A PISO-like algorithm to simulate superfluid helium flow with the two-fluid model

Cyprien Soulaine [a,*], Michel Quintard [a,b], Hervé Allain [c], Bertrand Baudouy [d],
Rob Van Weelderen [c]

[a] Université de Toulouse; INPT, UPS; IMFT (Institut de Mécanique des Fluides de Toulouse), Allée Camille Soula, F-31400 Toulouse, France
[b] CNRS; IMFT, F-31400 Toulouse, France
[c] CERN, TE-CRG, 1211 Geneva 23, Switzerland
[d] CEA Saclay, Irfu/SACM, F-91191 Gif-sur-Yvette, France

ABSTRACT

This paper presents a segregated algorithm to solve numerically the superfluid helium (He II) equations using the two-fluid model. In order to validate the resulting code and illustrate its potential, different simulations have been performed. First, the flow through a capillary filled with He II with a heated area on one side is simulated and results are compared to analytical solutions in both Landau and Gorter–Mellink flow regimes. Then, transient heat transfer of a forced flow of He II is investigated. Finally, some two-dimensional simulations in a porous medium model are carried out.

1. Introduction

The physical phenomena of superfluid liquid helium can be macroscopically represented in a general framework best illustrated by Landau's two fluids model [1,2]. In this model, He II is seen as though it were a mixture of two different fluids. One of these is a normal viscous fluid, the other is a superfluid and moves with zero viscosity along a solid surface. Besides this particularity, it must be noted that the superfluid flow does not carry entropy. Moreover, the momentum equations include a thermo-mechanical force that occurs when a temperature gradient exists. When the superfluid velocity reaches a certain critical value, some turbulence phenomena arise and the above equations are no longer valid. In such a case, one introduces the so-called Gorter–Mellink mutual friction term, which has been proposed to estimate the interaction between the two components [3,4].

Only few multi-dimensional codes that solve the complete two-fluid model are reported in the literature. Actually, as pointed out by Kitamura et al. [5], authors have met numerical difficulties, probably due to the fact that the thermo-mechanical and the

Gorter–Mellink mutual friction terms are several orders of magnitude larger than the other terms in the superfluid momentum equation. The heat wave propagation intrinsic to superfluidity, the so-called second sound, may also lead to additional instabilities. To overcome these numerical difficulties, authors have either focused on one-dimensional solutions [6,7] or proposed simplified multi-dimensional models. For instance, Ramadan and Witt [8] have modified the momentum equations as the thermo-mechanical and the mutual friction terms are larger than the others in *balancing* each other (hence they can be dropped from the equations) to create a heat flux with the «familiar» $|\nabla T|^{1/3}$ dependence. Assuming also that these two terms are approximately equal, Kitamura et al. [5], and later Suekane et al. [9] and Pietrowicz and Baudouy [10], proposed a simplified model made of a conventional continuity equation, a modified Navier–Stokes equation for the total velocity and a heat equation for temperature. Tatsumoto et al. [11] proposed a numerical segregated solution to solve superfluid equations. In their algorithm, the total fluid velocity and the pressure fields are first solved, then the superfluid momentum is computed from the resulting pressure field, after which the normal velocity is deduced point-wise. Finally, the temperature equation is solved and the whole sequence iterated for a new time step.

Actually, the discussion regarding the choice of using segregated or coupled algorithms is still an open debate. On one hand,

* Corresponding author.
  E-mail address: csoulain@stanford.edu (C. Soulaine).

the superfluid equations are strongly coupled, mainly from the thermo-mechanical and the mutual friction terms, and coupled algorithms offer suitable solutions [12,7]. On the other hand, segregated approaches tend to consume less memory precisely because the entire problem is never assembled. However, in that case, how can one choose the optimal sequence? Why should the normal velocity be deduced from the superfluid velocity as suggested by Tatsumoto et al. [11] and not the opposite? In this paper, we propose an alternative segregated approach to solve superfluid equations. Our algorithm is an extension to the two-fluid model of the Pressure Implicit Operator Splitting (PISO) algorithm by Issa [13]. In our method, a pressure equation is directly derived from the total mass balance and both momentum equations. We called this algorithm Super-PISO. It is at the core of *HellFOAM*, the superfluid code we have developed using the OpenFOAM® technology.

The paper is organized as follows. In Section 2, we present the superfluid equations and the main assumptions. In Section 3, we introduce the discretization of the superfluid equations and the Super-PISO algorithm. Then in Section 4, we solve numerically the superfluid equations in different configurations to validate and illustrate the potential of *HellFOAM* code.

2. Mathematical model and assumptions

In this section, we introduce the mathematical model on which our numerical study relies. In what follows we denote by subscripts $_n$ and $_s$ all quantities related to the normal and superfluid flow respectively. Hence, ρ_n and ρ_s are the normal and superfluid densities and their sum is the actual density of He II,

$$\rho = \rho_n + \rho_s. \tag{1}$$

These quantities vary according to the temperature: ρ_s vanishes at the λ- point where the fluid becomes fully normal, ρ_n is null at absolute zero. Moreover, as a classical fluid, the normal component has a fluid viscosity μ_n and a fluid heat conductivity k_n. Besides these variables, the model also requires the Gorter–Mellink coefficient A introduced later. All these physical properties are assumed to be a function of temperature only and the pressure dependency is neglected. Their values are provided by polynomial interpolations from the HePak thermodynamic package by Cryodata Inc (http://www.cryodata.com/).

The mass conservation for the whole fluid reads

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho_n \mathbf{v}_n + \rho_s \mathbf{v}_s) = 0. \tag{2}$$

Introducing Γ, the rate at which the superfluid particles become normal during the heating process, we can write the continuity equations for the normal and superfluid components as

$$\frac{\partial \rho_n}{\partial t} + \nabla \cdot (\rho_n \mathbf{v}_n) = \Gamma, \tag{3}$$

and

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot (\rho_s \mathbf{v}_s) = -\Gamma \tag{4}$$

in which Γ is an unknown variable that has to be determined and varies in time and space. One notices that $\Gamma = 0 \text{ kg/m}^3/\text{s}$ when no species conversion occurs. In fact, it is not granted that the rate of superfluid particles becoming normal (r_{sn}) is equal to the rate of normal particles that move to the superfluid state (r_{ns}). Ideally, these two phenomena should be expressed separately and Γ should be replaced by $\Gamma = r_{sn} - r_{ns}$ in the continuity equations (3)–(4). One example of derivation of such a non-equilibrium theory can be found in Woods [14] where separate expressions are given for r_{sn} and r_{ns}. However, in the absence of such additional

equation linking r_{sn} and r_{ns} the problem has an extra unknown variable compared to the number of equations available and the problem turns ill-posed. In what follows, we will consider a symmetry of normal/superfluid mass transfer (i.e., $r_{sn} = -r_{ns}$).

