A Lagrangian-Eulerian finite element algorithm for advection-diffusion-reaction problems with phase change

Beñat Oliveira[a,*], Juan Carlos Afonso[a], Sergio Zlotnik[b]

[a]*Australian Research Council Centre of Excellence for Core to Crust Fluid Systems/GEMOC, Department of Earth and Planetary Sciences, Macquarie University, Sydney, Australia.*
[b]*Laboratori de Càlcul Numèric, Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports, Universitat Politècnica de Catalunya, Barcelona, Spain.*

Abstract

This paper presents a particle-based Lagrangian-Eulerian algorithm for the solution of the unsteady advection-diffusion-reaction heat transfer equation with phase change. The algorithm combines a Lagrangian formulation for the advection + reaction problem with the Eulerian-based heat source method for the diffusion + phase change problem. The coupling between the Lagrangian and Eulerian subproblems is achieved with a phase change detector scheme based on a local latent heat balance and a consistent/conservative interpolation technique between Lagrangian particles and the Eulerian grid. This technique makes use of an auxiliary (finer) Eulerian grid that provides a simple and efficient method of tracking internal heterogeneities (e.g. phase boundaries), allows the use of higher order integration quadratures, and facilitates the implementation of multiscale techniques. The performance of the proposed algorithm is compared against one- and two-dimensional benchmark problems, i.e. pure rigid-body advection, isothermal and non-isothermal phase change, two-phase advective heat transfer and chemical reactions coupled with diffusion and advection. The numerical results confirm that the proposed solution method is accurate, oscillation-free and useful for and applicable to a wide range of fully coupled problems in science and engineering.

Keywords: Advection-Diffusion-Reaction, Phase Change, Heat Source Method, Lagrangian-Eulerian FEM, Interpolation

1. Introduction

Phase change transformations arise in many natural and industrial applications, e.g. metal casting and molding, composites manufacture, solar energy storage, groundwater transport problems, magmatic systems, planetary convection, etc. In addition, this phenomenon often takes place simultaneously with other physical processes such as advection, diffusion or chemical reactions. This is the case e.g. in geoscience, biological and planetary processes, where both solid-solid and solid-liquid transformations are coupled to chemical reactions over a range of spatial and temporal scales. Given the relevance of advection-diffusion-reaction (ADR) and phase change processes in both natural systems and industrial processes, much effort has been invested in the development of sophisticated numerical methods capable of providing accurate temporal and spatial solutions to these problems.

Among the many existing numerical methods to deal with ADR problems, Lagrangian-Eulerian (LE) approaches have become very popular [[1] [2] [3] [4] [5]]. In general, the LE approach takes advantage of appropriate operator splitting techniques to solve different aspects of the physical model with the most suitable Lagrangian or Eulerian formalism, avoiding thus the need of e.g. correction terms (e.g. SUPG, SCM, etc) to stabilize advection-reaction-dominated problems [[6] [7]]. Moreover, the use of Lagrangian particles instead of Lagrangian meshes mitigate (and in most cases, eliminate) the problem associated with the distortion of the mesh when large deformations are modelled [8]. For instance, the Particle Transport Method [5] computes the reactive term along moving (Lagrangian) particles

*Corresponding author
Email address: benat.oliveira-bravo@mq.edu.au (Beñat Oliveira)

while the diffusion is solved with standard finite element (FE) methods in the Eulerian mesh. However, despite their many advantages, particle-based LE methods have the drawback of having to transfer information from the Lagrangian particles to the Eulerian nodes (typically an overdetermined problem) and vice-versa (underdetermined problem) throughout the simulation, which inevitably introduces errors into the transferred fields. The transfer of information is commonly achieved by interpolation or projection techniques (e.g. global vs local information, damped least-squares, etc, [9],[10]), which are generally problem-dependent and require careful design, as seemingly small errors during the interpolation process can accumulate rapidly due to the large number of interpolations performed during a typical simulation [11]. Also, while splitting the physical operators facilitates the implementation of different solvers within the numerical scheme (lending itself to parallel computing), and greatly simplifies the numerical procedure, traditional splitting techniques are typically unable to achieve high-order accuracy (i.e. of order more than two). This, however, can be mitigated by using iterative splitting approaches [[12] [13]]. The authors refer to [14] and references therein for more details about the existing operator-splitting techniques.

Contrary to the vast literature on LE methods for traditional ADR problems, the extension of particle-based LE approaches to deal with ADR problems that include phase changes is a much less explored field. Furthermore, the authors are unaware of any study of operator-splitting particle-based LE approach with phase change. Current numerical strategies to deal with phase change problems are generally grouped into fixed-domain (fixed-grid) and front-tracking methods (e.g. adaptive, moving mesh or deforming grid methods). The latter provide accurate predictions of both isothermal and non-isothermal phase changes by explicitly modelling the location of the phase change interface [15] [16]. The main drawback of such adaptive methods is the high computational cost, as sophisticated re-meshing strategies are needed in order to accurately track the phase change boundaries. Moreover, when multiple scales are involved in the phase change problem of interest (e.g. dendritic interfaces within a much larger convecting system), front-tracking methods can become impractical. Fixed-domain methods offer a more general solution as they account for the phase change conditions implicitly. "Effective heat capacity" [17] [18], "enthalpy method" [19] [20] [21] [22], "heat source method" [23] [24] [25], "capacitance method" [21] [26] [27], "Temperature Transformation Methods (TTM)" [28] and "temperature recovery techniques" [29] are some examples of fixed-domain methods (see [30] for a review).

Among fixed domain methods, the "heat source method", also referred to as the "fictitious heat-flow method" [23, 24, 25], represents an attractive procedure for several practical reasons outlined below. The central idea behind this method is to treat the absorption or release of thermal energy accompanying phase transformations as an internal heat source (thus moved to the right-hand side of the energy equation) [23, 30]. The resulting energy equation becomes non-linear, as the localized heat source is temperature dependent. However, this non-linearity can be dealt with efficiently through an iterative procedure which adapts the release or absorption of heat until all the available latent heat is consumed by the phase change [23]. The nodal temperature is essential for this method since it is not only used to compute the release or absorbtion of the latent heat, but also identifies phases and tracks the phase change front without any additional computational effort. Compared to other approaches, the heat source method often reaches similar accuracy with significantly fewer iterations (rapid convergence) and it has no restrictions as to the nature (isothermal vs non-isothermal) of the phase transition. It also offers a more flexible platform to introduce algorithm modifications (as in this paper, see also [31] [32]), it allows relatively coarse meshes and large time steps, and it is stable and robust. In spite of the mentioned advantages over other approaches, the extension of the heat source method to work under particle-based Lagrangian-Eulerian approaches for general ADR problems is a poorly studied topic.

In this paper, we present a particle-based LE algorithm for the solution of general ADR problems including phase change. For this purpose, the general problem is divided into two subproblems: a) advection + reaction and b) diffusion + phase change. The former is solved with a particle-based Lagrangian method, and the latter with the heat source FEM [23]. Since the original heat source method is a purely transient Eulerian approach, initial conditions at any time step are the nodal solutions from the previous one. This is not the case in our LE approach, where nodal values need to be initialized with the information coming from the Lagrangian particles, including the phase change front position. The particle-to-node interpolation is performed by a fast local linear interpolation technique, and the node-to-particle interpolation by a new scheme able to i) preserve sharp discontinuities, ii) account for their natural diffusivity, and iii) minimize inconsistencies in interpolated magnitudes between Lagrangian particles and Eulerian nodes. Both interpolation schemes make use of an auxiliary finer grid, which also facilitates the implementation of higher accuracy numerical integration quadratures and/or recently developed multiscale techniques (e.g. MsFEM

[33]) within the general algorithm. However, as with any LE approach, this interpolation technique can introduce some errors in the nodal temperature field. Since the the original heat source method depends on nodal temperatures, the coupling of both methods requires two additional steps. Firstly, we propose a phase detection technique that uses the evolution of the released latent heat during phase changes rather than nodal temperatures as in [23]. This overcomes the problem associated with inaccurate nodal temperatures resulting from interpolation techniques and effectively initializes the Eulerian heat source FEM with the particle latent heat information. Secondly, we introduce a local node-to-particle latent heat balance which ensures its exact conservation and accurately tracks the phase change front in our algorithm.

The rest of the paper is organised as follows. Section 2 introduces the governing equations and problem statement. A detailed explanation of the Lagrangian-Eulerian FEM coupled with the heat source method is introduced in Section 3, where the particle-to-grid and grid-to-particle projection procedures are described. Numerical examples and benchmarks are presented in Section 4 to illustrate the performance and accuracy of the algorithm to solve both individual and coupled ADR problems with phase change. The applicability of the presented Lagrangian-Eulerian method to model chemical reactions is also tested in Section 4. Finally, future work and main conclusions are discussed in Section 5.

2. Governing Equations. Problem Statement

Let us consider the thermal advection-diffusion-reaction equation with phase change,

$$\begin{aligned} \rho c_p \left(\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T \right) &= \nabla \cdot (k \nabla T) + Q_{PC} + Q && \text{in } \Omega \times (0, t_{end}) \\ T(\mathbf{x}, 0) &= T_0 && \text{in } \Omega \\ T &= T_D && \text{on } \Gamma_D \times (0, t_{end}) \\ k\,(\mathbf{n} \cdot \nabla T) &= h && \text{on } \Gamma_N \times (0, t_{end}) \end{aligned} \tag{1}$$

where $\mathbf{x}$ $\{\mathbf{x} : \Omega \subset \mathbb{R}^d, d = 1, 2, 3\}$ is the space variable, $T = T(\mathbf{x}, t)$ is temperature, $\rho = \rho(\mathbf{x}, t)$ is the density, $c_p = c_p(\mathbf{x}, t)$ is the heat capacity, $k = k(\mathbf{x}, t)$ is the thermal conductivity, $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ is the given velocity field on $\Omega \times [0, t_{end}]$ and, time t varies in the interval $[0, t_{end}]$. For simplicity, throughout the paper we assume $d = 2$ and $\nabla \cdot \mathbf{v} = 0$, but these conditions can be easily relaxed. The body source term is broken into the phase change contribution, $Q_{PC} = Q_{PC}(\mathbf{x}, t)$, and any other heat source, $Q = Q(\mathbf{x}, t)$, (e.g. enthalpy of chemical reactions, shear heating, radioactive decay, etc). Γ_D and Γ_N are portions of the domain boundary $\partial\Omega$ ($\Gamma_D \cup \Gamma_N = \partial\Omega$), where Dirichlet and Neumann boundary conditions are respectively defined. $\mathbf{n}$ is the unit vector normal to the boundary, and h is a constant.

Lastly, the heat source term related to phase change processes is described as a function of latent heat L as

$$Q_{PC} = \rho \frac{\partial}{\partial t} \left(\int_\Omega L \, d\Omega \right) \tag{2}$$

3. Numerical Method

The original problem stated in (1) is divided into two subproblems, an *advection + reaction subproblem* and a *diffusion + phase change subproblem*. In the following two sections we describe the general LE numerical method; the coupling of the phase change problem is addressed in section 3.3; a brief algorithm summary is given in section 3.4.

3.1. Lagrangian-Eulerian approach

The *advection + reaction subproblem* is defined by

$$\begin{aligned} \rho c_p \left(\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T \right) &= Q && \text{in } \Omega \times [t_{m-1}, t_m] \\ T(\mathbf{x}, t) &= T(\mathbf{x}, t_{m-1}) && \text{in } \Omega \times \{t = t_{m-1}\} \\ T &= T_D && \text{on } \Gamma_D \times [t_{m-1}, t_m] \end{aligned} \tag{3}$$

whereas the *diffusion + phase change subproblem* reads

$$\begin{aligned} \rho c_p \frac{\partial T}{\partial t} &= \nabla \cdot (k \nabla T) + Q_{PC} && \text{in } \Omega \times [t_{m-1}, t_m] \\ T(\mathbf{x}, t) &= T^* (\mathbf{x}, t_{m-1}) && \text{in } \Omega \times \{t = t_{m-1}\} \\ T &= T_D && \text{on } \Gamma_D \times [t_{m-1}, t_m] \\ k (\mathbf{n} \cdot \nabla T) &= h && \text{on } \Gamma_N \times [t_{m-1}, t_m] \end{aligned} \tag{4}$$

The time interval $[0, t_{end}]$ is divided into M (possibly irregular) subintervals $[t_{m-1}, t_m]$, $m = 1, 2, ..., M$, where $t_0 = 0$ and $t_M = t_{end}$. Both subproblems are subsequently solved at each time step until $t_m = t_{end}$. The solution of subproblem (3), $T^* (\mathbf{x}, t_{m-1}) = T(\mathbf{x}, t_m)$, is used as the initial condition for (4). Therefore, a single time step of our LE method consists of the following five steps: (i) solve the *advection + reaction subproblem* with a Lagrangian scheme, (ii) transfer the intermediate solution $T^* (\mathbf{x}, t_{m-1})$ from the particles onto the Eulerian grid, (iii) solve the *diffusion + phase change subproblem* with the Eulerian FEM, (iv) project the final nodal temperatures onto the advected particles and (v) update temperature-dependent physical properties at the particles.

3.1.1. Lagrangian subproblem

The continuum is discretized into a set of Lagrangian moving particles, p, which carry (time-dependent) information of the physical properties of the medium. Using a Lagrangian notation, eq. (3) is rewritten for each particle p as

$$\frac{\mathrm{d}T^p}{\mathrm{d}t} = Q^p \tag{5}$$

which is solved subject to appropriate initial conditions. For time step $m \neq 1$, the initial conditions are the discrete values of the function T computed in the *diffusion + phase change subproblem* for each particle at their initial positions $(\mathbf{X}^p_{m-1})$,

$$T^p_{m-1} = T^p \left(\mathbf{X}^p_{m-1}, t_{m-1} \right) \tag{6}$$

It is clear from (5) that the source term Q^p represents an integrated effect along the trajectories of the individual particles. Particle trajectories are defined as,

$$\frac{\mathrm{d}\mathbf{X}^p}{\mathrm{d}t} = \mathbf{v}^p (\mathbf{X}^p, t) \tag{7}$$

In absence of boundary conditions, eq. (7) is a simple ODE. In order to find the new particles' positions after a certain time step, one solves eq. (7) on the time interval $[t_{m-1}, t_m]$, $m = 1, 2, ..., M$ with the initial conditions at time t_{m-1}

$$\mathbf{X}^p_{m-1} = \mathbf{X}^p (t_{m-1}) \tag{8}$$

Equations (5)-(6) and (7)-(8) can be integrated in time by any ODE solver (e.g. explicit Runge-Kutta, n-stage explicit Padé, etc) to obtain $T^p_{m^*} = T^p \left(\mathbf{X}^p_m, t_m \right)$, i.e. the particle temperature for the *advection + reaction subproblem*. The order of the chosen method is usually dictated by accuracy considerations, and since particles are independent from each other, schemes of different order could be used for different particles. Stiff solvers could be used if the reactive problem requires it. Higher accuracy can also be obtained reducing the time step whenever needed.

3.1.2. *Eulerian subproblem*

The standard Galerkin FE discretization is used to transform eq. (4) into the matrix form

$$\mathbf{M}\frac{\mathrm{d}\widehat{T}}{\mathrm{d}t} + \mathbf{K}\widehat{T} = \mathbf{Q}_{PC} \tag{9}$$

where $\widehat{T}$ is the vector containing the nodal temperature values T^I ($I = 1, ...$ number of nodes). $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ the stiffness matrix and $\mathbf{Q}_{PC}$ the right-hand-side vector (source term), which is computed as in the original heat source method of Rolph and Bathe [23]. An overview of the original heat source method can be found in Appendix A.

