A NEW NUMERICAL ALGORITHM FOR TWO-PHASE FLOWS DRIFT-FLUX MODEL WITH STAGGERED GRID IN POROUS MEDIA

Anouar Mekkas¹, Anne Charmeau² and Sami Kouraichi³

Abstract. FLICA4 is a 3D compressible code dedicated to reactor core analysis. FLICA4 solves for two-phase flows in porous media using a compressible drift-flux model [2]. To derive convective fluxes, FLICA4 uses a specific finite volume numerical method based on an extension of the Roe’s approximate Riemann colocated solver [3]. The method is developed to solve 1D flows. It requires adjustments to solve for 2D or 3D flows, at low Mach number, on a cartesian mesh. As Mach number goes to zero, in some extreme 2D or 3D cases, the native method does converge but the solution is not consistent with the analytical or experimental solution [4]. A so-called “pressure correction” is added to the initial solver to derive an accurate solution. However, this extra term produces checkerboard oscillations in space for most applications of interest, even when solving for 1D flows. These checkerboard oscillations are sometimes critical and may lead to unstable solutions or divergence of the solver. The paper presents the analysis of an alternative numerical algorithm to solve for compressible drift-flux model in the low Mach regime. The current study aims at investigating a new compressible scheme that is both accurate and robust at low Mach number on staggered grids. The actual behavior of the new scheme is tested against simplified nuclear core "boiling channel" study cases in the low Mach regime. The tests are extended to compressible regime for flows exhibiting both shock waves and no shock waves. The paper shows that the alternative numerical algorithm leads to numerical solutions that are very close to analytical solutions. Moreover, no more checkerboard oscillations are generated. Therefore, the algorithm is a real improvement to FLICA4 current numerical method.

¹ CEA/DEN/DM2S/STMF F-91191, Gif-sur-Yvette, France: anouar.mekkas@cea.fr
² CEA/DEN/DM2S/STMF F-91191, Gif-sur-Yvette, France: anne.charmeau@cea.fr
³ Université Paris 13, 99 Avenue Jean Baptiste Clément, 93430 Villetaneuse, France: sami.kouraichi@edu.univ-paris13.fr

© EDP Sciences, SMAI 2017
Résumé. FLICA4 est un logiciel de simulation 3D dédié à l’analyse des écoulements dans les coeurs de réacteurs nucléaires et qui résout un modèle compressible diphasique à 4 équations à l’échelle poreuse [2]. Le schéma numérique du code FLICA4 est basé sur une technique de volumes finis où les flux numériques convectifs sont calculés à l’aide d’un solveur colocalisé appelé Roe [3]. L’analyse de cette méthode numérique montre qu’à bas nombre de Mach, il est nécessaire d’introduire des modifications spécifiques aux géométries 2D ou 3D sur un maillage cartésien sans quoi la solution ne converge pas vers la bonne solution lorsque le nombre de Mach tend vers zéro [4]. C’est la raison pour laquelle une correction dite “correction de pression” est appliquée. Cette “correction de pression” nécessaire à la précision du schéma numérique à bas nombre de Mach pour des configurations 2D ou 3D sur un maillage cartésien introduit presque systématiquement des oscillations en espace de type mode en échiquier dans les configurations étudiées ici, surtout en 1D. Comme ces oscillations spatiales peuvent être très fortes dans certains cas et éventuellement conduire à une divergence de certains calculs. Nous étudions un nouvel algorithme numérique pour réoudre le modèle compressible diphasique à 4 équations. Le but de ce travail est de proposer un nouveau schéma numérique précis et robuste à bas nombre de Mach sur grilles décalées car les oscillations spatiales de type mode en échiquier sont inexistantes avec ce type de discrétisation [8]. Ce nouveau schéma numérique est vérifié en régime bas Mach avec des cas-tests décrivant un cœur nucléaire simplifié “canal bouillant”. Le comportement de ce schéma est également testé en régime compressible avec ou sans ondes de choc.

Introduction

FLICA4 est un 3D code compressible dédié à la simulation des coeurs de réacteurs nucléaires. FLICA4 est utilisé pour évaluer le Département d’Évaporation dans les analyses de sécurité pour les réacteurs nucléaires tels que les réacteurs à eau courante. Ces analyses nécessitent un modèle à 3D pour modéliser le système sur des maillages cartésiens. Le domaine de vitesse des réacteurs nucléaires nécessite de modéliser les régimes de vitesse faible. FLICA4 résout des écoulements diphasiques poreux en utilisant un modèle compressible [2]. Pour dériver les flux convectifs, FLICA4 utilise une méthode numérique finie volume spécifique basée sur une extension de la technique de Roe de Riemann pour la résolution de la méthode de Riemann colocalisée [3]. Le principe est développé pour résoudre des écoulements 1D. Il nécessite des ajustements pour résoudre des écoulements 2D ou 3D, au bas nombre de Mach, sur un maillage cartésien. Si le nombre de Mach tend vers zéro, dans certains cas extrêmes 2D ou 3D, le schéma initial ne converge pas mais la solution est non cohérente avec la solution analytique ou expérimentale [4]. Une correction de pression est ajoutée à l’initial solvant pour résoudre la solution physique. Alors que cette correction est nécessaire pour arriver à la précision requise, le ”correction de pression” produit des oscillations en espace de type mode en échiquier. Checkboard oscillations lead to unsteady solutions, or divergence of the solver, making the code not as robust as required. Improvement of the numerical solver is key for industrial studies : an alternative numerical algorithm to solve for the compressible drift-flux model in the low Mach regime is investigated. The study aims at developing a compressible solver on staggered grid. Using staggered grids guarantees that no checkerboard oscillations are generated [8]. The aim of this work is to present the alternative compressible scheme and verify its properties with extensive testing. The three types of tests for this analysis are a simplified nuclear core model with flow in the low Mach regime (boiling channel), tests in the compressible regime exhibiting shock waves and others in the compressible regime but with no shock waves.

First, we present the four balance equations which are solved with the FLICA4 code (§ 1). Then, time and space discretisation of these equations using finite volume approach on staggered grids is detailed in section 2. The time discretisation of the equations is based on a semi-implicit scheme. Since the equations are not linear, the solution at each time step is obtained by a Newton-Raphson iterative method. The new numerical method is described in section 3. It leads to a linear system of equations for the increments of the principal variables. First, the chosen solution algorithm [6] consists in eliminating velocity increments as a function of pressure increments.
by rewriting the momentum equations. Substituting the velocity increments into the non-linear system results in a system with pressure increments only. The successive removal of all scalar variables other than pressure leads to a linear system where the only unknown is pressure. Resolution of this linear system allows to solve for pressure, then velocity and finally all other variables. The numerical method uses non-conservative formulation of the four balance equations. It is expected that it might not exhibit the same performances on both low Mach regime and compressible flows. Both regime are tested through three practical cases, for which analytical solutions exist [7, 12]. The tests cover all types of flow regime (§ 4).