The momentum balance equation for normal component reads,

$$\frac{\partial \rho_n \mathbf{v}_n}{\partial t} + \nabla \cdot (\rho_n \mathbf{v}_n \mathbf{v}_n) = -\frac{\rho_n}{\rho} \nabla p - \rho_s s \nabla T + \nabla \cdot (\mu_n \nabla \mathbf{v}_n)$$
$$- A \rho_n \rho_s |\mathbf{v}_n - \mathbf{v}_s|^2 (\mathbf{v}_n - \mathbf{v}_s) + [r_{sn} \mathbf{v}_s - r_{ns} \mathbf{v}_n], \tag{5}$$

and for superfluid component reads,

$$\frac{\partial \rho_s \mathbf{v}_s}{\partial t} + \nabla \cdot (\rho_s \mathbf{v}_s \mathbf{v}_s) = -\frac{\rho_s}{\rho} \nabla p + \rho_s s \nabla T$$
$$+ A \rho_n \rho_s |\mathbf{v}_n - \mathbf{v}_s|^2 (\mathbf{v}_n - \mathbf{v}_s) - [r_{sn} \mathbf{v}_s - r_{ns} \mathbf{v}_n]. \tag{6}$$

In these equations, $\rho_s s \nabla T$ represents the thermo-mechanical force which occurs when a temperature gradient exists. It is responsible for creating the counterflow in which normal fluid moves down the temperature gradient from the heat source to the bath, while the superfluid flows towards the heat source. The term $A \rho_n \rho_s |\mathbf{v}_n - \mathbf{v}_s|^2 (\mathbf{v}_n - \mathbf{v}_s)$, which is present in the equation only at high velocities, represents the Gorter–Mellink mutual friction term. The last term on the right hand side of Eqs. (5)–(6) represents the momentum transfer between the two fluids. It means that the superfluid momentum decreases when particles turn to normal state. Landau and Lifshitz [15] suggest that there is no momentum transfer while Woods [14] who derived superfluid hydrodynamic equations from a thermodynamic point of view keeps it. With the assumption that the rate at which the normal fluid gains mass from the superfluid is equal to the rate at which the superfluid gains mass from normal fluid, then the momentum transfer term reads

$$[r_{sn} \mathbf{v}_s - r_{ns} \mathbf{v}_n] = \frac{\Gamma}{2} (\mathbf{v}_s + \mathbf{v}_n), \tag{7}$$

as proposed by Ramadan and Witt [8] in their numerical model. This is the formulation we use in our simulations. Actually, several variants of the two-fluid model may be found in the literature, in particular depending on the choice of the reference frame ([2], see discussion in [14]). The diffusion energy term $\frac{\rho_n \rho_s}{2\rho} \nabla |\mathbf{v}_s - \mathbf{v}_n|^2$ appears in the standard Landau's model which is not included in our analysis. In fact, we have included it in another version of our code without major mathematical problem and we found that it does not play a significant role in the test cases presented in this paper. It must be emphasized that other models for superfluid helium flow are under discussion, like for instance those derived from Extended Irreversible Thermodynamics [16,17] but this is beyond the scope of this paper.

Besides these two momentum conservation equations, we should consider the energy conservation which is only transported by normal fluid. We have an energy conservation equation that reads,

$$\frac{\partial \rho s}{\partial t} + \nabla \cdot (\rho s \mathbf{v}_n) = \nabla \cdot \left(\frac{k_n}{T} \nabla T \right) + \frac{A \rho_n \rho_s |\mathbf{v}_n - \mathbf{v}_s|^4}{T}, \tag{8}$$

where the last term on the right-hand side expresses the energy dissipation based on the mutual friction between the two components. This term may behave as a source term in case of motion of the superfluid and normal particles [18]. To obtain a temperature equation, Eq. (8) can be recast into the following form, assuming that s and ρ are not pressure-dependent,

$$\left(\rho \frac{\partial s}{\partial T} + s \frac{\partial \rho}{\partial T} \right) \left(\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{v}_n T) \right)$$
$$= \nabla \cdot \left(\frac{k_n}{T} \nabla T \right) + \frac{A \rho_n \rho_s |\mathbf{v}_n - \mathbf{v}_s|^4}{T} - \left(\rho s - T \frac{\partial \rho s}{\partial T} \right) \nabla \cdot \mathbf{v}_n. \tag{9}$$

We note that, besides the mutual friction term, Eqs. (2)–(9) are strongly coupled by the presence of a temperature gradient in the momentum equations. Additionally, the physical variables s, ρ_n, ρ_s, k_n and A depend on temperature, which also contribute to the coupling of the equations.

This model is valid beyond a critical velocity in the so-called superfluid turbulent regime. To simulate flow at smaller Reynolds number, one has just to set the Gorter–Mellink coefficient A equal to zero and hence obtain Landau's equations.

The superfluid equations must be completed with boundary conditions. They are defined as follows [15]. First, the normal component of the total mass flux must vanish at any surface at rest:

$$\mathbf{n} \cdot (\rho_n \mathbf{v}_n + \rho_s \mathbf{v}_s) = 0. \tag{10}$$

Moreover, the tangential component of the normal velocity $\mathbf{v}_n$ must be zero at a solid surface,

$$\mathbf{v}_n - \mathbf{n}\mathbf{n} \cdot \mathbf{v}_n = 0. \tag{11}$$

In the absence of heat transfer between the solid surface and the fluid, we write

$$\mathbf{n} \cdot (\rho s T \mathbf{v}_n - k_n \nabla T) = 0. \tag{12}$$

Neglecting the diffusive term $k_n \nabla T$, the component of $\mathbf{v}_n$ perpendicular to the surface is also zero. Hence, for this case, we obtain a no-slip boundary condition for the normal velocity ($\mathbf{v}_n = \mathbf{0}$) and a slip boundary condition for $\mathbf{v}_s$.

At the heater, the energy condition boundary becomes,

$$\mathbf{n} \cdot (\rho s T \mathbf{v}_n - k_n \nabla T) = \mathbf{n}.\mathbf{q}. \tag{13}$$

At the helium bath entrance, temperature and pressure are fixed values. Actually, the situation regarding the temperature boundary condition at the bath entrance may be more complicated since superfluid equations cause a heat wave phenomenon, the so-called second sound, that has to be carried away from the computational domain. This point will be discussed in paragraph Section 4.1.2.

3. Numerical approach to solve the problem

All numerical developments of *HellFOAM* are performed with OpenFOAM® version 2.3. This code is a C++ library that solves partial differential equations with the method of finite volumes [19,20]. It handles 3D geometry by default. One of its features is to solve equations using segregated approaches. Since the superfluid problem presents a strong coupling between all its unknown variables ($\mathbf{v}_n, \mathbf{v}_s, p, T$ and Γ), it is important to develop a suitable algorithm. We introduce in this section the Super-PISO algorithm, a PISO algorithm [13] adapted to superfluid equations. The question of boundary conditions is treated in a separate paragraph.

3.1. Discretization of the superfluid helium equations

The set of Eqs. (2)–(9) is transformed into a set of algebraic equations after application of any standard discretization procedure. Using the Euler implicit difference scheme and if k and $k + 1$ denote successive time levels, then the semi-discretized form of the normal momentum equation reads,

$$\mathcal{V} \left(\frac{\rho_n^{k+1}\mathbf{v}_{n_P}^{k+1} - \rho_n^k\mathbf{v}_{n_P}^k}{\delta t} \right) = -a'_{n_P}\mathbf{v}_{n_P}^{k+1} + \sum_{NP} a'_{n_{NP}}\mathbf{v}_{n_{NP}}^{k+1} - \rho_s s \nabla T$$
$$- \frac{\rho_n}{\rho}\nabla p + K_{ns}\mathbf{v}_{s_P}^k. \tag{14}$$

In this equation, $\mathcal{V}$ and δt stand for the cell volume and the time step respectively, subscript P denotes values at the cell center, and

$a'_{n_{NP}}$ accounts for the influence of neighbor nodes. They are mainly composed by convective and diffusive fluxes across cell faces. They also involve the normal part of the momentum exchange terms characterized by $K_{ns} = \frac{\Gamma}{2} + A\rho_n\rho_s|\mathbf{v}_n - \mathbf{v}_s|^2$. In the spirit of the Rhie and Chow procedure [21], the pressure gradient and the temperature gradient terms are not discretized at this stage.

