OpenFOAM® Journal

Volume 5 [Full Papers], Pages 90–114
ISSN: 2753-8168

OpenFOAM® 3D Solver for Viscous Fluids to Simulate Lava Flows

Elisa Biagioli[1,*], Mattia de' Michieli Vitturi[2], Fabio Di Benedetto[3], and Margherita Polacci[1]

[1]Department of Earth and Environmental Sciences, University of Manchester, Williamson Building, Oxford Road, M13 9PL Manchester, UK
Email address: elisa.biagioli@manchester.ac.uk

[2]Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Pisa, Via Cesare Battisti, 53 - 56125 Pisa, Italia

[3]Department of Mathematics, University of Genoa, Via Dodecaneso 35, 16146 Genova, Italy.

DOI: https://doi.org/10.51560/ofj.v5.128
Results with version(s): OpenFOAM® v1912
Repository: https://github.com/BiElisa/interThermalRadConvFoam

Abstract. In this work, we present a new OpenFOAM solver, called `interThermalRadConvFoam`, to simulate free-surface viscous fluids with temperature changes due to radiative, convective, and conductive heat exchanges. The solver is based on `interFoam` (available in OpenFOAM) and thus on the Volume of Fluid technique used to describe the multiphase dynamics of two incompressible, viscous, and immiscible fluids (based on the Interface Capturing strategy). In our model, the two fluids are the fluid of interest with high viscosity and the surrounding atmosphere. Being interested in temperature effects, we added to the mass and momentum equations from `interFoam` an equation for energy that models the thermal exchanges between the fluid and the environment. Furthermore, a temperature-dependent viscoplastic model is used for the final application to lava flows. Here we present some results of numerical tests performed with `interThermalRadConvFoam` for a benchmark from literature based on a laboratory experiment and an application to a real lava flow by simulating the Pico do Fogo 2014–2015 Eruption. For the simulation of the laboratory experiment, we also present simulations executed using a dynamic mesh with adaptive refinement.

1. Introduction

Volcanic eruptions are among the most threatening natural disasters on Earth and can affect people and infrastructures near volcanoes. In general, eruptions may be explosive, with a mixture of gas and pyroclastic material ejected at high speed into the atmosphere, or effusive, with lava flowing out more gently from the vent. Every eruption is an unstoppable event, but effusive phenomena are relatively slow in terms of propagation because the lava front generally advances at speeds not exceeding a few hundred meters per hour. So, most of the time, when an effusive event is underway, it is possible to respond promptly by simulating one or multiple scenarios for the current event and then preparing evacuation and safety plans. By using numerical simulations, scientists and civil protection can also prepare lava flow hazard maps before an eruption occurs to forecast future scenarios [1–15].

Reliable forecasts of lava flow propagation require a quantitative description of the physical processes governing the phenomenon based on the conservation laws for mass, momentum and energy. In addition to the physical laws, a "physics-based" model for lava emplacement must consider the effects of:

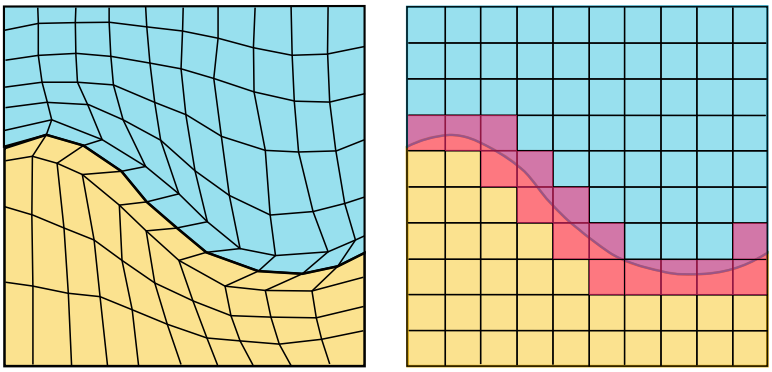
(*i*) topography or slope;
(*ii*) eruptive input conditions such as volumetric effusion rate, vent geometry (point or linear) and effusive temperature;
(*iii*) thermal exchanges at the top and bottom of the lava accounting for insulation, convection, radiation, and conduction;
(*iv*) physical properties of the lava, like density, thermal conductivity, and viscosity.

* Corresponding author
Received: 30 November 2023, Accepted: 20 January 2025, Published: 23 April 2025

In the past, several models for the simulation of lava flows have been developed from simple stochastic models [3, 5, 16–19] to more complex 2D [20–30] and 3D models [31–37]. If a description of the vertical distribution of variables within the flow is desired, 3D models should be adopted. Indeed, the 3D model approach is the only one capable of describing the whole vertical structure of the variables such as velocity, temperature, and viscosity. These variables, in particular, are coupled through the viscosity model, which plays a fundamental role in the dynamics. Therefore, a precise description of these variables might result in a more accurate viscosity model and, consequently, a better simulation. In most cases, the 3D description produces multiphase models because it requires solving for both the flow of interest and the overlying atmosphere. Due to the complexity of developing a multiphase 3D model from scratch, it is common to rely on existing codes, allowing for the implementation of customized numerical solvers. For these reasons, it was natural that the work we present here uses OpenFOAM as it offers many solvers for complex fluid dynamics problems. To our knowledge, only the following previous works adopted OpenFOAM to develop a lava flow model. Korotkii *et al.* [33] presented a steady-state model that uses an inverse problem solver to reconstruct thermal and flow characteristics by knowing the temperature and the heat flow on the lava surface obtained thanks to remote sensing measurements. Dietterich *et al.* [34] developed a model with temperature-dependent Newtonian rheology in which the cooling process occurs via interactions with the environment. Cataldo *et al.* [35] produced a steady-state, turbulent and incompressible model (by modifying simpleFoam) that describes the erosion process in channelled lava flows. Lee and Hwang [36] developed a model by modifying interFoam, like our approach, by adding a temperature equation with the diffusion only, and the viscosity is non-Newtonian and temperature independent. Finally, Park *et al.* [37] presented a transient model for multiphase flow (liquid-gas-particle mixture flow) that can be applied to the lava flows, although it does not consider temperature.

Our new 3D solver describes the dynamics of two incompressible, viscous, and immiscible fluids, representing the fluid of interest and the surrounding atmosphere, and it is based on the OpenFOAM solver `interFoam` [38]. The solver employs a segregated strategy and the PIMPLE algorithm [39] for the coupled solution of the governing equations, whose discretization is based on the Finite Volume Method. `interFoam`, and thus also our solver, uses the Volume of Fluid (VOF) technique [40] to deal with the multiphase dynamics; VOF is based on the Interface Capturing strategy, and hence solves for a transport equation for the volume fraction of one phase. To maintain a precise description of the interface between fluids, the solver employs the Multidimensional Universal Limiter for Explicit Solution (MULES) method [41] that implements the Flux-Corrected Transport (FCT) technique [42], proposing a mix of high and low order schemes.

The new solver we present here is called `interThermalRadConvFoam` because, with respect to `interFoam`, we added an equation for energy that models radiative and convective heat exchanges, and we implemented a new boundary condition for the heat conduction with the soil. A non-Newtonian temperature-dependent viscosity model is included in the approach and couples momentum and energy equations. Our work aims to improve the other existing models previously reported by collecting both a complete description of the thermal effects and an accurate viscosity model. The model can use digital elevation



(a) *Interface-tracking method.* **(b)** *Interface-capturing method.*

Figure 1. *Computational grid and phase interface between two fluids. Yellow* and *blue* represent the two fluids. (a) The thicker black line between the two fluids represents the interface, and the computational grid is adapted to the interface shape. (b) The cells highlighted in *pink* are those that determine the interface, and the grid is independent of that.

model (DEM) files to simulate flows over real topographies. To evaluate the performances of our model, we compared our simulations with a benchmark from [43]: the spreading and cooling of a viscous fluid on a flat bottom. For this test case, dynamic mesh refinement has also been used to dynamically refine the interface between fluids and obtain higher accuracy where necessary. In addition, we present an application of the solver to a real effusive eruption: the early hours of the 2014-2015 eruption of the Fogo volcano at Cape Verde, West Africa.

The paper is organized as follows: section 2 presents the physical model; section 3 describes the numerical approach implemented in our code; section 4 is devoted to simulations and section 5 focuses on conclusions and possible future developments.

2. 3D multiphase model

In this section, we present the equations that describe the 3D model. Our specific interest is in liquids with a free surface, and we choose a way to model them that alos takes into account the air above. This means that we deal with a *multiphase* model for the liquid-gas couple. Moreover, the two fluids are considered immiscible, incompressible, and with constant but different densities. Adopting the VOF method, we have a new variable that describes the phase volume fraction and allows us to distinguish where the two fluids are located and correctly associate their properties. Despite the multiphase (liquid-gas) approach to the model, our attention is mainly on the thermodynamic description of the liquid phase.

The main governing equations for this problem are the following:

$$\partial_t \alpha + \nabla \cdot (\mathbf{u}\alpha) = 0, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot \left(\rho \mathbf{u}\mathbf{u}^T\right) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} + \mathbf{f}_\Sigma, \tag{3}$$

$$\frac{\partial(\rho c_p T)}{\partial t} + \nabla \cdot (\rho c_p T \mathbf{u}) - \bar{\chi}_\Sigma \Delta(kT) = -\frac{\epsilon \sigma_{\mathrm{SB}} f A_{\mathrm{fs}}}{V}\left(T^4 - T_{\mathrm{env}}^4\right) - \frac{\lambda f A_{\mathrm{fs}}}{V}\left(T - T_{\mathrm{env}}\right). \tag{4}$$

A brief explanation of all governing equations is given here.

In describing the motion of two *immiscible* fluids, it is of fundamental importance a good treatment of the free surface that separates the two phases, which is often called *phase interface*, or simply *interface*, and its evolution is computed as a part of the solution. In the interface-capturing methods, the interface is defined by those cells filled by both phases as shown in Fig. 1b, and the computational grid is fixed or possibly refined. This approach differs, for example, from the one adopted by *interface-tracking methods* [44–47] that chase the phase interface as it moves. In the interface-tracking cases, indeed, the interface is geometrically defined by the computational grid that evolves in time and adapts, it is a boundary between subdomains of a moving mesh, as represented in Fig. 1a, so that each cell contains only one phase.

α is the new variable related to the volume fraction and Eqn. (1) (the "α-equation" that allows the calculation of the evolution of α distribution) is the associated transport equation, justified by the mass conservation principle. When a cell is full with one of the two fluids, then α is equal to one or zero. The interface is defined by those cells that see both phases inside, and a value $0 < \alpha < 1$ is hence assigned, as represented in Fig. 2. The VOF method treats the multiphase flow as a single fluid whose properties (density ρ and viscosity μ) vary in space according to the volumetric fraction of the two phases (which are denoted respectively with the subscripts l and g, that usually refer to liquid and gaseous phases)

$$\rho = \alpha \rho_l + (1 - \alpha)\rho_g, \tag{5}$$

$$\mu = \alpha \mu_l + (1 - \alpha)\mu_g, \tag{6}$$

and whose velocity, temperature and pressure fields are shared by the two fluids [38]. Hence, the conservation equations are solved for such single and "artificial" fluid, whose velocity and temperature are denoted with $\mathbf{u}$ and T, respectively. Other possible variants of VOF can be found in literature [40, 48]. Some alternatives to VOF, belonging as well to the group of interface-capturing methods, can be found in literature [49–51].

Equation (2) represents the continuity equation that reduces to a kinematic constraint under the assumption of constant density. Equation (3) models the linear momentum balance, where p is the pressure, $\mathbf{g}$ is the gravity acceleration, $\boldsymbol{\tau}$ is the viscous stress tensor defined as

$$\boldsymbol{\tau} = \mu \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}\right], \tag{7}$$

where $..^{\mathrm{T}}$ represents the matrix transpose. The last term of Eqn. (3), $\mathbf{f}_\Sigma$, describes the effect of surface tension as a continuum surface force [52] which is defined as follows:

$$\mathbf{f}_\Sigma \coloneqq \sigma_\Sigma \kappa_\Sigma \nabla\alpha, \quad \kappa_\Sigma = -\nabla \cdot \frac{\nabla\alpha}{|\nabla\alpha|}, \tag{8}$$

where σ_Σ is the surface tension constant and κ_Σ is the surface curvature.

Finally, Eqn. (4) is the energy conservation equation that is written in terms of the temperature T. We consider only the balance of the thermal energy, namely $E = \rho c_p T$, where c_p is the specific heat of the fluid, computed as the mass average of the specific heat of the two phases. We assume that (i) the pressure variations produce negligible effects on the thermodynamic variables and that (ii) the effects of the energy dissipation caused by viscosity are small enough to be neglected [20, 24, 26, 27, 32, 53, 54]. Because of these two hypotheses, which are common to many contexts, including lava flows, the pressure and the viscous terms that are usually present in the energy conservation equation are not accounted for. The Laplacian term on the left-hand side represents the conductive flux term that models the heat diffusion, with k being the thermal conductivity. The role of the coefficient $\bar{\chi}_\Sigma$ is explained at the end of this section.

The model accounts for the heat exchange phenomena that are important for our applications. Convection and radiation heat losses with the environment take place at the free surface, and their implementation is based on the assumption that the liquid phase is hotter than the gaseous phase. These two phenomena are modelled by the two source terms on the right-hand side of Eqn. (4).

The first source term in Eqn. (4) represents the radiative heat loss according to the Stefan-Boltzmann law [55, 56], where ϵ is the emissivity, σ_{SB} is the Stefan-Boltzmann constant, T_{env} is the environmental temperature and f is the fractional area of the exposed inner core; see Fig. 3. f is an empiric parameter which allows the model to account implicitly for the likely presence of crusts (namely solidified lava) [25, 57], by measuring the area of the free surface not covered by the crust with values between 0 and 1. The upper bound 1 represents the absence of crust, whereas the lower bound 0 represents a surface fully solidified through which there is no heat loss (our test case of the Fogo eruption is an intermediate case). The parameter f is constant throughout the eruption and at the entire flow extent (therefore it is not transported). It represents a quantitative magnitude associated with the crust formation without providing a detailed spatial characterization.

Having in mind that the equations of any model we choose will be solved on a discrete computational grid, for each control volume located at the interface of the two fluids, the radiative heat loss is proportional to the free surface area A_{fs} of such control volume (section 3 reports its computation, see in particular Listing 3); in addition, since the energy equation is expressed per unit volume, then it is necessary also to divide for the volumetric size of each control volume, denoted as V.

The second source term on the right-hand side of Eqn. (4) represents the convective heat loss, where λ is the heat transfer coefficient. Like the radiative term, also the convective term accounts for the presence of the crust by using the parameter f, is proportional to the free surface area A_{fs}, and is defined per unit volume, hence divided by V.