Supplemented with appropriate boundary and initial conditions, eq. (9) gives the solution to the discrete *diffusion + phase change subproblem* at the nodes. The initial conditions are obtained by interpolating the Lagrangian solution $T^p_{m^*}$ on the Eulerian grid. $\mathbf{M}$ and $\mathbf{K}$ are assembled at every time step using the auxiliary subcells $\{ij\}$ (see section 3.2.1). For sake of clarity, we describe the method a Cartesian Eulerian grid, but other type of grids can be also used.

The time derivative is discretized with standard finite differences and the resulting system (9) is solved with the most suitable implicit or explicit method.

3.2. *Interpolation techniques*

While physical information, B, is defined exclusively in the particles, solutions for temperature, T, are obtained at particles and Eulerian nodes. Moreover, the Eulerian FEM requires discrete physical information not only at the nodes but also at the Gaussian points to initialize the system (9). Therefore, in the following we define the necessary particle-to-grid and node-to-particle interpolation techniques.

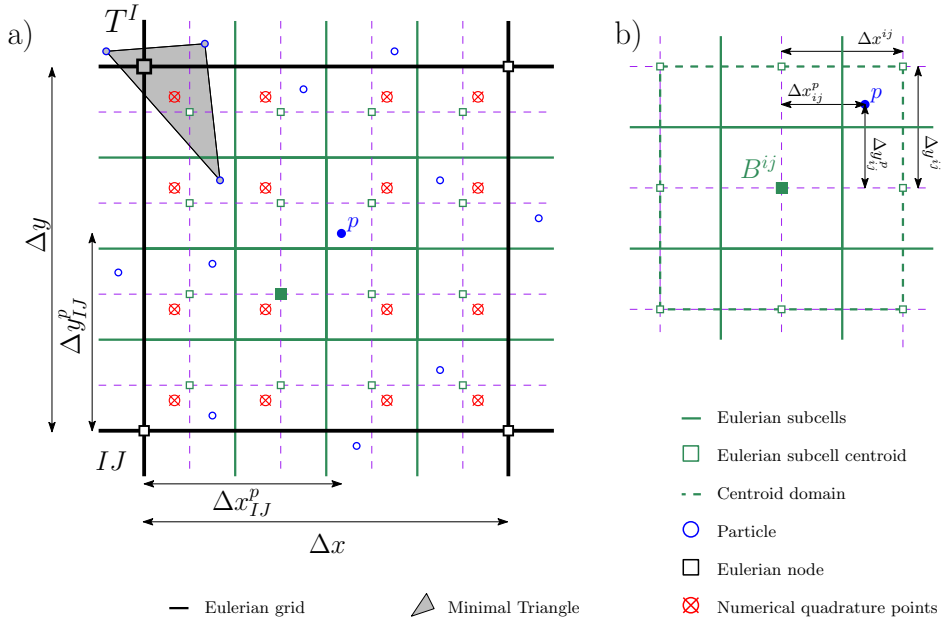

Figure 1: Representation of the Eulerian grid, Lagrangian particles and particle-to-grid interpolation. The Eulerian grid is formed by nodes and Eulerian subcells. a) Minimum triangles (gray) are employed as particle-to-node interpolation technique for the temperature (T^I); b) the particle-to-subcell communication is performed by bilinear interpolation for any other physical parameter (B^{ij}) at a given centroid. The numerical quadrature points are also shown, 4x4 in here, but this number may vary depending on the required accuracy at the integration.

3.2.1. *Particle-to-grid interpolation*

There are two different types of particle-to-grid interpolation. First, a *particle-to-node* interpolation is performed to obtain initial nodal temperatures. A minimal triangle is defined for each node by the triad of closest particles;

a Global Discrete Coordinate System (GDCS) is used to build these minimal triangles [5]. This results in a highly efficient algorithm to locate particles associated with Eulerian nodes [9]. The three closest particles define an unique plane in space and the nodal temperature T^I can be readily computed on it (see figure 1-A). In general, this local linear interpolation would introduce some smearing into the nodal temperature field (more details in section 3.3). This smearing can be minimized by using a higher particle density (albeit increasing the computational cost).

Second, we propose a *particle-to-subcell* interpolation of c_p, ρ, k in order to perform the numerical integration associated with the construction of the matrices in (9). Each subcell centroid receives information from the surrounding particles (figure 1) via the following bilinear expression [34],

$$B^{ij} = \frac{\sum_p B^p w_{ij}^p}{\sum_p w_{ij}^p}$$

$$w_{ij}^p = \left(1 - \frac{\Delta x_{ij}^p}{\Delta x^{ij}}\right) \cdot \left(1 - \frac{\Delta y_{ij}^p}{\Delta y^{ij}}\right)$$

where Δx_{ij}^p, Δy_{ij}^p, Δx^{ij} and Δy^{ij} are the distances defined in figure 1-b, and B refers to the variables ρ, c_p and k. While there are no real restrictions as to the number and shape of the Eulerian subcells, it is sensible to have at least one particle per subcell to take full advantage of the secondary mesh. In figure 1-b, the centroid domain coincides with the region delimited by the neighboring centroids, but the present method is not restricted to this case. The centroid domain can be expanded (adaptively) if the local particle density is not high enough. Alternatively, adaptive particle sets [5] can also be readily implemented.

In this manner, subgrid heterogeneities represented by the set of particles are explicitly considered/preserved at the subcell scale, rather than simply lumped at Eulerian nodes. The presence of the auxiliary subcell grid also facilitates the implementation of higher accuracy numerical integration quadratures or recently developed multiscale techniques (e.g. MsFEM [33]) within the general algorithm. In the first case, the integration points receive the information of the subcell they are situated in, thus allowing for more accurate integrations. In the case of multiscale techniques, the subcell grid can be used to compute the multiscale basis functions.

3.2.2. Node-to-particle interpolation

Once the *diffusion + phase change subproblem* is solved in the Eulerian mesh, the nodal temperature is transferred back to the particles to compute the next time step. An accurate node-to-particle interpolation is required in order to preserve the particle resolution. Simple linear interpolation schemes of absolute values smooth out any possible heterogeneities at the particle scale (see figure 2-b). Consequently, the interpolation of nodal increments is preferred. However, this last technique can lead to the creation of new maxima/minima under the presence of sharp fronts (see figure 2-c).

We overcome these problems with an alternative *node-to-particle* interpolation method based on nodal incremental values. The proposed technique aims to simulate the effects due to the *diffusion + phase change subproblem* at the particle scale. We define the particle increment, ΔT_m^p, as the sum of three terms: 1) a shape-preserving term $\left(\Delta T_m^p\right)_{sp}$, 2) a diffusion term $\left(\Delta T_m^p\right)_{diff}$, and 3) a conservative node-particle correction term $\left(\Delta T_m^p\right)_{corr}$. The first term preserves initial discontinuities at the particles after solving the Eulerian subproblem through a geometric operation. In this way, pre-existent particle heterogeneities and gradients are correctly preserved without creating new maxima or minima (figure 2-d). The second term accounts for a potential non-zero diffusivity of the initial discontinuity, thus simulating diffusion on a subgrid scale (without explicitly solving eq. (9) at the particles). Finally, the third term enforces that conserved quantities (e.g. heat, mass) are actually conserved during node-to-particle interpolations. For instance, in the case of eq. 1, this correction ensures that the global enthalpy change at Lagrangian particles (after node-to-particle interpolation) is consistent with that computed at Eulerian nodes.

Then, the updated particle temperature reads

$$T_m^p = T_{m^*}^p + \Delta T_m^p = T_{m^*}^p + \left(\Delta T_m^p\right)_{sp} + \left(\Delta T_m^p\right)_{diff} + \left(\Delta T_m^p\right)_{corr} \tag{10}$$

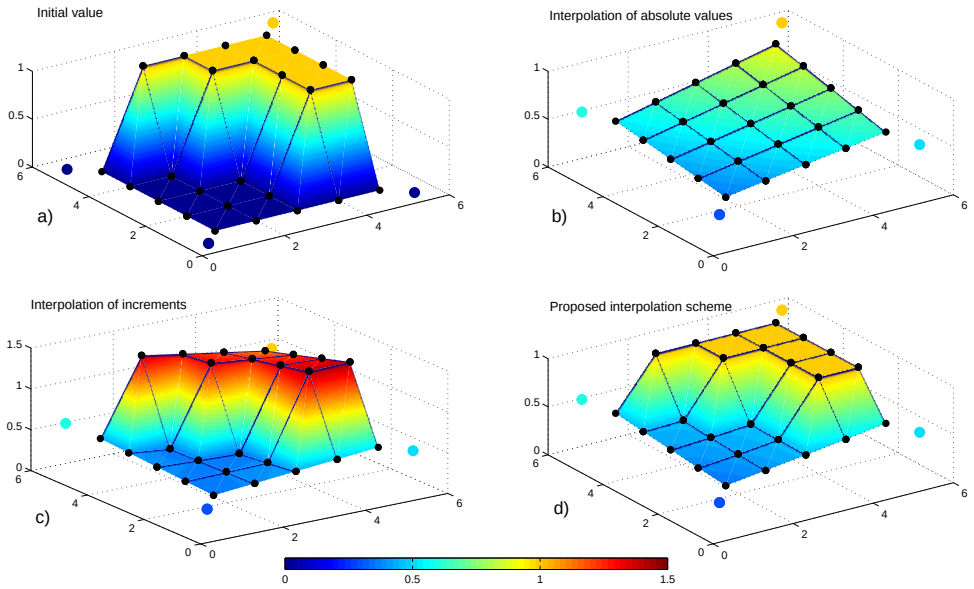

Figure 2: Representation of different node-to-particle interpolation techniques inside a rectangular element. The four large coloured circles refer to the Eulerian nodes, and the smaller black circles to the particles. a), Initial discontinuity (at t_{m-1}). b), the discontinuity is smoothed (reset of particles' values) during the interpolation of new absolute nodal values to particles (at t_m). c), creation of new maxima and minima through interpolation of increments with a standard bilinear scheme (at t_m). d), proposed weighted shape-preserving interpolation (t_m) technique where the discontinuity is preserved and no new maxima nor minima is created.

In addition, temperature dependent physical properties (ρ, c_p and k) are accordingly updated at each particle. More details on the physical basis and actual computation of $\left(\Delta T_m^p\right)_{sp}$, $\left(\Delta T_m^p\right)_{diff}$ and $\left(\Delta T_m^p\right)_{corr}$ can be found in Appendix B.

3.3. Coupling of the Heat Source method with the LE approach

The heat source FEM is an iterative procedure used to compute thermal effects related to phase change transformations [23]. Those phase change transformations occur at the nodes, which are understood as representative measures of their surrounding volumes. The iterative procedure allows us to update the non-linear source term, Q_{PC}, wherever phase transformations occur. This is done with the aid of two new variables, the amount of absorbed/released latent heat for each node Q_{CUMU}, and the total available nodal latent heat Q_{TOTL}. The former is used to compute the source term and it is only updated at nodes undergoing phase transformations after every iteration; the latter is used to identify nodes where the phase transformation has been completed. A phase transformation is considered to be completed when all the available latent heat has been consumed by a given node, and therefore, $Q_{CUMU} = Q_{TOTL}$. Unlike the temperature, both variables have to be conserved through the computation, as they are function of the latent heat, L.

In the original heat source method, the variables Q_{CUMU}, Q_{TOTL} and T remain unchanged between time steps (the values obtained at the end of time step $m-1$ coincide with the initial conditions at time step m). As it has been previously shown, this is not the case in our LE formulation. Consequently, the "phase status" of a given node (solid, liquid or undergoing phase transformation) needs to be initialized at the beginning of every time step m. Originally, a node is considered to undergo phase transformation if its temperature is between the liquidus (T_{liq}) and the solidus (T_{sol}) temperatures after any time step or iteration (see Appendix A for further details). Therefore, using the temperature as a phase change detector in our coupled LE approach would be the natural choice. However, as the local linear particle-to-node interpolation technique presented in section 3.2.1 inevitably introduces smearing, temperature is no longer a reliable variable to detect nodes undergoing phase change (see figure 3 for an isothermal phase change example). Therefore, we propose a phase change detector technique based on the latent heat (a conservative variable) and not the temperature (see section 3.3.1).

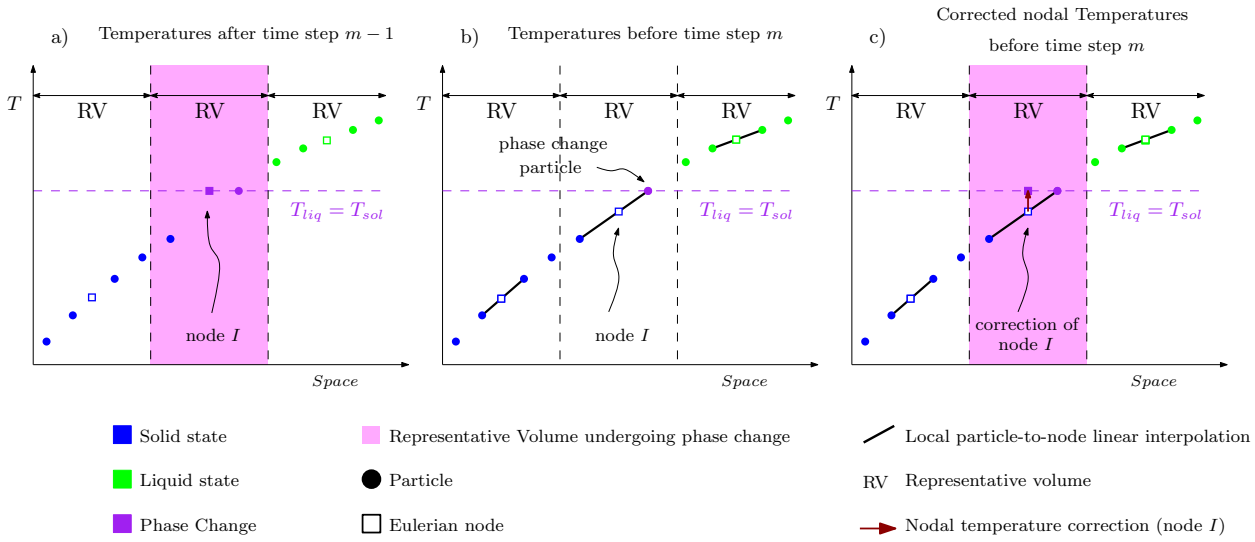

Figure 3: Illustration of a 1D isothermal phase change problem. For the sake of clarity, particles remain steady (no advection or reaction is computed). a), temperature distribution at nodes and particles after computing time step $m-1$. Here, a single particle is undergoing phase change, which is located inside the pink representative volume (RV). Node I is considered to undergo phase transition too ($T^I_{m-1} = T_{liq} = T_{sol}$), since not all the latent heat has been consumed by the phase transformation in its RV. b), the Eulerian grid is initialized at the beginning of time step m through the particle-to-node local linear interpolation scheme. Node I has not recovered the isothermal phase transformation temperature ($T^I_m \neq T_{liq} = T_{sol}$), as the nodal solution has been smeared due to the interpolation. According to the temperature, node I is not undergoing phase change despite the presence of particles undergoing phase change inside its RV. This is in contradiction with the original Rolph and Bathe heat source method [23], where a node is considered to have completed phase transformation when all the available latent heat in its RV has been consumed. Therefore, we proceed to time step m identifying particles undergoing phase change inside any RV and consequently correcting the nodal temperatures, as in c) (see section 3.3.1 for further explanation of the nodal temperature correction).