Introducing the vector $\mathbf{S} = \frac{\mathcal{V}\rho_n^{k+1}}{\delta t} - \rho_s s \nabla T + K_{ns}\mathbf{v}_{s_P}^k$ that contains all the explicit source terms (the source part of the transient term, the thermo-mechanical term $\rho_s s \nabla T$ and the superfluid part of the momentum exchange terms) apart from the pressure gradient, Eq. (14) can be rearranged as

$$\left(\frac{\mathcal{V}\rho_n^{k+1}}{\delta t} + a'_{n_P} \right) \mathbf{v}_{n_P}^{k+1} = \sum_{NP} a'_{n_{NP}}\mathbf{v}_{n_{NP}}^{k+1} + \mathbf{S} - \frac{\rho_n}{\rho}\nabla p. \tag{15}$$

We can now introduce the diagonal term $a_{n_P} = \frac{\mathcal{V}\rho_n^{k+1}}{\delta t} + a'_{n_P}$ of the full matrix resulting from the momentum equation discretization, and the operator $\mathbf{H}_n(\mathbf{X}) = \sum_{NP} a'_{n_{NP}}\mathbf{X}_{NP} + \mathbf{S}$. Therefore, Eq. (15) turns to,

$$a_{n_P}\mathbf{v}_{n_P}^{k+1} = \mathbf{H}_n(\mathbf{v}_n^{k+1}) - \frac{\rho_n}{\rho}\nabla p. \tag{16}$$

Following the same process, the semi-discretized superfluid momentum equation can be expressed as

$$a_{s_P}\mathbf{v}_{s_P}^{k+1} = \mathbf{H}_s(\mathbf{v}_s^{k+1}) - \frac{\rho_s}{\rho}\nabla p, \tag{17}$$

and the energy balance equation as

$$a_{T_P} T_P^{k+1} = \mathbf{H}_T(T_P^{k+1}). \tag{18}$$

Since the diffusive terms in Eqs. (6) and (9) are very small or non-existent, we discretized the convection terms in these equations with a first order upwind scheme. All the Laplacians are discretized with a second order scheme.

3.2. The Super-PISO algorithm

We present in this sub-section the algorithm we propose to solve the superfluid helium equations. It is based on the Pressure Implicit Operator-Splitting algorithm designed by Issa [13]. The algorithm is extended to the two-fluid model and a pressure equation is derived from the total mass conservation and both momentum equations. We have called this algorithm Super-PISO. The main steps are described below and sketched in Fig. 1.

1. At a time step, we start by updating all physical properties from the temperature field at the latest time step and get $\rho_n^{k+1}, \rho_s^{k+1}, \rho^{k+1}, s^{k+1}, k_n^{k+1}, \mu_n^{k+1}$ and A^{k+1}.
2. We continue by predicting Γ^{k+1} from the normal and superfluid velocities evaluated at the previous time step. From Eqs. (3) and (4) we have:

$$\Gamma^{k+1} = \frac{1}{2}\left(\mathcal{V}\left(\frac{\rho_n^{k+1} - \rho_n^k}{\delta t} \right) + \nabla \cdot (\rho_n\mathbf{v}_n)^k \right.$$
$$\left. - \mathcal{V}\left(\frac{\rho_s^{k+1} - \rho_s^k}{\delta t} \right) - \nabla \cdot (\rho_s\mathbf{v}_s)^k \right). \tag{19}$$

3. Then we evaluate the temperature field solving implicitly Eq. (18). It must be noticed that, at this stage, the temperature is transported by $\mathbf{v}_n$ of the previous time step.
4. We now solve the pressure-velocities ($\mathbf{v}_n, \mathbf{v}_s, p$) coupling by adapting the PISO algorithm [13]. We give here the main steps of the procedure. The first stage consists in the prediction of the normal and superfluid velocities, $\mathbf{v}_n^*$ and $\mathbf{v}_s^*$, by solving implicitly
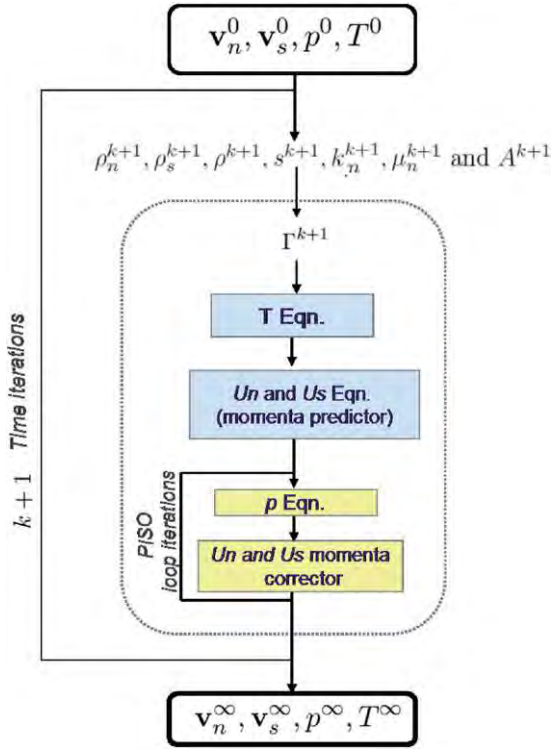
$$\mathbf{v}_n^0, \mathbf{v}_s^0, p^0, T^0$$

$$\rho_n^{k+1}, \rho_s^{k+1}, \rho^{k+1}, s^{k+1}, k_n^{k+1}, \mu_n^{k+1} \text{ and } A^{k+1}$$

$$\Gamma^{k+1}$$

T Eqn.

Un and Us Eqn.
(momenta predictor)

p Eqn.

Un and Us momenta
corrector

$$\mathbf{v}_n^\infty, \mathbf{v}_s^\infty, p^\infty, T^\infty$$

Fig. 1. Principle of the Super-PISO algorithm.

the momentum equations

$$a_{n_P} \mathbf{v}_{n_P}^* = \mathbf{H}_n\left(\mathbf{v}_n^*\right) - \frac{\rho_n^{k+1}}{\rho^{k+1}} \nabla p^k, \qquad (20)$$

$$a_{s_P} \mathbf{v}_{s_P}^* = \mathbf{H}_s\left(\mathbf{v}_s^*\right) - \frac{\rho_s^{k+1}}{\rho^{k+1}} \nabla p^k, \qquad (21)$$

where the gradient of the pressure field is evaluated from the values computed at the previous time step. This stage is called the momenta predictor. Actually, we have noticed that due to the small time-step requirement, it is not necessary to solve these equations. We only need to update a_{i_P} and $\mathbf{H}_i(\mathbf{v}_i)$ with the latest computed velocities.