1. Porous 4 Equations Model

FLICA4 is a two-phase flow code. The system of balance equation is a 4-equation system. The equations are mixture mass balance, phasic mass balance, mixture momentum balance and mixture energy balance. A drift-flux model is used to account for the slip between vapor and liquid phases. The fluid is compressible and the two-phases are assumed to be at the same pressure. The main assumption to write the system of equations is that one of the phases, the vapor phase, remains at saturation temperature. Hence, vapor specific enthalpy remains constant and equal to vapor enthalpy at saturation temperature. The non-conservative form of the two-phases are assumed to be at the same pressure. The main assumption to write the system of equations is:

\[ \rho = \mathcal{F}(P, h, C). \]  

The nomenclature is given as follows : \( t \) is the time, \( \phi \) is the porosity, \( \alpha \) is the void fraction of the vapor, \( \rho_v \) is the vapor density, \( \rho_l \) is the liquid density, \( \rho = \alpha \rho_v + (1 - \alpha) \rho_l \) is the mixture density, \( C = \frac{\alpha \rho_v}{\rho} \) is the vapor mass concentration, \( h_v \) is the vapor specific enthalpy, \( h_l \) is the liquid specific enthalpy, \( h = \rho C h_v + (1 - \rho C) h_l \) is the mixture specific enthalpy, \( V_v \) is the vapor velocity, \( V_l \) is the liquid velocity, \( V = \rho C V_v + (1 - \rho C) V_l \) is the mixture velocity, \( V_r = \frac{V_v - V_l}{1 - \beta} \) is the relative velocity between the two phases, \( P \) is the mixture pressure, \( e = h - \frac{P}{\rho} \) is the mixture internal energy, \( H_v = h_v + \frac{1}{2} (V_v)^2 \) is the vapor total enthalpy, \( H_l = h_l + \frac{1}{2} (V_l)^2 \) is the liquid total enthalpy, \( K_{cv} \) is the concentration dispersion coefficient, \( \Gamma_v(P, h, C, V) \) is the term represents the interphase mass exchange [2], \( \overline{g} \) is the gravity force, \( Q \) is the power density, \( \overline{\tau} \) is the stress tensor, \( \tau_f \) is the friction forces and \( q \) is the heat flux.

The relative velocity \( V_r \) is given by the Ishii model [9, 10]:

\[ V_r = \frac{(C_0 - 1) V + V_{v, lim}}{1 - \beta + C_0 (C - \alpha)} \]  

where \( C_0 \) is a user parameter and \( V_{v, lim} \) is the vapor velocity limit.

The Friction forces \( \tau_f = \tau_{fw} + \tau_{fs} \) where \( \tau_{fw} \) is the wall friction and \( \tau_{fs} \) is the singular friction:
\[
\tau_{fw} = -\frac{f_w}{2D_h} \rho \bar{V} |\bar{V}|,
\]
(4)

\[
\tau_{fs} = -\frac{K}{2} \rho \bar{V} |\bar{V}|
\]
(5)

where \(D_h\) is hydraulic diameter, \(K\) and \(f_w\) are friction coefficients.

We note that the hyperbolicity of the drift-flux model (Eq. 1) without taking into account the energy equation is studied in [16]. The paper states conditions on the phase difference between phases that ensure the hyperbolicity of the system.

We also note that the drift-flux model exhibits a lot of non-conservative terms: in the low Mach number regime, the non-conservative term in the energy equation is not an issue. On the opposite, it could be a source of difficulties to get precise shock wave solutions in the compressible regime. These difficulties can be overcome by replacing the internal energy equation in Eq. 1 by a total energy or total enthalpy equation. The non-conservative convection term of convection in the momentum equation eases derivation of the porosity (flow area over total area, including massive elements such as rods...). Writing the convection term in momentum equation in a non-conservative form is necessary for the pressure-based solver presented later (§3.2) to eliminate velocity increments (§3.2.1).

In the following work, we assume that contribution of the diffusive terms \(\nabla \cdot (\phi K_{ce} \nabla C), \nabla \cdot \bar{\tau}\) and \(\nabla \cdot \bar{q}\) is negligible.

2. Full time and space discretisation

For space discretisation, a staggered grid (see figure 1) is used where scalar variables (pressure, density, enthalpy, etc) are derived at the center of the control volumes. Velocity or momentum variables are computed on volume faces. It differs from a collocated grid, where all the variables are computed at the same position.

\[\text{Figure 1. Staggered grid in 2-dimensions}\]

Using a staggered grid is a simple way to avoid odd-even decoupling between pressure and velocity. Odd-even decoupling is a discretisation error that can occur on collocated grids and which can lead to checkerboard patterns in the computed solution [15].

2.1. Discretisation of the mixture mass equation

Time discretisation of the mixture mass equation is based on a semi-implicit scheme. Mass fluxes are approximated at the interface using a donor semi-implicit scheme where the velocity is implicit while the scalar variables are explicit. Space discretisation of the mass fluxes in the mixture mass equation in the three directions is obtained by approximating the density at the cell faces through a donor formulation.
Integration of the mixture mass equation over elementary control volume $M^K_c$ (see figure 2) and over a time interval from $t_n$ to $t_{n+1} = t_n + \Delta t$ gives:

$$
\int_{M^K_c} \int_{t_n}^{t_{n+1}} \left[ \frac{\partial \rho}{\partial t} + \text{\textbf{\nabla}}.(\phi \rho \text{\textbf{V}}) \right] \, d\Omega \, dt = 0.
$$

(6)

By using the divergence theorem, equation (6) becomes:

$$
\int_{M^K_c} \int_{t_n}^{t_{n+1}} \phi \frac{\partial \rho}{\partial t} \, d\Omega \, dt + \int_{\partial M^K_c} \int_{t_n}^{t_{n+1}} \phi \rho \text{\textbf{V}} \cdot \text{\textbf{n}} \, dS \, dt = 0
$$

(7)

or in discrete form:

$$
\frac{\rho_{M^K_c}^{n+1} - \rho_{M^K_c}^n}{\Delta t} + \frac{1}{(\phi \text{\textbf{V}})_{M^K_c}} \sum_{\sigma \in \varepsilon_K^{n+1}} \mathcal{F}^{n+1}_{\sigma} = 0
$$

(8)

Figure 2. Control volume $M^K_c$ in 2-dimensions

In (8), $\varepsilon_K$ is the set of faces of $M^K_c$, $\nabla M^K_c$ is the volume of $M^K_c$ and $\mathcal{F}^{n+1}_{\sigma}$ is the approximation of the flux at the interface $\sigma$ at time $t_{n+1}$.

To establish the discrete mixture mass equation, one needs to approximate the flux $\mathcal{F}^{n+1}_{\sigma}$ on the faces of a given cell $M^K_c$. To ensure the stability of the numerical scheme, we use a donor formulation of the convection term:

$$
\mathcal{F}^{n+1}_{\sigma} = \phi_{\sigma} S_{\sigma} \rho_{\sigma}^n \nabla^{n+1}_{\sigma} \cdot \text{\textbf{n}}_{\sigma}.
$$

(9)

In (9), $S_{\sigma}$ is the flow area of $\sigma$, $\nabla^{n+1}_{\sigma}$ is the unit normal vector of $\sigma$ oriented from $M^K_c$ to $M^K_L$ and $\rho_{\sigma}^n$ stands for the upwind mixture density at the interface $\sigma$ between a given cell $M^K_c$ and its neighboring cell $M^K_L$:

$$
\rho_{\sigma}^n = \begin{cases} 
\rho_{M^K_c}^n & \text{if } \nabla^{n+1}_{\sigma} \cdot \text{\textbf{n}}_{\sigma} > 0, \\
\rho_{M^K_L}^n & \text{otherwise}. 
\end{cases}
$$

As a result, the discrete form of the mixture mass equation (8) can be written as:

$$
F^1(P_{M^K_c}^{n+1} , \nabla^{n+1}_{\sigma} \in \varepsilon_K) = 0.
$$

(10)

Taking into account the equation of state (2), equation (10) becomes:

$$
F^1(P_{M^K_c}^{n+1} , h_{M^K_c}^{n+1} , C_{M^K_c}^{n+1} , \nabla^{n+1}_{\sigma} \in \varepsilon_K) = 0.
$$

(11)
2.2. Discretisation of the mixture internal energy balance equation

**Space and time discretisation** of the mixture internal energy balance equation follows the previous approach, used for the mixture mass equation, with a finite volume method.