Thermal heat loss at the fluid surface, which should be modelled through the thermal diffusion term, is compromised by the presence of spurious velocities [58]. Moreover, due to the nature of the Finite

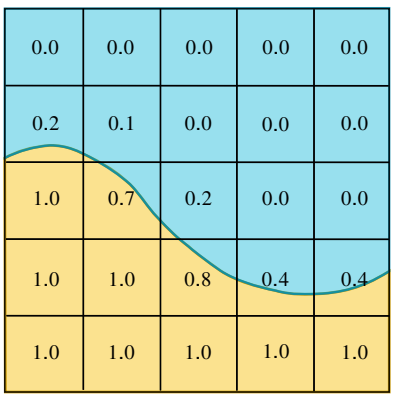


Figure 2. *Volume of Fluid. Yellow* and *blue* represent the two fluids. Defining the variable α as the phase volume fraction of the yellow fluid, the value assumed by α is reported in each cell.

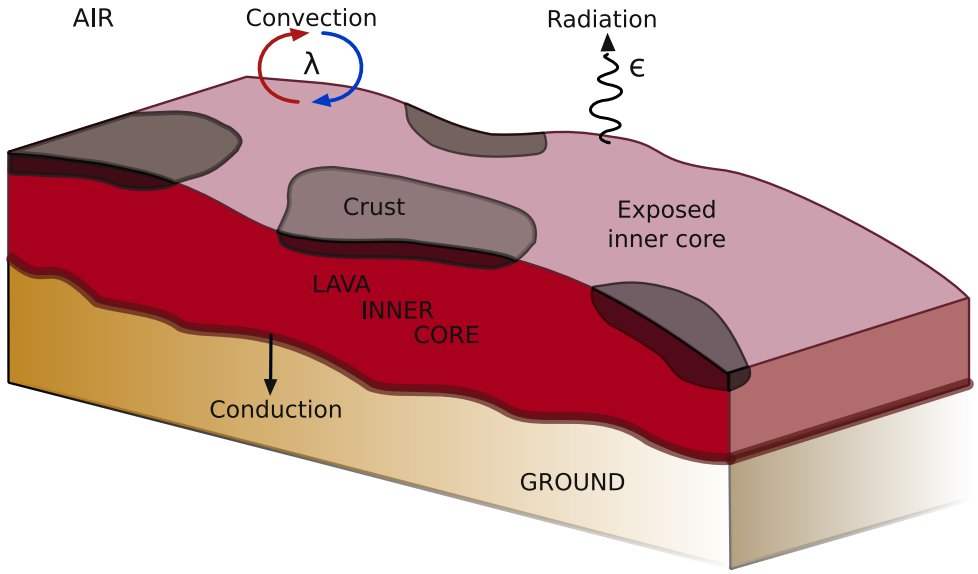


Figure 3. *Sketch of a lava flow (figure created by the authors).* The lava surface might have crusts (solidified colder lava) that prevent heat from exchanging with the atmosphere.

Volume VOF method, the cells at the interface host both phases, making it hard to select the correct value of diffusion to use. To overcome these inconveniences, the model suppresses the diffusion term on the fluid surface and, instead, considers the convective and radiative heat loss terms over there. By introducing the function $\chi_\Sigma(\mathbf{x}, t)$, which denotes the indicator function of the phase interface Σ, we use its complementary function $\bar{\chi}_\Sigma(\mathbf{x}, t)$ as a coefficient of the Laplacian term to override the diffusion at the phase interface.

Since conductive heat transfer with the bottom/soil is implemented as a boundary condition, it does not appear explicitly in the energy equation. We assume that the conductive heat flux at the fluid cells on the bottom boundary is equal to the heat flux in the bottom/soil, as described by the following equation:

$$-k\frac{T_c - T_{\text{wall}}}{d} = -k_{\text{wall}}\frac{T_{\text{wall}} - T_{\text{inf}}}{L_{\text{wall}}} \ . \tag{9}$$

The left-hand side of Eqn. (9) represents the conductive heat flux in the fluid (k is the fluid thermal conductivity). By considering a boundary cell, we denote as T_c, T_{wall} and d the temperature at the cell centre, the temperature on the face at the boundary, and the normal distance between the cell centre and the wall boundary, respectively. According to this, the left-hand side term is the discretization of the term $-k\nabla T \cdot \mathbf{n}$. The right-hand side of the equation depicts instead the conductive heat flux in the bottom/soil, where T_{inf} is the unchanged temperature in the bottom/soil, L_{wall} is the thickness of the thermal boundary layer, and k_{wall} is the thermal conductivity of the material. Rearranging the terms in Eqn. (9) reveals that this conductive boundary condition is of the Robin type as it can be expressed as a combination of Dirichlet and Neumann conditions:

$$\frac{k_{\text{wall}}}{L_{\text{wall}}}T_{\text{wall}} - k\nabla T \cdot \mathbf{n} = \frac{k_{\text{wall}}}{L_{\text{wall}}}T_{\text{inf}}. \tag{10}$$

For several materials, the viscosity is strongly temperature-dependent. In the present work, for the application to the real case of lava flows where this occurs, we adopt the non-Arrhenian Newtonian viscosity model described by the Vogel–Fulcher–Tammann (VFT) equation [54]:

$$\log \mu = A + \frac{B}{T - C}, \tag{11}$$

where μ (Pa s) is the dynamic viscosity, T is the temperature expressed in K, whereas A (Pa s), B (Pa s K^{-1}), and C (K) are empirical parameters depending on the lava's chemical composition. In particular, A is the value of $\log \mu$ at infinite temperature, it hence represents the high-temperature limit of silicate melt viscosity, whereas C is a characteristic temperature related to the glass transition temperature.

3. Implementation of the energy conservation equation and the non-Newtonian viscosity

The numerical discretization of our solver called `interThermalRadConvFoam` is based on that implemented in the OpenFOAM solver `interFoam` [38]. Our work consisted of modifying such a solver by adding the energy equation described before.

Following the structure of the original `interFoam` solver, we added a file for the temperature equation that we call `TEqn`, and its terms are assembled in the code excerpt shown in Listing 1.

```
volScalarField NoInterface(mixture.nearInterface ());
forAll(mesh.C(), celli) {
    if (NoInterface[celli] > 0) NoInterface[celli] = 0;
    else
        NoInterface[celli] = 1;
}

fvScalarMatrix TEqn (
    fvm::ddt(rhoCp,T)
    + fvm::div(rhoCpPhi,T)
    - NoInterface * fvm::laplacian(kappaf,T)
    ==
    mixture.calcSourceRadiation(U,T,RadiativeCoeff)
    + mixture.calcSourceForcedConvection(U,T)
);
```

Listing 1. Energy equation implementation with the `noInterface` field definition

The code starts by defining a volume scalar field named `NoInterface` that takes the value 0 at the interface cells and 1 otherwise, playing the role of the complementary indicator function $\bar{\chi}_\Sigma$ described in the previous section; the function `nearInterface()` is inherited from the `interfaceProperties` module. Then the temperature equation is introduced: the first term on the left-hand side of the equation is the transient term, the second is the advective term, and the third is the diffusive term, which is multiplied by the coefficient `NoInterface` which allows overriding the diffusion at the interface cells. The diffusion coefficient `kappaf` in the third term refers to the thermal conductivity, which is defined as $k = (c_p\,\nu\,\rho)/\mathrm{Pr}$, where c_p is the specific heat, the product $\nu\,\rho$ results in the dynamic viscosity μ, and Pr is the Prandtl number (a dimensionless number defined as the ratio of momentum diffusivity to thermal diffusivity). This multiphase model treats thermal conductivity as a fluid property that varies in space according to the phase volume fraction (like density and viscosity, Eqns. (5) and (6)) computed as follows:

$$k = \alpha\,k_l + (1-\alpha)\,k_g = \alpha\frac{c_{p_l}\,\rho_l\nu_l}{\mathrm{Pr}_l} + (1-\alpha)\frac{c_{p_g}\,\rho_g\nu_g}{\mathrm{Pr}_g}. \tag{12}$$

The code computes the face-interpolated values of conductivity, denoted as `kappaf`, as shown in the code excerpt of Listing 2.

```
Foam::tmp<Foam::surfaceScalarField>
Foam::incompressibleTwoPhaseMixture::kappaf() const {
    const surfaceScalarField alpha1f (
        min(max(fvc::interpolate(alpha1_), scalar(0)), scalar(1))
    );
    return tmp<surfaceScalarField> (
        new surfaceScalarField (
            "kappaf", (
                alpha1f * cp1_ * rho1_ * (1/Pr1_) * fvc::interpolate(nuModel1_->nu())
                + (scalar(1) - alpha1f) * cp2_ * rho2_ * (1/Pr2_)
                * fvc::interpolate(nuModel2_->nu())
            ) ) );
}
```

Listing 2. Computation of the thermal conductivity face-interpolated values for an incompressible two-phase mixture

First, the interpolation of the volume scalar field `alpha1_`, which represents α, is computed at the cell faces, generating a surface scalar field. The interpolated value is limited between 0 and 1 to maintain the result physically admissible. Then the surface scalar field is computed according to Eqn. (12). The parameters for density, specific heat, and Prandtl number are constant values for each phase. In contrast,

the kinematic viscosity `nu()` depends on the assumed viscosity models and is defined as a volume scalar field. Therefore, its interpolation at the cell faces is required to calculate the surface field `kappaf`.

The function `calcSourceRadiation` on the right-hand side in Listing 1 for `TEqn` computes the radiative heat transfer term between the surface of the fluid and the environment appearing in Eqn. (4). As usually done in OpenFOAM for source terms, we split the radiative term into two parts according to their numerical treatment, explicit and implicit, and we linearize the implicit part:

$$-\frac{\epsilon\sigma_{\mathrm{SB}} f A_{\mathrm{fs}}}{V}\left(T^4 - T_{\mathrm{env}}^4\right) = \frac{\epsilon\sigma_{\mathrm{SB}} f A_{\mathrm{fs}}}{V} T_{\mathrm{env}}^4 - \left(\frac{\epsilon\sigma_{\mathrm{SB}} f A_{\mathrm{fs}}}{V} T^3\right) T. \tag{13}$$

This formulation can be further modified to improve the numerical stability of the discretized radiative source term. Introducing the coefficient $\varepsilon := \left(\epsilon\sigma_{\mathrm{SB}} f A_{\mathrm{fs}}\right)/V$, adding and subtracting $\pm 4\varepsilon T^3\, T$, we can rearrange the terms as follows:

$$-\frac{\epsilon\sigma_{\mathrm{SB}} f A_{\mathrm{fs}}}{V}\left(T^4 - T_{\mathrm{env}}^4\right) = -\varepsilon\left(T^4 - T_{\mathrm{env}}^4\right) \pm 4\varepsilon T^3\, T = \varepsilon\left(T_{\mathrm{env}}^4 + 3T^4\right) - 4\varepsilon T^3\, T. \tag{14}$$

This procedure redistributes the source terms and makes the coefficient matrix resulting from the numerical discretization more diagonally dominant, thus improving the stability of the solver.

OpenFOAM offers namespaces of functions that automatically treat source terms: `Foam::fvm::Sp()` for implicit treatment, `Foam::fvm::SuSp()` for implicit/explicit discretization and `Foam::fvm::Su()` for explicit treatment.

The code excerpt of Listing 3 shows the implementation of the radiative heat transfer term in the function `calcSourceRadiation`.

```
Foam::tmp<Foam::fvScalarMatrix> Foam::addTRadImmiscibleIncompressibleTwo
PhaseMixture::calcSourceRadiation(
    const volVectorField& U,
    volScalarField& T,
    volScalarField& RadiativeCoeff
) {
    volScalarField Epsilon(nearInterface());
    volScalarField Afs(Epsilon);
    forAll(Afs, cellI) {
        if ( Afs[cellI] > 0 )
            Afs[cellI] = pow(U.mesh().V()[cellI],0.67);
    }
    forAll(Epsilon.boundaryField(), patchI) {
        forAll(Epsilon.boundaryField()[patchI], faceI ) {
            Epsilon.boundaryFieldRef()[patchI][faceI] = scalar(0.0);
        }
    }
    forAll(Epsilon, cellI) {
        if((T[cellI] > T_env_.value())) {
            Epsilon[cellI] = emissivity_.value()
                * fractionalAreaExposed_.value()
                * sigma_SB_.value() * Afs[cellI]
                / U.mesh().V()[cellI] ;
        }
        else {
            Epsilon[cellI] = scalar(0.0) ;
        }
    }
    dimensionedScalar dimCorr("dimCorr",dimMass/(pow4(dimTemperature)
    *pow3(dimTime)*dimLength),1);
    RadiativeCoeff = Epsilon * dimCorr ;
    return(
        Epsilon * dimCorr * pow4(T_env_)
        - Foam::fvm::Sp( 4 * Epsilon * dimCorr * pow3(T) , T)
        + Epsilon * dimCorr * 3 * pow4(T)
    ) ;
}
```

Listing 3. Implementation of radiative heat transfer term calculation at the interface between the two immiscible incompressible phases

In the code, variables ε and A_{fs} that appear in Eqn. (13) and Eqn. (14) are referred to as `Epsilon` and `Afs`. The former variable is initialized by the `nearInterface()` function and the latter as its copy. For each cell at the interface, defining accurately the free surface area `Afs` through which the heat loss occurs is an open problem, see discussion in section 5; in our code, as a first attempt, it is calculated by using the approximation $A_{\text{fs}} = V^{2/3}$ (which corresponds to the correct value only when the cell is a cube). The boundary conditions of `Epsilon` are set to zero on all faces of all patches, meaning that radiation is only considered inside the domain (not at the boundaries). For each cell at the fluids interface with a temperature higher than the environmental temperature, the `Epsilon` field is calculated according to the expression defined for ε, it is zero otherwise; this choice forces the radiative term to act only in the interface cells. Notice that the parameters ϵ, σ_{SB}, and f in the code are referred to as `emissivity`, `fractionalAreaExposed` and `sigma_SB` respectively. `dimCorr` is a dimensional constant used to normalize the units of `Epsilon` by multiplying them. The source term is then returned and calculated according to Eqn. (14). Lastly, we point out that we create the additional field `RadiativeCoeff` that mimics `Epsilon` to visualize with ParaView where the radiative term is active. This variable is passed in input and then updated within the function.