5. The predicted velocities $\mathbf{v}_n^*$ and $\mathbf{v}_s^*$ do not satisfy the continuity equation (2) and have to be corrected. This is achieved by looking for $\left(\mathbf{v}_n^{**}, \mathbf{v}_s^{**}, p^*\right)$ that obeys:

$$\mathbf{v}_{n_P}^{**} = \frac{1}{a_{n_P}}\left[\mathbf{H}_n\left(\mathbf{v}_n^*\right) - \frac{\rho_n^{k+1}}{\rho^{k+1}} \nabla p^*\right], \qquad (22)$$

$$\mathbf{v}_{s_P}^{**} = \frac{1}{a_{s_P}}\left[\mathbf{H}_s\left(\mathbf{v}_s^*\right) - \frac{\rho_s^{k+1}}{\rho^{k+1}} \nabla p^*\right], \qquad (23)$$

$$v\left(\frac{\rho^{k+1} - \rho^k}{\delta t}\right) + \nabla \cdot \left(\rho_n^{k+1}\mathbf{v}_n^{**} + \rho_s^{k+1}\mathbf{v}_s^{**}\right) = 0. \qquad (24)$$

Assembling these three equations, the pressure equation can be formulated as

$$\nabla \cdot \left(\left(\frac{\rho_n^{k+1}}{\rho^{k+1}}\frac{\rho_n^{k+1}}{a_{n_P}} + \frac{\rho_s^{k+1}}{\rho^{k+1}}\frac{\rho_s^{k+1}}{a_{s_P}}\right) \nabla p^*\right)$$
$$= \nabla \cdot \left(\frac{\rho_n^{k+1}\mathbf{H}_n(\mathbf{v}_n^*)}{a_{n_P}} + \frac{\rho_s^{k+1}\mathbf{H}_s(\mathbf{v}_s^*)}{a_{s_P}}\right) + v\left(\frac{\rho^{k+1} - \rho^k}{\delta t}\right)$$
$$\qquad (25)$$

and solved implicitly. The corrected velocities $\mathbf{v}_n^{**}$ and $\mathbf{v}_s^{**}$ are then computed point-wise from Eqs. (22) and (23). This fifth step may be repeated several times to force the convergence. The resulting values are assimilated to $(\mathbf{v}_n^{k+1}, \mathbf{v}_s^{k+1}, p^{k+1})$. This iteration process is called the Super-PISO loop.

As all the solution algorithms that belong to the PISO family, Super-PISO is not unconditionally stable. This is typical of the predictor–corrector feature of the method since as the solution to the equations is achieved approximately, the residual errors in these approximations may alter the stability characteristics of the overall scheme. To handle the instabilities, the time-step size required some restrictions. A rigorous analysis of the stability is actually a tough task and beyond the scope of this paper, especially with this strongly coupled physics that involves a lot of non-linearities. From a practical point of view, we have noticed that our simulations need very small time steps. Moreover we also noticed that a fixed Δt was much more stable than an adjustable time-step. The time step management will deserve further investigations.

3.3. How do we treat the boundary conditions ?

We discuss here the treatment of boundary conditions at walls, at the heated section and at the vicinity of the bath.

3.3.1. Walls

In the absence of heat transfer at the solid walls, boundary conditions are defined as follows in OpenFOAM®:

- $\mathbf{v}_n$: *fixedValue* with value equal to *uniform(0 0 0)* (no-slip boundary conditions)
- $\mathbf{v}_s$: *slip*
- p: *zeroGradient*
- T: *zeroGradient*.

3.3.2. Heated section

Regarding the boundary conditions at the heated area, we overcome the difficulty of imposing Eq. (10) to Eq. (13) by considering the heat source $\mathbf{n}.\mathbf{q} = q$ as a volume heat source in the energy equation that is distributed in some cells. Hence, the actual energy equation we have coded in OpenFOAM® is

$$\left(\rho\frac{\partial s}{\partial T} + s\frac{\partial \rho}{\partial T}\right)\left(\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{v}_n T)\right) = (\cdots) + \frac{qS_{heater}}{V_\alpha}\alpha \qquad (26)$$

where α is a phase indicator equal to 1 for the heated cells and 0 elsewhere. Hence, we can define a heated area within the computational domain. V_α is the volume of the heated area and S_{heater} is the surface of the heated boundary. This "trick" offers some advantages. First, Eq. (13) involves a non-linear term, $\rho sT\mathbf{v}_n$, which needs to be solved in the framework of a sequential algorithm. With the proposed method, this non-linear boundary condition is replaced by a volume source term and conditions of zero mass and heat fluxes at the boundary. First, it is no longer necessary to deal with boundary conditions for velocity fields at the heater since they are automatically set up at the α jump. Second, this method allows the heat flux to be relaxed over several cells, which improves the stability of the calculation. In particular, the creation of normal fluid component is now spread over several cells instead on a *singularity* at the heated surface. Moreover, this "trick" of a heat source distributed in a volume also allows more flexibility to set up He II simulations. For example, to simulate a bath of helium with a heater in the middle of the computational domain, it is not necessary to explicitly mesh the heater and specify boundary conditions. Instead, only a cartesian grid and a mask function that indicates the heated cells are needed.
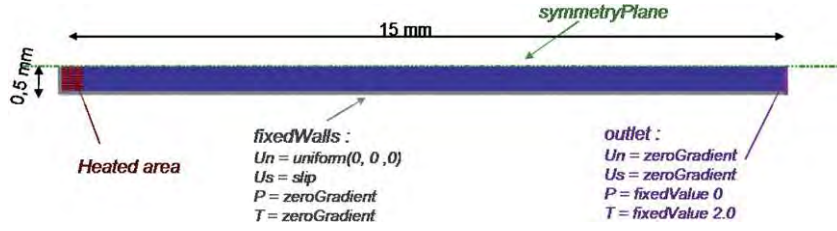
Fig. 2. Geometry and mesh of the capillary tube.

3.3.3. Helium bath

At the bath entrance, the boundary conditions are defined as follows in the OpenFOAM® framework

- $\mathbf{v}_n$: *directionMixed* to precise that zero tangential velocity and zero gradient in normal direction.
- $\mathbf{v}_s$: *zeroGradient*.
- p: *fixedValue* with value equal to *uniform 0*.
- T: *fixedValue* with value equal to bath temperature.

4. Some simulation results

In this section, we present some simulations we have performed with *HellFOAM*. The objective is two-fold: (i) a comparison of the numerical results with analytical ones when available, (ii) simulation examples for cases of current research interest. First, steady-state results for a capillary containing He II in both Landau and Gorter–Mellink flow regimes are compared to analytical solutions. Then transient simulation results of forced flow of He II are compared to the experimental data by Fuzier and Van Sciver [22]. The goal of these results is to confront *HellFOAM* code by comparing its results against known data, quantitatively or semi-quantitatively. Finally, we present some 2D simulations of He II in a porous medium model, as an example of a rather complex situation.

4.1. Capillary containing helium II in Landau's regime

This test consists in the simulation of a capillary (2h × L = 1 mm × 15 mm) tube filled with He II, heated at the left-hand side and connected to a He II bath at the right-hand side (see Fig. 2). For the sake of computing economy, we only simulate the flow on half the geometry using a symmetry plane boundary condition. The grid is made of 200 × 20 hexahedral cells. Bath is assumed to be at 2 K and so does the capillary tube initially. At this temperature, we have the following fluid properties: $\rho_n = 82.12$ kg/m^3, $\rho_s = 63.53$ kg/m^3, $\mu_n = 1.489 \times 10^{-6}$ kg/m/s, $k_n = 0.00387$ W/m/K and $s = 957.7$ J/kg/K.