Integration of mixture internal energy equation over the elementary control volume $M^K_c$ (see figure 2) and over a time interval from $t_n$ to $t_{n+1} = t_n + \Delta t$ gives:

$$
\int_{M^K_c} \int_{t_n}^{t_{n+1}} \left\{ \phi \frac{\partial p}{\partial t} + \nabla \cdot (\phi p \overrightarrow{V}) + P \nabla \cdot (\phi \overrightarrow{V}) + \nabla \cdot \left[ \left( \phi \rho C(1 - C)(H_v - H_l) \overrightarrow{V} \right) - \phi Q \right] \right\} d\Omega dt = 0. \tag{12}
$$

It may be written in a discrete form as:

$$
\frac{(pe)^{n+1}_{M^K_c} - (pe)^{n}_{M^K_c}}{\Delta t} + \frac{1}{(\phi \overrightarrow{V})_{M^K_c}} \sum_{\sigma \in \epsilon_K} \left[ \mathcal{F}^{n+1}_\sigma + P^{n+1}_{M^K_c} \mathcal{G}^{n+1}_\sigma + \mathcal{H}^{n+1}_\sigma \right] - Q^n_{M^K_c} = 0 \tag{13}
$$

where $\mathcal{F}^{n+1} \sigma$, $\mathcal{G}^{n+1} \sigma$ and $\mathcal{H}^{n+1} \sigma$ are fluxes approximations at the interface $\sigma$ at time $t_{n+1}$:

$$
\begin{align*}
\mathcal{F}^{n+1}_\sigma &= \phi_\sigma S_\sigma (pe)^n_{M^K_c} \overrightarrow{V}^{n+1}_\sigma \overrightarrow{n}_\sigma, \\
\mathcal{G}^{n+1}_\sigma &= \phi_\sigma \overrightarrow{n}_\sigma \nabla \cdot \overrightarrow{V}^{n+1}_\sigma, \\
\mathcal{H}^{n+1}_\sigma &= \phi_\sigma S_\sigma [\rho_r C_r (1 - C_r) (H_v - H_l)]_\sigma \overrightarrow{V}^{n+1}_\sigma \overrightarrow{n}_\sigma.
\end{align*}
$$

- $(pe)^n_\sigma$ and $[\rho_r C_r (1 - C_r) (H_v - H_l)]_\sigma^n$ are defined at the cells center. On the faces, they are approximated using the following donor formulation:

$$
(p e)^n_\sigma = \begin{cases} (pe)^{n}_{M^K_c} & \text{if } \overrightarrow{V}^{n+1}_\sigma \cdot \overrightarrow{n}_\sigma > 0, \\
(p e)^{n}_{M^K_c} & \text{otherwise.}
\end{cases}
$$

$$
[p_r C_r (1 - C_r) (H_v - H_l)]_\sigma^n = \begin{cases} [\rho C(1 - C)(H_v - H_l)]^{n}_{M^K_c} & \text{if } \overrightarrow{V}^{n+1}_\sigma \cdot \overrightarrow{n}_\sigma > 0, \\
[\rho C(1 - C)(H_v - H_l)]^{n}_{M^K_c} & \text{otherwise.}
\end{cases}
$$

- $(\overrightarrow{V})^{n+1}_\sigma$ is given by (3):

$$
(\overrightarrow{V})^{n+1}_\sigma = \frac{(C_0 - 1) \overrightarrow{V}^{n+1}_\sigma + \overrightarrow{V}_{v,lim}}{1 - C_0^a + C_0 (C_0^a - \alpha^a_0)} \tag{14}
$$

The concentration $C^n_\sigma$ and the void fraction $\alpha^n_\sigma$ at the interface $\sigma$ are approximated by also using a donor formulation.

Hence, the discrete form of the mixture internal energy equation (13) can be written as:

$$
F^2(p^{n+1}_{M^K_c}, \overrightarrow{t}^{n+1}_{M^K_c}, C^{n+1}_{M^K_c}, \overrightarrow{V}^{n+1}_{\sigma \in \epsilon_K}, (\overrightarrow{V})^{n+1}_\sigma) = 0. \tag{15}
$$

Taking into account the equation of state (2) and the formula (14), equation (15) becomes:

$$
F^2(P^{n+1}_{M^K_c}, \overrightarrow{t}^{n+1}_{M^K_c}, C^{n+1}_{M^K_c}, \overrightarrow{V}^{n+1}_{\sigma \in \epsilon_K}) = 0. \tag{16}
$$
2.3. Discretisation of the vapor mass equation

The previous approach is applied to the mixture mass equation to discretize in space and time the vapor mass balance equation.

Integration of vapor mass equation over the elementary control volume $M^K_c$ (see figure 2) and over a time interval from $t_n$ to $t_{n+1} = t_n + \Delta t$ gives:

$$\int_{M^K_c} \int_{t_n}^{t_{n+1}} \left\{ \phi \frac{\partial \rho C}{\partial t} + \nabla \cdot (\phi \rho C \mathbf{V}) + \nabla \cdot \left[ \phi \rho C (1 - C) \mathbf{V}^r \right] - \phi \mathbf{f}_V \right\} \mathrm{d}V \mathrm{d}t = 0. \quad (17)$$

The discrete form of (17) is:

$$\frac{(\rho C)^{n+1}_M - (\rho C)^n_M}{\Delta t} + \frac{1}{(\rho V)_M^K} \sum_{\sigma \in K} \left[ \mathcal{F}^{n+1}_\sigma + \mathcal{G}^{n+1}_\sigma \right] - (\Gamma_v)^{n+1}_M = 0 \quad (18)$$

where $\mathcal{F}^{n+1}_\sigma$ and $\mathcal{G}^{n+1}_\sigma$ are the fluxes approximations at the interface $\sigma$ at time $t_{n+1}$.

- $(\rho C)^n_\sigma$ and $(\rho \tau C_r (1 - C_r))^n_\sigma$ are derived at the cells center. On the faces, they are approximated by the following donor formulation:

$$\quad (\rho C)_\sigma^n = \begin{cases} (\rho C)^n_M & \text{if } (\mathbf{V}^{n+1}_\sigma, n_\sigma) > 0, \\ (\rho C)^n_M & \text{otherwise,} \end{cases}$$

$$[(\rho \tau C_r (1 - C_r))^n_\sigma] = \begin{cases} (\rho \tau C_r (1 - C))^n_M & \text{if } (\mathbf{V}^{n+1}_\sigma, n_\sigma) > 0, \\ (\rho \tau C_r (1 - C))^n_M & \text{otherwise.} \end{cases}$$

- $(\Gamma_v)^{n+1}_M$ is calculated using a correlation [2].