In order to complete the coupling of both methods, the absorbed or released nodal latent heat needs to be transferred back to the particles. An effective coupling of the heat source method with the LE approach depends on both the spatial accuracy of this node-to-particle Q_{CUMU} transference, and the conservation of Q_{CUMU} at every stage of the algorithm. We ensure these properties with a local balance of the nodal Q_{CUMU} inside each representative volume (RV) undergoing phase change (see section 3.3.2).

3.3.1. Phase Change detection technique

Firstly, Q_{CUMU} and Q_{TOTL} need to be lumped from the particles to the corresponding nodes (figure 4). Each node receives this information from the particles inside their RV. Secondly, nodes undergoing phase change are flagged using this lumped information rather than nodal temperatures, as done in the original heat source method ([23]). As mentioned above, this overcomes the problem of inaccurate nodal temperature values and enables an effective phase change detection mechanism (figure 3-c). The flagging system ($Q_{flag} = \{0,1,-1\}$) is defined as follows: -1 indicates liquid phase, 1 indicates solid phase, and 0 indicates a node undergoing phase change. At any stage of the simulation, three possible scenarios can occur at the nodes:

1. $Q_{CUMU} = 0$. The node has not been affected by a phase change at any previous time step. Therefore, $Q_{flag} = \pm 1$ depending on the problem (i.e. solidification or melting).
2. $|Q_{CUMU}| \leq Q_{TOTL}$. The node is undergoing phase change, so $Q_{flag} = 0$. In this case, the nodal temperature generally needs to be corrected to keep it consistent with the computed amount of latent heat released/absorbed. We use

$$T_{m-1} = T_{liq} - \frac{|Q_{CUMU}|}{Q_{TOTL}} \left(T_{liq} - T_{sol}\right) \qquad \text{solidification}$$

$$T_{m-1} = T_{sol} + \frac{|Q_{CUMU}|}{Q_{TOTL}} \left(T_{liq} - T_{sol}\right) \qquad \text{melting}$$

assuming linear evolution of latent heat with temperature [23].

3. $|Q_{CUMU}| = Q_{TOTL}$. All the available latent heat in the node has been previously released or absorbed (i.e. the phase change has been completed). Therefore, $Q_{flag} = \pm 1$ depending on the problem.

This flagging system requires accurate computations of both Q_{CUMU} and Q_{TOTL}. This is achieved with a conservative node-to-particle projection of these variables, as described below.

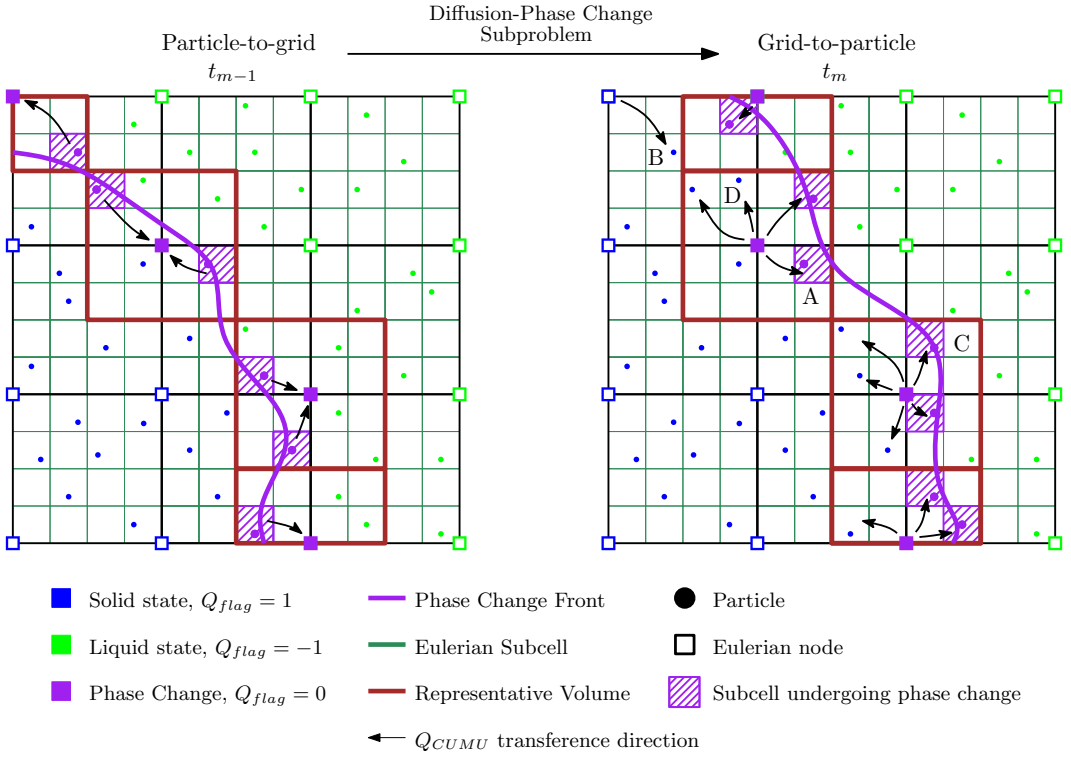

Figure 4: Particle-to-grid and grid-to-particle communication schemes for phase change related variables. At t_{m-1}, the accumulated latent heat is lumped from the particles to the Eulerian nodes, they are flagged and their temperature corrected according to the three possible scenarios described in section 3.3.1. Once the diffusion - phase change subproblem is complete, Q_{CUMU} is transferred back from the Eulerian nodes to the subcell grid at t_m. New nodes may undergo phase change at every time step, and Q_{CUMU} is distributed to the subcells surrounding these new phase change Eulerian nodes. A, B, C and D refer to the different possibilities in the node-to-particle Q_{CUMU} projection. This enables an accurate phase change front tracking system (in purple).

3.3.2. Conservative node-to-particle projection for phase change variables

Q_{CUMU} is projected from the nodes to the particles through a local balance of Q_{CUMU} inside their RVs. This technique is spatially accurate since it takes advantage of the Eulerian subcells (figure 1). It also takes into account the shape of the phase front. Moreover, the conservation of the latent heat is strictly ensured by this balance.

For this purpose, the spacial gradient of the ratio Q_{CUMU}/Q_{TOTL} is computed at each node using the 2D stencil shown in figure 5. The direction of the gradient and the phase change propagation direction are assumed to be the same (represented by the plane π in figure 5). Therefore, the spatial distribution of Q_{CUMU} in the subcells becomes a 1D problem and it is solved using

$$\sum_{sc} \left(A + \overline{x}_{ref_A - sc} \cdot \tan\alpha\right) Q^{sc}_{TOTL} = Q_{CUMU} \tag{11}$$

where A is the unknown ratio Q_{CUMU}/Q_{TOTL} for a reference subcell (ref_A), $\overline{x}_{ref_A - sc}$ is the distance between ref_A and the projected subcell positions into π, $\tan\alpha$ is the value of the Q_{CUMU}/Q_{TOTL} gradient, and the subindex sc refers to the individual subcells ($sc = 1, \ldots$ number of subcells) (see figure 5). Eq. (11) distributes the nodal Q_{CUMU} among the

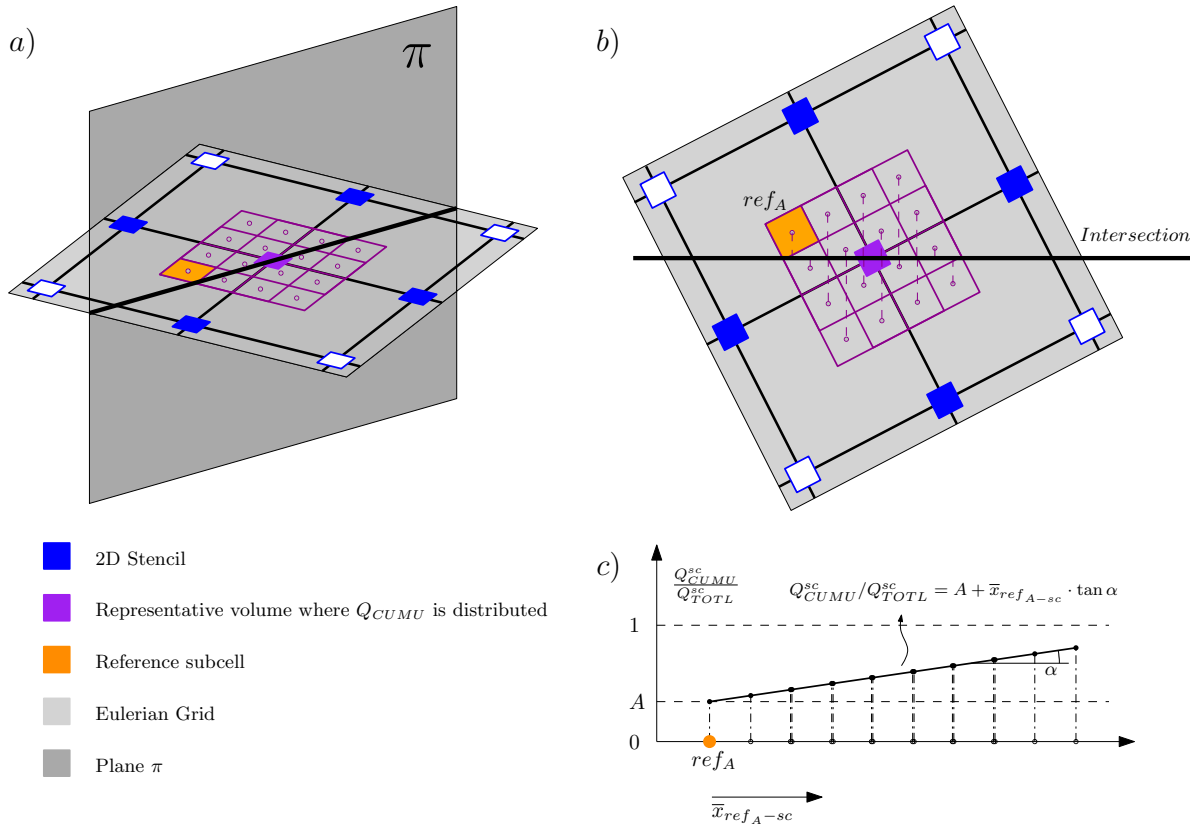

Figure 5: Main steps to distribute the absorbed/released latent heat by a certain node, Q_{CUMU}, back to the representative volume. First the spatial gradient of the ratio Q_{CUMU}/Q_{TOTL} is computed at nodes undergoing phase change using the 2D stencil represented by blue nodes in *a*). Then, the Eulerian grid is intersected by the plane π, which is any plane parallel to the gradient direction and containing the given node. We use the intersection between this plane π and the one containing the Eulerian grid to project the positions of the subcells centroids, as shown in *b*). In here, we compute $\overline{x}_{ref_A-sc}$ which is the projected distance between any subcell and the reference subcell onto the intersection (or plane π), (*c*), out of scale). The gradient value $\tan\alpha$ is used to distribute Q_{CUMU} according to $\sum_{sc}(A + \overline{x}_{A-sc} \cdot \tan\alpha)\, Q^{sc}_{TOTL} = Q_{CUMU}$ subjected to $0 \leq |Q^{sc}_{CUMU}|/Q^{sc}_{TOTL} \leq 1$, where the only unknown A refers to the ratio Q_{CUMU}/Q_{TOTL} at the reference subcell. After obtaining A, the remaining subcell values, $Q^{sc}_{CUMU}/Q^{sc}_{TOTL}$, are automatically given by $Q^{sc}_{CUMU}/Q^{sc}_{TOTL} = A + \overline{x}_{ref_A-sc} \cdot \tan\alpha$ (*c*))

subcells inside its representative volume, where $Q^{sc}_{CUMU}/Q^{sc}_{TOTL} = A + \overline{x}_{ref_A-sc} \cdot \tan\alpha$ and $0 \leq |Q^{sc}_{CUMU}|/Q^{sc}_{TOTL} \leq 1$. We obtain A from eq. (11) for any arbitrary ref_A, and the remaining $Q^{sc}_{CUMU}/Q^{sc}_{TOTL}$ are automatically given by the expression above.

The variable Q^{p}_{CUMU} is then updated at each particle $p \subset sc$, assuming the same ratio $\frac{Q^{p}_{CUMU}}{Q^{p}_{TOTL}} = \frac{Q^{sc}_{CUMU}}{Q^{sc}_{TOTL}}$ between the particle p and the subcell that contains it. Indeed, this projection technique conserves the latent heat over a given RV,

$$\sum_{p \subset RV} Q^{p}_{CUMU} = \sum_{sc \subset RV} Q^{sc}_{CUMU} = Q_{CUMU}$$

$$\sum_{p \subset RV} Q^{p}_{TOTL} = \sum_{sc \subset RV} Q^{sc}_{TOTL} = Q_{TOTL}$$

Figure 4 illustrates the four possible cases for transferring Q_{CUMU} from the nodes back to the particles at t_m. In the first case (A), a particle undergoing phase change will still be in transition after the current time step, as $|Q_{CUMU}| < Q_{TOTL}$. In case (B), the phase change has been completed at the particle, as $|Q_{CUMU}| = Q_{TOTL}$. Scenarios (C) and (D) are respectively the same as (A) and (B), but for particles that were not undergoing phase change at the previous time step. This procedure uses the same Global Discrete Coordinate System (GDCS) used in the particle-to-grid interpolation (section 3.2.1) to keep the algorithm simple and efficient.

Finally, consistency between temperature and absorbed/released latent heat at the particles is ensured with the following correction [23],

$$T_m^p = T_{liq} - \frac{|Q_{CUMU}^p|}{Q_{TOTL}^p}\left(T_{liq} - T_{sol}\right) \qquad \text{solidification}$$

$$T_m^p = T_{sol} + \frac{|Q_{CUMU}^p|}{Q_{TOTL}^p}\left(T_{liq} - T_{sol}\right) \qquad \text{melting}$$

3.4. Algorithm summary

The transient problem in eq. (1) is solved for m time steps until $t = t_{end}$. In each time step, our particle-based LE algorithm requires the following operations:

1. Solve the advection-reaction subproblem between t_{m-1} and t_m and get $T_{m^*}^p$ (section 3.1.1).
2. Interpolate c_p, ρ, k and $T_{m^*}^p$ from the set of particles onto the Eulerian nodes and subcells (section 3.2.1).
3. Lump Q_{CUMU} and Q_{TOTL} from the set of particles onto the Eulerian nodes and subcells (section 3.3.1).
4. Identify the nodes undergoing phase change from the previous time step and correct their temperatures (section 3.3.1).
5. Assemble $\mathbf{M}$ and $\mathbf{K}$ matrices and solve the eq. (9) (section 3.1.2).
6. Identify new nodes at t_m undergoing phase change (Appendix A).
7. If there are no nodes undergoing phase change, go to step 10.
8. Compute $\mathbf{Q}_{PC}$ and solve the system of equations (9) using the iterative heat source method (Appendix A). This can be done more efficiently by solving smaller versions of system (9) depending on the spatial distribution of nodes experiencing phase change.
9. Project Q_{CUMU} back to the subcells and particles (section 3.3.2).
10. Interpolate the temperature at particles, T_m^p, from the nodes (section 3.2.2). Correct the particle temperature, if necessary (section 3.3.2). Update c_p, ρ, k at the particles.
11. Advance time and loop back to step 1 for the next time step $m + 1$.