Initially, the superfluid and normal components are at rest ($\mathbf{v}_n = \mathbf{0}$ and $\mathbf{v}_s = \mathbf{0}$), the bath temperature at 2 K, and the pressure field is uniformly zero. The left-hand side is heated to $q = 1000$ W/m^2. Time-step is set up to 10^{-6} s and the simulation is carried out until steady-state is reached. In order to use *HellFOAM* code in Landau's regime, A is set to zero.

4.1.1. Steady-state

To avoid or limit the propagation of the heat waves intrinsic to superfluid equations, the so-called second sound [15], the left-hand side is gradually heated using a heat-up ramp, $q\left(1 - e^{-t/\tau}\right)$, where $\tau = 10^{-2}$ s. We plot in Fig. 3 the simulation results we obtain for the normal and superfluid velocities at steady-state. Normal velocity $\mathbf{v}_n$ has a parabolic profile while the superfluid component flows in a homogeneous manner at counter-current to the normal component.

At steady state, this test-case has analytical solutions [15]. Indeed, in the established regime, both velocities are aligned
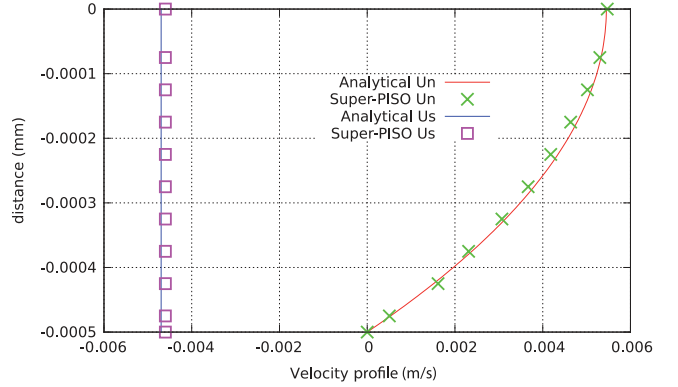

Fig. 3. Plot of the velocity profiles over a vertical cross-section at the middle of the tube in Landau's flow regime. Both components are flowing at counter-current. The normal velocity profile has a parabolic profile and the superfluid component is homogeneous. The results fit very well with the analytical solution.

along the x-axis. From the heat flux equation, the normal velocity averaged over a cross-section is estimated by $\overline{v_n} = \frac{q}{\rho s T} \approx 0.00358$ m/s. Moreover, noticing that the sum of the two momentum equations, Eqs. (5)–(6), leads to a Stokes equation for the couple $\left(p, v_{n_x}\right)$, one can deduce that the normal velocity has a parabolic profile, $v_{n_x}(y) = v_{n_{max}}\left(1 - \left(\frac{y}{h}\right)^2\right)$, where $v_{n_{max}} = \frac{3}{2}\overline{v_n} \approx 0.00537$ m/s. On the other hand, since there is no viscosity in the superfluid momentum equation, Eq. (6), then the superfluid velocity is constant over a cross-section. It can be predicted using the zero mass transfer relation $\overline{v_s} = -\frac{\rho_n}{\rho_s}\overline{v_n} \approx -0.0046$ m/s. We see in Fig. 3 that simulation results and analytical solutions are in very good agreement. The temperature elevation between the ends of the capillary tube can be evaluated using London's equation $\Delta T = \frac{1}{\rho s}\Delta p$ and the pressure loss / velocity relation in a 2D capillary $\overline{v_n} = \frac{h^2}{12\mu_n}\frac{\Delta p}{L}$. It follows that $\Delta T \approx 6.9 \times 10^{-9}$ K which is in good agreement with the temperature profile in the traverse direction obtained by simulation (see Fig. 4). It must be noted that for such a small temperature increase, the variations of the fluid properties ρ_n, ρ_s, μ_n, k_n and s are negligible in agreement with our analytical comparison.

4.1.2. Transient

We now turn our attention to the transient regime of the superfluid He II problem. One important characteristic of the superfluid equations is the existence of a heat wave called second sound: when a capillary is heated at one side, a temperature front travels towards the other side. Landau and Lifshitz [15] have estimated its propagation velocity at $u = \sqrt{\frac{Ts^2\rho_s}{c\rho_n}}$. Using the approximation that relates the compressibility and the entropy $c \approx 3s$ at very low temperature, we get $u \approx 22.2$ m/s. It means that the heat front crosses the capillary in approximately 7×10^{-4} s. To point out this phenomenon, we start again the simulation without the heat-ramp. As depicted in Fig. 5, a heat front travels along the capillary and reaches the bath at about 9×10^{-4} s in accordance with Landau's
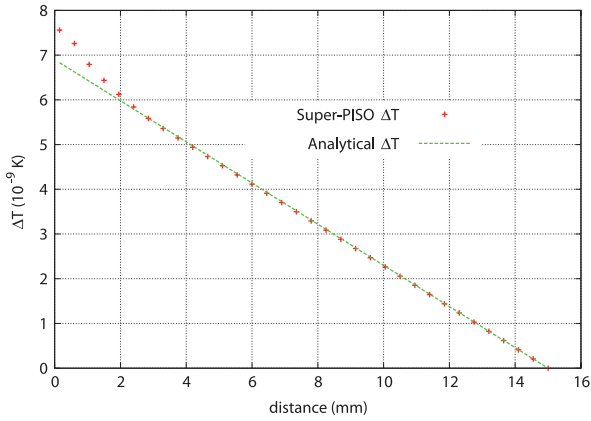
Fig. 4. Plot of the temperature profile along the capillary at steady-state. We notice a temperature increase of the order of magnitude of 7×10^{-9} K as expected by the theory.
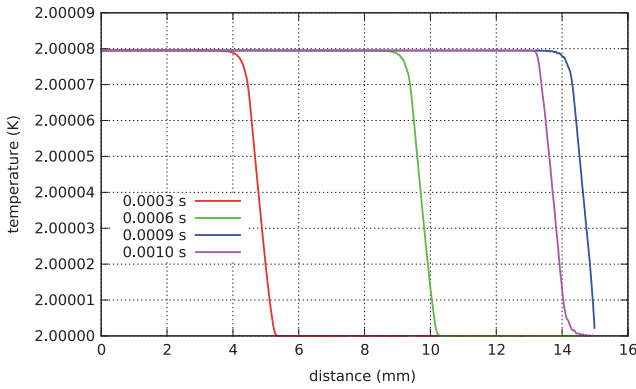

Fig. 5. Plot of heat front propagation along the capillary tube filled with He II in Landau's regime.

Fig. 6. Plot of the temperature profile along the capillary at steady-state. We notice a temperature increase up to 0.0009 K as expected by the theory.

Fig. 7. Plot of the velocity profiles over a vertical cross-section in the Gorter–Mellink flow regime. The normal velocity profile has still a parabolic shape while the superfluid component is no longer homogeneous. Indeed, the theory predicts that $v_{s_x}(y) = v_{n_x}(y) - \sqrt[3]{\frac{s}{\rho_n A} \frac{\Delta T}{L}}$. Numerical results are in very good agreements with the analytical formula.

predictions. However, we can notice that the heat front is then re-flected towards the heater (see 10×10^{-4} s for instance). While a re-flexion at any place where geometry changes is not unphysical, its amplitude here is probably unrealistic if we consider the presence of a bath starting at this boundary. And it is also very probable that the boundary conditions at the bath side, namely a Dirichlet condition for the temperature, cannot represent the dampening effect of a real bath. The use of a wave-transmissive boundary condition could overcome this problem but it has not been developed for our superfluid case. We also notice in Fig. 5 that the heat wave amplitude is much higher than the steady-state value predicted by the theory. Indeed, the dissipation mechanisms that lead to the steady-state value are quite slow (since $\frac{k_n}{\rho s} \approx 4.9 \times 10^{-8}$ m^2/s).