As a result, the discrete form of the vapor mass equation (18) can be written as:

$$F^3((\rho C)^{n+1}_M, C^{n+1}_M, V^{n+1}_\sigma, (\mathbf{V}^{n+1}_r)_\sigma) = 0. \quad (19)$$

Taking into account the equation of state (2) and the formula (14), the equation (19) becomes:

$$F^3((\rho)^{n+1}_M, C^{n+1}_M, V^{n+1}_\sigma, (\mathbf{V}^{n+1}_r)_\sigma) = 0. \quad (20)$$

2.4. Discretisation of the mixture momentum conservation equation

In its non-conservative form, mixture momentum equation of (1) becomes:

$$\phi \rho \left( \frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V} \right) + \nabla \cdot \left( \phi \rho C (1 - C) \mathbf{V}^r \otimes \mathbf{V}^r \right) + \phi \nabla P = \phi \mathbf{f} + \phi \rho \tilde{g}. \quad (21)$$

By rearranging Equation (21), porosity term $\phi$ only remains in the term: $\nabla \cdot \left( \phi \rho C (1 - C) \mathbf{V}^r \otimes \mathbf{V}^r \right)$. Since we assume constant porosity in the scope of this paper, the equation (21) becomes:

$$\rho \frac{\partial \mathbf{V}}{\partial t} + \rho \mathbf{V} \nabla \mathbf{V} + \nabla \cdot (\rho C (1 - C) \mathbf{V}^r \otimes \mathbf{V}^r) + \nabla P = \tilde{f} + \rho \tilde{g}. \quad (22)$$

The assumption needs to be addressed in future work.
Using the formula \( \rho \vec{V} \cdot \nabla \vec{V} = \nabla \cdot (\rho \vec{V} \otimes \vec{V}) - \vec{V} \nabla \cdot (\rho \vec{V}) \), mixture momentum equation (22) becomes:

\[
\rho \frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\rho \vec{V} \otimes \vec{V}) - \vec{V} \nabla \cdot (\rho \vec{V}) + \nabla \cdot (\rho (1-C) \vec{V}_r \otimes \vec{V}_r) + \nabla P = \tau_f + \rho g.
\]

In the following, we give the details of space and time discretisation of equation (23) projected in the direction \( \vec{e}_x \) (discretisation for directions \( \vec{e}_y \) and \( \vec{e}_z \), is derived by analogy to \( \vec{e}_x \)).

In direction \( \vec{e}_x \), equation (23 becomes:

\[
\rho \frac{\partial V_x}{\partial t} + \nabla \cdot (\rho V_x \vec{V}) - V_x \nabla \cdot (\rho \vec{V}) + \nabla \cdot [\rho C (1-C) V_x \vec{V}_r] + \frac{\partial P}{\partial x} = \tau_{f,x} + \rho g_x.
\]

Integration of equation (23) over an elementary control volume \( M_{u,K}^+ \) (see figure 3) and over a period from time \( t_n \) to \( t_{n+1} \) gives:

\[
\int_{M_{u,K}^+} \int_{t_n}^{t_{n+1}} \left\{ \rho \frac{\partial V_x}{\partial t} + \nabla \cdot (\rho V_x \vec{V}) - V_x \nabla \cdot (\rho \vec{V}) + \nabla \cdot [\rho C (1-C) V_x \vec{V}_r] + \frac{\partial P}{\partial x} - \tau_{f,x} - \rho g_x \right\} \, d\Omega dt = 0.
\]

The discrete form is:

\[
\rho^n_{M_{u,K}^+} \frac{(V_x)^{n+1}_{M_{u,K}^+} - (V_x)^n_{M_{u,K}^+}}{\Delta t} + \frac{1}{V_{M_{u,K}^+}} \sum_{\sigma \in \varepsilon_M} \left[ \mathcal{F}_\sigma^n - (V_x)^n_{M_{u,K}^+} \mathcal{F}_\sigma^n + \mathcal{H}_\sigma^n \right] + \left( \frac{\partial P}{\partial x} \right)^{n+1}_{M_{u,K}^+} - (\tau_{f,x})^{n+1}_{M_{u,K}^+} - (\rho g_x)^{n}_{M_{u,K}^+} = 0
\]

Figure 3. control volume \( M_{u,K}^+ \) in 2-dimensions

\( \varepsilon_M \) is the set of faces of \( M_{u,K}^+ \) and \( \varepsilon_{M_{u,K}^+} \) is the volume of \( M_{u,K}^+ \). \( \mathcal{F}_\sigma^n \), \( \mathcal{G}_\sigma^n \) and \( \mathcal{H}_\sigma^n \) are the approximations of the fluxes at the interface \( \sigma \) at time \( t_n \):

\[
\begin{align*}
\mathcal{F}_\sigma^n &= S_\sigma \rho^n_\sigma (V_x)^n_\sigma \vec{V}_\sigma \cdot \vec{n}_\sigma, \\
\mathcal{G}_\sigma^n &= S_\sigma \rho^n_\sigma \vec{V}_\sigma \cdot \vec{n}_\sigma, \\
\mathcal{H}_\sigma^n &= S_\sigma [\rho, C_r(1-C_r)]^n_\sigma (V_x)^n_\sigma (\vec{V}_r)^n_\sigma \cdot \vec{n}_\sigma.
\end{align*}
\]
As an example, each term in derived in 2D:

- Pressure gradient \( \left( \frac{\partial P}{\partial x} \right)_{M^{K+}} \) approximation:

\[
\left( \frac{\partial P}{\partial x} \right)_{M^{K+}}^{n+1} = \frac{P_{M^{Lx}}^{n+1} - P_{M^{K}}^{n+1}}{\Delta x}
\]

- Approximation of \( \bar{V}_{\sigma}^{n} \) and \( (V^x)^{n}_{\sigma} \) at the interfaces of \( M^{K+} \):

**Interface \( \sigma_x \)**

\[
\frac{\bar{V}_{M^{K+}}^{n} + \bar{V}_{M^{Lx}}^{n}}{2} = \frac{\bar{V}_{M^{K+}}^{n} + \bar{V}_{M^{Lx}}^{n}}{2}
\]

\[
(V^x)^{n}_{\frac{\sigma}{+}} = \begin{cases} (V^x)^{n}_{M^{K+}} & \text{if } \bar{V}_{\sigma}^{n} \bar{M}_{\sigma}^{n} > 0 \\ (V^x)^{n}_{M^{Lx}} & \text{otherwise} \end{cases}
\]

\[
(V^x)^{n}_{\frac{\sigma}{-}} = \begin{cases} (V^x)^{n}_{M^{K+}} & \text{if } \bar{V}_{\sigma}^{n} \bar{M}_{\sigma}^{n} > 0 \\ (V^x)^{n}_{M^{Lx}} & \text{otherwise} \end{cases}
\]