A MATLAB implementation of all the above steps is available from the authors upon request or downloadable from: http://research.science.mq.edu.au/mg3/software/. We note that the algorithm is highly parallelizable, e.g. the advection-reaction subproblem is solved independently for each particle. This implementation, however, is beyond the scope of the present work.

4. Numerical Examples

4.1. 1D convection-diffusion of a Gaussian hill: experimental convergence analysis

We use a 1D convection-diffusion of a Gaussian hill problem to experimentally discuss the order of accuracy of the presented LE particle-based method. The problem consists of solving the transient homogeneous linear convection-diffusion equation on the 1D domain $\Omega = [0, 1]$ [35]. Eq. (1) is solved with $\rho = c_p = 1$, $k = 0.005$ and $\mathbf{v} = (1, 0)$ and reaction terms $Q_{PC} = Q = 0$. The initial condition is,

$$u(x, t) = \frac{5}{7}\exp\left(-\left(\frac{\mathbf{x} - \mathbf{x}_0}{l}\right)^2\right)$$

where $\mathbf{x}_0 = 2/15$ and $l = 7\sqrt{2}/300$ and with boundary conditions $u(0, t) = 0$ and $u(1, t) = 0$. The exact solution (figure 6) is given by

$$u(x, t) = \frac{5}{7\sigma(t)}\exp\left(-\left(\frac{\mathbf{x} - \mathbf{x}_0 - t}{l\sigma(t)}\right)^2\right), \qquad \text{where } \sigma(t) = \sqrt{1 + 4\frac{k}{\rho c_p}t/l^2}$$

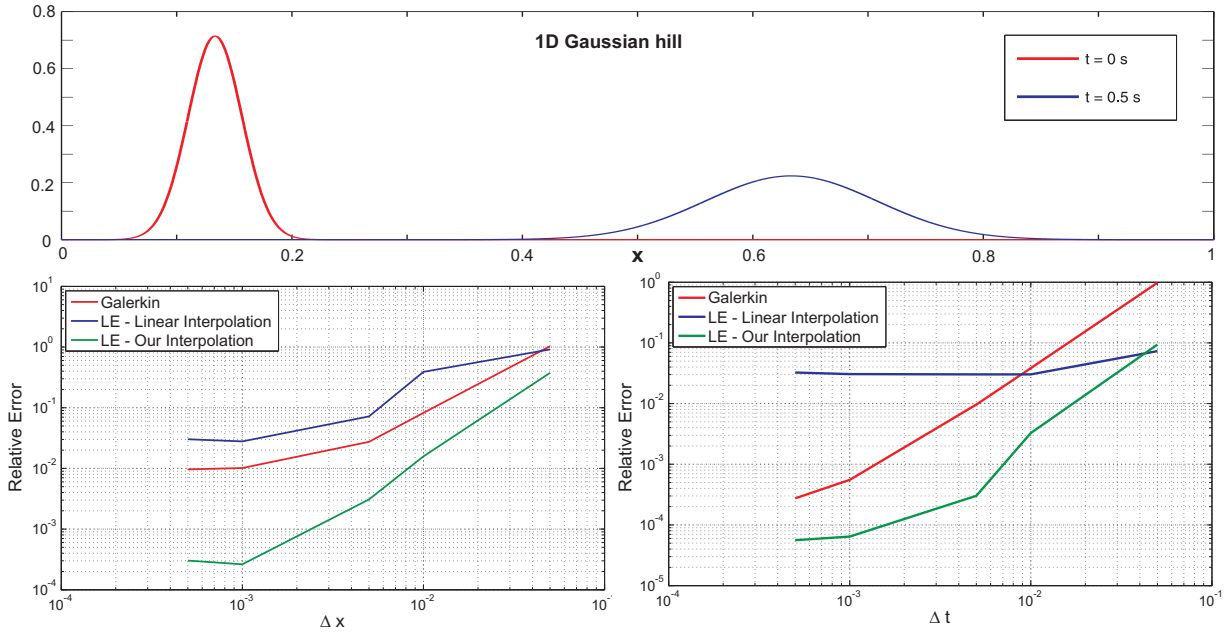

Figure 6: Analytical results for the 1D convection-diffusion of a Gaussian hill (top figure) and the spatial and temporal convergence study for our particle-based LE method (bottom left and right figures). The relative error is evaluated in the $H_1(\Omega)$ norm, at $t = 0.5s$. For the spatial convergence $\Delta t = 1/200s$ has been used and $\Delta x = 1/2000m$ for the time convergence. Computations have been carried out with 4 randomly distributed particles per element.

Figure 6 shows the evolution of the error against time and space discretization of our particle-base LE method. The error is evaluated in the $H_1(\Omega)$ norm, at $t = 0.5s$. Results are shown for a standard Galerkin approach and our algorithm with two different interpolation techniques with a second order Crank-Nicolson time discretization (figure 6). Contrary to Galerkin, no restriction on the timestep is needed in order to obtain oscillation-free results with our method (since the advection operator disappears from the Eulerian framework).

In general, the Galerkin formulation converges as expected, whereas the behavior of the particle-based LE method highly depends on the interpolation scheme used. In the spacial case (bottom left), the relative error evolves well for both Galerkin and our method. Here, our method (green line) improves the convergence rate obtained with Galerkin and yields significantly smaller relative errors. In the case of time convergence (bottom right), the type of interpolation employed has a major impact on the results. When our difference-based interpolation technique is used, second order accuracy is obtained. However, the convergence rate is drastically reduced for the case of linear interpolation, as the interpolation error exceeds the error in time.

The second order convergence obtained by the proposed method contrasts with the first order convergence of most splitting schemes. Traditional operator splitting methods cannot achieve higher-order accuracy in time (i.e. of order more than two); moreover the sequential splitting is of first order [14]. This is mainly due to the errors introduced by the decoupled equations. Several schemes have been proposed to improve the order of convergence; see for example [12] [13] [36]. All these methods introduce an iterative-type operator splitting that couples the different physical processes and therefore reduces the splitting errors. These studies, however, address splitting techniques where all the operators are based on the same type of spatial dicretization (Eulerian). To our knowledge, the order of particle-based Eulerian-Lagrangian methods have not been studied and no theoretical result are available. Our study shows that the expected first-order accuracy for a sequential splitting can be improved with proper interpolation techniques. In our method, the advected particles are corrected through interpolation with the information coming from the Eulerian diffusion, which mimics the procedure followed by higher-order splitting techniques.

Other splitting techniques such as Strang-Marchuk splitting (second order), the symmetrically weigthed sequential splitting (first order) and the higher-order iterative splitting have also been tested for this example. None of these methods significantly improves the results of the sequential-type scheme for any of the interpolation schemes.

Consequently, they have not been used in the following examples.

4.2. Three rigid body rotation

This example is an adaptation of the three body rotation transport problem in [37] and [38] to test the performance of our numerical scheme for advection and advection+diffusion problems in 2D.

Eq. (1) is solved in $\Omega = [0, 1]^2$, with $\rho = c_p = 1000$, $\mathbf{v} = (0.5 - \mathbf{x}(2), \mathbf{x}(1) - 0.5)$ and reaction terms $Q_{PC} = Q = 0$. Two different problems are solved: pure advection ($k = 0$) and advection+diffusion ($k = 100$). The initial setup consists of three bodies (top right contour plot in figure 7): a slotted cylinder, a conical body, and a smooth cone. They are centered at $\mathbf{x}_c = (0.5, 0.75)$, $\mathbf{x}_c = (0.5, 0.25)$ and $\mathbf{x}_c = (0.25, 0.5)$ respectively, with radius $r_0 = 0.15$ for the three bodies. Outside the bodies the initial value of the field variable is 0 and Dirichlet boundary conditions are imposed at the edges. Three auxiliary functions $r(\mathbf{x})$ are defined in Ω, one for each body

$$r(\mathbf{x}) = \frac{1}{r_0}\sqrt{(\mathbf{x}(1) - \mathbf{x}_c(1))^2 + (\mathbf{x}(2) - \mathbf{x}_c(2))^2}$$

The slotted cylinder, conical body and hump are given by

$$T(\mathbf{x}, 0) = \begin{cases} 1 & \text{if } r(\mathbf{x}) \leq 1, |\mathbf{x}(1) - \mathbf{x}_c(1)| \geq 0.0225 \text{ or } \mathbf{x}(2) \geq 0.85 \\ 0 & \text{else} \end{cases}$$

$$T(\mathbf{x}, 0) = 1 - r(\mathbf{x})$$

$$T(\mathbf{x}, 0) = \frac{1}{4}\left(1 + \cos\left(\pi \min\{r(\mathbf{x}), 1\}\right)\right)$$

In the simulations, three different rectangular regular grids have been used: 64x64, 128x128 and 256x256 Eulerian nodes. In each element, 4x4 equidistant and (initially) regularly distributed Lagrangian particles have been used. This results in 4,096, 16,384 and 65,536 degrees of freedom (including the Dirichlet nodes) in the Eulerian problem (eq. (4)) and, 63,504, 258,064 and 1,040,400 total particles respectively. With the given velocity field $\mathbf{v}$, the 3 bodies rotate counter-clockwise and it takes $t = 2\pi$ s for a full revolution. Three revolutions are computed for the advection problem and one for the advection+diffusion problem. Numerical solutions are compared using an error estimate $||E|| = \sqrt{\sum_I \sum_J E_{IJ}}/DOF$, where E_{IJ} is the nodal difference between the numerical solution and a reference solution for each case (last column in figure 7) and DOF means degrees of freedom (in this case = number of Eulerian nodes).

	Advection			Advection + Diffusion		
	64x64	128x128	256x256	64x64	128x128	256x256
IT1	2.208e-03	7.832e-04	2.636e-04	5.300e-04	8.481e-05	1.666e-05
IT2	0	0	0	4.353e-04	4.883e-05	2.468e-05
IT3	0	0	0	3.029e-04	4.089e-05	1.104e-05

Table 1: $||E||$ error estimation for the three body rotation problem. Results for the advection and the advection+diffusion problems for different mesh sizes (by columns) and interpolation techniques (by rows) are given. IT1, IT2 and IT3 refer to the interpolation technique 1, 2 and 3 respectively.

Figure 7 shows contour solutions of the three body rotation problem. Each column represents a different interpolation technique, whereas rows show different mesh sizes. Since solutions are shown for the Eulerian mesh, they all include the effects of the different particle-to-node-to-particle interpolation techniques discussed in sections 3.2.1 and 3.2.2. Table 1 shows the computed errors for each problem, mesh size, and used interpolation technique (linear interpolation of absolute values = IT1; linear interpolation of increments = IT2, proposed interpolation scheme = IT3).

For the pure advection problem (rows 1 and 2), interpolation IT1 results in an unrealistic smearing of the solution. As expected, this smearing is significantly reduced in the finer mesh, but increases with number of revolutions. On the other hand, for both interpolation techniques based on incremental values (IT2 and IT3), the initial condition is recovered exactly regardless of the total number of revolutions. The small irregularity visible in the slotted cylinder for

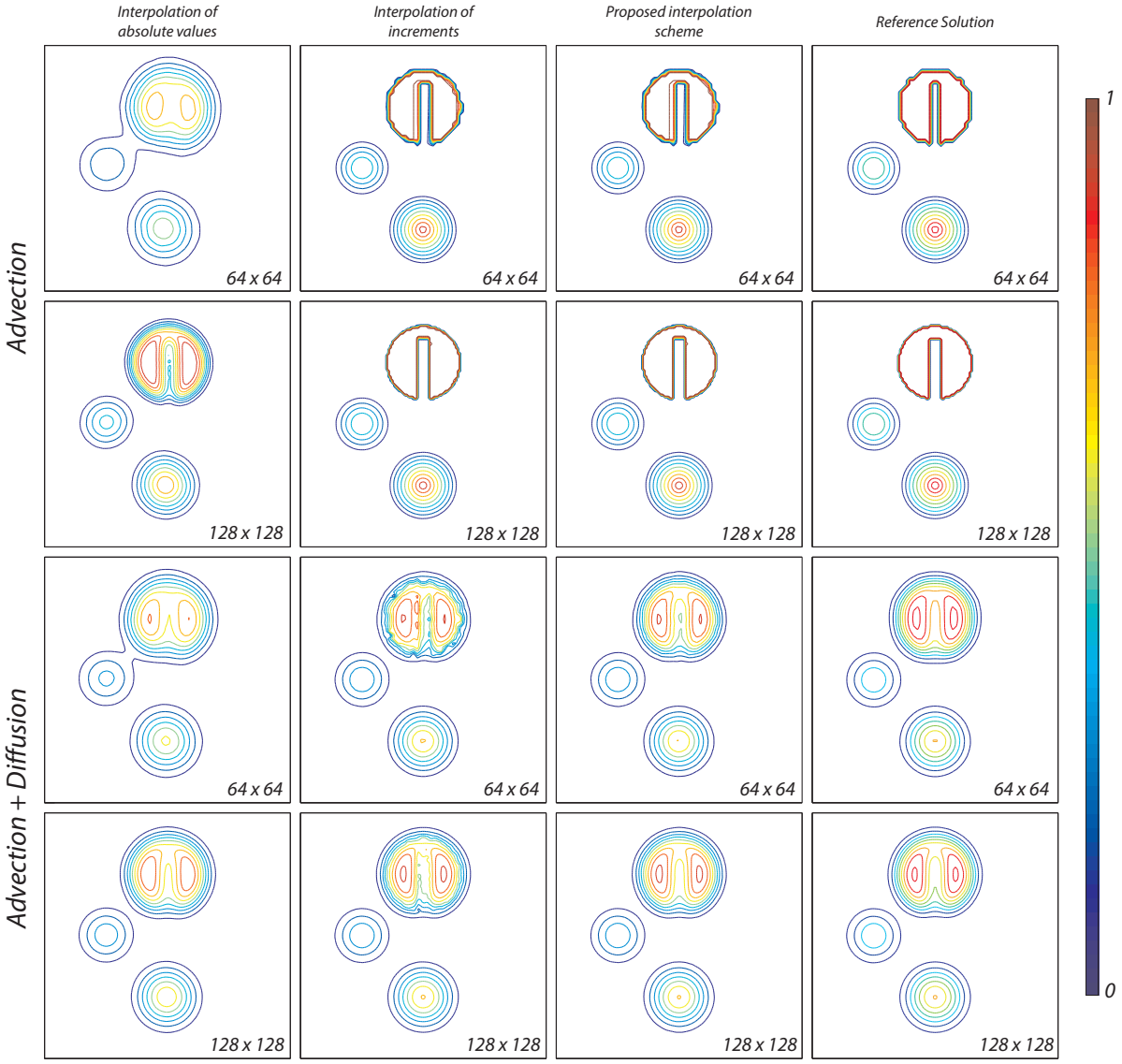

Figure 7: Contour plots for the three body rotation problem. Advection (3 revolutions) and Advection + Diffusion (1 revolution) problems (by rows) for different interpolation schemes (by columns) and mesh sizes. The reference solutions are also shown (last column).