The function `calcSourceForcedConvection` on the right-hand side in Listing 1 for `TEqn` computes the convective heat transfer term appearing in Eqn. (4). From a mathematical point of view, the convective term is linear, differently from the radiative term that required a linearization step for the numerical discretization. We introduce the coefficient $\mathcal{W} \coloneqq \lambda f A_{\text{fs}}/V$ and, as done for the radiative term, we split the convective term into two parts, one treated explicitly and the other implicitly:

$$-\frac{\lambda f A_{\text{fs}}}{V}\left(T - T_{\text{env}}\right) = \mathcal{W}T_{\text{env}} - \mathcal{W}T. \tag{15}$$

The code excerpt of Listing 4 shows the computation of the convective heat transfer term of Eqn. (15) in the function `calcSourceForcedConvection` (whose implementation is similar to that of `calcSourceForcedConvection`, Listing 3).

```
Foam::tmp<Foam::fvScalarMatrix>
Foam::addTRadConvImmiscibleIncompressibleTwoPhaseMixture::calcSourceForcedConvec
    tion(
    const volVectorField& U,
    volScalarField& T
) {
    volScalarField W(nearInterface());
    forAll(Afs, cellI) {
        if ( Afs[cellI] > 0 )
            Afs[cellI] = pow(U.mesh().V()[cellI],0.67);
    }
    forAll(W.boundaryField(), patchI) {
        forAll(W.boundaryField()[patchI], faceI ) {
            W.boundaryFieldRef()[patchI][faceI] = scalar(0.0) ; }
    }
    forAll(W, cellI) {
        if((T[cellI] > T_env_.value())) {
            W[cellI] = heatTransferCoeff_.value()
            * fractionalAreaExposed_.value()
            * Afs[cellI] / U.mesh().V()[cellI] ;
        }
        else
            W[cellI] = scalar(0.0) ;
    }
    dimensionedScalar dimCorr(
        "dimCorr",dimMass/(dimTemperature*pow3(dimTime)*dimLength),1);

    return(
        W * dimCorr * T_env_
        - Foam::fvm::Sp( W * dimCorr , T)  );
}
```

Listing 4. Implementation of convective heat loss term calculation at the interface between the two immiscible incompressible phases

In the code, variables $\mathcal{W}$ and A_{fs} that appear in Eqn. (15) are referred to as `W` and `Afs`. The former variable is initialized by the `nearInterface()` function, and the latter has already been initialized in Listing 3 and here is then updated by using again the approximation $A_{\text{fs}} = V^{2/3}$. The boundary conditions of `W` are set to zero on all faces of all patches, meaning that radiation is only considered inside the domain (not at the boundaries). For each cell at the fluids interface with a temperature higher than the environmental temperature, the `W` field is calculated according to the expression defined for $\mathcal{W}$, it is zero otherwise; this choice forces the convective term to act only in the interface cells. Notice that the parameters λ and f in the code are referred to as `heatTransferCoeff` and `fractionalAreaExposed`, respectively. `dimCorr` is a dimensional constant used to normalize the units of `W` by multiplying them. The source term is then returned and calculated according to Eqn. (15).

By rearranging the terms of Eqn. (9), we obtain the explicit expression of the temperature at the boundary T_{wall}:

$$T_{\text{wall}} = (1-\varphi)T_c + \varphi T_{\text{inf}}, \quad \varphi \coloneqq \frac{1}{1+\dfrac{k}{d}\dfrac{L_{\text{wall}}}{k_{\text{wall}}}}. \tag{16}$$

Equation (16) shows that the boundary condition to impose is a Robin condition as it is the sum of Neumann and Dirichlet conditions (found for $\varphi = 0$ and $\varphi = 1$ respectively). Implementing such a boundary condition is done by modifying the file `0/T` of the test case as follows. First, as shown in Listing 5, some reference parameters (that must be coherent with those specified in constant/transportProperties) are added:

```
dimensions      [0 0 0 1 0 0 0]; // kg m s K mol A cd
T_env           293; // environmental temperature (K)
fluidDensity    954;
Pr              34000;
cp              1500;
internalField   uniform $T_env;
```

Listing 5. Parameters for temperature initial and boundary conditions in file `0/T`

Secondly, we implement the Robin boundary condition with the boundary condition type called `exprMixed` as shown in Listing 6, where the value for the Dirichlet condition is defined by `valueExpr` and the value of φ is `fractionExpr`.

```
    lowerWall {
        type            exprMixed;
        value           $internalField;
        variables       "Tinf=$T_env; rho=$fluidDensity; cp=$cp; DT=1/$Pr;
                        k=DT*rho*cp; L_wall=0.002; k_wall=0.03";
        valueExpr       "Tinf";
        fractionExpr    "1.0/(1.0 + ( k/(mag(pos())) * (L_wall/k_wall) ))";
    }
```

Listing 6. Conductive heat loss as temperature boundary condition in file `0/T`

In the same code, `k` and `k_wall` are the fluid and bottom/soil conductivity (k and k_{wall} in Eqn. (16)), `L_wall` is the thickness of the thermal boundary layer in the bottom/soil (L_{wall}), and the cell centre-boundary distance d is approximated by `mag(pos())`.

To implement the temperature-dependent viscosity model VFT defined in Eqn. (11), we created a new transport model called `tdepGiordano`. The VFT model requires the values of the constant parameters A, B, and C, and its implementation also needs the minimum and maximum values of the kinematic viscosity ν_{min}, ν_{max} (the kinematic viscosity is defined as the dynamic viscosity divided by the density, $\nu \coloneqq \mu/\rho$ m^2 s^{-1}). The value of these five parameters is read at runtime from the `transportProperties` dictionary located in the test case input file constant/transportProperties. The constructor function is written in the C-file tdepGiordano.C located in src/transportModels/incompressible/viscosityModels/tdepGiordano is shown in Listing 7.

The constructor initializes the basic class `viscosityModel` with the parameters given, extracts the subdictionary `tdepGiordanoCoeffs`, then retrieves the configuration values (`A`, `B`, `C`, `nuMin`, `nuMax`) from the dictionary and assigns them to the corresponding members of the class. Moreover, if present, the

```
Foam::viscosityModels::tdepGiordano::tdepGiordano (
    const word& name,
    const dictionary& viscosityProperties,
    const volVectorField& U,
    const surfaceScalarField& phi
) : viscosityModel(name, viscosityProperties, U, phi),
    tdepGiordanoCoeffs_(viscosityProperties.subDict(typeName + "Coeffs")),
    A_(tdepGiordanoCoeffs_.lookup("A")),
    B_(tdepGiordanoCoeffs_.lookup("B")),
    C_(tdepGiordanoCoeffs_.lookup("C")),
    nuMin_(tdepGiordanoCoeffs_.lookup("nuMin")),
    nuMax_(tdepGiordanoCoeffs_.lookup("nuMax")),
    rho_(tdepGiordanoCoeffs_.lookupOrDefault("rho",
            dimensionedScalar("rho", dimensionSet(1,-3,0,0,0,0,0), 2700.0))),
    nu_ (
        IOobject (
            name,
            U_.time().timeName(),
            U_.db(),
            IOobject::NO_READ,
            IOobject::AUTO_WRITE ),
        calcNu_()
    ) {}
```

Listing 7. Constructor function of the VFT viscosity model

parameter `rho` (for the density) is retrieved, whereas the default value 2700 is adopted otherwise. The value 2700 kg m^{-3} is commonly attributed to basaltic lava density (ρ) [59]. Finally, the constructor calls the private member function `calcNu_` to compute the kinematic viscosity `nu_` based on Eqn. (11). The implementation of `calcNu_` is reported in Listing 8.

```
Foam::tmp<Foam::volScalarField>
Foam::viscosityModels::tdepGiordano::calcNu_() const {
    const volScalarField& T= U_.mesh().lookupObject<volScalarField>("T");
    return max (
        nuMin_,
        min (
            nuMax_,
            dimensionedScalar("n", dimensionSet(0,2,-1,0,0,0,0), 1.0)*(Foam::pow(
                10.0, A_ + B_/max( (T-C_),dimensionedScalar(
                    "minT", dimensionSet(0,0,0,1,0,0,0), 100.0) ))
            )/ rho_.value()
        )
    );
}
```

Listing 8. Implementation of the private function `calcNu_`

Equation (11), implemented in the code Listing 8, defines the dynamic viscosity; therefore, in the code, the division by the fluid density `rho` is needed to get the kinematic viscosity. Moreover, we highlight that Eqn. (11) is valid only for lava temperatures T significantly higher than C: when T approaches C, extremely high values of viscosity, which might not reflect the real physical behaviour anymore, are produced. This is due to the transition from the liquid towards the glassy state, and, in such a regime, the system approaches a solid-like state, so other models might become more appropriate. To overcome this problem, a lower limit for the difference $T - C$ is introduced, `minT`, which assumes the value 100 K [60].

4. Numerical simulations

This section presents some results of numerical tests performed with `interThermalRadConvFoam`. These tests are based on a benchmark proposed in [43] for lava flow simulations (denoted as BM3) and on the simulation of the first 24 hours of a real lava flow (Fogo volcano, 2014–2015 effusive eruption).

In all simulations presented, the air was modelled as a Newtonian incompressible fluid with density $\rho = 1\ \mathrm{kg\,m^{-3}}$ and kinematic viscosity $\nu = 1.48 \times 10^{-5}\ \mathrm{m^2 s^{-1}}$. Furthermore, the regime was set to laminar (no turbulence effects are modelled), and the solver runs in PIMPLE mode (a combination of the PISO and SIMPLE algorithms [61]).

4.1. **BM3: Axisymmetric cooling and spreading.** The first test simulates a hot viscous fluid injected into a horizontal plane, which starts spreading and cooling in an axisymmetric fashion. In this test, even if we model temperature changes, the viscosity is not temperature-dependent.

For this test we consider a Newtonian fluid (with kinematic viscosity $\nu = 3.56 \times 10^{-3}\ \mathrm{m^2\,s^{-1}}$ and density $\rho = 954\ \mathrm{kg\,m^{-3}}$) injected onto the plane with an effusion rate of $2.2 \times 10^{-8}\ \mathrm{m^3\,s^{-1}}$ from a hole of 4×10^{-3} m radius at the temperature $T_{\mathrm{vent}} = 42°\mathrm{C}$ while the environment is at $T_{\mathrm{env}} = 20°\mathrm{C}$ (the complete set of physical parameters is reported in Tab. 1). This test refers to an analogue experiment reported in [62], denoted as C14, where a hot silicone oil (Rhodorsil® 47V 5000 or 47V 12500, dyed red) is injected, at a constant volumetric supply rate $R = 2.2 \times 10^{-8}\ \mathrm{m^3 s^{-1}}$, onto a horizontal plane of polystyrene from a point source of 2–4 mm of radius.

Table 1. *BM3: Axisymmetric cooling and spreading.* Physical parameters of the hot silicone oil simulation. The symbol * refers to the parameters available in [62] and not provided in [43].

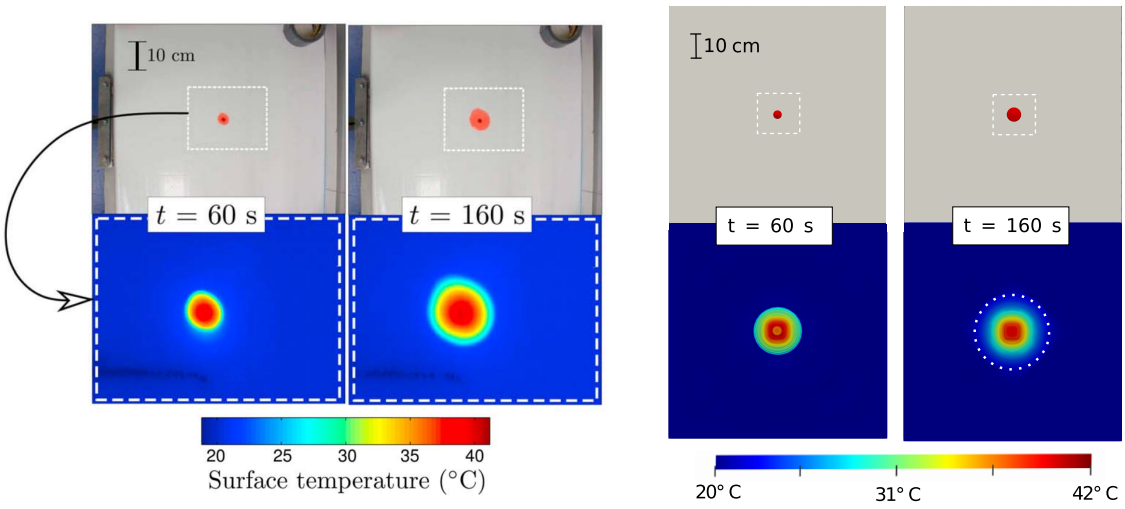
Symbol	Value	Definition	Unit
ρ	954	fluid density	$\mathrm{kg\,m^{-3}}$
μ	3.4	dynamic viscosity	$\mathrm{Pa\,s}$
c_p	1500	specific heat of fluid	$\mathrm{m^2\,s^{-2}\,K^{-1}}$
k	0.15	thermal conductivity of fluid	$\mathrm{W\,m^{-1}\,K^{-1}}$
κ	10^{-7}	thermal diffusivity of fluid *	$\mathrm{m^2\,s^{-1}}$
λ	2	convective heat transfer coeff.	$\mathrm{W\,m^{-2}\,K^{-1}}$
ϵ	0.96	emissivity	-
σ_{SB}	5.67×10^{-8}	Stefan-Boltzmann constant	$\mathrm{kg\,s^{-3}\,K^{-4}}$
T_{env}	293.15	temp. of environment	K
T_{vent}	315.15	temp. of fluid at the vent	K
T_{soil}	293.15	temp. of soil	K
k_{soil}	0.03	thermal conductivity of soil *	$\mathrm{W\,m^{-1}\,K^{-1}}$
κ_{soil}	6×10^{-7}	thermal diffusivity of soil *	$\mathrm{W\,m^{-2}\,J^{-1}}$
R	2.2×10^{-8}	effusion rate	$\mathrm{m^3 s^{-1}}$

The 3D computational domain that we adopt is 16×10^{-2} m long and wide, and only 10^{-2} m high because the fluid propagates in a very thin layer. Such a rectangular mesh is generated with `blockMesh`, then `snappyHexMesh` refines the mesh around the centre of the lower wall, and with `topoSet` and `createPatch` we create the hole at its centre. These preprocessing steps, provided in the `Allrun` file, are followed by the execution of the solver. The high-quality mesh and relatively low propagation speed allow the solver to run with a minimalist configuration, requiring only one outer and one inner correction loop.

The main aim of the suite of simulations performed for this test is to study the performance obtained by using the adaptive mesh refinement (that helps to keep a sharp interface and to reduce the computation time with respect to a uniform, static, fine mesh). In addition, we discuss the scalability performance of the parallel execution, and, finally, we test the capability of the terms implemented in the energy equation of the model to describe the fluid heat losses properly.