**Interface \( \sigma_y \)**

\[
\frac{\bar{V}_{M^{K+}}^{n} + \bar{V}_{M^{Lx}}^{n}}{2} = \frac{\bar{V}_{M^{K+}}^{n} + \bar{V}_{M^{Lx}}^{n}}{2}
\]

\[
(V^x)^{n}_{\frac{\sigma}{+}} = \begin{cases} (V^x)^{n}_{M^{K+}} & \text{if } \bar{V}_{\sigma}^{n} \bar{M}_{\sigma}^{n} > 0 \\ (V^x)^{n}_{M^{Lx}} & \text{otherwise} \end{cases}
\]

\[
(V^x)^{n}_{\frac{\sigma}{-}} = \begin{cases} (V^x)^{n}_{M^{K+}} & \text{if } \bar{V}_{\sigma}^{n} \bar{M}_{\sigma}^{n} > 0 \\ (V^x)^{n}_{M^{Lx}} & \text{otherwise} \end{cases}
\]

- Approximation of relative velocity \((V^x)^{n}_{\sigma} \) and \( (V^x)^{n}_{\sigma} \) at the interfaces of \( M^{K+} \):

\[
(V^x)^{n}_{\sigma} = \left( C_0 - 1 \right) \bar{V}_{\sigma} + \bar{V}_{\sigma \lim}
\]

\[
(V^x)^{n}_{\sigma} = \begin{cases} (V^x)^{n}_{M^{K+}} & \text{if } \bar{V}_{\sigma} \bar{M}_{\sigma} > 0 \\ (V^x)^{n}_{M^{Lx}} & \text{otherwise} \end{cases}
\]

\[
(V^x)^{n}_{\sigma} = \begin{cases} (V^x)^{n}_{M^{K+}} & \text{if } \bar{V}_{\sigma} \bar{M}_{\sigma} > 0 \\ (V^x)^{n}_{M^{Lx}} & \text{otherwise} \end{cases}
\]

- Approximation of scalar variables at the interfaces of \( M^{K+} \):

\[
\rho^{n}_{\sigma^{x}} = \rho_{M^{K}}^{n}, \quad \rho^{n}_{\sigma^{x}} = \rho_{M^{Lx}}^{n}
\]

\[
\rho^{n}_{\sigma^{x}} = \begin{cases} \rho_{M^{K}}^{n} + \rho_{M^{Lx}}^{n} & \text{if } \bar{V}_{\sigma} \bar{M}_{\sigma} > 0 \\ \rho_{M^{Lx}}^{n} + \rho_{M^{Lx}}^{n} & \text{otherwise} \end{cases}
\]

\[
\rho^{n}_{\sigma^{y}} = \begin{cases} \rho_{M^{K}}^{n} + \rho_{M^{Lx}}^{n} & \text{if } \bar{V}_{\sigma} \bar{M}_{\sigma} > 0 \\ \rho_{M^{Lx}}^{n} + \rho_{M^{Lx}}^{n} & \text{otherwise} \end{cases}
\]

The same method is applied to derive \( [\rho C(1 - C)]^{n}_{\sigma} \):

\[
[\rho C(1 - C)]^{n}_{\sigma^{x}} = \frac{[\rho C(1 - C)]^{n}_{M^{K}} + [\rho C(1 - C)]^{n}_{M^{Lx}}}{2}, \quad [\rho C(1 - C)]^{n}_{\sigma^{y}} = \frac{[\rho C(1 - C)]^{n}_{M^{Lx}} + [\rho C(1 - C)]^{n}_{M^{Lx}}}{2}
\]

\[
[\rho C(1 - C)]^{n}_{\sigma^{y}} = \begin{cases} \frac{[\rho C(1 - C)]^{n}_{M^{K}} + [\rho C(1 - C)]^{n}_{M^{Lx}}}{2} & \text{if } \bar{V}_{\sigma} \bar{M}_{\sigma} > 0, \\ \frac{[\rho C(1 - C)]^{n}_{M^{Lx}} + [\rho C(1 - C)]^{n}_{M^{Lx}}}{2} & \text{otherwise} \end{cases}
\]
\[
[r_{CR} (1 - C_r)]_{s-y}^n = \begin{cases} 
\frac{[\rho C (1-C)]_{s-y}^n \rho_{MK} C + [\rho C (1-C)]_{s-y}^{n+L_x}}{2} & \text{if } (\vec{V}_1^y)_{s-y} \cdot \vec{n}_{s-y} > 0, \\
\frac{[\rho C (1-C)]_{s-y}^{n+L_x} + [\rho C (1-C)]_{s-y}^{n-L_x}}{2} & \text{otherwise.}
\end{cases}
\]

At the interfaces \(\sigma_x^+\) and \(\sigma_y^-\), we easily get the same result as for interfaces \(\sigma_y^+\) and \(\sigma_y^-\) when substituting \(y\) by \(z\).

- Friction forces \((\tau_x^f)^{n+1}_{MK^+}\) approximation:

\[
(\tau_x^{f})^{n+1}_{MK^+} = (\tau_x^{f_w})^{n+1}_{MK^+} + (\tau_x^{f_s})^{n+1}_{MK^+}
\]

where \(\tau_x^{f_w}\) is the wall friction and \(\tau_x^{f_s}\) is the singular friction. It becomes:

\[
(\tau_x^{f_w})^{n+1}_{MK^+} = \frac{-f^x_w}{2(D^x_h)_{MK^+}} \rho_{MK^+}^{n} (V_x^{n+1}_{MK^+})^2
\]

\[
(\tau_x^{f_s})^{n+1}_{MK^+} = \frac{-K^x}{2} \rho_{MK^+}^{n} (V_x^{n+1}_{MK^+})^2
\]

where \(K^x\) and \(f^x_w\) are friction coefficients.

Finally, discretisation of the mixture momentum equation in direction \(e_x\) (25) can be written as:

\[
F^4(P_{MK^+}^{n+1}, P_{MK^+}^{n+L_x}, (V_x^{n+1}_{MK^+} = 0). (26)
\]

By analogy, discretisation in directions \(e_y\) and \(e_z\) is:

\[
F^5(P_{MK^+}^{n+1}, P_{MK^+}^{n+L_y}, (V_y^{n+1}_{MK^+} = 0), (27)
\]

\[
F^6(P_{MK^+}^{n+1}, P_{MK^+}^{n+L_z}, (V_z^{n+1}_{MK^+} = 0). (28)
\]

3. Solution Algorithm

3.1. Construction of the linear system to solve

\((S)\) is the non-linear system to solve for each time step. \((S)\) is:

\[
(S) = \begin{cases} 
F^1(P, h, C, \vec{V}) = 0, \\
F^2(P, h, C, \vec{V}) = 0, \\
F^3(P, h, C, \vec{V}) = 0, \\
F^4(P, \vec{V}) = 0, \\
F^5(P, \vec{V}) = 0, \\
F^6(P, \vec{V}) = 0.
\end{cases} (29)
\]

