# solidificationMeltingSource: A Built-in fvOption in OpenFOAM® for Simulating Isothermal Solidification

Mahdi Torabi Rad

**Abstract** In this chapter, I introduce, document, and verify solidificationMeltingSource: a built-in fvOption in OpenFOAM® for simulating isothermal solidification. The main challenge in simulating isothermal solidification is the incorporation of movements of the solidification front. To overcome this challenge, solidificationMeltingSource adds source terms to the momentum and energy equations. First, I rigorously derive the equations for these source terms and outline their implementation in the source code. Then, I verify solidificationMeltingSource by simulating a well-known numerical benchmark for isothermal solidification. Finally, I end the chapter by suggesting possible future extensions for solidificationMeltingSource.

**Keywords** Phase change · fvOption · Isothermal solidification solidificationMeltingSource

## 1 Introduction

### 1.1 fvOptions

fvOptions is a flexible framework in OpenFOAM® that allows users to add source terms to equations at run time [1]. fvOptions is easy to use since users do not have to modify source code. Some of the implemented fvOptions are: (1) explicit porosity source; to simulate flow in a domain with porous subzones; (2) MRF source: to simulate flow on both stationary and rotating frames; (3) fixed temperature constraint: to fix temperature at given locations to constant or time-varying values. The reader is referred to reference [1] for a complete list of available fvOptions and example syntaxes on how to use them.

solidificationMeltingSource [2] is a fvOption that can be used to simulate isothermal phase-change (solidification and melting) problems such as windshield defrost-

M. Torabi Rad (✉)
Seamans Center, 3131, Iowa City, IA 52242, USA
e-mail: mahdi-torabirad@uiowa.edu

ing or solidification of phase-change materials (PCM). solidificationMeltingSource adds source terms to: (1) the momentum equation, to account for the drag force due to the presence of a solid in a liquid, and (2) the energy equation to account for latent heat release during the phase change.

## 1.2 Background on Isothermal Solidification

Pure materials solidify at a single temperature, whereas alloys solidify within a range of temperatures. This chapter focuses only on solidification of pure materials: isothermal solidification. Figure 1 shows a schematic of a system under isothermal solidification. Cooling is from the left. The domain consists of a solid region on the left, with $T < T_f$, where $T_f$ is the melting temperature, and a liquid region, with $T > T_f$, on the right. The two regions are separated by the sharp and moving solidification front located at $x = x^*$. For solidification to continue, latent heat released at the solidification front has to be dissipated by net conduction away from the solidification front. Heat balance at the solidification front implies

$$-\rho L_f \frac{\mathrm{d}x^*}{\mathrm{d}t} = \left(-k_s \frac{\partial T}{\partial x}\right)^* - \left(-k_l \frac{\partial T}{\partial x}\right)^*. \tag{1}$$

In solidification literature, Eq. (1) is typically referred to as the classical Stefan condition [3]. The term on the left-hand side represents the latent heat released at $x = x^*$ (due to solidification). On the right-hand side, the first and second terms are the heat fluxes through the solid and liquid, respectively. The superscript * indicates that the fluxes are evaluated at $x = x^*$.

The main challenge in simulating isothermal solidification problems is to incorporate movements of the solidification front. Numerical methods for overcoming this challenge can be categorized as: (1) moving boundary methods, and (2) enthalpy methods. In the moving boundary methods, the location of the solidification front is tracked *explicitly*, and two different energy equations are solved, the first one for the solid side and the second one for the liquid side. In the enthalpy methods, there is no need for explicit tracking of the solidification front. Instead, the energy equation is written in a form that is valid on both (solid and liquid) sides of the solidification front. The latter methods are numerically easier to implement. However, one needs to modify the local single-phase equations so that they are valid on both sides of the solidification front. solidificationMeltingSource uses an enthalpy method, which is discussed in this chapter.

The objective of the present chapter is to introduce, document, and verify solidificationMeltingSource. Equations implemented in solidificationMeltingSource are rigorously derived and their implementation in the source code is outlined. solidificationMeltingSource is verified through the simulation of a numerical benchmark for isothermal solidification and the results are compared with data available in the literature.