Before presenting a more quantitative analysis of our results, we show a qualitative comparison between the results of our simulations and those of the laboratory experiment putting them alongside in Fig. 4. Even though the pictures that represent the experiment and the simulation are not directly comparable, both of them show that the temperature gradient, from the centre to the front of propagation, is more diffused at time $t = 160$ s with respect to time $t = 60$ s. Notice that the temperature values only give qualitative information because, due to the nature of the Finite Volume VOF method, they come from an average between the temperature of the two phases, and therefore, the value of the fluid temperature at the interface cannot be precisely determined.

Whereas in Fig. 4 we presented a 2D view from the top of simulation results, the simulations are fully 3D, and in Fig. 5 we present a three-dimensional view of the liquid free surface (represented by the contour of $\alpha = 0.5$) at time $t = 60$, 160, 380 s respectively, obtained from a simulation that uses an



(a) Picture taken from [62]. **(b)** Results of our simulation.

Figure 4. *Axisymmetric cooling and spreading.* (a) Optical (*top row*) and infrared (*bottom row*) images taken during an analog laboratory experiment at time $t = 60, 160$ s. The dashed rectangle in the optical image corresponds to the field of view of the infrared image below. (b) Top views of the fluid free-surface at time $t = 60, 160$ s. *Top row:* contour of $\alpha = 0.5$ coloured in red. *Bottom row:* zoom on the temperature field. The dashed square in the *top row* corresponds to the field of view of the *bottom row*. The white dotted line in the *bottom row* at time $t = 160$ s represents the fluid extension.

adaptive mesh with up to 4 levels of refinement, starting from a uniform grid with $\Delta x = 0.005$ m. The 3D plots clearly show that the grid refinement dynamically follows the liquid/air interface.

In Figs. 6 to 8 we show and compare results and mesh refinements for simulations using different computational grids and dynamic mesh with adaptive refinement at two times $t = 20, 380$ s. Using a dynamic mesh with adaptive refinement means that the mesh may change during the computation and might be locally refined or coarsened according to some parameters given by the user. The user can set the dynamic and adaptive refinement conditions inside the dictionary dynamicMeshDict located inside the folder case/constant related to the test case of interest. In Listing 9, we report an extract of the conditions adopted for our test.

```
refineInterval   10;
field            alpha.fluid;
lowerRefineLevel 0.01;
upperRefineLevel 0.99;
nBufferLayers    1;
maxRefinement    3;
```

Listing 9. Mesh refinement settings based on `alpha` field for the for BM3 test

For our tests, we refine the mesh based on the `alpha.fluid` field, refining regions that host the phase interface, namely where $0.01 \leq \alpha \leq 0.99$, by setting `lowerRefineLevel` and `upperRefineLevel` accordingly. The refinement is done for a maximum of 3 levels (`maxRefinement`) and is updated every 10 simulation steps (`refineInterval`). Moreover, we opted for a 1 buffer layer (`nBufferLayers`) of cells adjacent to the refined ones that should also be refined to ensure a smooth transition in the mesh and improve numerical stability.

It is worth recalling that the Courant number is defined as $\mathrm{Co} = \max\{\mathrm{u}\Delta\mathrm{t}/\Delta\mathrm{x}\}$, where Δt is the time step, u is the fluid velocity, and Δx is the cell dimension, see [63, 64], and [65, §4.4]. So, as Co depends on the size of the spatial discretization, the refinement leads to a reduction of the time step determined by the CFL condition.

Figure 6 and 7 show the results of simulations obtained with adaptive meshes that use up to 3 and 4 levels of refinement respectively, but that initially started with the same uniform mesh discretized with a grid resolution of 5×10^{-3} m. As one might expect, the interface is described more sharply in the simulation with the highest level of refinement, i.e. 4, represented in Fig. 7. Figure 8 depicts the result of a simulation computed with the use of dynamic refinement up to level 3 and with an initial uniform

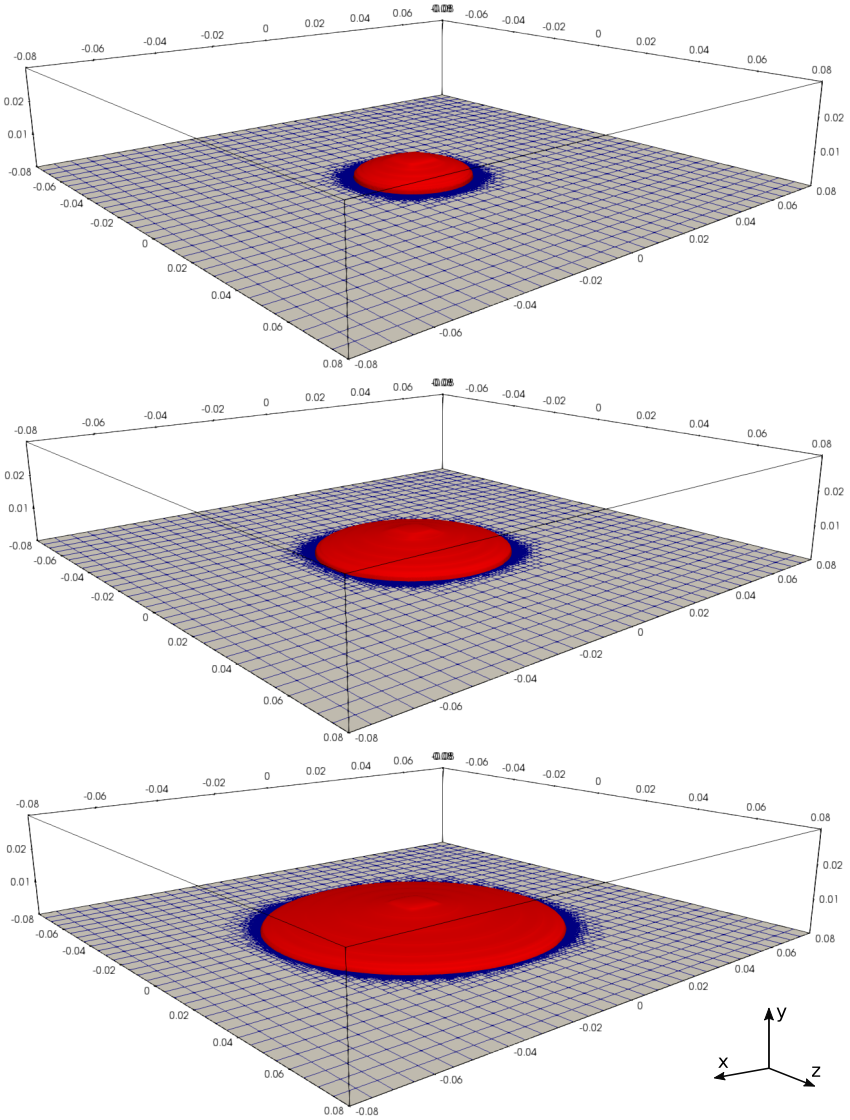


Figure 5. *Axisymmetric cooling and spreading.* Fluid phase free-surface and grid refinement of the horizontal plane $y = 0$. The computational domain $[-8; 8]$ cm $\times$ $[-8; 8]$ cm is outlined. Simulation at time $t = 60, 160, 380$ s obtained with adaptive mesh refinement up to 4 levels of refinement. Because of the very small thickness of the surface, we used a vertical scale increased by a factor 3, in order to better appreciate the three-dimensional nature of the flow surface.

mesh discretized with a grid resolution of 2.5×10^{-3} m (half the resolution of the previous simulations). The results of this last case are comparable to those obtained with an initial grid resolution of 5×10^{-3} m and 4 refinement levels, both in terms of sharpness of the phase interface and execution time of the code as the number of cells is similar (see Tab. 2).

Table 2. *Axisymmetric cooling and spreading.* Elapsed execution time for the simulations of 380 s with different mesh refinements.

initial Δx	ref. level	time	cells number
0.005 m	3	8112.01 s	246 394
0.005 m	4	31 486.70 s	832 186
0.0025 m	3	32 348.10 s	904 385

In every case depicted in Figs. 6 to 8, it is possible to appreciate both the refinement of the meshes (which occurs as the phase interface propagates), and the subsequent mesh coarsening. Furthermore, we also notice that in Fig. 8 there is a refinement in correspondence of the origin that does not coarse. This

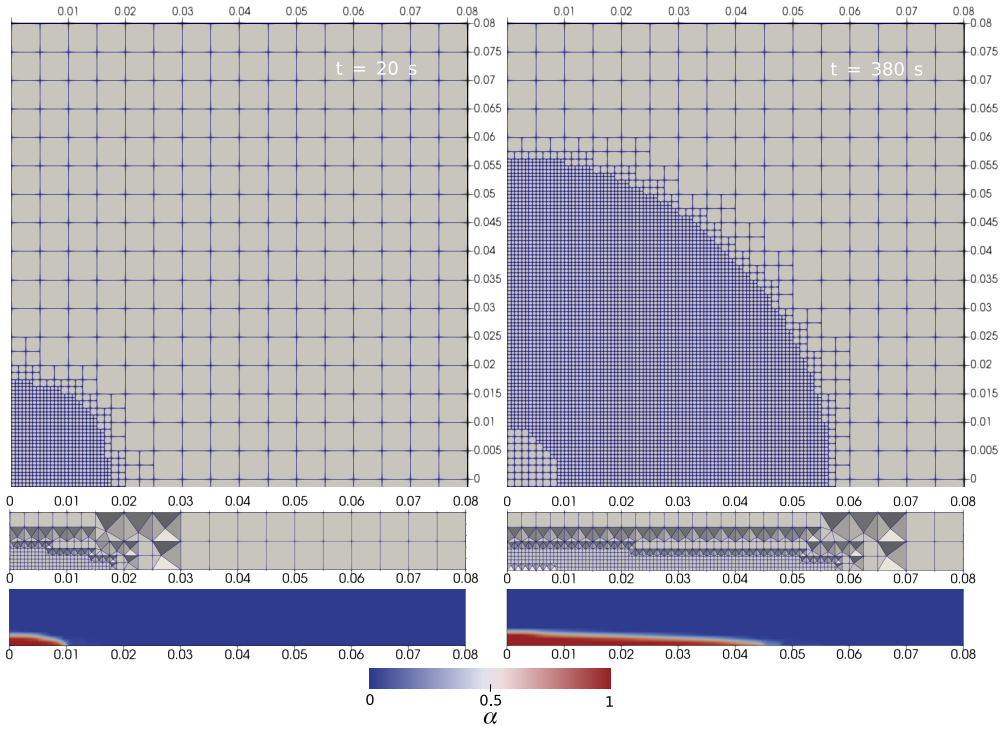


Figure 6. *Axisymmetric cooling and spreading.* Simulation computed with the dynamic mesh refinement up to 3 refinement levels over an initial uniform mesh defined by $\Delta x = 0.005$ m (that is also the dimension of the coarsest discretization depicted) represented at time $t = 20, 380$ s. *Top row:* bottom view of the discretization grid, plane $y = 0$. *Middle row:* side view of the discretization grid on the plane $x = 0$. *Bottom row:* side view of the α-field on the plane $x = 0$.

is caused by the fact that, for this simulation, we used a finer initial grid in correspondence of the inlet hole, namely centered in $(0, 0, 0)$ and with radius 4×10^{-3} m, to better capture the circular shape of the hole.

In addition, we investigated the effect of the grid size by comparing a simulation obtained with a uniform static mesh with a grid resolution of 6.25×10^{-4} m to a simulation obtained with a dynamic mesh with 3 refinement levels and an initial grid resolution of 5×10^{-3} m. The resolution of the uniform static mesh corresponds to the minimum cell size obtained with the dynamic mesh refinement. From Fig. 9 we can see that the results of the two simulations are very similar, both in terms of runout and sharpness of the interface. It is important to notice that in this case also the computational time required is similar since the simulation over the uniform static mesh required 5994.02 s and the other 8112.01 s. This is because the 3rd refinement level occupies a big part of the computational domain (see Fig. 6) because the thickness of the domain is small and the refinement is still relatively coarse, therefore there is no time-saving in the use of the dynamic refinement. With a real (non-flat) topography the vertical extent would be larger, and the advantages of the dynamic refinement would be more evident.

As a further investigation, we study the time-series analysis of the radius. We use the analytical expression of the radial flow advance

$$x(t) \approx 0.715 \left[\frac{g\rho R^3}{3\mu}\right]^{1/8} t^{1/2}, \tag{17}$$

determined by Huppert [66], for a convergence study, Fig. 10. Starting from a uniform grid with $\Delta x = 0.005$ m, we adopted respectively 2, 3, and 4 levels of dynamic refinement. The results show a good convergence of the simulations with the finer level of refinement 3 and 4, so, in the following, we do not use the case with 2 levels of refinement. Moreover, we can appreciate that the result obtained with 4 levels of refinement is completely superimposable to that obtained with a simulation that starts with a uniform grid of $\Delta x = 0.0025$ m and that evolves with 3 levels of dynamic refinement. Indeed, the finest cells in these two cases have the same dimensions.

