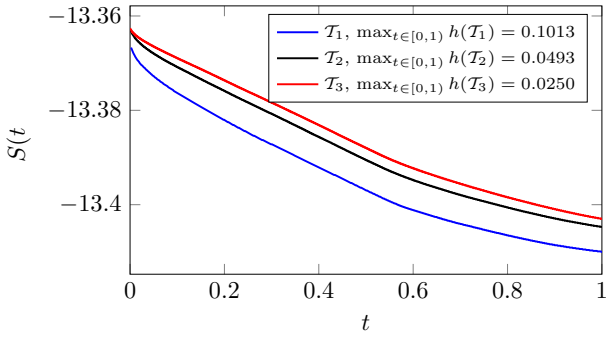


FIGURE 11. Evolution of $S[\mathbf{u}_h(t,\cdot)]$ from (5.2).

is non-increasing. We therefore check this property for the approximate solution $\mathbf{u}_h$. It can be seen in Figure 11 that the new algorithm leads to a perfectly monotone decreasing total entropy.

6. CONCLUSIONS

We have presented a moving mesh method to approximate weak solutions of conservation laws that contain undercompressive shocks. The approach generalizes earlier work on the scalar one-dimensional case to systems in two spatial dimensions. Basic invariance properties of the exact solutions are shown to transfer to the numerical scheme.

The conceptional approach can be extended to the three-dimensional case in a straightforward manner, however becomes intricate due to the complexitiy of the three-dimensional remeshing. From the practical point

of view it would be extremely interesting to extend the approach to other examples with undercompressive waves which would only require to construct case-dependent Riemann solvers.

A higher order method of the algorithm is possible but requires some changes. First, the interface curve is part of the unknowns. A higher order approximation would require a higher order polynomial for the representation of the interface, *i.e.* elements with curved edges. Second, at the interface not a single Riemann problem in normal direction has to be solved but the higher order evaluation of the surface integral in (3.3) requires multiple evaluations of interface solvers.

References

[1] R. Abeyaratne and J.K. Knowles, Kinetic relations and the propagation of phase boundaries in solids. *Arch. Ration. Mech. Anal.* **114** (1991) 119–154.

[2] A.L. Bertozzi, A. Münch and M. Shearer, Undercompressive shocks in thin film flows. *Phys. D* **134** (1999) 431–464.

[3] B. Boutin, C. Chalons, F. Lagoutière and P.G. LeFloch, Convergent and conservative schemes for nonclassical solutions based on kinetic relations. I. *Interfaces Free Bound.* **10** (2008) 399–421.

[4] C. Chalons, Transport-equilibrium schemes for computing nonclassical shocks. Scalar conservation laws. *Numer. Methods Partial Differ. Equ.* **24** (2008)1127–1147.

[5] C. Chalons, F. Coquel, P. Engel and C. Rohde, Fast relaxation solvers for hyperbolic-elliptic phase transition problems. *SIAM J. Sci. Comput.* **34** (2012A) 1753–A1776.

[6] C. Chalons, P. Engel and C. Rohde, A conservative and convergent scheme for undercompressive shock waves. *SIAM J. Numer. Anal.* **52** (2014) 554–579.

[7] C. Chalons and P.G. LeFloch, Computing undercompressive waves with the random choice scheme. Nonclassical shock waves. *Interfaces Free Bound.* **5** (2003) 129–158.

[8] C. Chen and H. Hattori, Exact Riemann solvers for conservation laws with phase change. *Appl. Numer. Math.*, **94** (2015) 222–240.

[9] A. Chertock, S. Karni and A. Kurganov, Interface tracking method for compressible multifluids. *ESAIM: M2AN* **42** (2008) 991–1019.

[10] R.M. Colombo and F.S. Priuli, Characterization of Riemann solvers for the two phase p-system. *Commun. Partial Differ. Equ.* **28** (2003) 1371–1389.

[11] R.M. Colombo and M.D. Rosini, Pedestrian flows and non-classical shocks. *Math. Methods Appl. Sci.* **28** (2005) 1553–1567.

[12] Y. Di, R. Li, T. Tang and P. Zhang, Moving mesh finite element methods for the incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.* **26** (2005) 1036–1056.

[13] A. Dressel and C. Rohde, A finite-volume approach to liquid-vapour fluids with phase transition. In *Finite volumes for complex Applications V*. Iste, London (2008) 53–68.

[14] S. Fechter, C.-D. Munz, C. Rohde and C. Zeiler, A sharp interface method for compressible liquidvapor flow with phase transition and surface tension. *J. Comput. Phys.* **336** (2017) 347–374.

[15] R.P. Fedkiw, The ghost fluid method for numerical treatment of discontinuities and interfaces. *Godunov Methods*. Edited by E. Toro. Springer US (2001) 309–317.