**Fig. 1** Schematic of a system under isothermal solidification. The red, solid blue, and dashed blue lines show solid fraction, temperature, and enthalpy, respectively. The solid region on the left, with $T < T_f$ and $g_l = 0$, is separated from the liquid region on the right, with $T > T_f$ and $g_l = 1$, by the solidification front (dashed line). Heat fluxes are represented by arrows
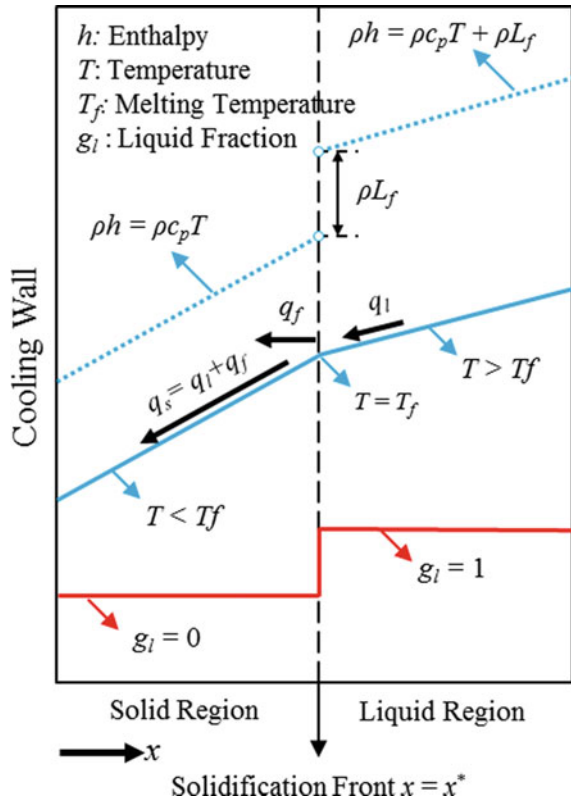


## 2 Governing Equations

Equations governing isothermal solidification are listed below. These equations are taken from Voller and Prakesh [4].

### 2.1 Conservation Equations

#### 2.1.1 Continuity

$$\nabla \cdot v = 0, \tag{2}$$

where $v$ is the flow velocity.

### 2.1.2 Momentum

$$\rho_0 \frac{\partial \boldsymbol{v}}{\partial t} + \rho_0 \nabla \cdot (\boldsymbol{v}\boldsymbol{v}) = -\nabla p + \mu_0 \nabla^2 \boldsymbol{v} + \boldsymbol{S}_b + \boldsymbol{S}_d, \tag{3}$$

where $\rho_0$ and $\mu_0$ are the density and the dynamic viscosity (both taken to be constant here), $\boldsymbol{S}_b$ is the buoyancy source term, and $\boldsymbol{S}_d$ is a source term added to force the velocities in the solid region to zero. These source terms will be derived in the next subsection.

### 2.1.3 Energy

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p \nabla \cdot (\boldsymbol{v}T) = \nabla \cdot (k\nabla T) - S_h, \tag{4}$$

where $c_p$ is the specific heat and $k$ is the thermal conductivity, and $S_h$ is a source term that accounts for the latent heat release during solidification. The equation for this source term will be also derived in the next section.

## 2.2 Derivation of the Equations for Source Terms

The momentum and energy equations, listed in the previous subsection, have source terms $\boldsymbol{S}_b$, $\boldsymbol{S}_d$, and $S_h$. The equations for these source terms are derived below.

### 2.2.1 Momentum Source Terms: $\boldsymbol{S}_b$ and $\boldsymbol{S}_d$

Equation (3) has two source terms: $\boldsymbol{S}_b$ and $\boldsymbol{S}_d$. The first one is the buoyancy source term, which is given by the Boussinesq approximation:

$$\boldsymbol{S}_b = \rho_0 \mathbf{g} \beta_T (T - T_{\text{ref}}), \tag{5}$$

where $\mathbf{g} = -9.8\hat{\mathbf{j}}$ is the acceleration due to gravity, $\beta_T$ is the thermal expansion coefficient, and $T_{\text{ref}}$ is the reference temperature.

The second source term in Eq. (3) is added to make sure that the velocities in the solid region will become zero. An acceptable relation for $\boldsymbol{S}_d$ should satisfy two conditions: (1) in the liquid region, $\boldsymbol{S}_d$ should be zero so that Eq. (3) reduces to the normal single-phase Navier–Stokes equations, and (2) in the solid region, $\boldsymbol{S}_d$ should dominate all other terms in the momentum equation and should result in $\boldsymbol{v} = 0$. Motivated by the Darcy law for flow through porous media, Voller and Prakesh [4] have suggested

$$S_d = A\boldsymbol{v}, \tag{6}$$

where $A$ is inversely proportional to permeability $K$:

$$A \sim \frac{1}{K}. \tag{7}$$

Permeability $K$ is related to liquid fraction $g_l$ through the well-known Carman–Koseny equation [5]:

$$K \sim \frac{g_l^3}{(1 - g_l)^2}. \tag{8}$$

The reader should note that, for isothermal solidification, permeability has no physical significance; it is introduced here only as a numerical technique to force the velocities in the solid region to zero. In fact, Voller and Prakesh [4] stated that $A \sim 1/K$, in Eq. (7), is not the only correct relation. Any relation that gives $A = 0$ in the liquid region and a relatively high value for $A$ in the solid region will be equally correct.

