

## A Space-Time Discontinuous Galerkin Spectral Element Method for the Stefan problem

Chaoxu Pei · Mark Sussman ·  
M. Yousuff Hussaini

Received: / Accepted:

**Abstract** A space-time Discontinuous Galerkin (DG) spectral element method is presented to solve the Stefan problem in an Eulerian coordinate system. In this scheme, the level set method describes the time evolving interface. To deal with the prior unknown interface, two global transformations, a backward transformation and a forward transformation, are introduced in the space-time mesh. The idea of the transformations is to combine an Eulerian description, i.e. a fixed frame of reference, with a Lagrangian description, i.e. a moving frame of reference. The *backward transformation* maps the unknown time-varying interface in the fixed frame of reference to a known stationary interface in the moving frame of reference. In the moving frame of reference, the transformed governing equations, written in the space-time framework, are discretized by a DG spectral element method in each space-time slab. The *forward transformation* is used to update the level set function and then to project the solution in each phase onto the new corresponding time-dependent domain. Two options for calculating the interface velocity are presented, and both methods exhibit spectral accuracy. Benchmark tests in one spatial dimension indicate that the method converges with spectral accuracy in both space and time for both the temperature distribution and the interface velocity. The interrelation between the interface position and the temperature makes the Stefan problem a nonlinear problem; a Picard iteration algorithm is introduced in order to solve the nonlinear algebraic system of equations and it is found that only a few iterations lead to convergence.

**Keywords** Space-time · Discontinuous Galerkin · Spectral accuracy · Cut-cell · Mapping method · the Stefan problem

**Mathematics Subject Classification (2010)**

---

C. Pei · M. Sussman (✉) · M.Y. Hussaini  
Department of Mathematics, Florida State University, Tallahassee, FL, 32306, USA.  
E-mail: sussman@math.fsu.edu (✉)

## 1 Introduction

The Stefan problem is a moving boundary problem that is used to model phase change, e.g. the freezing and thawing process for a solid-liquid system [1,2]. A key issue for the Stefan problem is that the interface between different phases is evolving in time. This requires solving the heat equations for different phases, i.e. the ice and water phases, while the phase boundary separating the two phases is transported with a velocity that is proportional to the jump of the normal heat flux at the evolving and prior unknown boundary. In other words, the problem requires one to find the solutions in a prior unknown domain and to compute the shape of the unknown domain as a part of the solution. Hence, these problems are nonlinear due to the coupling between unknown geometric quantities, i.e. the growing interfacial shape, and the temperature distribution. Such a problem arise in numerous industrial and technological applications, such as metal solidification, food freezing, dendritic crystal growth in a super-cooled liquid and ice accretion on aircraft [3–7].

Various numerical methods have been proposed for solving the Stefan problem. These methods can be broadly classified into two categories by the description of the phase interface, which can be represented explicitly or implicitly. With explicit representation, i.e. front tracking and Lagrangian methods, the interface is tracked as it moves through the computational domain. Rønquist and Patera [8] presented a continuous Galerkin spectral element method combined with an interface-local transformation, in which the element boundaries coincide with the phase interface. They showed exponential convergence in space and first order in time in one dimensional numerical tests. In contrast to generating a conforming grid, the following authors [9–11] proposed a hybrid method in an Eulerian coordinate system, in which the interface is tracked sharply while the treatment of the interface is done by the immersed boundary method [3]. One drawback for tracking the interface explicitly is that special care has to be taken when topological changes such as merging or breaking occur. To overcome this difficulty, interface capturing methods, such as the phase-field method [12,13] or the level set method [14,15], have been introduced to solve the Stefan problem. The phase-field method is a diffuse interface method in which the interface is given a certain thickness. However, two drawbacks of the phase-field method for the Stefan problem were pointed out by the authors [16]. One is that the grid resolution requirements within the interfacial region is much more stringent with the phase-field method than with the level set method. The other drawback is that the high dependence on the physical parameters in the phase-field method makes this method a challenge when one adds new physics to the model.

In this paper, we use a sharp interface model, the level set method, which enable one to exactly locate the interface. Other examples of sharp interface methods are the immersed interface method [17,18], the ghost fluid method [19], the XFEM method [20], the virtual node method [21,22], the coupled level set and volume of fluid method [23], the coupled level set and moment of fluid method [24], the cell integrated Semi-Lagrangian with moment of fluid ap-

proach [25] and sharp interface Cartesian grid finite volume method [26]. The earliest level set method for solidification was presented in [5]. Their method combined the level set method with a boundary integral method. Chen et al. [27] introduced a simple approach in an Eulerian coordinate system for the Stefan problem, which involved the level set method and a finite difference approach for solving the heat equation in each phase and for evaluating the interface velocity. The interface boundary condition was imposed directly by constructing interpolating polynomials on both sides of the interface. Improvements were made by Gibou et al. [28, 29]. In contrast to [27], the discretizations in [28, 29] led to a symmetric linear system. In addition, Gibou et al. [28, 29] pointed out the necessity to redefine the grid points that are swept over by the interface, i.e. a liquid node crossed by the interface must be converted to a ice node with an appropriate temperature value. Later, a high order nonsymmetric approach, a third order accurate method, was presented for solving the Stefan problem by Gibou and Fedkiw [30], in which a high order extrapolation approach [31] was applied to redefine the values at nodes that are swept over by the interface.