4.2. Capillary containing He II in Gorter–Mellink regime

We now turn our attention to the investigation of He II flow dynamics in the Gorter–Mellink regime. We keep the same geometry and fluid properties as those used in Section 4.1. At 2 K, the Gorter–Mellink coefficient is equal to $A = 1280$ m.s/kg. As for the other fluid properties, even if they are estimated as a function of the temperature in *HellFOAM*, it is reasonable to consider their value constant in the following analytical formula for a temperature increase below 1 mK. Initially, both components are at rest at the temperature of the He II bath and the relative pressure field is uniformly zero. We heat the left hand side to 10^4 W/m^2. Time step is set up to 5×10^{-6} s and simulation is stopped when steady-state is reached.

In the Gorter–Mellink regime, the source term in the energy equation, Eq. (9), behaves as a strong dissipative process. As
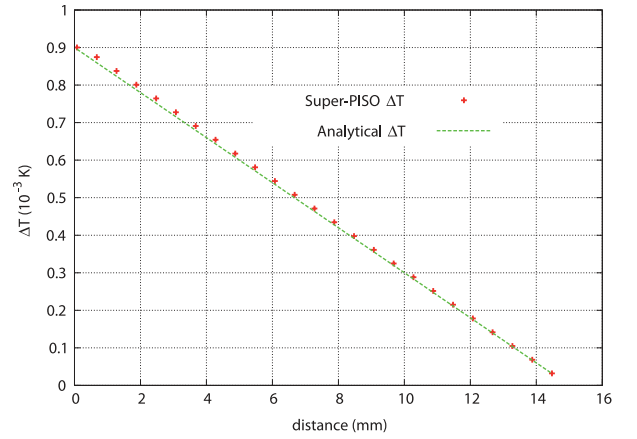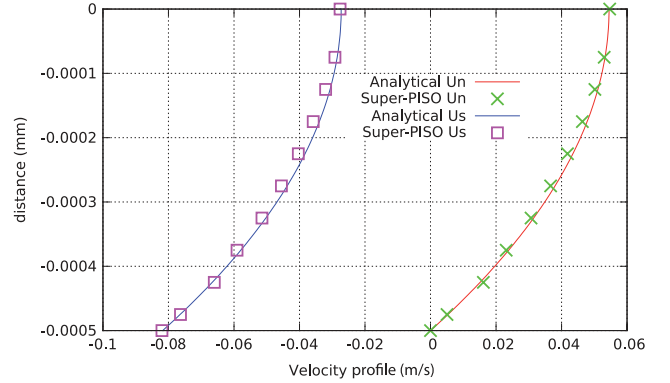
expected, the temperature field reaches its steady-state quasi-immediately after the heat-up time, and there are no longer heat wave propagations. The Gorter–Mellink dissipation terms have absorbed these waves. The temperature gradient we found (see Fig. 6) is in agreement with the value analytically obtained by the formula $\Delta T \approx \frac{A \rho_n L}{\rho_s^3 s^4 T^3} q^3 \approx 0.0009$ K [23].

Regarding the velocity fields, as plotted in Fig. 7, the normal velocity field remains a parabolic profile while the superfluid velocity field is no longer homogeneous over the cross-section and some higher values are noticed in the walls vicinity. Profiles may be explained merely from the two-fluid model. Indeed, adding the two momentum equations, Eqs. (5)–(6), and assuming that the velocities are aligned with the x-axis, one gets a Stokes equation for the normal velocity. Consequently, the normal velocity has a parabolic profile as in Landau's regime. Regarding the superfluid momentum, it is now well-known that the thermomechanical effect and mutual friction terms are the controlling terms. Neglecting the other ones, we can derive the relationship, $v_{s_x}(y) = v_{n_x}(y) - \sqrt[3]{\frac{s}{\rho_n A} \frac{\Delta T}{L}}$, that relates both velocities. Numerical results and the analytical theory fit very well, as illustrated by the results presented in Fig. 7.

4.3. Fuzier and Van Sciver's experiments

In this section, we simulate transient heat transfer for a forced flow of He II at high Reynolds number inspired from the setup
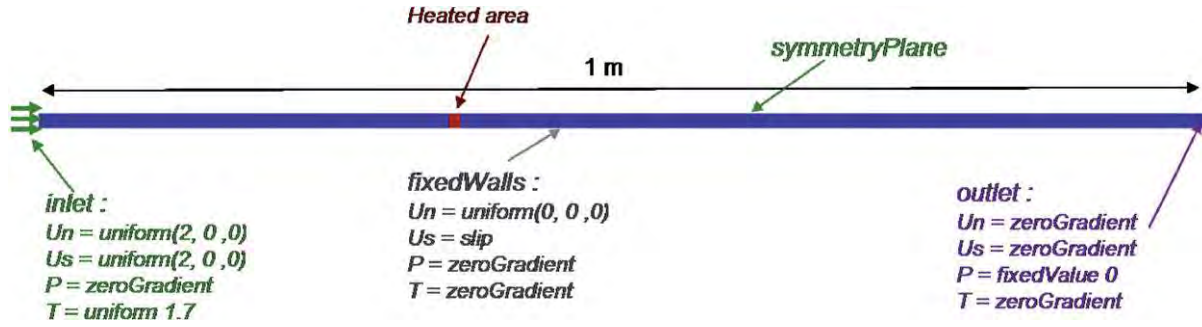
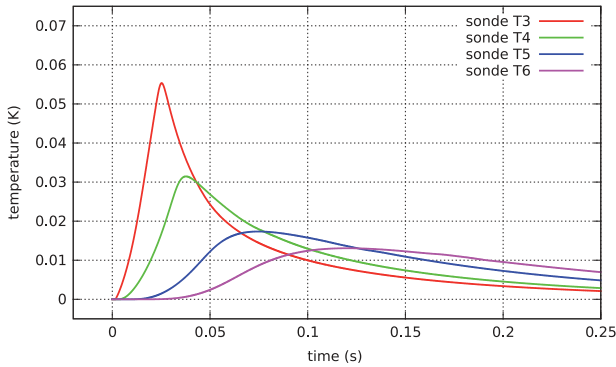**Fig. 8.** Geometry and boundary conditions of the domain.

Fig. 9. Transient temperature profile recorded by several probes placed along the tube in case of a forced convection flowing at 2 m/s.
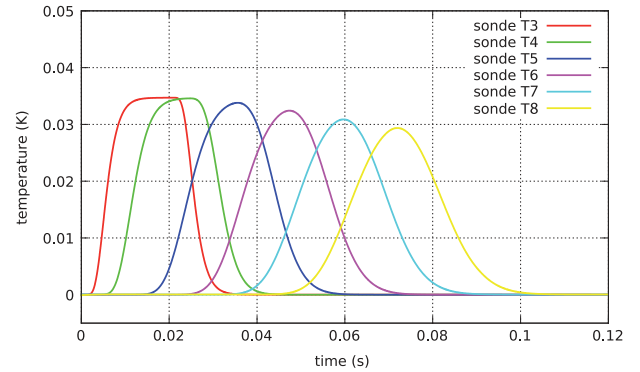

Fig. 10. Transient temperature profiles recorded by several probes placed along the tube in case of a forced convection flowing at 8 m/s.