Now, after substituting Eqs. (7) and (8) into Eq. (5), we get the final equation for $S_d$ as

$$S_d = -C\frac{(1 - g_l)^2}{g_l^3}\boldsymbol{v}. \tag{9}$$

It is apparent from this equation that in the liquid region (where $g_l = 1$), $S_d$ will be zero; in the solid region, where $g_l$ is a sufficiently small number (on the order of $10^{-6}$), $S_d$ will be high enough to dominate all other terms in the momentum equation. Therefore, the momentum equation will reduce to $S_d = 0$, and, consequently, $\boldsymbol{v} = 0$.

### 2.2.2 Energy Source Term: $S_h$

The source term in the energy equation, $S_h$, represents the latent heat released during solidification. The equation for $S_h$ is derived below.

For a system with phase change, the energy equation reads as

$$\frac{\partial}{\partial t}(\rho h_{\text{tot}}) + \nabla \cdot (\rho \boldsymbol{v} h_{\text{tot}}) = \nabla \cdot (k \nabla T), \tag{10}$$

where $h_{\text{tot}}$ is the total enthalpy. In the solid region, $h_{\text{tot}} = c_p T$; in the liquid region, $h_{\text{tot}} = c_p T + L_f$. To have a single relation valid in both solid and liquid sides, we write

$$h_{\text{tot}} = c_p T + g_l L_f, \tag{11}$$

where $g_l$ is the liquid fraction equal to zero in the solid region and equal to one in the liquid region.

Now, if we substitute Eq. (11) into (10) and subtract Eq. (4), we get

$$S_h = \rho_0 L_f \left[ \frac{\partial g_l}{\partial t} + \nabla \cdot (\boldsymbol{v} g_l) \right], \tag{12}$$

which is the final equation for the energy source term $S_h$.

Finally, we need a relation that will transform $g_l = 1$ into $g_l = 0$ as temperatures become lower than the melting temperature, i.e., $T < T_f$. Discretizing the energy equation explicitly, and taking the temperature derivative of both sides (i.e., $\partial/\partial T$), will give us

$$\frac{\partial g_l}{\partial T} = -\frac{c_p}{L_f}. \tag{13}$$

From Eq. (13), we can write

$$g_l^{n+1} = g_l^n + \frac{\gamma c_p}{L_f} (T - T_f)$$
$$g_l^{n+1} = \max \left[ 0, \min \left( 1, g_l^{n+1} \right) \right], \tag{14}$$

where the superscripts $n + 1$ and $n$ refer to the present and last iteration levels, respectively, and $\gamma$ is an under-relaxation factor (of order 1). Equation (14) updates the liquid fractions as follows: initially, we have $g_l^n = 1$ and $T > T_f$; therefore, the first line in Eq. (14) will give us $g_l^{n+1} > 1$, which, in the second line, will be reset back to $g_l^{n+1} = 1$. For temperatures slightly lower than $T_f$, $\gamma c_p (T - T_f)/L_f$ will be negative, and therefore Eq. (14) will result in $g_l^{n+1} < 1$: the liquid fraction will start to decrease. The liquid fraction will keep decreasing until it reaches zero, and after that, it will no longer change.

## 3   Implementation in solidificationMeltingSource

Listings (1, 2, and 3) are code pieces from the solidificationMeltingSource source code, in which implementation of the source terms outlined in the previous section is shown. In listing (1), lines 337 and 338 are the implementations of Eq. (9) and Eq. (5), respectively. In listing (2), line 54 is the implementation of Eq. (12). In listing (3), line 216 is the implementation of Eq. (14).

```
337              scalar S = -Cu_*sqr(1.0 - alpha1c)/(pow3(alpha1c) + q_);
338              vector Sb = rhoRef_*g*beta_*deltaT_[i];
339
340          Sp[cellI] += Vc*S;
341          Su[cellI] += Vc*Sb;
```
Listing (1): Code piece showing how momentum source terms are added.

```
50        if (eqn.psi().dimensions() == dimTemperature)
51        {
52            // isothermal phase change - only include time derivative
53            // eqn -= L/Cp*(fvc::ddt(rho, alpha1_) + fvc::div(phi, alpha1_));
54            eqn -= L/Cp*(fvc::ddt(rho, alpha1_));
55        }
```

Listing (2): Code piece showing how an energy source term is added.

```
214           scalar Tc = T[cellI];
215           scalar Cpc = Cp[cellI];
216           scalar alpha1New = alpha1_[cellI] + relax_*Cpc*(Tc - Tmelt_)/L_;
```

Listing (3): Code piece showing how a liquid fraction, represented by alpha in the code, is updated.