The second part of this section focuses on the parallel computation and the scalability performances of this test. For our parallel calculations, we used the High-Performance Computing Super Micro cluster

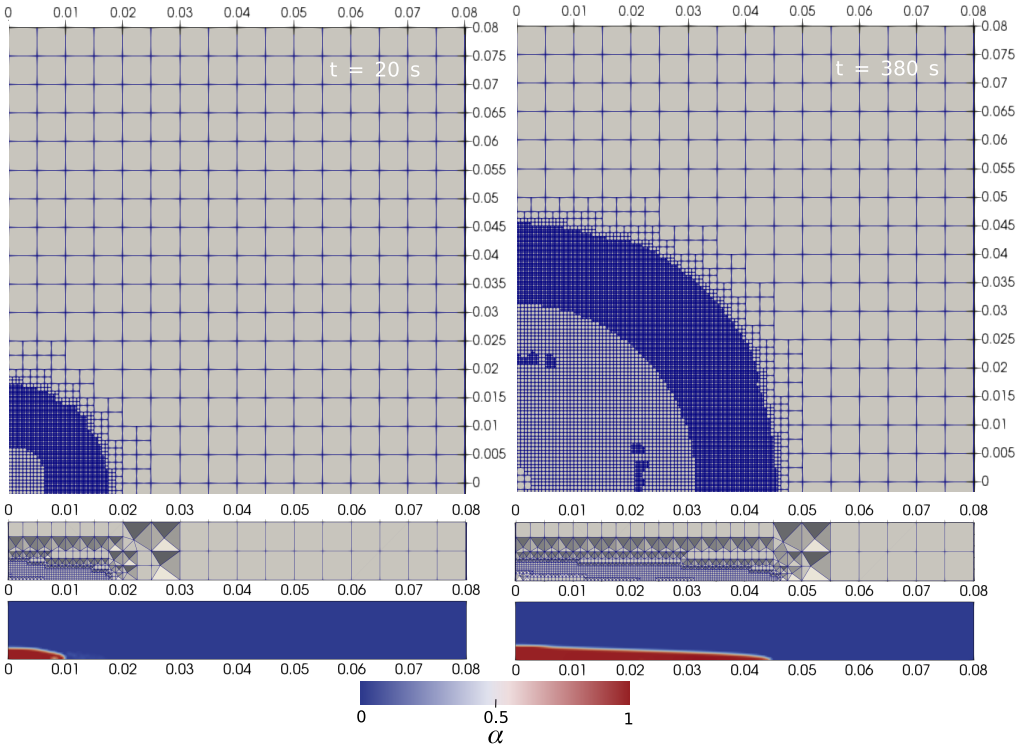


Figure 7. *Axisymmetric cooling and spreading.* Simulation computed with the dynamic mesh refinement up to 4 refinement levels over an initial uniform mesh defined by $\Delta x = 0.005$ m (that is also the dimension of the coarsest discretization depicted) represented at time $t = 20, 380$ s. *Top row:* bottom view of the discretization grid, plane $y = 0$. *Middle row:* side view of the discretization grid on the plane $x = 0$. *Bottom row:* side view of the α-field on the plane $x = 0$.

laki (16 nodes and 256 core Intel Xeon 2.40GHz, interconnection InfiniBand 40 Gbps at low latency) hosted at INGV, Section of Pisa. We parallelized the test that uses the adaptive mesh up to 4 refinement levels starting from an initial uniform grid with $\Delta x = 5 \times 10^{-3}$ m (Fig. 7). OpenFOAM parallelization relies on domain decomposition and the subsequent distribution of fields. For our tests, we divided the domain (along the horizontal directions) into 2, 4, 8, and 16 subdomains, respectively. Table 3 reports the elapsed time needed to compute the simulations for each decomposition, the execution time necessary for the serial computation, and the (approximate) amount of cells per core.

As expected, serial computation requires a longer time than the other cases. Moreover, the execution distributed on 4 cores took half the time needed for the calculation on 2 cores. However, as the number of cores increases, the decrease in execution time does not persist because using 8 and 16 cores requires more time than using 4 (and this time is still less than that used for the serial computation). This limit in the scalability process is likely due to the geometric domain decompositions. The performance improvement when we move from 1 to 2 and then to 4 cores is due to the symmetric domain decomposition (2×1 and 2×2) that matches the dynamics symmetry, see Fig. 11 on the left as an example; each core has indeed a similar computational load (which is reported in Tab. 3). Passing to a higher number of cores, 8 and 16, with the actual strategy, the cores closer to the vent have a computation overload with respect to the others because they have to deal with many more cells due to the dynamic mesh refinement that expands with the fluid propagation (at least at this stage of the simulation), see Fig. 11 on the right as an example. Indeed, the cores close to the vent have assigned approximately 2×10^5 cells while the others 10^3, Tab. 3. We point out that some specific studies showed that the optimum cell count per core ranges from approximately 10 000 to approximately 50 000 and is highly dependent on the solver being used; the reader can refer to the study in [67]). We also presume that, in the simulation's longer term, when the fluid spreads more widely on the domain, the performances with 8 and 16 cores might improve. Possible solutions to overcome these scalability limitations in future studies are to use other options provided in OpenFOAM for the domain decomposition and to apply load-balancing strategies.

We conclude this section by analysing the temperature behaviour, discussing the influence of the different thermal heat loss processes on the final temperature distribution within the liquid, similar to the study proposed by Costa and Macedonio [25] for the shallow water model they proposed. For this

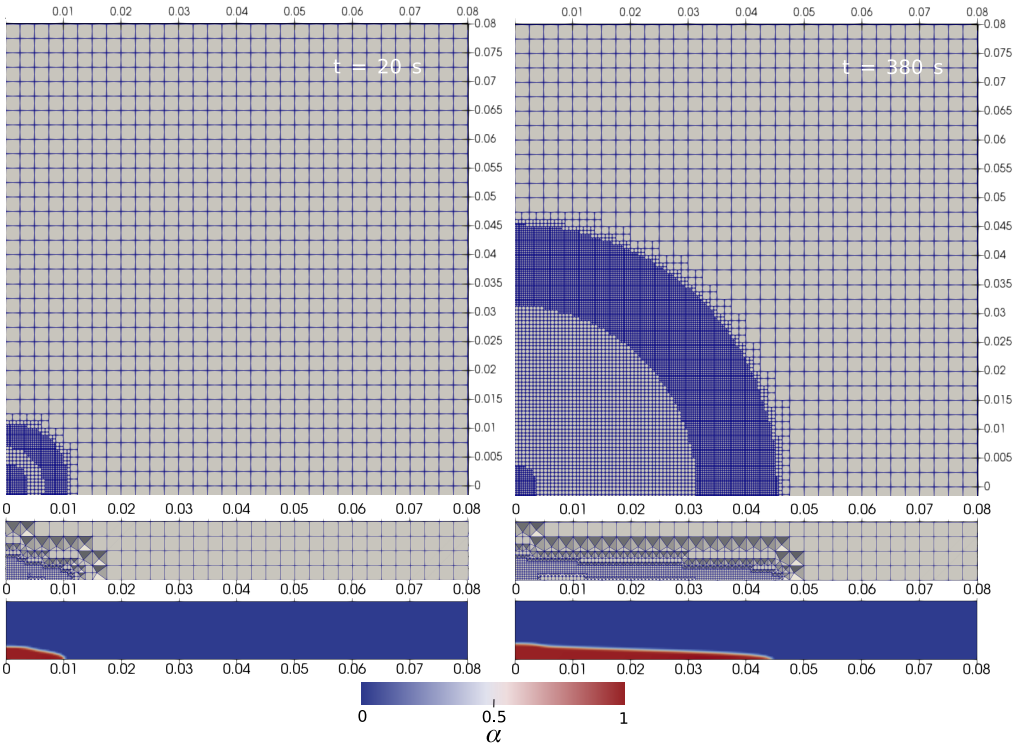


Figure 8. *Axisymmetric cooling and spreading.* Simulation computed with the dynamic mesh refinement up to 3 refinement levels over an initial uniform mesh defined by $\Delta x = 0.0025$ m (that is also the dimension of the coarsest discretization depicted) represented at time $t = 20$, 380 s. *Top row:* bottom view of the discretization grid, plane $y = 0$. *Middle row:* side view of the discretization grid on the plane $x = 0$. *Bottom row:* side view of the α-field on the plane $x = 0$.

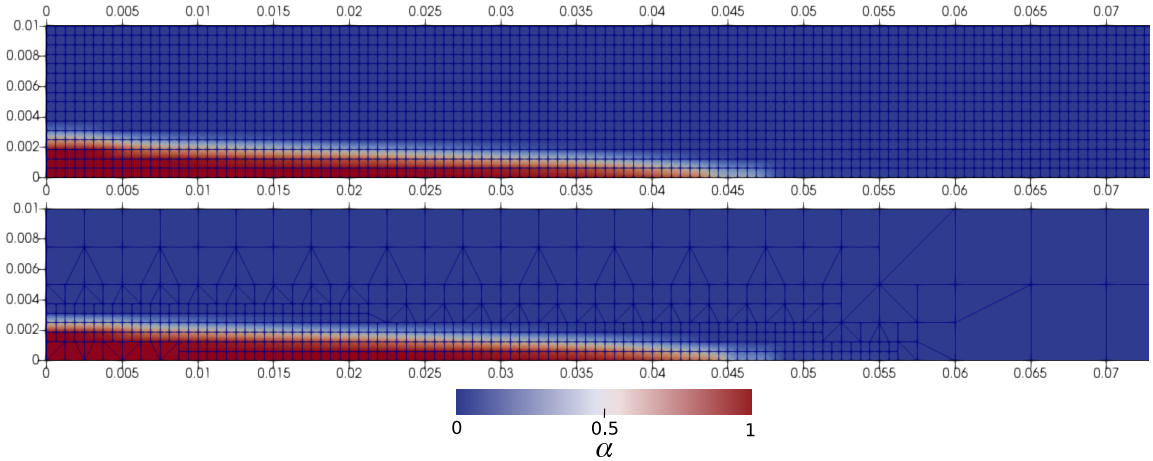


Figure 9. *Axisymmetric cooling and spreading.* Comparison between simulations computed with static (*top*) and dynamic (*bottom*) mesh. Side view of the α-field at time $t = 380$ s on the vertical plane $x = 0$. The dimension of the coarsest cells is 0.005 m, and the dimension of the finest is 6.25×10^{-4} m.

purpose, we consider the simulation results obtained using the adaptive mesh with up to 4 refinement levels at time $t = 380$ s (namely the case shown in Fig. 7). We compare the differences in temperature distribution when all heat loss mechanisms are active versus cases where only some are, see Fig. 12. We have selected the following conditions: (a) all the heat loss processes are active; (b) only convection; (c) only radiation; (d) radiation and convection but no conduction; (e) only conduction.

We observe that convection and conduction have a relatively minor impact on thermal heat loss compared to radiation. This is evident from cases (a), (c) and (d), which exhibit nearly identical behaviour and similar average temperature, 23.4–24.3 °C. In all these three cases the radiative heat loss is

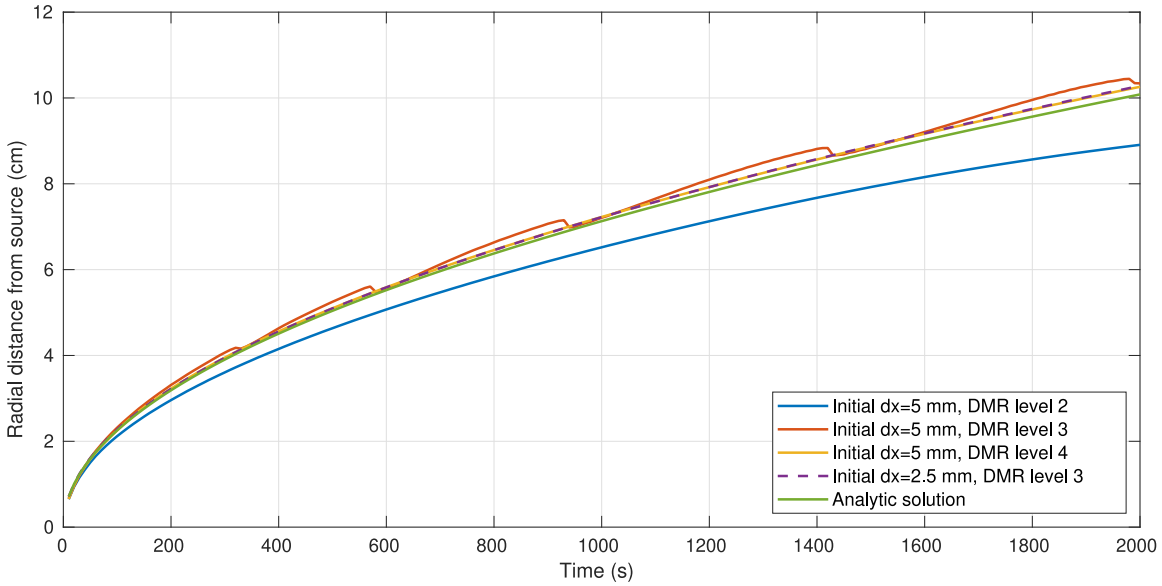


Figure 10. *Axisymmetric cooling and spreading.* Convergence study obtained comparing the theoretical front position, Eqn. (17), and the results of simulations. The interface is defined from the value of $\alpha = 0.5$.

Table 3. *Axisymmetric cooling and spreading.* Comparison of serial and parallelized simulations. The table indicates the number of cores used, the domain decompositions, the elapsed execution times to simulate 380 s, the percentage of the time saved by the various parallelized simulations compared to the serial one, and the (approximate) number of cells per core. When the distribution of the cells is not uniform among the decomposition parcels, we report the interval indicating the minimum and maximum number of cells assigned.

Cores	Decomposition	Time	Relative performance	Cells number per core
1	–	31 486.72 s	–	8×10^5
2	2×1	28 184.20 s	10%	4×10^5
4	2×2	14 115.90 s	56%	2×10^5
8	4×2	26 704.01 s	15%	$[10^3,\ 2 \times 10^5]$
16	4×4	26 682.20 s	15%	$[10^3,\ 2 \times 10^5]$

active and the temperature reaches environmental conditions at a distance of (*a*) 3.7, (*c*) 4.3 and (*d*) 3.8 cm from the centre, considering that the fluid extends up to 4.4 cm (we remark that the viscosity is not temperature-dependent in this test, the dynamics are therefore not affected by temperature distributions). Convection alone, case (*b*), has a mild impact as the average temperature is 29.8 °C, and the minimum temperature reaches 23 °C. The different impacts that convection and radiation (see cases (*b*) and (*c*)) have on heat loss depend on the fluid temperature itself. Considering the two terms, Eqn. (13) and Eqn. (15), their relative magnitude changes while temperature decreases. For example, for $T \approx T_{\text{vent}}$, the radiative term is one order of magnitude higher than the convective term, whereas for $T \approx 300$ K (27 °C), the two terms have the same magnitude. Although in this setup the conduction alone has the least impact on the heat loss (see case (*e*)), its contribution is not negligible, as proven by the test that shows an effective drop in the temperature of almost 2 °C close to the bottom where the minimum temperature reaches 40.5 °C. The thermal diffusion process is always active, and it is a dominant process in the only conduction case (*e*) where the temperature gradient is small.

4.2. **Natural case: the Pico do Fogo 2014–2015 Eruption.** In this test, we apply our model to simulate a natural effusive eruption. As a reference, we adopt the last eruption that occurred at Pico do Fogo volcano, which started on 23 November 2014 and ended on 8 February 2015. The actual topography of Fogo and the characteristic data of such an event, like the vent location, effusion rate, and effusive temperature, are used to simulate the eruption's earliest hours. This test aims to show the model's actual applicability to a real-case scenario.