IT2 and IT3 is due to the particle-to-node interpolation, but the particle information remains unchanged at every time-step. Therefore, the computed error shown in table 1 is 0 for both cases. The reference solution for the computation of the error is the initial condition (last column in figure 7) for each mesh size.

For the combined advection and diffusion problem, the IT1 interpolation performs better than in the previous case (especially in fine meshes), as the numerical diffusion is balanced with the physical diffusion (figure 7). However, a significant smearing still is perceived. The solution with IT2 shows the numerical instability described in section 3.2.2, whereas IT3 shows no instabilities. The use of a finer mesh does not mitigate these instabilities for IT2, where new nonphysical maximum and minimum temperatures are generated in the vicinity of sharp fronts. In some cases,

these maxima exceed 50% the theoretical maxima. This could be avoided with an initial smoothing of the sharp fronts, but this is in fact a linear interpolation of nodal absolute values to the particles, and therefore it is subject to the same loss of information and smearing as IT1. In contrast, there is no appreciable numerical instability associated with the proposed interpolation IT3. IT3 also shows good phase accuracy, since the particle with initially the highest value inside the cone remains with the highest value after a full revolution. The reference solution for the computation of the error is the transient diffusion problem solved in each Eulerian mesh for the same initial condition (last column in figure 7).

4.3. 1D semi-infinite solidification with phase change

Property	1D isothermal	1D non-isothermal	2D semi-infinite	Unit
Thermal conductivity - k	100	100	1	W/(m °C)
Specific Heat Capacity - c_p	400	400	1	J/(kg °C)
Density - ρ	6900	6900	1	kg/m^3
Latent Heat - L	130000	130000	0.1923	J/kg
Solidus Temperature - T_{sol}	400	395	-0.15	°C
Liquidus Temperature - T_{liq}	400	405	-0.15	°C
Initial Temperature - T_0	450	450	1	°C
Boundary Temperature - T_{wall}	0	0	-45	°C

Table 2: Material properties, initial and boundary conditions.

This benchmark aims to test the accuracy of our numerical scheme for both, isothermal and non-isothermal phase change problems ([39] [40]). The problem set up is displayed in figure 8 and the material properties in Table 2. Each representative volume in the Eulerian grid is divided in 4x4 subcells. Temperature history, temperature profiles and the location of the phase change front are given in figure 9 for the isothermal case. Figure 10 shows the temperature history of the non-isothermal case at x = 10 mm. Analytical solutions are only available for the isothermal case [41] and have been summarized in Appendix C.

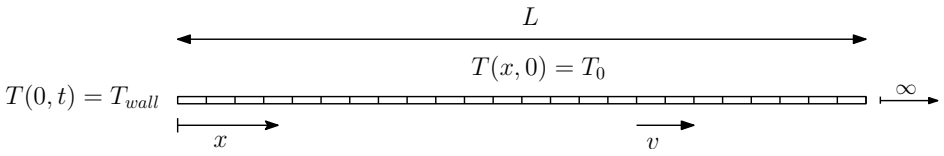

Figure 8: Mesh, geometry, boundary conditions and initial conditions for the 1D solidification problem

Results from our algorithm differs little from those obtained with the original heat source method in the case of non-advective phase change problems [23]. The inherent oscillations in temperature histories appearing in the original method close to the phase change front also occur in our formulation and the analytical solutions are recovered with the same accuracy at the nodes. Since particles remain in their positions during the whole computation, the absorbed or released latent heat is projected (and lumped) back and forward from particles to nodes without additional difficulty. Moreover, the phase change front can advance more than one Eulerian cell in a given time step without lost of accuracy. Another advantage of our method is that the the phase front is defined at the particle level, which improves the resolution for the location of the phase change front (figure 9).

For the case of non-isothermal phase change, results in figure 10 indicate that our algorithm performs well, giving satisfactory solutions (as compared with those in the finest mesh) even for coarse meshes. As expected, better solutions are obtained for finer meshes. The temperature oscillations close to the phase change front are mitigated in this case (except for the coarsest mesh) due to the non-isothermal nature of the problem. Similar non-isothermal phase change cases reported in the literature [23] have been successfully reproduced with our method (not shown here).

4.4. Two-phase convective heat transfer in a 1D semi-infinite slab

This problem couples both advection and the phase change problems. Temperature histories and profiles are shown in figure 11. Analytical solutions are only available for cases with v = 0 [41], which are denoted by solid lines

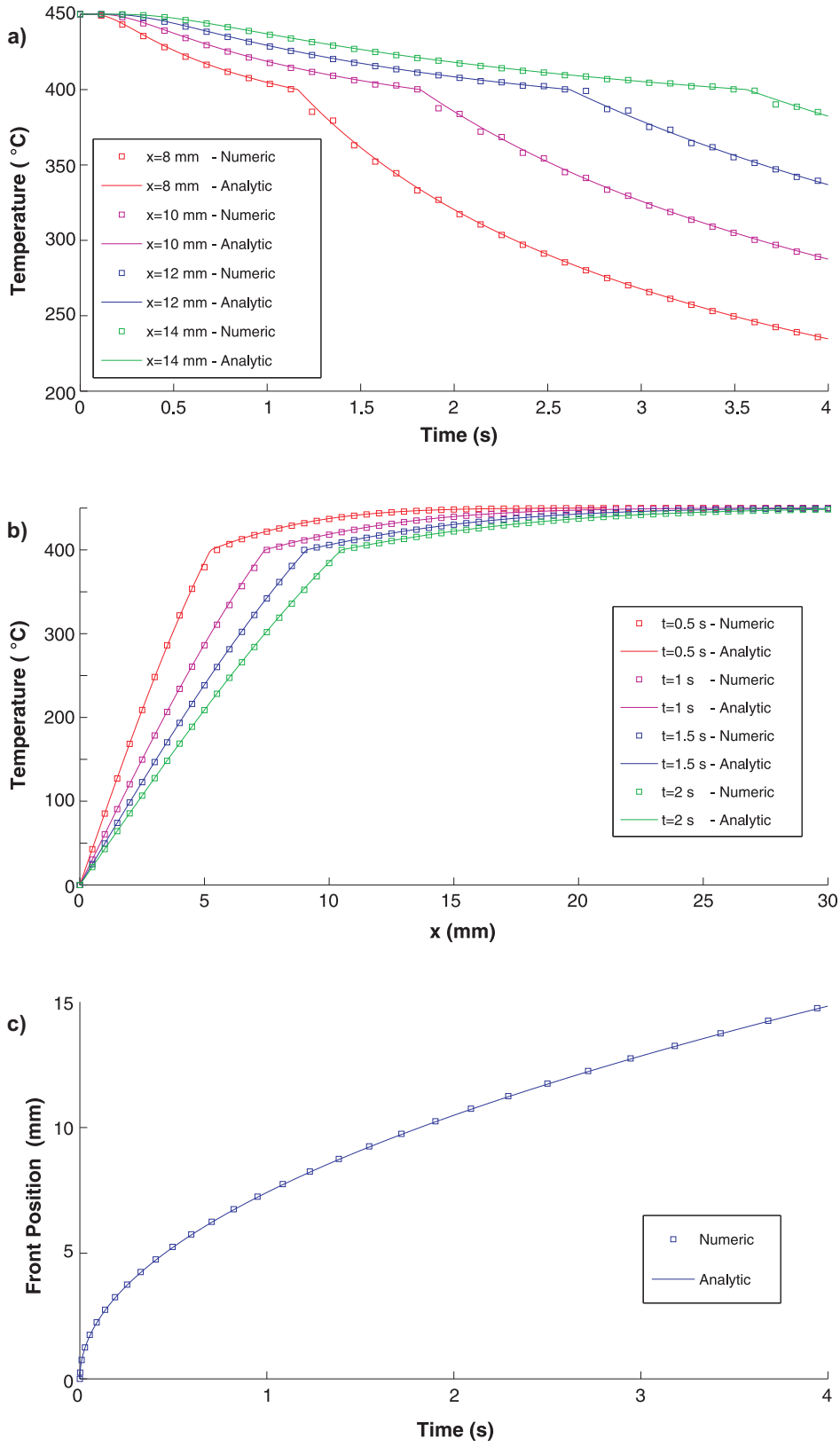

Figure 9: Results for the isothermal phase change problem. a) Temperature histories for different positions. b) Temperature profiles at different times. c) Evolution of the phase change front over time.

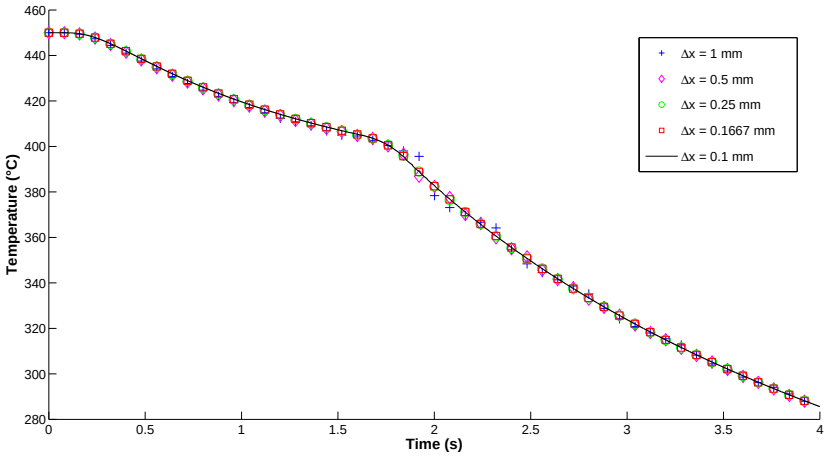

Figure 10: Temperature history at $x = 10$ mm for non-isothermal phase change problem and different element lengths (Δx).

in figure 11. We show these reference solutions for qualitative comparison only, as our simulations involve $v \neq 0$. Boundary conditions are treated as follows. For $v > 0$, particles entering the system at $x = 0$ mm have $T_m^p = 0°C$, whereas for $v < 0$ mm/s, particles entering the system at $x = 60$ mm have $T_m^p = 450°C$.

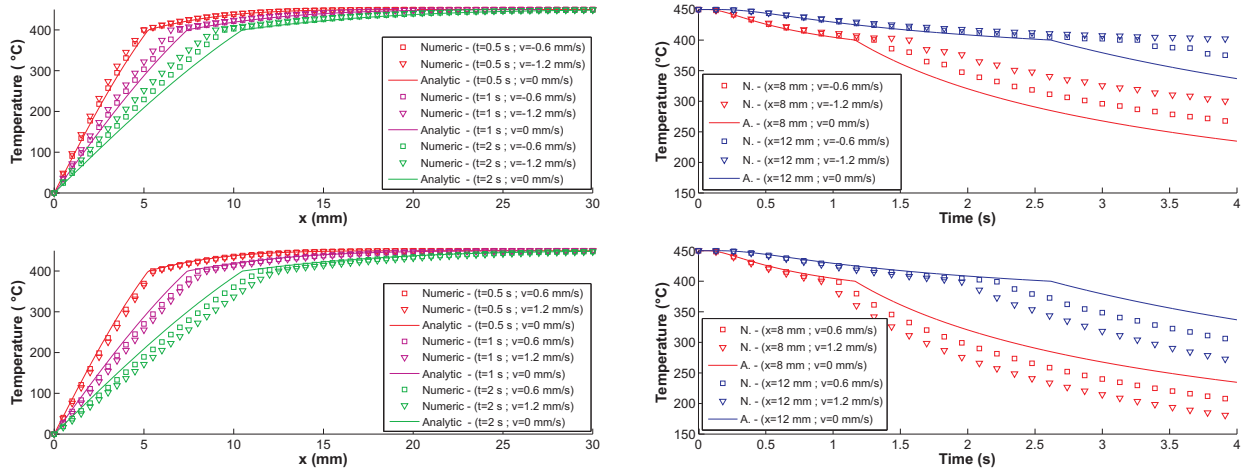

Figure 11: Temperature profiles (left plots) and temperature histories (right plots) for different times (t = [0.5 s; 1 s; 2 s]) and discrete points (x = [8 mm; 12 mm]) for $\Delta x = 0.5mm$ and $\Delta t = 0.015s$. The top figures correspond to negative velocities (v = [-0.6 mm/s; -1.2 mm/s]), whereas the bottom figures show solutions for positive velocities (v = [0.6 mm/s; 1.2 mm/s]). The solutions are compared to the analytical solution without advection (v = 0 mm/s)

Results in figure 11 indicate that our algorithm also performs well for the case of advection with solidification, regardless of the direction of advection. Our results are also in agreement with those presented by [40] for a similar test case.

4.5. Two-phase heat transfer in a 2D semi-infinite domain

This problem is the extension of the 1D isothermal phase change for a semi-infinite rectangular corner [42] [23] [39]. Here, the problem is slightly modified by adding localized heat sources. Consequently, our method is challenged by the development of a highly irregular phase-change front. Boundary conditions and additional heat sources are depicted in figure 12-E, whereas physical properties and the initial constant temperature are summarized in table 2.

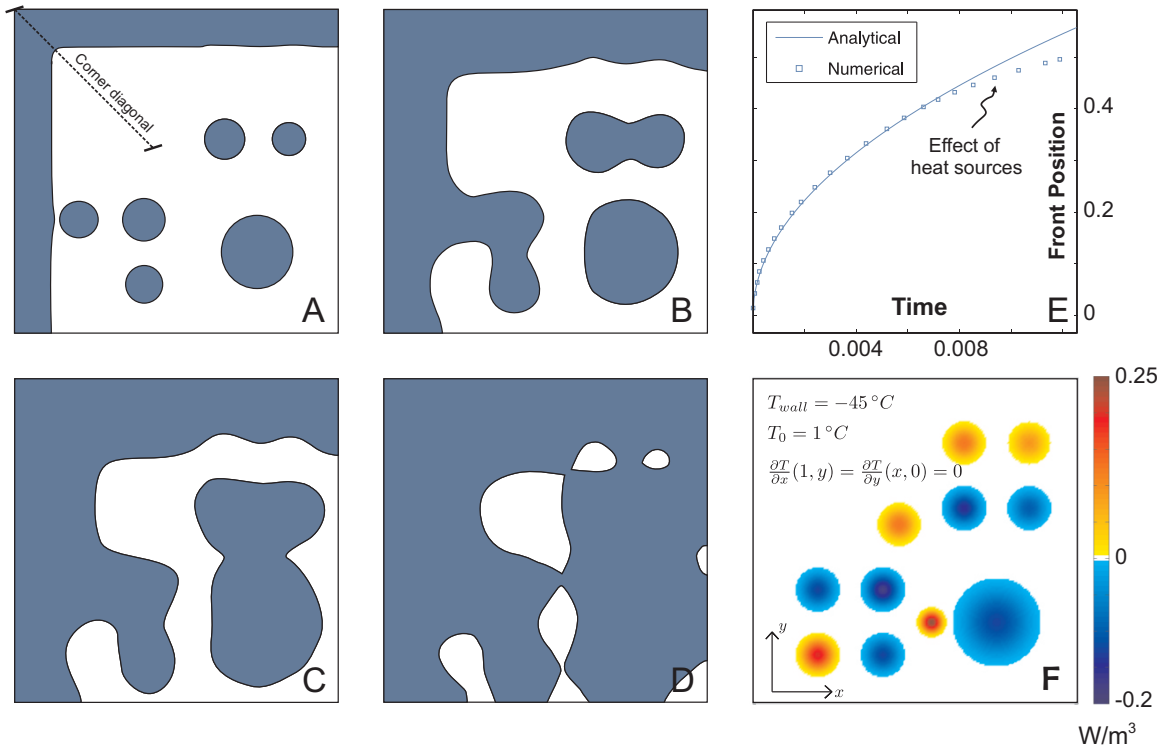

Figure 12: Sequence A-B-C-D: phase-front evolution of a two-phase heat transfer problem in a 2D semi-infinite domain (white = liquid phase, coloured = solid phase). Figure-E shows the evolution of the phase front along the diagonal highlighted in A, whereas initial and boundary conditions are summarized in figure-F, where the heat sources are also defined.