## 4  Problem Statement and Simulation Setup

The isothermal solidification problem studied here is the benchmark introduced in Voller and Prakesh [4] and is sketched in Fig. (2). It consists of a square cavity initially filled with liquid above its freezing temperature. The cavity cools from the left wall, while the top and bottom walls are thermally isolated. Solidification starts from the left, with the solidification front moving to the right as time proceeds.

The OpenFOAM® solver that is used is buoyantBoussinesqPimpleFoam. The mesh size and time step are 2.5 cm and 1 s, respectively. In the fvOptions file, we have $T_{melt} = 0$, $L = 5$, thermoMode = lookup, beta = 0, and rhoRef = 1. Note that the value of beta in the fvOptions file is set to zero since buoyant Boussinesq Pimple Foam already accounts for the buoyancy source term. The properties and boundary conditions are taken from Voller and Prakesh [4] and, due to space limitations, are not listed here.



**Fig. 2** Schematic of the benchmark problem introduced in Voller and Prakesh [4] and simulated here
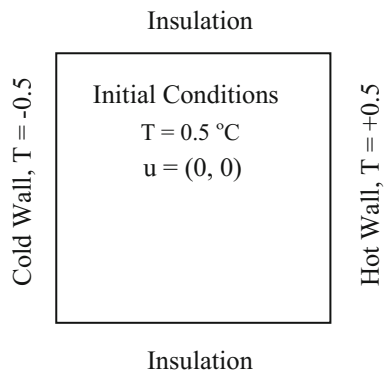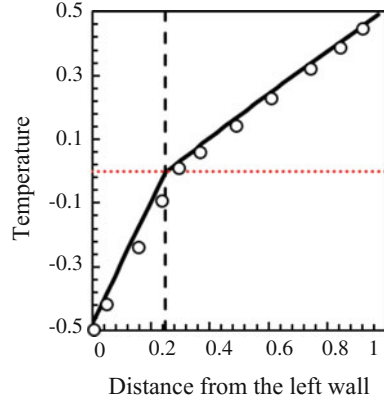
**Fig. 3** Temperature
distributions at $t = 500$ s
calculated in the present
study (solid line) and
reported in Voller et al. [6].
The dashed line is the
solidification front and the
red dotted line shows the
melting temperature ($T_f = 0$)



## 5 Results

In the absence of flow, the results are compared with data in Voller et al. [6]. Figure (3) shows a comparison of temperature profiles reported in Voller et al. [6] (markers), with the temperatures calculated in the present study (solid line at $t = 500$ s). The dashed line is the solidification front and the dotted line shows the melting temperature ($T_f = 0$). The two temperatures are in good agreement, which verifies solidificationMeltingSource.

In the presence of flow, results are shown in Fig. (4). This figure shows the evolution of the solidification over time. Columns represent different times. In the top row, color represents the solid fraction $g_s(= 1 - g_l)$ and the vectors represent the liquid velocity. In the bottom row, color represents temperature, and the white line, which is the solidification front (isoline $g_s = 0.5$), is superimposed from the solid fraction contours at the top. One can easily notice that this line lies on the freezing temperature, i.e., $T = 0$. Furthermore, regions with $T > 0$ in the bottom row have $g_s = 0$ in the top row; these regions are fully liquid. Similarly, regions with $T < 0$ in the bottom row have $g_s = 1$ in the top row; these regions are fully solid. These agreements verify solidificationMeltingSource in the presence of flow.