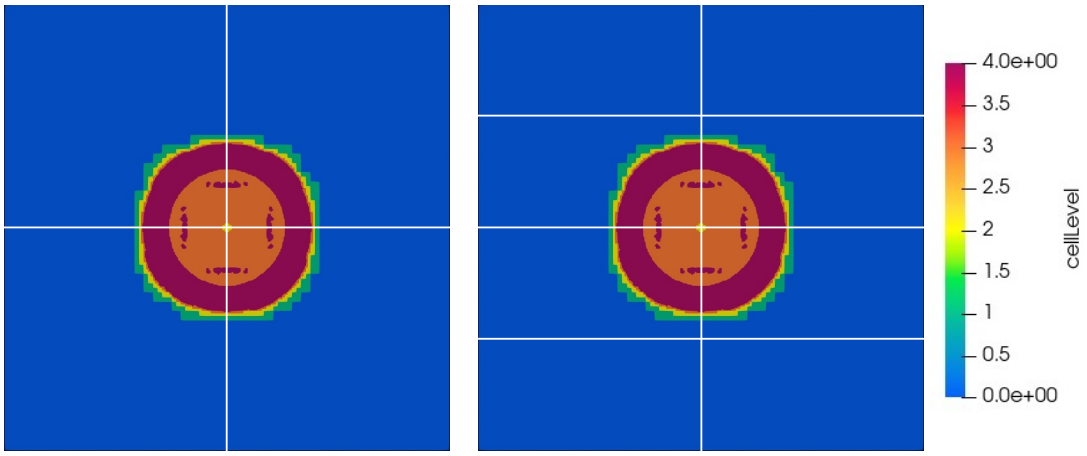


Figure 11. *Axisymmetric cooling and spreading.* Examples of parallelization decompositions among the cores (view from the domain bottom): *left* 2×2 decomposition, *right* 4×2 decomposition. The colour scale represents the level of mesh refinement.

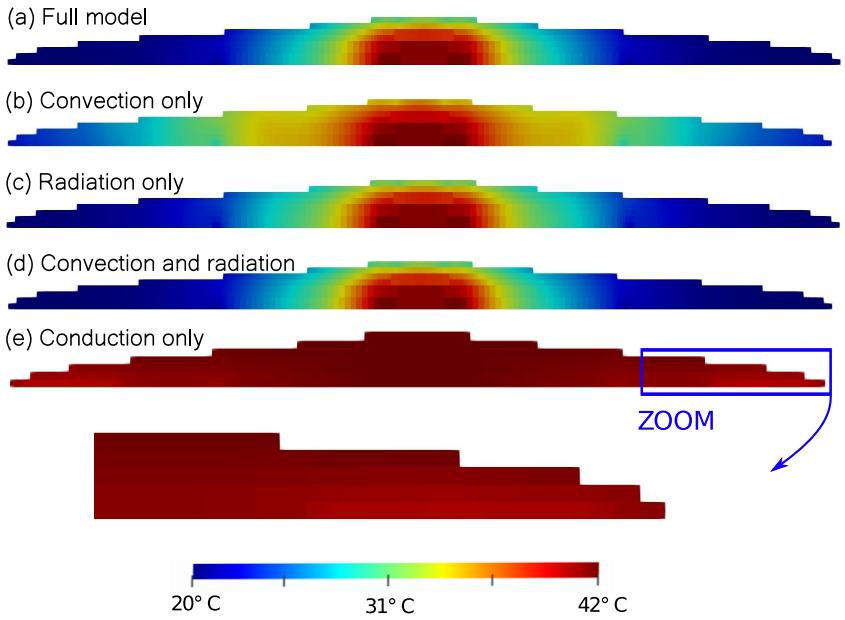


Figure 12. *Axisymmetric cooling and spreading.* Temperature distribution inside the liquid ($0.5 \leq \alpha \leq 1$) at time $t = 380$ s, side view at $x = 0$. At the bottom, there is the zoom of the front of the liquid in the conduction-only case, in order to better appreciate the effects of conductive heat loss. A vertical scale magnified by a factor 2 was used to enhance the visualization (to appreciate the actual proportion between the horizontal and vertical directions, the reader can refer to Fig. 7-right and Fig. 9-bottom).

Cape Verde is an archipelago with volcanic origins located west of the Western Atlantic coast of Africa [68]. Fogo Island stands between Brava and Santiago islands and is the highest with 2829 m above the sea level of Pico do Fogo, see Fig. 13b. An active volcano stands in the island's centre, presents a 9 km wide caldera, Chã das Caldeiras ("Plain of the Calderas"), and has a summit at Pico do Fogo. Bordeira is an enormous crater rim up to 1 km high that encircles the caldera on the western side, Fig. 13a. Fogo is the youngest and most active volcano of the archipelago [69, 70]. During the eruption, the lava emission started from a fissure on the southwest flank of Pico do Fogo, and the flow propagated in Chã das Caldeiras, producing two main branches aligned along the direction North West-South East, Fig. 13c.

Topography first and vent location second impact the simulation of real events. Therefore, the more accurate such data are, the more reliable the results are. Our simulation uses the topography that originates from a DEM (Digital Elevation Model) file generated from SAR satellite with data acquired

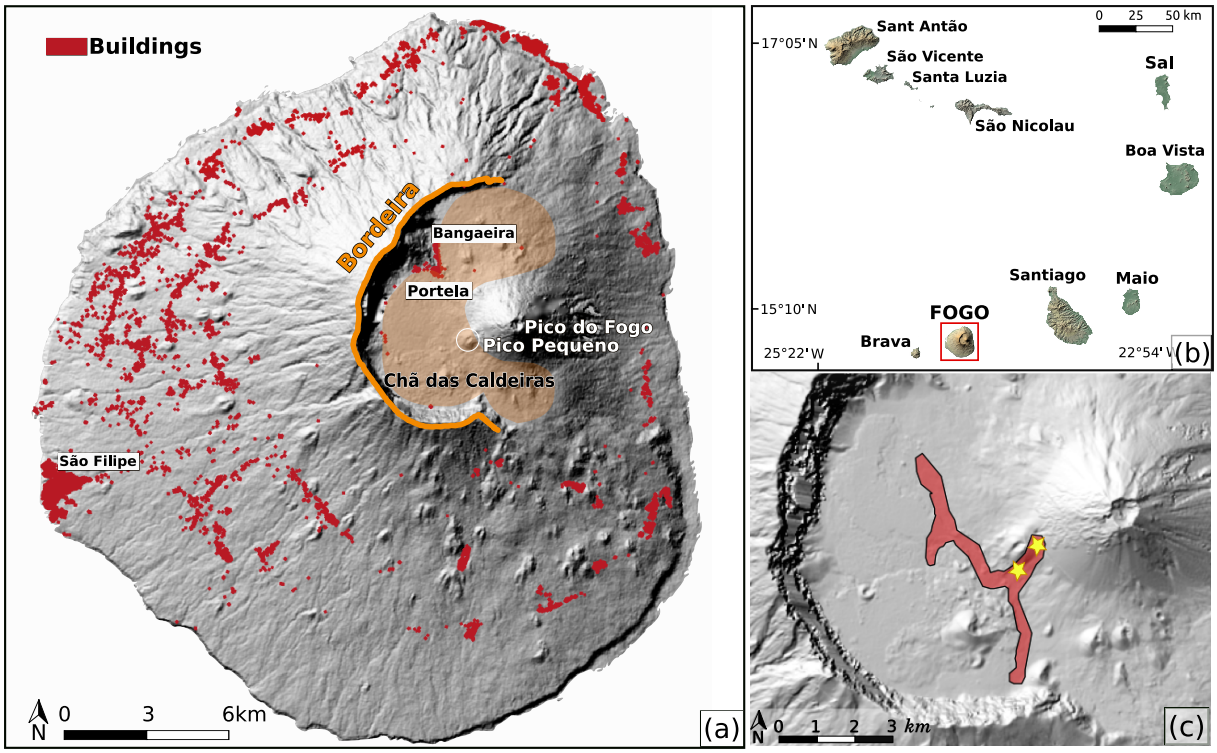


Figure 13. (a) Map of Fogo Island. (b) Archipelago of Cape Verde, of which Fogo is a part. (c) Lava emplacement of the real event after one day of the eruption, on 24 November 2014, is represented by an outline extracted from [71] defined based on field mapping and satellite images. The stars represent the extrema of the fissural vent. Images from [30].

in 2011–2013. It has a horizontal resolution of 12 m (the DEM of Fogo is a TanDEM-X WorldDEM* data [72] provided by the German Aerospace Center (DLR) through data proposal DEM_GEOL_1522, PI Nicole Richter). The original format of such a DEM file is GeoTIFF (Georeferenced Tagged Image Format File), and we converted it to the STL format, a suitable extension to OpenFOAM, using the plugin DEMto3D available for QGis [73].

As stated previously, the eruptive source was a fissure. In Richter *et al.* [12] the probabilistic code DOWNFLOW [3, 16] was used to estimate the lava flow hazard at Fogo by using a single vent, corresponding to the highest end of the fissure (DMS coordinates: 14° 56′ 40.56″ N - 24° 21′ 12.28″ W; UTM coordinates: East 784689.69 - North 1653895.03, zone 26P). Instead, in [71] it was observed that the other end of the fissure, the lowest, was actually the main source of lava. Having this discordant information, we decided to take into account both the vents, using the following position for the second, DMS coordinates: 14°56′27.15″ N - 24°21′22.96″ W; UTM coordinates: East 784375.00 - North 1653479.0, zone 26P. Reference [30] proves that using both vents for simulations of the first day of this eruption produces a better result. Therefore, we did the same and maintained their settings: both vents are circular with a radius of 20 meters.

Reference [71] used HOTSAT, a satellite thermal monitoring system, to retrieve details about the eruption. For the first day of the eruption, they registered a mean effusion rate of 10.5 $m^3\,s^{-1}$ and estimated the extrusion temperature to be 1265°C (1538 K). For our simulations, we adopted their results.

We adopted the value of 2700 $kg\,m^{-3}$ for the density ρ and 1150 $J\,kg^{-1}\,K^{-1}$ for specific heat capacity c_p; these values are typical of basaltic magma, which is a proper choice because it respects the characteristics of the magma for Fogo.

We report the whole set of thermophysical parameters adopted for our tests in Tab. 4. The values of the emissivity, atmospheric heat transfer, and thermal conductivity are typical parameters for Etna lava

*The TSX/TanDEM-X mission is for the creation of a global, consistent, and high-resolution Digital Elevation Model (DEM) obtained by exploiting the interferometric capabilities of the two twin SAR satellites TerraSAR-X and TanDEM-X, which fly in a close orbit formation. The work for the creation of this global DEM lasted from December 2010 to September 2016.

Table 4. Parameters of lava flow simulation.

Symbol	Value	Definition	Unit
ρ	2700	density of lava	$\mathrm{kg\,m^{-3}}$
c_p	1150	specific heat of lava	$\mathrm{J\,kg^{-1}\,K^{-1}}$
Pr	28 750	Prandtl number of lava	–
f	0.7	fractional area of exposed inner core	–
ϵ	0.8	emissivity of lava	$\mathrm{m^{-2}}$
λ	70	atmospheric heat transfer coefficient	$\mathrm{W\,m^{-2}\,K^{-1}}$
σ_{SB}	5.67×10^{-8}	Stefan-Boltzmann constant	$\mathrm{kg\,s^{-3}\,K^{-4}}$
T_{vent}	1538	temperature of lava at the vent	K
T_{env}	300	temperature of environment	K
k_{wall}	2.0	thermal conductivity of soil	$\mathrm{W\,m^{-1}\,K^{-1}}$
L_{wall}	0.5	thermal boundary layer in soil	m
T_{inf}	300	fixed temperature deep in the soil	K
A	-5.94	rheological parameter of VFT model	Pa s
B	5500	rheological parameter of VFT model	$\mathrm{Pa\ s\ K^{-1}}$
C	610	rheological parameter of VFT model	K
ν_{min}	3.7	rheological parameter of VFT model implementation	$\mathrm{m^2\,s{-}1}$
ν_{max}	100	rheological parameter of VFT model implementation	$\mathrm{m^2\,s{-}1}$

flows [25], and since both the lavas of Etna and Fogo are basaltic, this choice of values is consistent. Soil thermal conductivity is taken from [74].

To perform this simulation, a coarse mesh with a resolution of $\Delta x = 160$ m is created. Subsequentially, the mesh is refined starting close to topography until level 4 so that the terrain is approximated with a discretization step of the order of 10 m (no dynamic mesh refinement was adopted for this case, differently from the previous test case). With a spatial discretization of 10 m, if the lava flow thickness is less than 10 m, the volumetric fraction $\alpha = 1$ is nowhere assumed. If the lava flow is 1 m thick, for example, the value of α would be 0.1. Following such considerations, we lower the threshold of α for which we consider that lava is present in the cell to 0.005. This threshold, $\alpha = 0.005$, could also represent, for example, the case where a lava flow with a thickness of 0.5 m advances 1 m into a computational cell. Since the mesh adapts to the topography, we adjusted the PIMPLE configuration compared to the previous test: the solver performs one outer loop, two inner correction loops, and one non-orthogonal corrector.

Figure 14 shows the simulation results after 4, 12, and 24 hours since the eruption began, respectively.

The comparison between the simulation and the observed emplacement at 24 h (see Fig. 14.c) shows a good agreement in the flow extent at the southern branch; the flow propagation at the northern branch is instead considerably underestimated by the simulation. In fact, the simulation seems to produce two branches, north and south going respectively, of comparable lengths, whereas the northern branch propagates further in the real case. In addition to the flow emplacement, our simulation gives further information, allowing us to appreciate how lava motion and cooling evolved to obtain the final result.

Concerning the lava motion, we observe that in the earliest stage of the eruption (4 h since the beginning), the lava propagates from the vents towards the southwest following the topography and creates two distinct lobes at the edge of the flow propagation. From that stage on, the lava flow propagates mainly in two directions, namely northwest and southeast, kept in subsequent times. It is interesting to notice that the length of the two branches develops at the same speed, reaching 1 km in both directions 16 h after the beginning of the eruption.

Concerning the evolution of lava temperature, we observe that at the beginning of the eruption (after 4 h), the temperature recorded at the lava surface ranges from the peak at the vent (over 1500 K) to a value above 500 K reached at the edge of the flow propagation. After that, we see that the temperature at the fronts of the branches reaches a minimum of 300 K. Moreover, because of the temperature-dependent viscosity model adopted, the colder fronts present a higher viscosity and, therefore, tend to slow the flow propagation.