The sequence A-D in figure 12 shows the temporal evolution of the phase front, which is in agrement with the position and magnitude of the heat sources. The proposed method is able to track irregular phase-change fronts and permits the creation/removal of a phase inside another. Figure 12-F compares the position of the phase front along the corner diagonal with the reference solution obtained from [39]. The influence of the additional heat sources is evident at t > 0.008s, where the phase front is retarded due to the effect of the nearby positive heat sources. In absence of heat sources, the position of the phase front is accurately recovered (not shown here).

4.6. ADR system

In this example, we demonstrate how our approach can also be used to model systems in which chemical reactions are coupled with advection and diffusion. As a test case, we consider a closed system held at constant temperature and volume in which three different chemical species exist and interact through chemical reactions. Depending on the relative values of the rate constants describing each reaction, this initial value problem can be stiff and thus difficult to solve by standard approaches [43] [44]. Identical systems have been studied in the literature, e.g. the modelling of ozone's concentration in the atmosphere [45]. The reaction mechanism for the three chemical species is described as a coupled system of four reactions

$$
\begin{aligned}
C_1 + C_2 &\xrightarrow{k_1} C_3 \\
C_1 + C_3 &\xrightarrow{k_2} 2C_2 \\
C_2 &\xrightarrow{k_3(t)} 2C_1 \\
C_3 &\xrightarrow{k_4(t)} C_1 + C_2
\end{aligned}
\tag{12}
$$

where k_j ($j = 1, 2, 3, 4$) are the reaction rate constants. Eq. (12) can be reformulated in terms of the three reaction rates, $dC_i/dt = R_i$ ($i = 1, 2, 3$) as

$$
\begin{aligned}
R_1 &= -k_1 C_1 C_2 - k_2 C_1 C_3 + 2k_3(t) C_2 + k_4(t) C_3 \quad \text{in } \Omega_i \times (0, t_{end}) \\
R_2 &= -k_1 C_1 C_2 + 2k_2 C_1 C_3 - k_3(t) C_2 + k_4(t) C_3 \quad \text{in } \Omega_i \times (0, t_{end}) \\
R_3 &= k_1 C_1 C_2 - k_2 C_1 C_3 - k_4(t) C_3 \quad \text{in } \Omega_i \times (0, t_{end})
\end{aligned}
\tag{13}
$$

We use the following values for the reaction constants

$$
\begin{aligned}
k_1 &= 1.63 \times 10^{-16} \\
k_2 &= 4.66 \times 10^{-16} \\
k_3(t) &= \begin{cases} \exp\left(-22.62/\sin(wt)\right) & \text{if } \sin(wt) > 0 \\ 0 & \text{else} \end{cases} \\
k_4(t) &= \begin{cases} \exp\left(-7.601/\sin(wt)\right) & \text{if } \sin(wt) > 0 \\ 0 & \text{else} \end{cases}
\end{aligned}
\tag{14}
$$

With the above considerations, the complete advection-diffusion-reaction to be solved reads

$$
\begin{aligned}
\frac{\partial C_i}{\partial t} + \mathbf{v} \cdot \nabla C_i &= \nabla \cdot (D_i \nabla C_i) + R_i \quad \text{in } \Omega_i \times (0, t_{end}) \\
C_1(\mathbf{x}, 0) &= \begin{cases} 10^6 & \text{if } 0.3 \le \mathbf{x}(1) \le 0.6 \\ 0 & \text{else} \end{cases} \\
C_2(\mathbf{x}, 0) &= \begin{cases} 3.7 \times 10^6 & \text{if } 0.3 \le \mathbf{x}(1) \le 0.6 \\ 0 & \text{else} \end{cases} \\
C_3(\mathbf{x}, 0) &= \begin{cases} 10^{12} & \text{if } 0.3 \le \mathbf{x}(1) \le 0.6 \\ 0 & \text{else} \end{cases} \\
C_i(0, t) &= 0
\end{aligned}
\tag{15}
$$

where C_i is the concentration of the ith chemical species ($i = 1, 2, 3$). For simplicity, we assume a constant diffusivity $D_1 = D_2 = D_3 = D$. Note that $k_3(t)$ and $k_4(t)$ have a periodic behavior, with maximum values of 1.5×10^{-10} and 5×10^{-4}, respectively, for $t = 3/2$ s and $w = \pi/3$. These conditions simulate the dissociation of species C_2 and C_3 by an external time-dependent factor (e.g. sunlight).

The operator splitting is performed as in section 3, dividing (15) into an advection+reaction and diffusion sub-problems. The Eulerian and Lagrangian approaches are linked with the same interpolation technique described in sections 3.2.1 and 3.2.2 for each component.

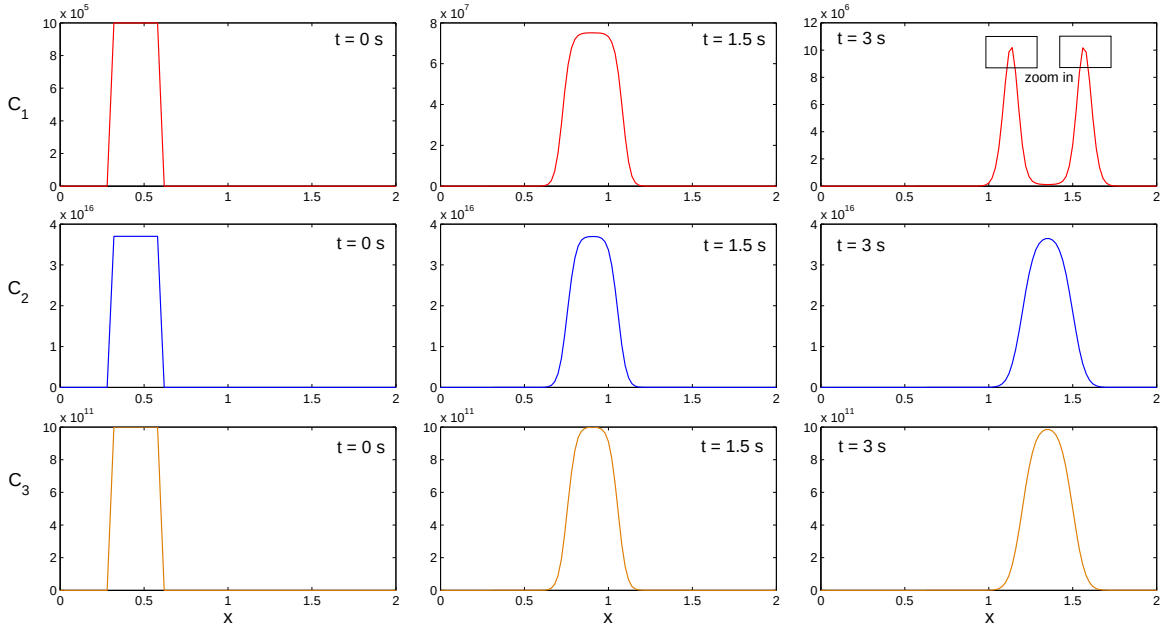

Figure 13: Spatial distribution of species at three different time steps. The curves denote concentration values at Eulerian nodes. The zoom in area is shown in 14 and includes the concentration values in the Lagrangian particles.

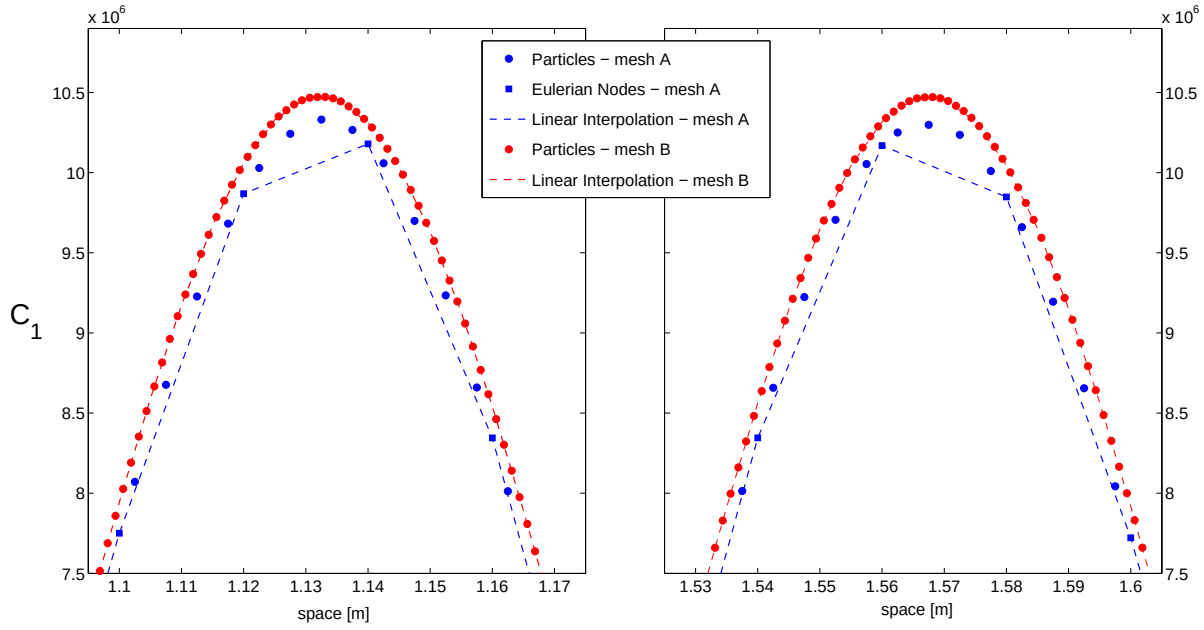

Figure 14: Zoom in area in figure 13. Mesh A [100 elements x 400 particles] is compared with the mesh B [400 elements x 1600 particles]. The particles in mesh A capture the solution at the subgrid scale.

The system (15) is solved in 1D, with $\Omega = [0, 2]$, a diffusion coefficient $D = 0.5 \times 10^{-3}$ and an advection velocity $\mathbf{v} = 0.3$. We use 100 elements in the Eulerian mesh and a total of 400 Lagrangian particles (4 particles per element). The time step Δt is set to 0.04 s. Figure 13 show the spatial distribution of each chemical species at times $t = 0, 1.5, 3$ s at the Eulerian nodes, whereas a zoom-in is shown in figure 14. Results in figure 13 are in perfect agreement with

the expected evolution of the system, including the double-peak profile of C_1 due to the structure of the rate constants $k_3(t)$ and $k_4(t)$ [45].

The performance of our interpolation scheme is shown in figure 14, where the results with mesh A [100 elements x 400 particles] are compared with a more accurate solution obtained with a finer mesh B [400 elements x 1600 particles]. Also, the importance of adding the conservative node-particle correction term (section 3.2.2 and Appendix B) to enforce mass conservation is illustrated in figure 15. The combination of particles with our grid-to-particle interpolation scheme provides a superior solution (with subgrid resolution) than the more common linear interpolation. The solution captured by the particles in mesh A reproduces closely the (correct) solution obtained with the finer mesh, but at a significantly lower computational cost. Importantly, there is no phase error between different mesh sizes, as the Lagrangian approach computes the advection exactly. The local particle-to-node interpolation, however, smooths the initial condition of the Eulerian subproblem at every time step. As a consequence, some minor clipping [46] is observed, resulting in the lost of amplitude in concave and convex zones (figure 14). Although beyond of the scope of the present paper, we mention that this effect can be easily minimized at a low cost by a local correction for the particles' values in the problematic zones using second nodal derivatives in space and the particle's *a priori* information.

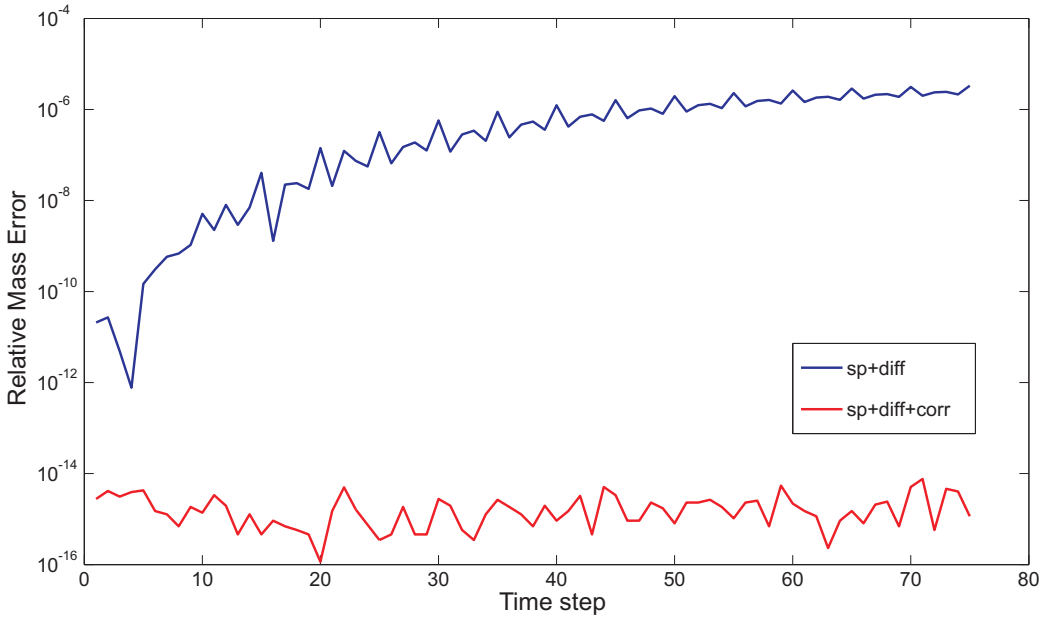

Figure 15: Relative mass error, ε_{rel}, at every time step for the different steps of our node-to-particle interpolation scheme in logarithmic scale. The relative error is computed as $\varepsilon_{rel} = \left| \frac{\sum_i \sum_p \left(C^p_{i_m} - C^p_{i_{m-1}} \right)}{\sum_i \sum_p C^p_{i_{m-1}}} \right|$. *sp+diff* refers to the particle's concentration with the shape-preserving and diffusive terms; the third step in section 3.2.2 is added in *sp+diff+corr*.