## 6 Conclusions

In this chapter, I introduced, documented and verified solidificationMeltingSource, which is a built-in fvOption in OpenFOAM® for simulating isothermal solidification problems. The main challenge in simulating these problems is the incorporation of movements of the solidification front. To account for these movements, solidificationMeltingSource adds source terms to the momentum and energy equations. I rigorously derived the equations for these source terms and outlined their implemen-
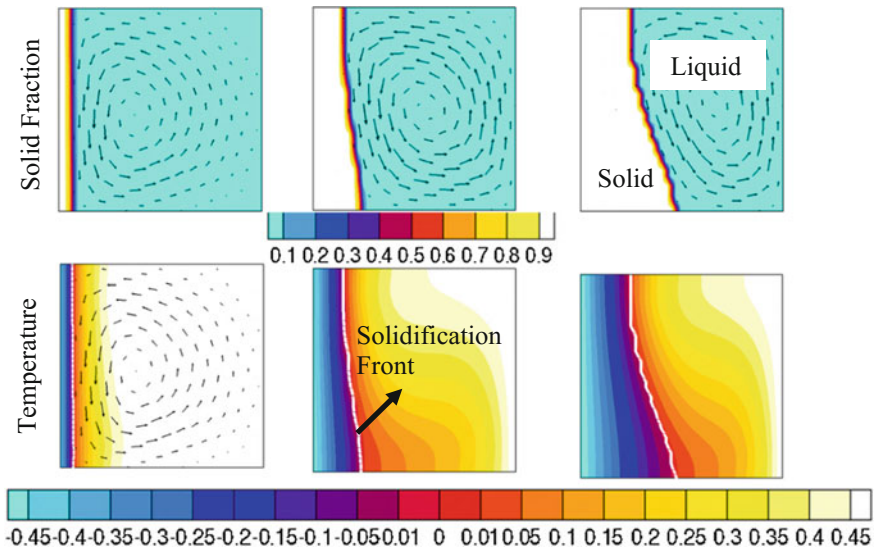
**Fig. 4** Time evolution of the solid fraction (top row) and temperature (bottom row) during solidi-fication. Left, middle, and right columns show different times. The vectors in the top row and the white line in the bottom row represent liquid velocity and the solidification front (isoline $g_s = 0.5$), respectively

tation in the source code. I simulated a benchmark for isothermal solidification and compared the results with the data in the literature. The agreement was found to be good, and therefore solidificationMeltingSource is verified.

Clearly, the solidification problem investigated here was a highly simplified one. This problem was chosen because realistic and more interesting solidification problems, such as the solidification of alloys, can only be represented by multiphase/-scale mathematical models [7], which need to be handled through complex numerical techniques [7–10] and cannot be numerically solved solely by adding source terms to the equations. The material presented in this chapter should be viewed as a starting guide for FOAMers interested in solidification problems and, more importantly, as an attempt to help with future efforts to extend OpenFOAM®'s built-in capabilities in simulating solidification problems. As an example of these extensions, implementation of algebraic lever and Scheil solidification models [3] in solidificationMeltingSource is suggested.

# References

1. http://openfoam.org/release/2-2-0/fv-options/, OpenFOAM®, 2016. Accessed 15 Oct 2016
2. https://github.com/OpenFOAM/OpenFOAM-dev/blob/master/src/fvOptions/sources/derived/solidificationMeltingSource/solidificationMeltingSource Accessed 15 Oct 2016

3. Dantzig J A, Rappaz M (2009) Solidification. CRC Press, Lausanne
4. Voller VR, Prakash C (1987) A fixed grid numerical modeling methodology for convection-diffusion mushy region phase-change problems. International Journal of Heat and Mass Transfer 30:1709–1719
5. Torabi Rad M, Kotas P, Beckermann C (2013) Rayleigh Number Criterion for Formation of A-Segregates in Steel Castings and Ingots. Metall. Mater. Trans. A 44:4266–4281
6. Voller VR, Cross M, Markatos NC (1987) An enthalpy method for convection/diffusion phase change. International Journal of Numerical Methods in Engineering 24:271–284
7. Torabi Rad M, Beckermann C (2016) Validation of a Model for the Columnar to Equiaxed Transition with Melt Convection. In: Nastac L et al. (eds) CFD Modeling and Simulation in Materials Processing 2016, TMS, John Wiley & Sons Inc., Hoboken, NJ, pp. 85–92
8. Mohaghegh F, Udaykumar H (2016) Comparison of sharp and smoothed interface methods for simulation of particulate flows I: Fluid structure interaction for moderate Reynolds numbers. Computers and Fluids, 140, 39–58
9. Mohaghegh F, Udaykumar H (2016) Comparison of sharp and smoothed interface methods for simulation of particulate flows II: Fluid structure interaction for moderate Reynolds numbers. Computers and Fluids, 143, 103–119
10. Mohaghegh F, Mousel J, Udaykumar H (2014) Comparison of sharp interface and smoothed profile methods for laminar flow analysis over stationary and moving boundaries. Paper presented at the American Society of Mechanical Engineers, Fluids Engineering Division (Publication) FEDSM, 1A https://doi.org/10.1115/fedsm2014-21171