Fig. 11. Transient temperature profile recorded by several probes placed along the tube in case of a forced convection flowing at 16 m/s.

of the experiments performed by Fuzier and Van Sciver [22]. In their work, authors measured temperature profiles in a forced flow of superfluid helium in a 1 m long, 9.8 mm inside diameter, smooth tube. The liquid is pushed from a bellows pump to reach a velocity up to 22 m/s. Fuzier and Van Sciver's experiments cannot be entirely reproduced by our numerical model mainly because these experiments were performed at relatively high normal fluid Reynolds numbers, hence with potentially turbulence effects. The two-fluid model developed in this paper is able to simulate inertia effects but not a fully developed turbulence, such as the kind of flow that develop in a straight tube, once hydrodynamic instabilities have started. Therefore, the comparison here will be more qualitative, remembering that the simulated flow remains laminar. In addition, only a 2D geometry will be considered. The computational domain is depicted in Fig. 8. It consists in a 1 m long, 4.9 mm thick, 2D half tube. The grid is made of 500×10 hexahedral cells. The heater is placed between 0.30 and 0.31 m from the left-hand side. Liquid helium enters into the domain from the left-hand side at 2 m/s and flows out at the right hand side of the tube. Initially, we assume that the superfluid and normal velocities are equal to 2 m/s within the tube and the temperature is 1.7 K. Several probes are placed along the tube. They correspond to the temperature measurements T3 to T8 of Fuzier and Van Sciver [22]. Simulations are performed with the full two-fluid model involving Gorter–Mellink mutual friction terms. The time step is set up to $\Delta t = 5 \times 10^{-6}$ s. After a preliminary simulation where we let the flow to be established without warming, we start the heater at $q = 9.9$ W/cm^2 for 20 ms and we record the temperature evolution until 250 ms for the different probes as depicted in Fig. 9. The results are in very good qualitative agreement with the experimental results shown in Fig. 5 of Fuzier and Van Sciver [22]: the temperature peak for each probe is comparable both in amplitude and in time. This comparison example is preliminary and semi-quantitative in the sense that some assumptions do not correspond exactly to the actual situation (for example: 2D geometry as opposed to a cylindrical geometry, laminar flow
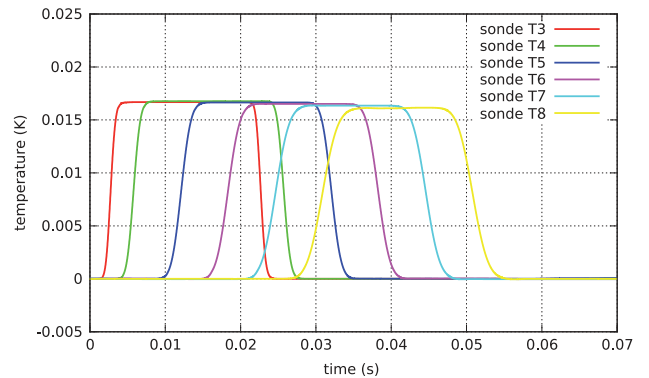
while some turbulence may have arisen in Fuzier and Van Sciver's experiments). It is intended to show that the expected physics is captured by the proposed numerical tool, i.e, correct evolution of physical variables. Further work is in progress towards a more comprehensive comparison. Additional discussions regarding this comparison can be found in another work by the same authors [18].

Then, following the same procedure, transient heat transfer simulation of He II flowing at 8 m/s is performed. Temperature records are plotted in Fig. 10. Once again, results are in good agreement with Fuzier's experiments (see Fig. 6 of [22]).

Finally, simulation results of forced flow of He II at 16 m/s are plotted in Fig. 11. Results are clearly comparable to Fuzier's experiments (see [22], Fig. 7).

These results lead to two conclusions : (i) they confirm the possibility of simulating laboratory experiments for relatively simple geometries with the use of a complete two-fluid model, (ii) they comfort the conclusions of the analytical validation tests for the *HellFOAM* code.
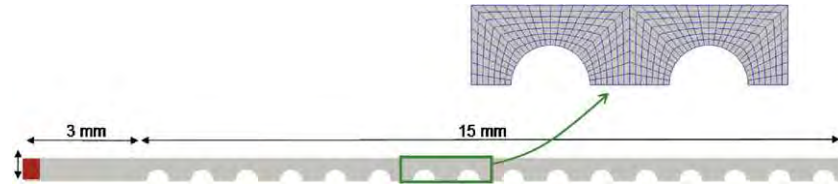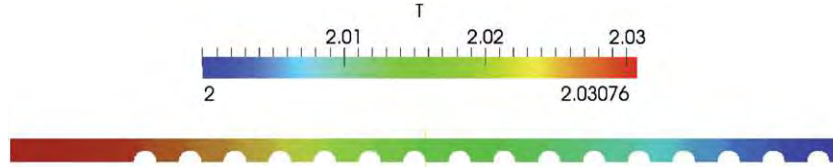
Fig. 12. Geometry and mesh of the domain.

Fig. 13. Plot of the temperature profile at steady state in a capillary tube filled by 16 beads. The presence of solid materials inside the capillary leads to an increase of the ΔT.
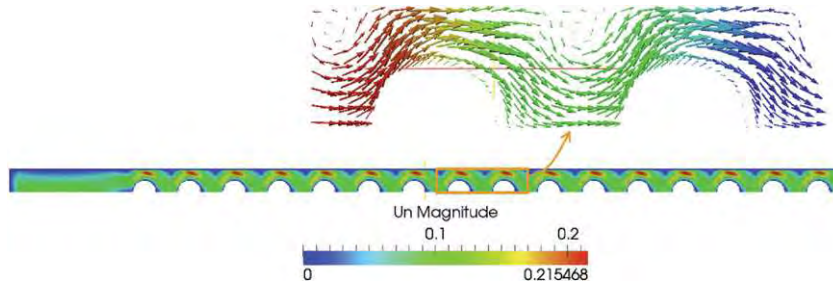

Fig. 14. Plot of the normal velocity magnitude and of the normal velocity vectors in two adjacent REVs. We clearly notice cyclic flow patterns.
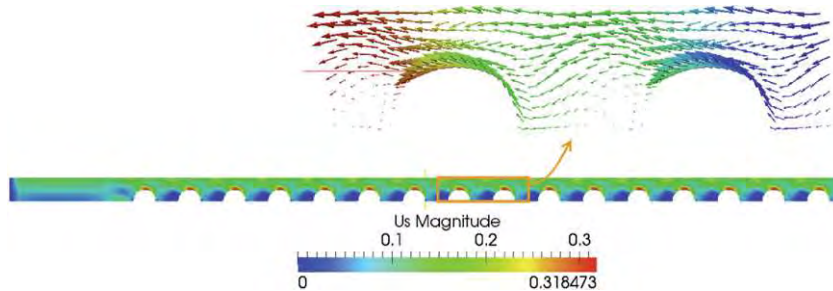

Fig. 15. Plot of the superfluid velocity magnitude and of the superfluid velocity vectors in two adjacent REVs. We clearly notice cyclic flow patterns.