System \((S)\) is solved by using a Newton-Raphson iterative method which consists in solving a linearisation of \((S)\) at each iteration. For each iteration \(k\) of the Newton-Raphson algorithm, we solve for the linear system:

\[
\mathcal{J}(\mathcal{U}^k) \Delta \mathcal{U}^{k+1} = S(\mathcal{U}^k, \mathcal{U}^k) (30)
\]
where \( k \) is the number of the iteration, \( U = (P, h, C, V^x, V^y, V^z)^t \) is unknown vector and \( \Delta U^{k+1} = U^{k+1} - U^k \) is increment of \( U \), the matrix \( J \) is the Jacobian matrix of the system \((S)\) and the vector \( S = -(F^1, F^2, F^3, F^4, F^5, F^6)^t \) is the right-hand-side vector containing the residuals of equations (29) evaluated at the previous iteration. The solution \( U^{n+1} \) is \( U_k \) when \( k \to +\infty \). We choose \( U_k = 0 = U^n \) as the starting point of the Newton-Raphson algorithm. The criteria that is used to determine whether the Newton-Raphson method has converged is:

\[
\max || \Delta U^{k+1} || < \epsilon = 10^{-5}.
\]

### 3.2. The ”pressure-based solver” method

A “semi-implicit” scheme is used to discretize the four equations (1). It leads to significant simplification of the terms of the matrix from the momentum equations \( F^4, F^5 \) and \( F^6 \). Hence, velocity increments are expressed as a function of pressure increments (see section 3.2.1). They can be easily substituted. Section §3.2.2 shows how to get rid of all scalar variables (but pressure increments) and derive a linear equation where the only unknown is pressure increment.

#### 3.2.1. Suppression of velocity increments

The purpose of this step is to write velocity increments as a function of pressure increments. Only the momentum equations are taken into account. It can be viewed as a “partial” or simplified linear system:

\[
\begin{bmatrix}
\frac{\partial F^4}{\partial P} & \frac{\partial F^4}{\partial h} & \frac{\partial F^4}{\partial C} & \frac{\partial F^4}{\partial V^x} & \frac{\partial F^4}{\partial V^y} & \frac{\partial F^4}{\partial V^z} \\
\frac{\partial F^5}{\partial P} & \frac{\partial F^5}{\partial h} & \frac{\partial F^5}{\partial C} & \frac{\partial F^5}{\partial V^x} & \frac{\partial F^5}{\partial V^y} & \frac{\partial F^5}{\partial V^z} \\
\frac{\partial F^6}{\partial P} & \frac{\partial F^6}{\partial h} & \frac{\partial F^6}{\partial C} & \frac{\partial F^6}{\partial V^x} & \frac{\partial F^6}{\partial V^y} & \frac{\partial F^6}{\partial V^z}
\end{bmatrix}
\begin{bmatrix}
\Delta U^1 \\
\Delta U^2 \\
\Delta U^3 \\
\Delta U^4 \\
\Delta U^5 \\
\Delta U^6
\end{bmatrix}
= 
\begin{bmatrix}
S^4 \\
S^5 \\
S^6
\end{bmatrix}
\tag{31}
\]

According to (26),(27) and (28), functions \( F^4, F^5 \) and \( F^6 \) do not depend on mixture enthalpy \( h \) and mass vapor concentration \( C \) at time step \( t^{n+1} \). Thus, their derivatives with respect to these variables are equal to zero. Hence:

- the derivatives of \( F^4 \) with respect to \( V^y \) and \( V^z \) are equal to zero,
- the derivatives of \( F^5 \) with respect to \( V^x \) and \( V^z \) are equal to zero,
- the derivatives of \( F^6 \) with respect to \( V^x \) and \( V^y \) are equal to zero.

System (31) then becomes:

\[
\begin{bmatrix}
\frac{\partial F^4}{\partial P} & 0 & 0 & \frac{\partial F^4}{\partial V^x} & 0 & 0 \\
\frac{\partial F^5}{\partial P} & 0 & 0 & 0 & \frac{\partial F^5}{\partial V^y} & 0 \\
\frac{\partial F^6}{\partial P} & 0 & 0 & 0 & 0 & \frac{\partial F^6}{\partial V^z}
\end{bmatrix}
\begin{bmatrix}
\Delta U^1 \\
\Delta U^2 \\
\Delta U^3 \\
\Delta U^4 \\
\Delta U^5 \\
\Delta U^6
\end{bmatrix}
= 
\begin{bmatrix}
S^4 \\
S^5 \\
S^6
\end{bmatrix}
\tag{32}
\]

Writing equation (32) over cells \( M_s^{K+}, M_v^{K+} \) and \( M_w^{K+} \) leads to the following equations system:
By analogy, we determine velocity increments \((\Delta V_x)_{M^K-}, (\Delta V_y)_{M^K-}\) and \((\Delta V_z)_{M^K-}\) as a function of pressure increments.

Through this step, velocity increments \(\Delta u_1, \Delta u_2\) and \(\Delta u_6\) are derived on the faces of the meshing as a function of pressure increments. This formulation is used for the next step (see §3.2.2) to write a linear pressure equation (where pressure increments are the only unknowns).

### 3.2.2. Triangulation

This step aims at eliminating scalar variables increments. The three scalar variable equations (mixture mass, mass vapor concentration and mixture energy) are used as follow:

\[
\begin{equation}
\begin{pmatrix}
\frac{\partial F^1}{\partial P} & \frac{\partial F^1}{\partial h} & \frac{\partial F^1}{\partial C} & \frac{\partial F^1}{\partial V_x} & \frac{\partial F^1}{\partial V_y} & \frac{\partial F^1}{\partial V_z} \\
\frac{\partial F^2}{\partial P} & \frac{\partial F^2}{\partial h} & \frac{\partial F^2}{\partial C} & \frac{\partial F^2}{\partial V_x} & \frac{\partial F^2}{\partial V_y} & \frac{\partial F^2}{\partial V_z} \\
\frac{\partial F^3}{\partial P} & \frac{\partial F^3}{\partial h} & \frac{\partial F^3}{\partial C} & \frac{\partial F^3}{\partial V_x} & \frac{\partial F^3}{\partial V_y} & \frac{\partial F^3}{\partial V_z}
\end{pmatrix}
\begin{pmatrix}
\Delta U_1 \\
\Delta U_2 \\
\Delta U_3 \\
\Delta U_4 \\
\Delta U_5 \\
\Delta U_6
\end{pmatrix}
= \begin{pmatrix} S^1 \\ S^2 \\ S^3 \end{pmatrix}
\end{equation}
\]

By applying algebraic operations, the linear system (34) can be written in the following form:

\[
\begin{equation}
\begin{pmatrix}
J^{1,1} & 0 & J^{1,4} & J^{1,5} & J^{1,6} & 0 \\
J^{2,1} & J^{2,2} & 0 & J^{2,4} & J^{2,5} & J^{2,6} \\
J^{3,1} & J^{3,2} & J^{3,3} & J^{3,4} & J^{3,5} & J^{3,6}
\end{pmatrix}
\begin{pmatrix}
\Delta U_1 \\
\Delta U_2 \\
\Delta U_3 \\
\Delta U_4 \\
\Delta U_5 \\
\Delta U_6
\end{pmatrix}
= \begin{pmatrix} D^1 \\ D^2 \\ S^3 \end{pmatrix}
\end{equation}
\]

By developing the first row of system (35) over cell \(M^K_e\), we get the following equation:

\[
(J^{1,1})_{M^K_e} (\Delta P)_{M^K_e} + (J^{1,4})_{M^K_e} (\Delta V_x)_{M^K_e} + (J^{1,5})_{M^K_e} (\Delta V_y)_{M^K_e} + (J^{1,6})_{M^K_e} (\Delta V_z)_{M^K_e} -
+ (J^{1,5})_{M^K_e} (\Delta V_y)_{M^K_e} + (J^{1,6})_{M^K_e} (\Delta V_z)_{M^K_e} = D^1_{M^K_e}
\]