5. Conclusion

In this paper we have presented a versatile particle-based Lagrangian-Eulerian algorithm coupled with a heat source method for the solution of a large class of unsteady advection-diffusion-reaction (ADR) problems that include phase changes. The method combines the individual advantages of Lagrangian formulations for advection-dominated and coupled chemical reaction problems with those of Eulerian formulations for diffusion and phase change problems. The coupling between the Lagrangian and Eulerian subproblems is achieved with a phase change detector scheme based on a latent heat balance and a consistent/conservative interpolation technique between Lagrangian particles and the Eulerian grid. The use of an auxiliary Eulerian mesh facilitates the coupling of our method with the so-called Multiscale Finite Element Method (MsFEM) [33], as we could compute the multiscale basis functions on this

auxiliary mesh with the information coming from the particles. In addition, our algorithm offers flexibility in terms of the number of integration points to use, since the subgrid has spatially accurate information transferred from the particles, and is highly parallelizable. Adaptivity schemes for both, the Lagrangian particles and the Eulerian mesh can also be implemented in order to improve the accuracy of the method.

Numerical examples show that no numerical diffusion or oscillations are introduced with this approach for exclusively advective problems, even in the presence of sharp gradients. Our new coupling of the LE algorithm with the heat source method not only maintains the original advantages of the latter for phase change problems (e.g. relatively coarse meshes and large time steps can be used, efficiency, isothermal and non-isothermal problems), but also permits a more detailed tracking (at particle level) of highly irregular phase transition fronts in both isothermal and non-isothermal cases with or without advection. The method also performs well in advection+diffusion problems with phase change, regardless of the advection direction. Advection-diffusion with coupled chemical reactions have also been presented. These results show that our method is general, efficient, and can handle a great variety of challenging problems that arise in many scientific disciplines, including those that involve advecting sharp fronts. We believe that the simplicity, versatility and efficiency of our algorithm make it an attractive candidate for future developments of coupled codes that also include the solution of the mass and momentum equations in the modelling of complex ADR natural processes (e.g. planetary convection, groundwater reactive flow, bacterial pollution, etc).

Acknowledgements

We thank two anonymous reviewers for useful suggestions that helped us to significantly improve our manuscript. B.O. thanks support from an International Macquarie Research Excellent Scholarship, Obra Social "la Caixa" and Casa Asia. The work of JCA has been supported by two Discovery Grants from the Australian Research Council (DP110104145 and DP120102372). This is contribution XX from the Australian Research Council Centre of Excellence for Core to Crust Fluid Systems (http://www.ccfs.mq.edu.au) and YY in the GEMOC Key Centre (http://www.gemoc.mq.edu.au).

Appendix A. Heat Source Method

The internal heat source or sink term in eq. (1), Q_{PC}, can be written as,

$$Q_{PC} = \rho \frac{\mathrm{d}}{\mathrm{d}t}\left(\int_{\Omega} L \mathrm{d}\Omega\right) \tag{A.1}$$

where the density, ρ, is multiplied by the change in the amount of absorbed or released latent heat, L, over time. As the latent heat is absorbed or released according to change of temperature, the problem is non-linear. Therefore, an iterative procedure is required to compute Q_{PC}. The following lines are devoted to the description of one iteration, k, for any time step m $[t_{m-1}, t_m]$ of the original heat source method by Rolph and Bathe [23].

For every iteration k, we compute,

1. *Recover known data* from the previous iteration $k-1$, or time step, $m-1$ (table (1)).
2. *Solve* the transient diffusion-phase change problem (A.2), by a suitable solver, in order to get T_m^k,

$$\mathbf{M}\frac{\mathrm{d}\widehat{T}_m^k}{\mathrm{d}t} + \mathbf{K}\widehat{T}_m^k = \mathbf{Q}_{PC}^{k-1} \tag{A.2}$$

3. *Detection of new nodes undergoing phase change*.
   For the nodes where $Q_{flag}^{k-1} \neq 0$ and

$$T_m^{k-1} > T_{liq} > T_m^k \qquad \text{solidification}$$

$$T_m^{k-1} < T_{sol} < T_m^k \qquad \text{melting}$$

   We do $Q_{flag}^k = 0$.

Variable	Description
T_{m-1}	Temperature at each node at time t_{m-1}.
T_m^k	Temperature at each node at time t_m for the iteration k.
T_{corr}^k	Corrected temperature at each node undergoing phase change. Computed at the end of each iteration k.
T_{liq}	Liquidus temperature at each node. Initialized once in the beginning of the computation.
T_{sol}	Solidus temperature at each node. Initialized once in the beginning of the computation.
$\widehat{c_p}$	Effective heat capacity for nodes undergoing phase change.
Q_{flag}	Flag variable. -1 for liquid, 0 node undergoing phase change and 1 for solid.
Q_{MASS}	Lumped mass in each node. Initialized once in the beginning of the computation.
Q_{TOTL}	Total latent heat available in each node. Initialized once in the beginning of the computation. $Q_{TOTL} = Q_{MASS} \cdot L$
Q_{INCR}	Amount of latent heat absorbed or released for each node. Set to zero in the beginning of each iteration k
Q_{CUMU}	Amount of latent heat absorbed or released for each node. Set to zero in the beginning of the computation and updated in the iterative procedure, until $\lvert Q_{CUMU}\rvert \geq \lvert Q_{TOTL}\rvert$
Q_{PC}	Heat source vector. Set to zero in every time step and updated in the iterative procedure. The absolute value of Q_{PC} is upper limited by $\lvert Q_{TOTL}\rvert/\Delta t$, where $\Delta t = t_m - t_{m-1}$

Table A.3: Description of the variables

4. *Compute the absorbed or released latent heat* at each iteration.

   For nodes undergoing phase change for the first time at the iteration k ($Q_{flag}^{k-1} \neq 0$ and $Q_{flag}^{k} = 0$), Q_{INCR} is calculated as,

$$Q_{INCR} = \widehat{c_p}\left(T_{liq} - T_m^k\right) Q_{MASS} \qquad \text{solidification}$$

$$Q_{INCR} = \widehat{c_p}\left(T_{sol} - T_m^k\right) Q_{MASS} \qquad \text{melting}$$

   where,

$$\widehat{c_p} = \frac{1}{\left(\frac{T_{liq}-T_{sol}}{L} + \frac{1}{c_p}\right)}$$

   For the nodes that where already undergoing phase change from the previous iteration $k-1$, we do,

$$Q_{INCR} = \widehat{c_p}\left(T_{m-1} - T_m^k\right) Q_{MASS}$$

5. *Update of Q_{CUMU} and Q_{PC}* using Q_{INCR}

$$Q_{PC}^k = Q_{PC}^{k-1} + \frac{Q_{INCR}}{\Delta t}$$

$$Q_{CUMU}^k = Q_{CUMU}^{k-1} + Q_{INCR}$$

6. *Temperature correction* for the nodes undergoing phase change, $Q_{flag}^k = 0$,

$$T_{corr}^k = T_{liq} - \frac{\lvert Q_{CUMU}^k\rvert}{Q_{TOTL}}\left(T_{liq} - T_{sol}\right) \qquad \text{solidification}$$

$$T^k_{corr} = T_{sol} + \frac{|Q^k_{CUMU}|}{Q_{TOTL}}\left(T_{liq} - T_{sol}\right) \qquad \text{melting}$$

and reset the temperature values for those nodes doing $T^k_m = T^k_{corr}$.

7. *Detection of nodes with complete phase change*
   Nodes with $|Q^k_{CUMU}| \geq |Q_{TOTL}|$ have released or absorbed all the available latent heat, and they are considered to have completed the phase transition. They are correspondingly flagged $Q^k_{flag} = \pm 1$.
8. *Check convergence and prepare next iteration* using T^k_m and T^{k-1}_m.
   If convergence is not achieved or $k = 1$ go back to step 1, and do $k = k + 1$.
   Otherwise, $T_m = T^k_m$, and the iterative procedure is over.

Appendix B. Interpolation

This appendix describes the computation of the terms introduced in section 3.2.2: the shape-preserving term, $\left(\Delta T^p_m\right)_{sp}$, the subgrid diffusion, $\left(\Delta T^p_m\right)_{diff}$, and the conservative node-particle correction, $\left(\Delta T^p_m\right)_{corr}$.

Appendix B.1. Shape-preserving interpolation

Initial sharp temperature gradients, exothermic chemical reactions, phase changes etc. may lead to a heterogeneous particle temperature distribution. Through this shape-preserving term we aim to preserve this *a priori* heterogeneity at the particles after solving the Eulerian subproblem. We propose a geometrical operation based on temperature values of the particles relative to the temperatures of the surrounding nodes. Figure 2 d) shows the contribution of the shape-preserving increment to the particles, where the sharp gradient is preserved and each of the particle's value is adapted to the new nodal values.

In order to define the relative values or ratios, we require the initial nodal values, T^{IJ}_{m-1}, the particles' values after solving (5), $T^p_{m^*}$, and the linear/bilinear interpolation of initial nodal values into the particles' new positions, $T^{p_{in}}_{m-1}$. For each particle p and a **neighboring** node IJ, the expression for the ratio X^p_{IJ} reads,

$$X^p_{IJ} = \frac{T^p_{m^*} - T^{IJ}_{m-1}}{T^p_{m^*} - T^{p_{in}}_{m-1}}$$

where T^{IJ}_{m-1} is the value of the surrounding node, T^p_{m-1} is the particle value and $T^{p_{in}}_{m-1}$ is the linear/bilinear interpolation of nodal values into the particle's position at time t_{m-1}. From the previous expression we can recover the particle's initial value as,

$$T^p_{m^*} = \frac{T^{p_{in}}_{m-1} X^p_{IJ} - T^{IJ}_{m-1}}{X^p_{IJ} - 1}$$

In analogy to the expression above and using the notation related with the new nodal values (subindex m), we could preserve a specific ratio related to one node, X^p_{IJ}, and write a particle's new value at t_m as,

$$T^p_m = T^p_{m^*} + \Delta T^p_{IJ} = \frac{T^{p_{in}}_m X^p_{IJ} - T^{IJ}_m}{X^p_{IJ} - 1}$$

where $T^{p_{in}}_m$ is the linear/bilinear interpolation of new nodal values into the particle new position. After some algebra we can obtain the incremental value needed to preserve a certain nodal ratio X^p_{IJ}, ΔT^p_{IJ},

$$\Delta T^p_{IJ} = \frac{\left(T^{p_{in}}_m - T^{p_{in}}_{m-1}\right) X^p_{IJ} - \left(T^{IJ}_m - T^{IJ}_{m-1}\right)}{X_{IJ} - 1}$$

This incremental value is, in general, different for each X^p_{IJ}. This means that a different ΔT^p_{IJ} should be added to a given particle p in order to preserve all the ratios X^p_{IJ}. Since this is impossible to do, we propose the following weighted expression,

$$\left(\Delta T_m^p\right)_{sp} = \sum_{IJ} w_{IJ}^p \Delta T_{IJ}^p$$

where $w_{IJ}^p = \left(1 - \frac{\Delta x_{IJ}^p}{\Delta x}\right)\left(1 - \frac{\Delta y_{IJ}^p}{\Delta y}\right)$, being Δx_{IJ}^p and Δy_{IJ}^p the cartesian distance of each particle to the surrounding nodes (see figure 1).

Appendix B.2. Subgrid diffusion

The previous shape-preserving interpolation is a "rigid" transformation (see figure 2). The present subgrid diffusion term, $\left(\Delta T_m^p\right)_{diff}$, mimics the effect of diffusion at the particles' scale without having to solve the system of equations (4). It is computed using the actual thermal diffusivity of the particles as follows,

$$\left(\Delta T_m^p\right)_{diff} = \Delta T_\diamond^p \left[1 - \exp\left(-d\frac{\Delta t}{\Delta t_{diff}}\right)\right] \tag{B.1}$$

where $\Delta T_\diamond^p = T_m^{p_{in}} - \left(T_{m^*}^p + \left(\Delta T_m^p\right)_{sp}\right)$ is the difference between the temperatures predicted by the shape-preserving interpolation (i.e. no diffusion) and by a linear interpolation (i.e. infinite diffusion), $\Delta t_{diff} = \min\{\Delta x, \Delta y\}^2 / (2\kappa)$, $\Delta t = t_m - t_{m-1}$, κ is the diffusivity and d is a dimensionless scalar (here $d = 0.1$). The nature of the previous expression resides in the analytical heat flow solution for 1D semi-infinite domains for constant initial temperature ([41], page 59), where $-d\frac{\Delta t}{\Delta t_{diff}}$ is a dimensionless parameter controlling how fast temperature diffuses. At this stage, for $\kappa \to \infty$, we get from (10) $T_m^p = T_m^{p_{in}}$. This means that for very diffusive problems this step recovers the linear/bilinear interpolation solution. On the contrary, for $\kappa = 0$ we obtain $\left(\Delta T_m^p\right)_{diff} = 0$, and no subgrid diffusion is introduced to the particles. See section 4.5 for the case of mass diffusion.

Appendix B.3. Conservative node-particle correction

ADR processes that explicitly account for changes in thermal energy and/or mass should be conservative in these quantities. This last correction term ensures a conservative node-to-particle balance in our particle-based LE scheme. We first describe the correction for enthalpy and then extend it for the case of mass conservation.

The governing equations for diffusion + phase change are explicitly solved only in the Eulerian mesh. The total enthalpy change after each time step is

$$\Delta H = m \cdot c_p \cdot \Delta T \tag{B.2}$$

where m and c_p are the total mass and the average heat capacity of the system, and ΔT is the temperature change between two consecutive time steps. A thermodynamically-consistent node-to-particle interpolation scheme requires that the ΔH computed in the Eulerian subproblem is consistently transferred to the particles. In other words, any temperature field associated with the particles must satisfy the constraint of being consistent with the global ΔH computed in the Eulerian subproblem. Since the previous two terms ($\left(\Delta T_m^p\right)_{sp}$ and $\left(\Delta T_m^p\right)_{diff}$) are independent of this constrain, there would be a residual R_H between the ΔH computed at the Eulerian mesh and that obtained from the particles

$$R_H = \sum_p m^p \cdot c_p^p \cdot \left(\Delta T_m^p\right)_{corr} = \sum_I m^I \cdot c_p^I \cdot \Delta T^I - \sum_p m^p \cdot c_p^p \cdot \left(\left(\Delta T_m^p\right)_{sp} + \left(\Delta T_m^p\right)_{diff}\right) \tag{B.3}$$

R_H represents the enthalpy residual that needs to be distributed to the particles. This correction can be computed in several ways. For instance, we could uniformly transfer this residual to the particles. However, this approach would (incorrectly) change the temperature of particles located in regions where no real temperature change has been computed (i.e. their neighbour nodal temperature has not been modified after the Eulerian approach). This leads to

physically meaningless results at the particle scale, and the possible creation of new maxima/minima. Therefore, we propose a local correction that is applied *only* if $p \subset [\left(\Delta T_m^p\right)_{sp} \cup \left(\Delta T_m^p\right)_{diff} \neq 0]$.