[16] M. Hantke, W. Dreyer and G. Warnecke, Exact solutions to the Riemann problem for compressible isothermal Euler equations for two-phase flows with and without phase transition. *Quart. Appl. Math.* **71** (2013) 509–540.

[17] A. Harten and J. M. Hyman, Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.* **50** (1983) 235–269.

[18] B.T. Hayes and P.G. LeFloch, Nonclassical shocks and kinetic relations: finite difference schemes. *SIAM J. Numer. Anal.* **35** (1998) 2169–2194.

[19] T.Y. Hou, P. Rosakis and P. LeFloch, A level-set approach to the computation of twinning and phase-transition dynamics. *J. Comput. Phys.* **150** (1999) 302–331.

[20] F. Kissling and C. Rohde, The computation of nonclassical shock waves in porous media with a heterogeneous multiscale method: The multidimensional case. *Multiscale Model. Simul.* **13** (2015) 1507–1541.

[21] A. Kluwick, E. Cox and S. Scheichl, Non-classical kinematic shocks in suspensions of particles in fluids. *Acta Mech.* **144** (2000) 197–210.

[22] D. Kröner, *Numerical schemes for conservation laws*. Wiley-Teubner Series Advances in Numerical Mathematics. John Wiley and Sons Ltd., Chichester (1997).

[23] P.G. LeFloch, Propagating phase boundaries: formulation of the problem and existence via the Glimm method. *Arch. Rational Mech. Anal.* **123** (1993) 153–197.

[24] P.G. LeFloch, Hyperbolic systems of conservation laws. The theory of classical and nonclassical shock waves. *Lecture in Mathematics*. ETH Zürich, Birkhäuser, Basel (2002).

[25] P.G. LeFloch and C. Rohde, High-order schemes, entropy inequalities, and nonclassical shocks. *SIAM J. Numer. Anal.* **37** (2000) 2023–2060.

[26] P.G. LeFloch and M.D. Thanh, Non-classical Riemann solvers and kinetic relations. II. An hyperbolic-elliptic model of phase-transition dynamics. *Proc. R. Soc. Edinburgh Sect. A* **132** (2002) 181–219.

[27] R.J. LeVeque, Finite volume methods for hyperbolic problems, vol. 31. Cambridge university press (2002).
[28] R. Li and T. Tang, Moving mesh discontinuous Galerkin method for hyperbolic conservation laws. *J. Sci. Comput.* **27** (2006) 347–363.
[29] C. Merkle and C. Rohde, Computation of dynamical phase transitions in solids. *Appl. Numer. Math.* **56** (2006) 1450–1463.
[30] C. Merkle and C. Rohde, The sharp-interface approach for fluids with phase change: Riemann problems and ghost fluid techniques. *ESAIM: M2AN* **41** (2007) 1089–1123.
[31] J. Ning, X. Yuan, T. Ma and C. Wang, Positivity-preserving moving mesh scheme for two-step reaction model in two dimensions. *Comput. Fluids* **123** (2015) 72–86.
[32] C. Rohde and C. Zeiler, A relaxation Riemann solver for compressible two-phase flow with phase transition and surface tension. *Appl. Numer. Math.* **95** (2015) 267–279.
[33] R. Sanders, The moving grid method for nonlinear hyperbolic conservation laws. *SIAM J. Numer. Anal.* **22** (1985) 713–728.
[34] J. M. Stockie, J. A. Mackenzie and R.D. Russell, A moving mesh method for one-dimensional hyperbolic conservation laws. *SIAM J. Sci. Comput.* **22** (2000) 1791–1813.
[35] F. Svensson, Moving meshes and higher order finite volume reconstructions. *Int. J. Finite Vol.* **3** (2006) 27.
[36] H. Tang and T. Tang. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM J. Numer. Anal.* **41** (2003) 487–515.
[37] The CGAL Project, *CGAL User and Reference Manual*. CGAL Editorial Board, 4.7 edition (2016).
[38] L. Truskinovsky, Kinks versus shocks. In Shock Induced Transitions and Phase Structures in General Media, edited by J. Dunn. Vol. 52 of *The IMA Volumes in Mathematics and its Applications*. Springer (1993) 185–229.
[39] C.J. van Duijn, L.A. Peletier and I.S. Pop, A new class of entropy solutions of the Buckley-Leverett equation. *SIAM J. Math. Anal.* **39** (2007) 507–536.
[40] M. Yvinec, 2D triangulation. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.7 edition (2015).
[41] C. Zeiler, *Liquid Vapor Phase Transitions: Modeling, Riemann Solvers and Computation*. Ph.D. thesis. Universität Stuttgart (2015).
[42] X. Zhong, T.Y. Hou and P.G. LeFloch, Computational methods for propagating phase boundaries. *J. Comput. Phys.* **124** (1996) 192–216.