Using velocity increments from equation (33), equation (36) becomes:
\[
A(\Delta P)_{MK} + B(\Delta P)_{MK_{+Ls}} + C(\Delta P)_{MK_{-Ls}} + D(\Delta P)_{MK_{+Ly}} + E(\Delta P)_{MK_{-Ly}} + F(\Delta P)_{MK_{+Ls}} + G(\Delta P)_{MK_{-Ls}} = S
\]

where:

\[
A = (J^{1.1})_{MK} - (J^{1.4})_{MK_{+Ls}}\frac{\partial F_{MK}^4}{\partial P_{MK}} - (J^{1.4})_{MK_{-Ly}}\frac{\partial F_{MK}^4}{\partial V_{MK}^S}\]

\[
B = -(J^{1.4})_{MK_{+Ls}}\frac{\partial F_{MK}^4}{\partial P_{MK}} + (J^{1.4})_{MK_{-Ly}}\frac{\partial F_{MK}^4}{\partial V_{MK}^S}
\]

\[
C = -(J^{1.4})_{MK_{-Ls}}\frac{\partial F_{MK}^4}{\partial P_{MK}} + (J^{1.4})_{MK_{+Ly}}\frac{\partial F_{MK}^4}{\partial V_{MK}^S}
\]

\[
S = D_{MK}^1 - \left[ (J^{1.4})_{MK_{+Ls}}S_{MK_{+}}^4 + (J^{1.4})_{MK_{-Ly}}S_{MK_{-}}^4 + (J^{1.5})_{MK_{+Ls}}S_{MK_{+}}^5 + (J^{1.5})_{MK_{-Ly}}S_{MK_{-}}^5 \right] + \left[ (J^{1.6})_{MK_{+Ls}}S_{MK_{-}}^6 + (J^{1.6})_{MK_{-Ly}}S_{MK_{+}}^6 \right]
\]

Pressure increments \(\Delta U_1\) are derived by solving pressure equation (37). Then, velocity increments \(\Delta U_4, \Delta U_5\) and \(\Delta U_6\) are derived from (33).

3.2.3. **Enthalpy and concentration Increments**

Enthalpy increments are solved when developing the second row of system (35):

\[
(J^{2.1})_{MK}(\Delta P)_{MK} + (J^{2.2})_{MK}(\Delta h)_{MK} + (J^{2.4})_{MK_{+Ls}}(\Delta V^x)_{MK_{+}} + (J^{2.4})_{MK_{-Ls}}(\Delta V^x)_{MK_{-}} + (J^{2.5})_{MK_{+Ly}}(\Delta V^y)_{MK_{+}} + (J^{2.5})_{MK_{-Ly}}(\Delta V^y)_{MK_{-}} + (J^{2.6})_{MK_{+Ls}}(\Delta V^z)_{MK_{+}} + (J^{2.6})_{MK_{-Ls}}(\Delta V^z)_{MK_{-}} = D_{MK}^2
\]
In equation (38), pressure increments and velocity increments are now known (see §3.2.2). Then only remain unknowns are enthalpy increments. They are derived with:

\[
(\Delta h)_{M^k} = \frac{1}{(J^{2.1})_{M^k}} \left[ D^2_{M^k} - (J^{2.1})_{M^k} (\Delta P)_{M^k} - (J^{2.4})_{M^k} (\Delta V^z)_{M^k} - (J^{2.4})_{M^k} (\Delta V^x)_{M^k} - (J^{2.4})_{M^k} (\Delta V^y)_{M^k} - (J^{2.5})_{M^k} (\Delta V^y)_{M^k} - (J^{2.5})_{M^k} (\Delta V^x)_{M^k} - (J^{2.6})_{M^k} (\Delta V^z)_{M^k} - (J^{2.6})_{M^k} (\Delta V^x)_{M^k} \right].
\]

The same method applies for concentration increments that can be computed using the third row of system (35):

\[
(\Delta C)_{M^k} = \frac{1}{\partial C_{M^k}} \left[ S^h_{M^k} - \partial F^3_{M^k} / \partial P_{M^k} (\Delta P)_{M^k} - \partial F^3_{M^k} / \partial h_{M^k} (\Delta h)_{M^k} - \partial F^3_{M^k} / \partial (V^x)_{M^k} (\Delta V^x)_{M^k} - \partial F^3_{M^k} / \partial (V^y)_{M^k} (\Delta V^y)_{M^k} - \partial F^3_{M^k} / \partial (V^z)_{M^k} (\Delta V^z)_{M^k} - \partial F^3_{M^k} / \partial (V^x)_{M^k} (\Delta V^x)_{M^k} - \partial F^3_{M^k} / \partial (V^y)_{M^k} (\Delta V^y)_{M^k} - \partial F^3_{M^k} / \partial (V^z)_{M^k} (\Delta V^z)_{M^k} \right].
\]

4. Numerical tests

4.1. Low Mach regime

In this section, we check the efficiency of the numerical method described in sections §2 and §3 by carrying out various 1D tests. For all these tests, we used the stiffened gas law for the equation of state [5] and we solved for the homogeneous model (\( \vec{V} = 0 \), in Eq. 1). Newton-Raphson method requires only a few iterations (typically less than 5) to get convergence. The time step \( \Delta t \) follows as:

\[
\Delta t = CFL \frac{\Delta x}{\max(|V|)}
\]

where \( \Delta x \) is the mesh spacing, \( V \) is the fluid velocity and \( CFL = 1 \) is the Courant-Friedrichs-Lewy number.

4.1.1. Channel with varying porosity

**Test description:** Stationary liquid flows in a 1D channel with varying porosity. The channel is 4.2 m long. Porosity in the middle of the channel (1.4 \( \leq z \leq 2.8 \)) is set to \( \phi_1 = 0.5, \phi_2 = 1 \) elsewhere. Pressure at the outlet of the channel is set at \( P_s = 155 \) bar and liquid velocity at the inlet of the channel is \( V_e = 1 \) m s\(^{-1}\). Friction wall is set to zero and gravity is neglected. Here, the goal is to test the ability of space discretisation to preserve momentum with varying porosity. This test is very important because the porosity is a key ingredient to simulate the flow in a nuclear reactor core. Indeed, there is some section reduction (porosity) in a nuclear reactor core. The numerical solution is computed over 1000 cells.