4.4. Superfluid flow through an array of beads

A lot of devices involving He II superfluid can be seen as a porous medium involving two separate characteristic scales: a pore scale and a macro-scale. At the smaller scale, the flows and other transfer phenomena are simulated through the exact geometry of the material. Basically, to a cell of the mesh grid corresponds either the fluid or the solid structure. On the other hand, at the macro-scale, the flow can be described by volume-averaged equations. For coupled, non-linear equations, developing a macro-scale theory may benefit strongly of the quantitative view offered by direct pore-scale numerical results, which point in the right direction for performing necessary approximation. Such an illustration of this kind of approximation can be found in the paper by [24] in the case of Landau's regime. In this section, we show the capability of *HellFOAM* to simulate pore-scale flow of superfluid helium through a model porous medium.

The geometry (see Fig. 12) consists in a capillary (same dimensions as paragraph Sections 4.1 and 4.2) that contains 16 solid beads (*mm* diameter). While of course the geometry is a very sketchy representation of a porous medium, it has all the fundamental features, i.e., it contains several representative unit cells with some geometrical tortuosity. The heater is placed in a small extension of the capillary at the left-hand side of the array of beads. The capillary is initially filled with He II at 2 K. The left-hand side is then warmed up to a heat flux density of 10^4 W/m^2 and the flow is simulated using the Gorter–Mellink model. The computational domain is gridded with 3500 hexahedral cells. The time-step is set up to $\Delta t = 10^{-6}$ s.

Figs. 13–15 depict the temperature, the normal and the superfluid velocities at steady-state. We can notice in Fig. 13 that the temperature increase is much higher (30 times) than in the case of a straight capillary tube. This result is directly related to the presence of the solid structure within the tube and the related tortuosity effect. From Figs. 14 and 15, we clearly observe cyclic flow patterns. This suggests that one may define a Representative Elementary Volume (REV) of the pore-scale physics and then apply some upscaling technique like a volume-averaging methodology for instance in order to derive macro-scale equations. Moreover, the results indicate that further simulations could be restricted to a single REV with some kind of periodicity condition for velocities and temperature and pressure deviations. This can be used to significantly reduce the CPU time and, more fundamentally, this suggests also that some kind of decomposition technique in terms of

macro-scale variables and deviations can be performed in the context of upscaling. This has already been used in the paper by [24] in Landau's regime, our results indicate that this can be extended to more inertial flows with also significant Gorter–Mellink effects.

The inserts plotted in Figs. 14 and 15 represent the normal and superfluid velocity vectors, respectively, in two adjacent REVs. In Fig. 14 we can note the presence of an eddy in the region in between two beads and the top wall. In comparison, a Navier–Stokes simulation at high Reynolds number in the same geometry will give two recirculations (one in each symmetrical part) in the area in between two adjacent cylinders. Actually, this kind of flow patterns has been observed experimentally using cryogenic visualization technique for a superfluid flow past a cylinder (first by [25] and more recently by [26,27]). No clear explanations of the phenomenon have been proposed yet and research in this area is still a hot topic. The very positive point is that we are able to capture these eddies with our simulations, which means that all the necessary physics is included in the Gorter–Mellink model. The two major differences between the superfluid problem and the Navier–Stokes problem are: first, the superfluid velocity has a slip condition at the solid boundaries, second, the Gorter–Mellink term in the momentum equations will introduce a strong dissipation whenever the velocity difference is important. These differences are probably responsible of this unconventional flow pattern. This also illustrates the potential complexity of such flows in porous media and deserves further studies, in particular with the perspective to apply a full upscaling methodology.

5. Conclusions

We have presented in this paper the first version of *HellFOAM*, a helium superfluid simulator based on the OpenFOAM® technology. Given the sequential nature of OpenFOAM® algorithms, we had, in order to solve superfluid equations, to develop a new algorithm (called Super-PISO) extending the type of PISO algorithm used in OpenFOAM®.

Using the implemented code, different scenarii were simulated which correspond to situations of current research interest and with significant complexity. Solutions on straight tubes for which analytical (or partially analytical) solutions are known were used to check the validity of the numerical tool. Other simulations showed the interest of performing numerical simulations as a tool to interpret experiments, or in the framework of a multiscale analysis of superfluid flow through porous media. These various simulations have proven the robustness and efficiency of *HellFOAM*. They also showed that Gorter–Mellink two fluids model can well capture the physics in case of forced flow of helium II in a pipe and in case of superfluid flow past a cylinder.

References

[1] L. Landau, Phys. Rev. 60 (1941) 356.
[2] I.M. Khalatnikov, An Introduction to the Theory of Superfluidity, W. A. Benjamin, New York, 1965.
[3] C. Gorter, J. Mellink, Physica 15 (1949) 285.
[4] C. Gorter, P. Kasteleijn, J. Mellink, Physica 16 (1950) 113.
[5] T. Kitamura, K. Shiramizu, N. Fujimoto, Y. Rao, K. Fukuda, Cryogenics 37 (1997) 1.
[6] Y. Rao, Y. Inaba, T. Noda, K. Fukuda, Cryogenics 36 (1996) 219. International Symposium on Safety of Superconductors and Related Heat Transfer at Low Temperatures.
[7] P. Zhang, M. Murakami, R. Wang, Int. J. Heat Mass Transfer 49 (2006) 1384.
[8] N. Ramadan, R. Witt, Cryogenics 34 (1994) 563.
[9] T. Suekane, M. Sekiguchi, S. Hirai, T. Okamura, Cryogenics 43 (2003) 125.
[10] S. Pietrowicz, B. Baudouy, Cryogenics 53 (2013) 72. Workshop "CHATS on Applied Superconductivity 2011".
[11] H. Tatsumoto, K. Fukuda, M. Shiotsu, Cryogenics 42 (2002) 19.
[12] L. Bottura, C. Rosso, Internat. J. Numer. Methods Fluids 30 (1999) 1091.
[13] Issa, J. Comput. Phys. 62 (1985) 40.
[14] L.C. Woods, The Thermodynamics of Fluid Systems, Clarendon Press, Oxford, 1975.
[15] Landau, Lifshitz, Fluid mechanics, Course Theor. Phys. 6 (1969).
[16] M. Mongioví, Meccanica 29 (1994) 223.
[17] M.S. Mongioví, R.A. Peruzza, Z. Angew. Math. Phys. 54 (2003) 566.
[18] C. Soulaine, M. Quintard, H. Allain, B. Baudouy, R. Van Weelderen, Numerical investigation of heat transfer in a forced flow of He II, in Proceedings of the 15th International Heat Transfer Conference, IHTC-15 August 10-15, 2014, Kyoto, Japan, 2014.
[19] H. Jasak, Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows, (Ph.D. thesis), Department of Mechanical Engineering Imperial College of Science, Technology and Medicine, 1996.
[20] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, Comput. Phys. 12 (1998).
[21] C. Rhie, W. Chow, A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, in: Fluid Dynamics and Co-located Conferences, American Institute of Aeronautics and Astronautics, 1982.
[22] S. Fuzier, S. Van Sciver, Cryogenics 48 (2008) 130.
[23] V. Arp, Cryogenics 10 (1970) 96.
[24] H. Allain, M. Quintard, M. Prat, B. Baudouy, Int. J. Heat Mass Transfer 53 (2010) 4852–4864.
[25] T. Zhang, S. Van Sciver, Nat. Phys. 1 (2005) 36–38.
[26] T.V. Chagovets, S.W. Van Sciver, Phys. Fluids 25 (2013) 1.
[27] D. Duda, M. La Mantia, M. Rotter, L. Skrbek, J. Low Temp. Phys. 175 (2014) 331–338.