5. **Conclusions, comparisons and future developments**

In this paper, we presented `interThermalRadConvFoam`, a novel OpenFOAM solver specifically designed for modelling free-surface viscous fluids experiencing temperature variations due to radiative, convective, and conductive heat exchanges. Because of the temperature exchanges with the atmosphere

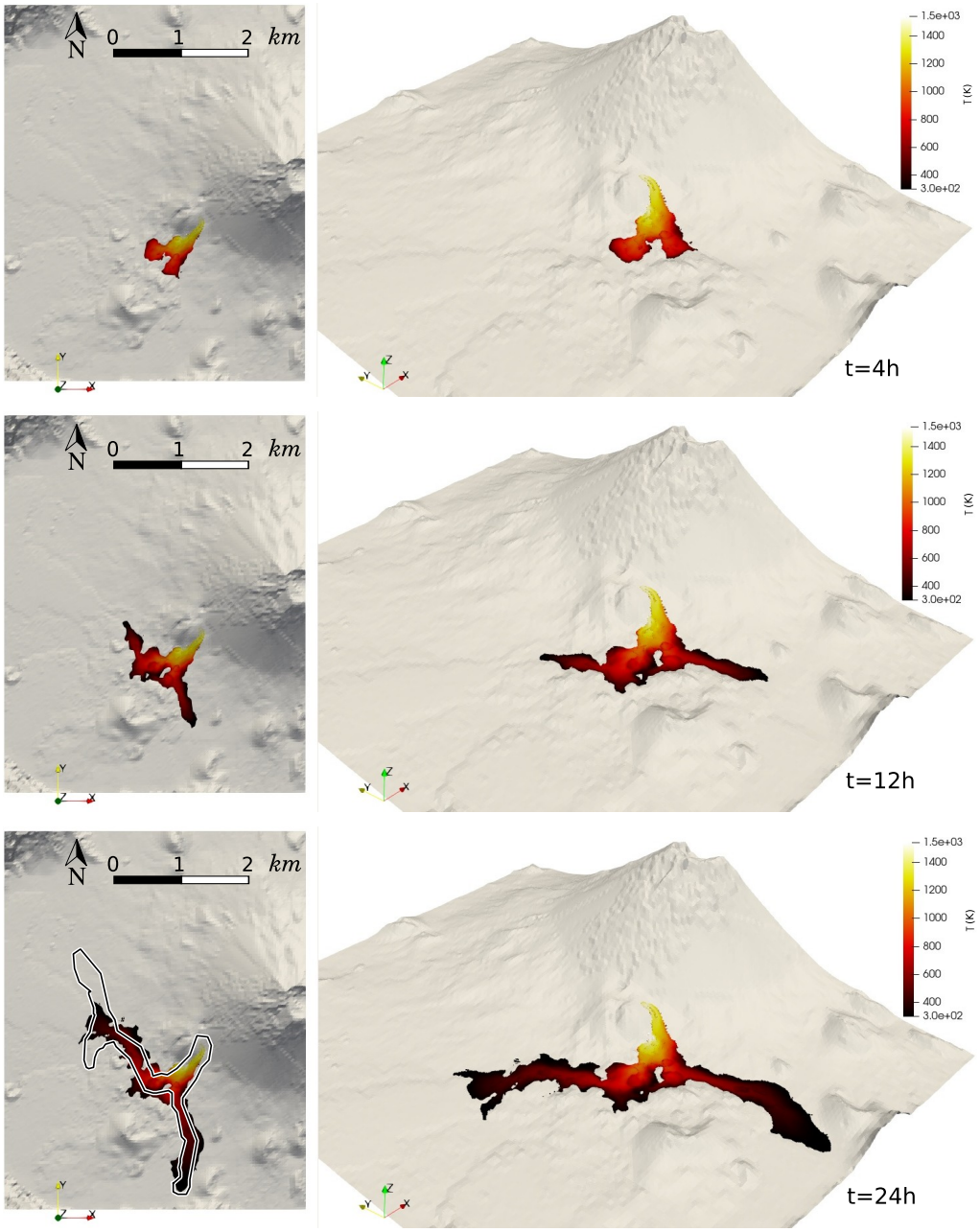


Figure 14. *Real case, Fogo eruption.* Lava flow propagation after 4, 12, and 24 h, respectively, since the beginning of the eruption. *Left:* 2D view from the top. The 2D result after 24 h is directly compared with the outline of the natural lava flow (displayed in Fig. 13c). *Right:* 3D view from south-west. These plots are obtained considering the values of α between 0.005 and 1. Temperature spans from 300 K to 1538 K.

and the need to properly model the free surface, the problem requires a 3D multiphase model, described by a system of partial differential equations for mass, momentum, energy and a transport equation for the volumetric fraction of one of the phases. To numerically solve this system of PDEs, we used the solvers and libraries available in OpenFOAM for multiphase flows and modified the solver `interFoam` by adding the desired features. `interFoam` adopts the Volume of Fluid technique, based on the Interface Capturing strategy, which is important to guarantee an accurate description of the interface between the fluid of interest and the atmosphere. With this approach, the two fluids are treated as a single fluid whose properties (such as density and viscosity) vary in space according to the volumetric fraction of each phase. The MULES method was then used to help in keeping the phase interface sharper.

As previously stated, an energy equation has been implemented in the 3D model to describe thermal energy transport throughout the fluids, together with the radiative and convective heat loss at the interface between the phases. In addition, the heat conduction with the ground was modelled as a boundary condition. Finally, a library for the dynamic mesh refinement has been adopted for some test cases, allowing us to start a simulation with a relatively coarse mesh and refine it dynamically during the simulation in regions where smaller cells are needed (for example, close to the interface between the two phases).

Some tests have been performed (a widely used benchmark of a laboratory test related to axisymmetric cooling and spreading and the simulation of a real lava flow) to look at the capability of the solver to keep a sharp interface, to examine the efficiency of the dynamic mesh refinement, and to analyze the performances of parallelization.

The benchmark test highlights that, with an appropriately fine mesh, the numerical scheme maintains an accurate description of the phase interface. Furthermore, we saw that by using the dynamic refinement, we obtained results that are perfectly comparable to those obtained with a uniform static mesh defined by the finest size of spatial discretization of the dynamic mesh, but with the critical difference of a sensible reduction in the computational time for the dynamic mesh case. However, even though this advantage on the execution time is true in most cases, this might not occur for simulations where the cells involved in the refinement occupy a significant part of the domain.

The real test results show that a sufficiently accurate numerical model can give specific information on the evolution of the fluid of interest concerning motion and cooling and their intertwining relation due to the temperature-dependent viscosity. This information is useful for enriching the available data, which often involves only the final emplacement.

The radiative and convective terms of computation depend on the surface area of the phase interface, which requires the knowledge of the surface area of each interface cell. Currently, we use a rough approximation to compute the surface area from the volume based on the assumption of an aspect ratio close to 1, as suggested by [75]. If the cell aspect ratio was significantly smaller or larger than 1, the approximation could be overestimated or underestimated, with a consequent error in the computation of the heat loss. From this fact, we think that the user should pay attention to having grid cells with an aspect ratio as close as possible to 1 to have a good description of the phase interface and proper modelling of the heat loss processes occurring at the interface between the phases. Further work should be done to validate this claim and, if confirmed, to increase the flexibility and capability of the code to capture the effects of thermal processes on the dynamics through a better estimate of the interface surface area with different strategies, independent from the aspect ratio of the cells, but based, for example, on the gradients of the volumetric phase fraction field.

Acknowledgements

The research leading to these results has received funding from UKRI FLF project 4DVOLC (MR/V023985/1). The third author is a member of the Research Group GNCS-INDAM, and his work is partially supported by the MIUR Excellence Department Project (CUP D33C23001110001). Some simulations of the BM3 tests were performed on the High-Performance Computing Super Micro cluster *laki* installed at INGV, Section of Pisa, to support the fluid-dynamic and volcanology modelling activity. The authors acknowledge Prof. Einat Lev, Lamont-Doherty Earth Observatory at Columbia University, New York, for the precious collaboration and helpful discussions. The authors thank the reviewers whose observations contributed to improving the work's presentation. TanDEM-X WorldDEM data were provided by the German Aerospace Center (DLR) through data proposal DEM_GEOL_1522, whose Principal Investigator is Nicole Richter, whom the authors thank for her gentleness and availability to collaborate.

Author Contributions: Conceptualisation, M.M.V. and E.B.; methodology, E.B. and M.M.V.; software, E.B.; validation, E.B.; formal analysis, E.B., M.M.V. and F.D.B.; investigation, E.B. and M.M.V.; resources, E.B. and M.M.V.; data curation, E.B.; writing—original draft preparation, E.B., F.D.B. and M.M.V.; writing—review and editing, E.B., F.D.B., M.M.V. and M.P.; visualisation, E.B. and M.M.V.; supervision, F.D.B., M.M.V. and M.P.; project administration, F.D.B., M.M.V. and M.P.; funding acquisition, F.D.B. and M.P. All authors have read and agreed to the published version of the manuscript.

References

[1] J. Kauahikaua, K. Cashman, T. Mattox, C. Heliker, K. Hon, K. Mangan, and C. Thornber, "Observation on basaltic lava streams in tubes from Kilauea Volcano, island of Hawai'i," *J. Geophys. Res.*, vol. 103, pp. 27,303–27,324, 1998.

[2] A. Felpeto, V. Araña, R. Ortiz, M. Astiz, and A. García, "Assessment and modelling of lava flow hazard on Lanzarote (Canary Islands)," *Nat. Hazard*, vol. 23, pp. 247–257, 2001.

[3] M. Favalli, M. Pareschi, A. Neri, and I. Isola, "Forecasting lava flow paths by a stochastic approach," *Geophysical Research Letters*, vol. 35, 2005. [Online]. Available: https://doi.org/10.1029/2004GL021718

[4] S. K. Rowland, H. Garbeil, and A. J. L. Harris, "Lengths and hazards from channel-fed lava flows on Mauna Loa, Hawaii, determined from thermal and downslope modeling with FLOWGO," *Bull. Volcanol.*, vol. 67, pp. 634–647, 2005.

[5] K. Bonne, M. Kervyn, L. Cascone, S. Njome, E. Van Ranst, E. Shu, S. Ayonghe, P. Jacobs, and G. Ernst, "A new approach to assess long-term lava flow hazard and risk using GIS and low-cost remote sensing: the case of Mount Cameroon, West Africa," *International Journal of Remote Sensing*, vol. 29, pp. 6539–6564, 2008. [Online]. Available: https://doi.org/10.1080/01431160802167873

[6] G. D. Chirico, M. Favalli, P. Papale, E. Boschi, M. T. Pareschi, and A. Mamou-Mani, "Lava flow hazard at Nyiragongo Volcano, DRC 2. Hazard reduction in urban areas," *Bull. Volcanol.*, vol. 71, pp. 375–387, 2009.

[7] M. Favalli, G. D. Chirico, P. Papale, M. T. Pareschi, and E. Boschi, "Lava flow hazard at Nyiragongo volcano, D.R.C. 1. Model calibration and hazard mapping," *Bull. Volcanol.*, vol. 71, pp. 363–374, 2009.

[8] M. Favalli, S. Tarquini, P. Papale, A. Fornaciai, and E. Boschi, "Lava flow hazard and risk at Mt. Cameroon volcano," *Bulletin of Volcanology*, vol. 74, pp. 423–439, 2012.

[9] C. Del Negro, A. Cappello, M. Neri, G. Bilotta, A. Herault, and G. Ganci, "Lava flow hazards at Mount Etna: constraints imposed by eruptive history and numerical simulations," *Sci. Rep.*, vol. 3, pp. 1–8, 2013.

[10] S. Tarquini and M. Favalli, "Uncertainties in lava flow hazard maps derived from numerical simulations: the case study of Mount Etna," *Journal of Volcanology and Geothermal Research*, vol. 260, pp. 90–102, 2013.

[11] A. Cappello, V. Zanon, C. Del Negro, T. J. L. Ferreira, and M. G. P. S. Queiroz, "Exploring lava-flow hazards at Pico Island, Azores Archipelago (Portugal)," *Terra Nov.*, vol. 27, pp. 156–161, 2015.

[12] N. Richter, M. Favalli, E. de Zeeuw-van Dalfsen, A. Fornaciai, R. M. da Silva Fernandes, N. M. Pérez, J. Levy, S. V. Silva, and T. R. Walter, "Lava flow hazard at Fogo Volcano, Cabo Verde, before and after the 2014–2015 eruption," *Nat. Hazards Earth Syst. Sci.*, vol. 16, pp. 1925–1951, 2016. [Online]. Available: https://doi.org/10.5194/nhess-16-1925-2016

[13] S. Mossoux, M. Kervyn, and F. Canters, "Assessing the impact of road segment obstruction on accessibility of critical services in case of a hazard," *Nat. Hazards Earth Syst. Sci.*, vol. 19, pp. 1251–1263, 2019.

[14] M. O. Chevrel, M. Favalli, N. Villeneuve, A. J. L. Harris, A. Fornaciai, N. Richter, A. Derrien, P. , A. Di Muro, and A. Peltier, "Lava flow hazard map of Piton de la Fournaise volcano," *Natural Hazard and Earth System Sciences*, 2021.

[15] G. B. M. Pedersen, M. A. Pfeffer, S. Barsotti, S. Tarquini, M. de' Michieli Vitturi, B. Óladóttir, and R. H. Þrasstarson, "Lava flow hazard modelling during the 2021 Fagradalsfjall eruption, Iceland: Applications of MrLavaLoba," *Natural Hazards and Earth System Sciences*, vol. 2022, pp. 1–38, 2022.