For this purpose, let us subdivide the particles into two groups depending on whether their $\left(\Delta T_m^p\right)_{diff}$ is positive or negative. Then, if $R_H > 0$ (i.e. temperature/enthalpy deficit), only those particles for which $\left(\Delta T_m^p\right)_{diff} > 0$ are corrected by adding more subgrid diffusion. Conversely, if $R_H > 0$ (i.e. temperature/enthalpy excess), only those particles with $\left(\Delta T_m^p\right)_{diff} < 0$ are corrected by subtracting subgrid diffusion. Besides being physically realistic, this scheme guarantees that no artificial minimum/maxima will be created in the particles. The actual correction is achieved with the aid of two proportionality constants X_{add} and X_{rem}, defined as

$$\begin{aligned} X_{add} &= \frac{R_H}{\sum_{p\subset **} m_p \cdot c_p \cdot \Delta T_\diamond^p} \quad \text{if } R_H > 0 \\ X_{rem} &= \frac{R_H}{\sum_{p\subset \star\star} m_p \cdot c_p \cdot \Delta T_\diamond^p} \quad \text{if } R_H < 0 \end{aligned} \tag{B.4}$$

where $(**)$ refers to $\left(\Delta T_m^p\right)_{diff} > 0$ and $(\star\star)$ to $\left(\Delta T_m^p\right)_{diff} < 0$. $\Delta T_\diamond^p$ has been defined in B.2 and it represents a maximum attainable diffusion for each particle. The correction $\left(\Delta T_m^p\right)_{corr}$ is thus defined as

$$\left(\Delta T_m^p\right)_{corr} = \begin{cases} X_{add} \cdot \Delta T_\diamond^p & \text{if } p \subset \left(\Delta T_m^p\right)_{diff} > 0 \\ X_{rem} \cdot \Delta T_\diamond^p & \text{if } p \subset \left(\Delta T_m^p\right)_{diff} < 0 \\ 0 & \text{else} \end{cases} \tag{B.5}$$

Since each particle has a different value of $\Delta T_\diamond^p$, $\left(\Delta T_m^p\right)_{corr}$ will in general be different for every particle. Notice that $\left(\Delta T_m^p\right)_{corr}$ enforces the desired consistency between the enthalpy change computed at the Eulerian nodes and that transferred into the particles,

$$\sum_p m^p \cdot c_p^p \cdot \left(\Delta T_m^p\right)_{corr} = \sum_p m^p \cdot c_p^p \cdot X \cdot \Delta T_\diamond^p = \sum_p \left(\frac{m^p \cdot c_p^p \cdot R_H \cdot \Delta T_\diamond^p}{\sum_p m^p \cdot c_p^p \cdot \Delta T_\diamond^p} \right) = R_H$$

The above procedure can be equally applied for a mass conservation problem (e.g. the ADR system involving chemical reactions between different species in section 4.6). For instance, the residual in B.3 is rewritten as,

$$R_H = -\sum_i \sum_p \Delta C_{i_m}^p \tag{B.6}$$

where $\Delta C_{i_m}^p = C_{i_m}^p - C_{i_{m-1}}^p$ refers to the change in mass of species i in a single time step (note that $C_{i_m}^p$ already includes the shape preserving and diffusion corrections). R_H then represents the mass residual that needs to be distributed to the particles. Similarly to B.4 and B.5, we define a correction over $\Delta C_{i_m}^p$ as

$$\begin{aligned} \Delta C_{i_m}^p &= X_i^{total} \Delta C_{i_m}^p \\ X_i^{total} &= \frac{R_H}{\sum_p |\Delta C_{i_m}^p|} \end{aligned} \tag{B.7}$$

Appendix C. Analytical Solution for melting and solidification in 1D semi-infinite domain

Solidification and melting are the most important cases in which a substance changes phase while emitting or absorbing heat. There is a moving surface of separation between the two phases and its evolution needs to be determined

as heat is absorbed or released in there. There are few exact solutions for this problem [41]. The governing equations to be satisfied for a one dimensional isothermal melting or solidification problems in linear flow are,

$$T_{p1} = T_{p2} = T_{pc} \quad \text{when } x = \chi(t) \tag{C.1}$$

$$k_{p1}\frac{\partial T_{p1}}{\partial x} - k_{p2}\frac{\partial T_{p2}}{\partial x} = L\rho\frac{d\chi}{dt} \tag{C.2}$$

$$\frac{\partial^2 T_{p1}}{\partial x^2} - \frac{1}{\kappa_{p1}}\frac{\partial T_{p1}}{\partial t} = 0 \tag{C.3}$$

$$\frac{\partial^2 T_{p2}}{\partial x^2} - \frac{1}{\kappa_{p2}}\frac{\partial T_{p2}}{\partial t} = 0 \tag{C.4}$$

where c, k and κ are the thermal constants for heat capacity, thermal conductivity and diffusivity, and the subindexes $p1$ and $p2$ refer to the solid and liquid phase respectively. Volume change due to the phase change is neglected in C.1 (density, ρ, remains constant in both phases). $\chi(t)$ is the position of the interface between the solid and liquid phases, where the region $x < \chi(t)$ contains solid at temperature $T_{p1} < T_{pc}$ and $x > \chi(t)$ contains liquid at temperature $T_{p2} < T_{pc}$. T_{pc} is the temperature where the phase change occurs.

Equation C.1 has an analytical solution for the following case: liquid at initial constant temperature T_0, 0 surface temperature in $x = 0$ for $t > 0$, and additional boundary conditions $T_{p2} \to T_0$, as $x \to \infty$ and $T_{p1}(0, t) = 0$. This particular analytical solution reads,

$$T_{p1} = \frac{T_{pc}}{\text{erf}\,[\lambda]}\text{erf}\left[\frac{\mathbf{x}}{2\left(\kappa_{p1}t\right)^{\frac{1}{2}}}\right] \tag{C.5}$$

$$T_{p2} = T_0 - \frac{\left(T_0 - T_{pc}\right)}{\text{erfc}\left[\lambda\left(\kappa_{p1}/\kappa_{p2}\right)^{\frac{1}{2}}\right]}\text{erfc}\left[\frac{\mathbf{x}}{2\left(\kappa_{p2}t\right)^{\frac{1}{2}}}\right] \tag{C.6}$$

where λ is a numerical constant determined by,

$$\frac{e^{-\lambda^2}}{\text{erf}\lambda} - \frac{k_{p2}}{k_{p1}}\frac{\kappa_{p1}{}^{\frac{1}{2}}}{\kappa_{p2}{}^{\frac{1}{2}}}\frac{\left(T_0 - T_{pc}\right)e^{-\lambda^2\frac{\kappa_{p1}}{\kappa_{p2}}}}{T_{pc}\text{erfc}\left[\lambda\left(\frac{\kappa_{p1}}{\kappa_{p2}}\right)^{\frac{1}{2}}\right]} = \frac{\lambda L\pi^{\frac{1}{2}}}{c_{p1}T_{pc}} \tag{C.7}$$

References

[1] M. Celia, T. Russell, I. Herrera, R. Ewing, An eulerian-lagrangian localized adjoint method for the advection-diffusion equation, Advances in water resources 13 (4) (1990) 187–206.

[2] A. Staniforth, J. Côté, Semi-lagrangian integration schemes for atmospheric models-a review, Monthly Weather Review 119 (9) (1991) 2206–2223.

[3] E. Varoglu, W. Liam Finn, Finite elements incorporating characteristics for one-dimensional diffusion-convection equation, Journal of Computational Physics 34 (3) (1980) 371–389.

[4] J. Douglas, Jr, T. F. Russell, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, SIAM Journal on Numerical Analysis 19 (5) (1982) 871–885.

[5] A. Smolianski, O. Shipilova, H. Haario, A fast high-resolution algorithm for linear convection problems: particle transport method, International journal for numerical methods in engineering 70 (6) (2007) 655–684.

[6] T. J. Hughes, L. P. Franca, G. M. Hulbert, A new finite element formulation for computational fluid dynamics: Viii. the galerkin/least-squares method for advective-diffusive equations, Computer Methods in Applied Mechanics and Engineering 73 (2) (1989) 173–189.

[7] R. Codina, Comparison of some finite element methods for solving the diffusion-convection-reaction equation, Computer Methods in Applied Mechanics and Engineering 156 (1) (1998) 185–210.

[8] L. Moresi, F. Dufour, H.-B. Mhlhaus, A lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials, Journal of Computational Physics 184 (2) (2003) 476 – 497. doi:http://dx.doi.org/10.1016/S0021-9991(02)00031-1.
URL `http://www.sciencedirect.com/science/article/pii/S0021999102000311`

[9] O. Shipilova, H. Haario, A. Smolianski, Particle transport method for convection problems with reaction and diffusion, International Journal for Numerical Methods in Fluids 54 (10) (2007) 1215–1238. doi:10.1002/fld.1438.
URL http://dx.doi.org/10.1002/fld.1438
[10] D. Sulsky, S.-J. Zhou, H. L. Schreyer, Application of a particle-in-cell method to solid mechanics, Computer Physics Communications 87 (1) (1995) 236–252.
[11] S. Bardenhagen, Energy conservation error in the material point method for solid mechanics, Journal of Computational Physics 180 (1) (2002) 383–403.
[12] I. Faragó, J. Geiser, Iterative operator-splitting methods for linear problems, International Journal of Computational Science and Engineering 3 (4) (2007) 255–263.
[13] J. Geiser, Iterative operator-splitting methods with higher-order time integration methods and applications for parabolic partial differential equations, Journal of Computational and Applied Mathematics 217 (1) (2008) 227–242.
[14] I. Faragó, A modified iterated operator splitting method, Applied mathematical modelling 32 (8) (2008) 1542–1551.
[15] C.-J. Ho, R. Viskanta, Heat transfer during melting from an isothermal vertical wall, Journal of heat transfer 106 (1) (1984) 12–19.
[16] A. Gadgil, D. Gobin, Analysis of two-dimensional melting in rectangular enclosures in presence of convection, Journal of heat transfer 106 (1) (1984) 20–26.
[17] K. Morgan, A numerical analysis of freezing and melting with convection, Computer Methods in Applied Mechanics and Engineering 28 (3) (1981) 275–284.
[18] J. Hsiao, An efficient algorithm for finite-difference analyses of heat transfer with melting and solidification, Numerical Heat Transfer 8 (6) (1985) 653–666.
[19] A. J. Dalhuijsen, A. Segal, Comparison of finite element techniques for solidification problems, International Journal for Numerical Methods in Engineering 23 (10) (1986) 1807–1829. doi:10.1002/nme.1620231003.
URL http://dx.doi.org/10.1002/nme.1620231003
[20] K. Morgan, R. W. Lewis, O. C. Zienkiewicz, An improved algrorithm for heat conduction problems with phase change, International Journal for Numerical Methods in Engineering 12 (7) (1978) 1191–1195. doi:10.1002/nme.1620120710.
URL http://dx.doi.org/10.1002/nme.1620120710
[21] R. Lewis, P. Roberts, Finite element simulation of solidification problems, Applied Scientific Research 44 (1-2) (1987) 61–92. doi:10.1007/BF00412007.
URL http://dx.doi.org/10.1007/BF00412007
[22] E. Lemmon, Multidimensional integral phase change approximations for finite element conduction codes, Numerical methods in heat transfer.(A 82-28551 13-34) Chichester, Sussex, England and New York, Wiley-Interscience, 1981, (1981) 201–213.
[23] W. D. Rolph, K.-J. Bathe, An efficient algorithm for analysis of nonlinear heat transfer with phase changes, International Journal for Numerical Methods in Engineering 18 (1) (1982) 119–134. doi:10.1002/nme.1620180111.
URL http://dx.doi.org/10.1002/nme.1620180111
[24] J. Roose, O. Storrer, Modelization of phase changes by fictitious-heat flow, International Journal for Numerical Methods in Engineering 20 (2) (1984) 217–225. doi:10.1002/nme.1620200203.
URL http://dx.doi.org/10.1002/nme.1620200203
[25] H.-C. Huang, A. S. Usmani, Finite element analysis for heat transfer, second ed Edition, American Mathematical Society, Clarendon Press, 1994.
[26] S. R. Runnels, G. F. Carey, Finite element simulation of phase change using capacitance methods, Numerical Heat Transfer, Part B Fundamentals 19 (1) (1991) 13–30.
[27] W. Bushko, I. Grosse, New finite element method for multidimensional phase change heat transfer problems, Numerical Heat Transfer, Part B Fundamentals 19 (1) (1991) 31–48.
[28] Y. Cao, A. Faghri, A numerical analysis of phase-change problems including natural convection, Journal of heat transfer 112 (3) (1990) 812–816.
[29] Y. Chen, Y.-T. Im, Z.-H. Lee, Three dimensional finite element analysis with phase change by temperature recovery method, International Journal of Machine Tools and Manufacture 31 (1) (1991) 1–7.
[30] V. Voller, C. Swaminathan, B. Thomas, Fixed grid techniques for phase change problems: a review, International Journal for Numerical Methods in Engineering 30 (4) (1990) 875–898.
[31] C. Swaminathan, V. Voller, A general enthalpy method for modeling solidification processes, Metallurgical Transactions B 23 (5) (1992) 651–664. doi:10.1007/BF02649725.
URL http://dx.doi.org/10.1007/BF02649725
[32] C. Swaminathan, V. Voller, Towards a general numerical scheme for solidification systems, International journal of heat and mass transfer 40 (12) (1997) 2859–2868.
[33] T. Y. Hou, X.-H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, Journal of computational physics 134 (1) (1997) 169–189.
[34] B. Fornberg, A practical guide to pseudospectral methods, Vol. 1, Cambridge university press, 1998.
[35] J. Donea, A. Huerta, Finite element methods for flow problems, John Wiley & Sons, 2003.
[36] J. F. Kanney, C. T. Miller, C. Kelley, Convergence of iterative split-operator approaches for approximating nonlinear reactive transport problems, Advances in Water Resources 26 (3) (2003) 247 – 261. doi:http://dx.doi.org/10.1016/S0309-1708(02)00162-8.
URL http://www.sciencedirect.com/science/article/pii/S0309170802001628
[37] V. John, E. Schmeyer, Finite element methods for time-dependent convection–diffusion–reaction equations with small diffusion, Computer methods in applied mechanics and engineering 198 (3) (2008) 475–494.
[38] R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM Journal on Numerical Analysis 33 (2) (1996) 627–665.
[39] K. Davey, R. Mondragon, A non-physical enthalpy method for the numerical solution of isothermal solidification, International Journal for

Numerical Methods in Engineering 84 (2) (2010) 214–252.
[40] K. Davey, N. Rodriguez, A control volume capacitance method for solidification modelling with mass transport, International journal for numerical methods in engineering 53 (12) (2002) 2643–2671.
[41] H. Carslaw, J. Jaeger, Conduction of Heat in Solids, second ed Edition, Oxford Science Publications, Clarendon Press, 1992.
[42] K. A. Rathjen, L. M. Jiji, Heat conduction with melting or freezing in a corner, Journal of Heat Transfer 93 (1) (1971) 101–109.
[43] P. Colella, A. Majda, V. Roytburd, Theoretical and numerical structure for reacting shock waves, SIAM Journal on Scientific and Statistical Computing 7 (4) (1986) 1059–1080.
[44] B. Zhang, H. Liu, F. Chen, J. H. Wang, The equilibrium state method for hyperbolic conservation laws with stiff reaction terms, Journal of Computational Physics 263 (2014) 151–176.
[45] D. Kahaner, C. Moler, S. Nash, G. Forsythe, Numerical methods and software, Prentice-Hall series in computational mathematics, Prentice Hall, 1989.
URL http://books.google.com.au/books?id=jipEAQAAIAAJ
[46] D. Kuzmin, Explicit and implicit fem-fct algorithms with flux linearization, Journal of Computational Physics 228 (7) (2009) 2517 – 2534. doi:http://dx.doi.org/10.1016/j.jcp.2008.12.011.
URL http://www.sciencedirect.com/science/article/pii/S0021999108006475