**Results:** Figure 4 shows velocity and pressure profiles along the \( z \)-axis for the analytical solution and the solution computed with the pressure-based solver. Both give the same solution. The analytical solution is
obtained by assuming that the density of the fluid $\rho$ is constant (low Mach regime and unheated fluid). Taking account of the above assumptions, the analytical solution is:

$$V(z) = \begin{cases} \frac{\phi_2}{\phi_1} V_e & \text{if } 1.4 \leq z \leq 2.8, \\ V_e & \text{otherwise}. \end{cases}$$  \tag{40}$$

$$P(z) = \begin{cases} P_s + \frac{1}{2} \rho V_e^2 \left(1 - \frac{\phi_2^2}{\phi_1^2}\right) & \text{if } 1.4 \leq z \leq 2.8, \\ P_s & \text{otherwise}. \end{cases}$$  \tag{41}$$

The velocity (40) and the pressure (41) profiles along the $z$-axis are solutions of the system:

$$\begin{cases} \frac{\partial(\rho V)}{\partial z} = 0, \\ \rho V \frac{\partial V}{\partial z} + \frac{\partial P}{\partial z} = 0. \end{cases}$$

Figure 4. Channel with varying porosity: Pressure-based solver (red), analytic solution (green)

4.1.2. Channel with singular charge loss

**Test description:** Stationary liquid flows in a 1D channel with a pressure drop (modeling the effect of a mixing grid for example) located in the middle of the channel. The channel is 4.2 m long. Pressure at the outlet of the channel is set at 155 bar and liquid velocity at the inlet of the channel is 1 m s$^{-1}$. Friction wall is set to zero and gravity is neglected. The singular friction factor is $K = 100$, which is a rather drastic pressure loss. The numerical solution is computed over 1000 cells.

**Results:** Figure 5 shows pressure profile along the $z$-axis. Pressure decreases after the flow passed through the mixing grid, as expected.
4.1.3. Boiling channel

**Test description:** Conditions for this test are close to operating conditions for PWRs (Pressurized Water Reactors). A 4.2 m long channel is heated with a uniform thermal flux $Q = 1.8\times 10^8$ W m$^{-2}$ with the following boundary conditions: inlet concentration $C = 0$, inlet enthalpies $h_l = 1300$ kJ kg$^{-1}$ and $h_v = 2600$ kJ kg$^{-1}$, inlet velocities $u_l = u_v = 1$ m s$^{-1}$ and outlet pressure $P_s = 155$ bar. Wall friction is negligible, as well as gravity. The numerical solution is computed with 250 cells.

**Results:** In the figure 6, we compare the results obtained with our pressure solver to an analytical solution obtained with the low Mach mixture model proposed in [5]. Both results are equivalent.
4.2. Compressible regime

To test our Pressure-based Solver in the compressible regime, we study test cases described in [12] and presented in Table 1. All tests are performed with an ideal gases $\rho = \frac{\gamma}{\gamma - 1} P$, with a constant $\gamma = 1.4$. All chosen data consist of two constant states separated by a discontinuity at $x = x_0$. The position of the discontinuity is stated in the legend. The spatial domain is $0 \leq x \leq 1$ and the numerical solution is computed with 100 and 5000 cells. We obtain convergence of Newton-Raphson method in a few iterations (typically less than 10). The time step $\Delta t$ follows as:

$$\Delta t = \frac{\Delta x}{2 \max(|V + c|, |V - c|)}$$

where $\Delta x$ is the mesh spacing, $V$ is the fluid velocity and $c$ is the sound speed.
For test 1, the Sod’s Shock Tube problem [12] is modified slightly. The solution of the problem has a right shock wave, a right travelling contact wave and a left sonic rarefaction wave. The purpose of this test is to assess the entropy satisfaction property of the numerical methods. The results for this test are shown in Figure 7 against the exact results. We can see that the Pressure-based Solver struggles slightly with the internal energy, but otherwise performs very close to the exact Riemann solver with particularly accurate result around the sonic point unlike Roe scheme [12].

Test 2 consists of two symmetric rarefaction waves and a trivial contact wave, with the star region between the non-linear waves close to vacuum. This problem is a good assessment of the performance of numerical methods for low-density flows. The results for this test are shown in Figure 8 against the exact results and we can see that the accuracy of the numerical results is nearly from those of the exact Riemann solver unlike Roe scheme which fails near low-density flows [13].

<table>
<thead>
<tr>
<th>Test</th>
<th>$\rho_L$ ($kg.m^{-3}$)</th>
<th>$V_L$ ($m.s^{-1}$)</th>
<th>$P_L$ (Pa)</th>
<th>$\rho_R$ ($kg.m^{-3}$)</th>
<th>$V_R$ ($m.s^{-1}$)</th>
<th>$P_R$ (Pa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>1</td>
<td>0.75</td>
<td>1</td>
<td>0.125</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>Test 2</td>
<td>1</td>
<td>-2</td>
<td>0.4</td>
<td>1</td>
<td>2</td>
<td>0.4</td>
</tr>
<tr>
<td>Test 3</td>
<td>1</td>
<td>0</td>
<td>1000</td>
<td>1</td>
<td>0</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 1. Data for three Riemann problem tests for testing the Pressure-based Riemann solver.

**Figure 7.** Pressure-based Riemann solver applied to test 1 of Table 1: Pressure-based solver (red and blue lines), Roe solver (sky blue line) and exact (dash) solutions compared at time 0.2

Test 2 consists of two symmetric rarefaction waves and a trivial contact wave, with the star region between the non-linear waves close to vacuum. This problem is a good assessment of the performance of numerical methods for low-density flows. The results for this test are shown in Figure 8 against the exact results and we can see that the accuracy of the numerical results is nearly from those of the exact Riemann solver unlike Roe scheme which fails near low-density flows [13].
Figure 8. Pressure-based Riemann solver applied to test 2 of Table 1: Numerical (red and blue lines) and exact (dash) solutions compared at time 0.12

The robustness and accuracy of the Pressure-based solver is tested with Test 3. The solution of Test 3 consists of a strong shock wave, a contact surface and a left rarefaction wave. We can see from Figures 9 (100 cells) that the Pressure-based solver struggles slightly with the density, but otherwise performs very close to the exact solution. We can also see that the results obtained with the Pressure-based and Roe solvers are near. However, when the mesh is refined (1000 cells), we see that the Pressure-based solver do not converge to the exact solution.

Figure 9. Pressure-based Riemann solver applied to test 3 of Table 1: Pressure-based solver (red and blue lines), Roe solver (sky blue line) and exact (dash) solutions compared at time 0.012
5. Conclusion

A pressure-based solver is developed to overcome FLICA4 current solver shortcomings. The code main drawback includes robustness difficulties because of checkerboard oscillations. Generation of checkerboard oscillations arise in the low Mach number regime, due to a so-called "pressure correction" in combination with the finite volume numerical method based on an extension of the Roe's approximate Riemann colocated solver.

A new solver is developed on staggered grids to guarantee that no checkerboard oscillations are to be generated. The pressure-based solver is tested against three types of numerical tests. The first series consists in simplified nuclear core models, close to actual systems of interest, and include sharp axial variations of the source terms (porosity, pressure drop, ...) for low Mach number regime. The pressure based-solver solutions are equivalent to the analytical ones. These results are very promising since the new solver improves FLICA4 code robustness without degrading its accuracy. However, the equations are in a non-conservative form, it is expected that it might not be as accurate in the compressible regime. Based on two test-cases, we demonstrate that the pressure-based solver is fairly accurate as long as the meshing remains relatively coarse. Because of the domain of application of the FLICA4 code, coarse meshing and low Mach regime, the new pressure-based solver described in this paper is a real alternative to the current solver. The new solver should greatly improve the use of the code in an industrial context for nuclear reactor safety analysis.

References

[1] O. Grégoire, Établissement formel d’un modèle diphasique macroscopique à 6 équations, Lien avec le modèle macroscopique à 4 équations FLICA4, Note Technique CEA 2004, SFME/LETTR/RT/04-007/A.