[16] S. Tarquini and M. Favalli, "Mapping and DOWNFLOW simulation of recent lava flow fields at Mount Etna," *Journal of Volcanology and Geothermal Research*, vol. 204, pp. 27–39, 2011. [Online]. Available: https://doi.org/10.1016/j.jvolgeores.2011.05.001

[17] M. L. Damiani, G. Groppelli, G. Norini, E. Bertino, A. Gigliuto, and S. Nucita, "A lava flow simulation model for the development of volcanic hazard maps for Mount Etna (Italy)," *Computers & Geosciences*, vol. 32, pp. 512–526, 2006. [Online]. Available: https://doi.org/10.1016/j.cageo.2005.08.011

[18] A. Felpeto, J. Marti, and R. Ortiz, "Automatic GIS-based system for volcanic hazard assessment," *J Volcanol Geotherm Res*, vol. 166, pp. 106–116, 2007. [Online]. Available: https://doi.org/10.1016/j.jvolgeores.2007.07.008

[19] M. de' Michieli Vitturi and S. Tarquini, "MrLavaLoba: A new probabilistic model for the simulation of lava flows as a settling process," *EGU General Assembly Conference Abstracts*, vol. 21, 2021. [Online]. Available: https://doi.org/10.1016/j.jvolgeores.2017.11.016

[20] A. Vicari, H. Alexis, C. Del Negro, M. Coltelli, M. Marsella, and C. Proietti, "Modeling of the 2001 lava flow at Etna volcano by a Cellular Automata approach," *Environmental Modelling & Software*, vol. 22, no. 10, pp. 1465–1471, oct 2007. [Online]. Available: https://doi.org/10.1016/j.envsoft.2006.10.005

[21] G. M. Crisci, S. Di Gregorio, O. Pindaro, and G. A. Ranieri, "Lava flow simulation by a discrete cellular model: first implementation." *International Journal of Modelling & Simulation*, vol. 6, pp. 137–140, 1986. [Online]. Available: https://doi.org/10.1080/02286203.1986.11759975

[22] P. Young and G. Wadge, "FLOWFRONT: Simulation of a lava flow," *Computers & Geosciences*, vol. 16, no. 8, pp. 1171–1191, jan 1990. [Online]. Available: https://doi.org/10.1016/0098-3004(90)90055-X

[23] L. J. Connor, C. B. Connor, K. Meliksetian, and I. Savov, "Probabilistic approach to modeling lava flow inundation: a lava flow hazard assessment for a nuclear facility in Armenia," *J. Appl. Volcanol.*, vol. 1, no. 1, p. 3, 2012. [Online]. Available: https://doi.org/10.1186/2191-5040-1-3

[24] K. Kelfoun and S. V. Vargas, "VolcFlow capabilities and potential development for the simulation of lava flows," *Geological Society, London, Special Publications*, vol. 426, no. 1, pp. 337–343, 2016. [Online]. Available: https://doi.org/10.1144/sp426.8

[25] A. Costa and G. Macedonio, "Numerical simulation of lava flows based on depth-averaged equations," *Geophysical Research Letters*, vol. 32, no. 5, 2005. [Online]. Available: https://doi.org/10.1029/2004gl021817

[26] N. Bernabeu, P. Saramito, and C. Smutek, "Modelling lava flow advance using a shallow-depth approximation for three-dimensional cooling of viscoplastic flows," *Geological Society, London, Special Publications*, vol. 426, no. 1, pp. 409–423, 2016. [Online]. Available: https://doi.org/10.1144/SP426.27

[27] C. J. Conroy and E. Lev, "A discontinuous Galerkin finite–element model for fast channelized lava flows v1.0," *Geoscientific Model Development*, vol. 14, pp. 3553–3575, 2021. [Online]. Available: https://doi.org/10.5194/gmd-14-3553-2021

[28] D. M. R. Hyman, H. R. Dietterich, and M. R. Patrick, "Toward Next–Generation Lava Flow Forecasting: Development of a Fast, Physics–Based Lava Propagation Model," *JGR: Solid Earth*, vol. 127, p. e2022JB024998, 2022. [Online]. Available: https://doi.org/10.1029/2022JB024998

[29] B. D. Michel, B. Piar, J. C. Babik, F. Latche, G. Guillard, and C. D. Pascale, "c," *Proceedings of the OECD Workshop on Ex-Vessel Debris Coolability*, vol. 6475, pp. 235–245, 2000.

[30] E. Biagioli, M. de' Michieli Vitturi, F. Di Benedetto, and M. Polacci, "Benchmarking a new 2.5D shallow water model for lava flows," *Journal of Volcanology and Geothermal Research*, vol. 444, 2023. [Online]. Available: https://doi.org/10.1016/j.jvolgeores.2023.107935

[31] E. Fujita and M. Nagai, "LavaSIM: its physical base and applicability," *Geological Society, London, Special Publications*, vol. 426, 2015.

[32] V. Zago, G. Bilotta, A. Cappello, R. A. Dalrymple, L. Fortuna, G. Ganci, A. Herault, and C. D. Negro, "Preliminary validation of lava benchmark tests on the GPUSPH particle engine," *Annals of Geophysics*, vol. 62, 2019. [Online]. Available: https://doi.org/10.4401/ag-7870

[33] A. Korotkii, D. Kovtunov, A. Ismail-Zadeh, I. Tsepelev, and O. Melnik, "Quantitative reconstruction of thermal and dynamic characteristics of lava flow from surface thermal measurements," *Geophysical Journal International*, vol. 205, no. 3, pp. 1767–1779, 2016.

[34] H. R. Dietterich, E. Lev, J. Chen, J. A. Richardson, and K. V. Cashman, "Benchmarking computational fluid dynamics models of lava flow simulation for hazard assessment, forecasting, and risk management," *Journal of Applied Volcanology*, vol. 6, no. 1, p. 9, 2017.

[35] V. Cataldo, D. Williams, M. Schmeeckle, and G. Leone, "Vallis schröteri, moon: Results of first 3-d model of thermal erosion by turbulent lava reveal how erosion likely shaped the inner rille," in *50th Annual Lunar and Planetary Science Conference*, no. 2132, 2019, p. 2297.

[36] S. Lee and K. Hwang, "Numerical simulations of lava flow using multiphase and non-Newtonian fluid model," *J. Korean Soc. Hazard Mitig.*, vol. 17, no. 2, pp. 117–125, 2017.

[37] K. M. Park, H. S. Yoon, and M. I. Kim, "CFD-DEM based numerical simulation of liquid-gas-particle mixture flow in dam break," *Communications in Nonlinear Science and Numerical Simulation*, vol. 59, pp. 105–121, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1007570417303830

[38] S. S. Deshpande, L. Anumolu, and M. F. Trujillo, "Evaluating the performance of the two-phase flow solver interFoam," *Computational Science & Discovery*, vol. 5, 2012. [Online]. Available: https://doi.org/10.1088/1749-4699/5/1/014016

[39] T. Holzmann, "Mathematics, Numerics, Derivations and OpenFOAM®," 2019. [Online]. Available: https://holzmann-cfd.de

[40] C. W. Hirt and B. D. Nicholls, "Volume of fluid (VOF) method for dynamics of free boundaries," *J. Comput. Phys.*, vol. 39, pp. 201–221, 1981. [Online]. Available: https://doi.org/10.1016/0021-9991(81)90145-5

[41] S. Márquez Damián, "An extended mixture model for the simultaneous treatment of short and long scale interfaces," Ph.D. dissertation, Universidad Nacional del Litoral, 2013. [Online]. Available: https://doi.org/10.1002/fld.3906

[42] J. P. Boris and D. L. Book, "Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works," *Journal of Computational Physics*, vol. 11, pp. 38–69, 1973. [Online]. Available: https://doi.org/10.1016/0021-9991(73)90147-2

[43] B. Cordonnier, E. Lev, and F. Garel, "Benchmarking lava-flow models," *Geological Society, London, Special Publications*, vol. 426, pp. 425–445, 2015. [Online]. Available: https://doi.org/10.1144/SP426.7

[44] T. Hino, "Computation of viscous flows with free surface around an advancing ship," in *Proc. 2nd Osaka Int. Colloquium on Viscous Fluid Dynamics In Ship and Ocean Technology.* Osaka Univ., 1992.

[45] G. D. Raithby, W. X. Xu, and G. D. Stubley, "Prediction of incompressible free surface flows with an element-based finite volume method," *Comput. Fluid Dynamics J.*, vol. 4, pp. 353–371, 1995.

[46] J. L. Thé, G. D. Raithby, and G. D. Stubley, "Surface-adaptive finite-volume method for solving free-surface flows," *Numer. Heat Transfer*, vol. 26, pp. 367–380, 1994. [Online]. Available: https://doi.org/10.1080/10407799408914935

[47] Z. Lilek, "Ein Finite-Volumen Verfahren zur Berechnung von inkompressiblen und kompressiblen Stromungen in komplexen Geometrien mit beweglichen Randern und freien OberflEhen," in *Dissertation.* University of Hamburg, Germany, 1995.

[48] T. Kawamura and H. Miyata, "Simulation of nonlinear ship flows by density-function method," *J. Soc. Naval Architects Japan*, vol. 176, pp. 1–10, 1994.

[49] F. H. Harlow and J. E. Welsh, "Numerical calculation of time dependent viscous incompressible flow with free surface," *Phys. Fluids*, vol. 8, pp. 2182–2189, 1965. [Online]. Available: https://doi.org/10.1063/1.1761178

[50] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *J . Comput. Phys.*, vol. 79, pp. 12–49, 1988.

[51] C. Min, "On reinitializing level set functions," *Journal of Computational Physics*, pp. 2764–2772, 2010. [Online]. Available: https://doi.org/10.1016/j.jcp.2009.12.032

[52] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modeling surface tension," *Journal of computational physics*, vol. 100, pp. 335–354, 1992. [Online]. Available: https://doi.org/10.1016/0021-9991(92)90240-Y

[53] I. Avramov, "Pressure and temperature dependence of viscosity of glassforming and of geoscientifically relevant system," *J. Volcanol. Geotherm. Res.*, vol. 160, pp. 165–174, 2007.

[54] D. Giordano, J. K. Russell, and D. B. Dingwell, "Viscosity of magmatic liquids: A model," *Earth and Planetary Science Letters*, vol. 271, pp. 123–134, 2008. [Online]. Available: https://doi.org/10.1016/j.epsl.2008.03.038

[55] J. Stefan, "Über die Beziehung zwischen der Wärmestrahlung und der Temperatur," *Sitzungsberichte der Mathematisch-naturwissenschaftlichen Classe der Kaiserlichen Akademie der Wissenschaften*, vol. 79, pp. 391–428, 1879.

[56] L. Boltzmann, "Ableitung des Stefan'schen Gesetzes, betreffend die Abhängigkeit der Wärmestrahlung von der Temperatur aus der electromagnetischen Lichttheorie," *Annalen der Physik und Chemie*, vol. 258, no. 6, pp. 291–294, 1884.

[57] J. Crisp and S. Baloga, "A model for lava flows with two thermal components," *J. Geophys. Res.*, vol. 95, pp. 1255–1270, 1990.

[58] B. Eltard Larsen, D. R. Fuhrman, and J. Roenby, "Performance of interFoam on the simulation of progressive waves," *Coastal Engineering Journal*, vol. 61, no. 3, pp. 380–400, 2019.

[59] R. S. J. Sparks and H. E. Huppert, "Density changes during the fractional crystallization of basaltic magmas: fluid dynamic implications," *Contributions to Mineralogy and Petrology*, vol. 85, pp. 300–309, 1984.

[60] A. Drozd-Rzoska and S. J. Rzoska, "Consistency of the Vogel — Fulcher — Tammann (VFT) Equations For The Temperature-, Pressure-, Volume-and Density- Related Evolutions of Dynamic Properties in Supercooled and

Superpressed Glass Forming Liquids/Systems"," in *Metastable Systems under Pressure*, 2010. [Online]. Available: https://doi.org/10.1007/978-90-481-3408-3_7

[61] H. Jasak, "Error analysis and estimation for the finite volume method with applications to fluid flows," Ph.D. dissertation, Imperial College, University of London, 1996.

[62] F. Garel, E. Kaminski, S. Tait, and A. Limare, "An experimental study of the surface thermal signature of hot subaerial isoviscous gravity currents: Implications for thermal monitoring of lava flows and domes," *Journal of Geophysical Research: Solid Earth*, vol. 117, 2012. [Online]. Available: https://doi.org/10.1029/2011JB008698

[63] R. Courant, K. O. Friedrichs, and H. Lewy, "Über die partiellen differenzengleichungen der mathematischen physik," *Math. Ann.*, vol. 100, pp. 32–74, 1928.

[64] ——, "On the partial difference equations of mathematical physics," *IBM J.*, vol. 11, pp. 215—-234, 1967.

[65] R. J. LeVeque, *Finite volume methods for hyperbolic problems*. Cambridge Texts in Applied Mathematics, 2002.

[66] H. E. Huppert, "The propagation of two–dimensional and axisymmetric viscous gravity currents over a rigid horizontal surface," *Journal of Fluid Mechanics*, vol. 121, pp. 43–58, 1982.

[67] S. Keough, "Optimising the parallelisation of OpenFOAM simulations," 2014.

[68] S. J. Day, S. I. N. Heleno da Silva, and J. F. B. D. Fonseca, "A past giant lateral collapse and present-day flank instability of Fogo, Cape Verde Islands," *J. Volcanol. Geotherm. Res.*, vol. 94, pp. 191–218, 1999. [Online]. Available: https://doi.org/10.1016/S0377-0273(99)00103-1

[69] R. C. Courtney and R. S. White, "Anomalous heat flow and geoid across the Cape Verde Rise: Evidence for dynamic support from a thermal plume in the mantle," *Geophys. J. R. Astron. Soc.*, vol. 87, pp. 815–867, 1986. [Online]. Available: https://doi.org/10.1111/j.1365-246X.1986.tb01973.x

[70] S. M. Dionis, N. M. Pérez, P. A. Hernández, G. Melián, F. Rodríguez, E. Padrón, H. Sumino, J. Barrrancos, G. D. Padilla, P. Fernandes, Z. Bandomo, S. V. Silva, J. M. Pereira, H. Semedo, and J. Cabral, "Diffuse CO_2 degassing and volcanic activity at Cape Verde islands, West Africa," *Earth Planets Space*, vol. 67, 2015. [Online]. Available: https://doi.org/10.1186/s40623-015-0219-x

[71] A. Cappello, G. Ganci, S. Calvari, N. M. Pérez, P. A. Hernández, S. V. Silva, and J. C. D. N. Cabral, "Lava flow hazard modeling during the 2014–2015 Fogo eruption, Cape Verde," *J. Geophys. Res. Solid Earth*, vol. 121, pp. 2290–2303, 2016. [Online]. Available: https://doi.org/10.1002/2015JB012666

[72] P. Rizzoli, M. Martone, C. Gonzalez, C. Wecklich, D. Borla Tridon, B. Bräutigam, M. Bachmann, D. Schulze, T. Fritz, M. Huber, B. Wessel, G. Krieger, M. Zink, and A. Moreira, "Generation and performance assessment of the global TanDEM-X digital elevation model," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 132, pp. 119–139, 2017. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2017.08.008

[73] M. A. Ureña Cámara, F. J. Venceslá Simón, and F. J. Ariza-López, "Demto3d, the new tool that joins gis and 3d printing," *REVISTA INTERNACIONAL MAPPING*, vol. 24, no. 170, p. 24–28, Sep. 2018. [Online]. Available: https://ojs.revistamapping.com/MAPPING/article/view/59

[74] J. Darkwa, G. Suba, and G. Kokkogiannakis, "An investigation into the thermophysical properties and energy dynamics of an intensive green roof," *Journal of Heat and Mass Transfer*, vol. 7, pp. 65–84, 2013.

[75] S. Almeland, "Implementation of an air-entrainment model in interFoam," *Proceedings of CFD with OpenSource Software, 2018*, 2018. [Online]. Available: https://doi.org/10.17196/OS_CFD#YEAR_2018