To solve the Stefan problem numerically, we develop a space-time Discontinuous Galerkin (DG) spectral element method in an Eulerian coordinate system, which is combined with the level set method. In order to redefine the values at nodes in the phase change region, we introduce two global transformations: a backward transformation and a forward transformation. The idea of the transformations is similar to a semi-Lagrangian method, which distinguishes our method from the interface-local transformation [8]. In the space-time mesh, we first apply the backward transformation to map the unknown time-varying interface in the Eulerian coordinate system to a known stationary interface in a new coordinate system. Such a mapping transforms the heat equation, coupled with the interface condition, defined in a prior unknown time-dependent domain, into a nonlinear convection-diffusion equation defined in a known stationary domain. To solve the transformed nonlinear equation, we employ a Picard iteration, i.e. [32, 33]. At each iteration, we therefore solve a linear advection-diffusion equation by a space-time DG spectral element method, and the level set function is updated by the forward transformation associated with the previous interface velocity. After the interface position is updated in the Eulerian coordinate system, the solution is then projected upon the updated time-dependent domain.

There are several reasons why a DG spectral element method is preferred over a Continuous Galerkin spectral element method, such as [8]. Firstly, the piecewise continuous basis functions enable one to apply the DG spectral element method on a nonconforming grid. In other words, our method allows for hanging nodes that can appear due to a non-continuous interface reconstruction or appear at the junction of coarse and fine grids when Adaptive Mesh Refinement is employed [34–37]. Secondly, the high degree of locality enable one to choose different values of the polynomial order in different elements, which is referred to as  $p$ -refinement. Thirdly, the auxiliary variables introduced in a DG method for discretizing the diffusion term [38, 39] can be used directly

to compute the interface velocity by the weak formulation, which enables the temperature distribution and the interface velocity to be computed simultaneously. This is different from the method presented in [8], in which the interface velocity was computed by a strong form, i.e. a spectral collocation method. A comparison of the two methods for calculating the interface velocity is shown in section 6 below.

DG methods in space have been developed for solving a wide range of physical problems, especially for dealing with flows with discontinuities computed on unstructured meshes. Detailed surveys can be found in e.g. [40–44]. Recently, space-time DG finite element methods have been explored by the following authors [45–49] for problems which require moving and deforming meshes. Space-time DG finite element methods, which automatically satisfy the Geometric Conservation Law, are unconditionally stable. Space-time DG schemes result in implicit time integration schemes and naturally allow for time adaptivity. Motivated by these properties of space-time DG finite element methods, we present a space-time DG spectral element method for solving the Stefan problem, which leads to spectral accuracy in both space and time.

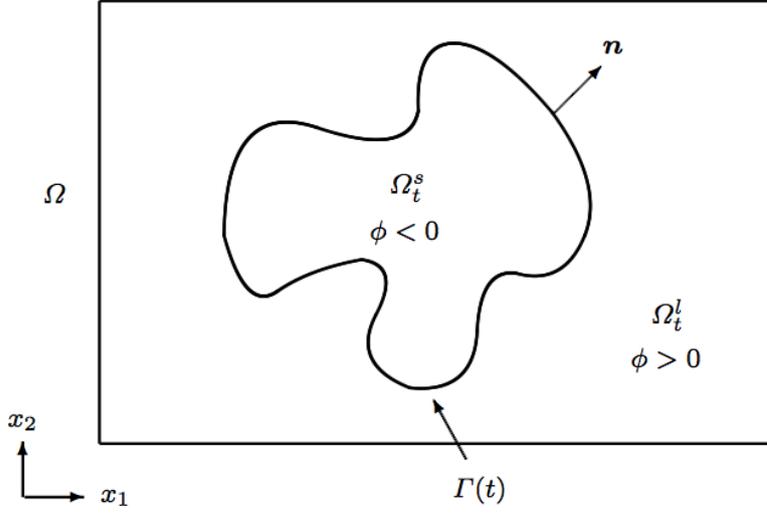
The organization of the paper is as follows. In section 2, we first present the Governing equations of the Stefan problem and introduce the level set function for capturing the phase boundary. The space-time discretization is introduced in section 3. Transformation techniques are then introduced in section 4. In section 5, the DG spectral element method is presented for the transformed equations in the space-time framework. Then two new algorithms for determining the interface velocity are described in section 5.4. Numerical tests are presented in one spatial dimension in section 6 to gauge the accuracy and efficiency of our new numerical scheme.

## 2 Mathematical model

The Stefan problem describes the temperature distribution of a pure material undergoing a phase transition. Let us consider a square domain,  $\Omega \in \mathbb{R}^d$ , of a pure material but at different states, such as ice and water. Let  $\mathbf{x} = (x_1, x_2, \dots, x_d)$  be the spatial variable, where  $d$  is the spatial dimension. The domain  $\Omega$  is composed of a time-dependent solid region, denoted as  $\Omega_t^s$ , and a time-dependent liquid region, denoted as  $\Omega_t^l$ . The interface, separating different phases, is a prior unknown free boundary, which is denoted as  $\Gamma(t)$  such that  $\Gamma(t) = \Omega_t^s \cap \Omega_t^l$ . The description of the Stefan problem on a Cartesian domain,  $\Omega \in \mathbb{R}^2$ , is illustrated in Fig. 1.

### 2.1 Governing equations

We assume that convection, thermal expansion and buoyancy are not considered. Since there is no flow in the liquid region, the energy for the Stefan



**Fig. 1** The Stefan problem on a Cartesian domain  $\Omega \in \mathbb{R}^2$ .  $\Omega_t^s$  and  $\Omega_t^l$  are time-dependent domains at time  $t$ , while domain  $\Omega$  is fixed, such that  $\Omega = \Omega_t^s \cup \Omega_t^l$ .  $\Gamma(t)$  is the moving interface at time  $t$ , and  $\mathbf{n} = (n_1, n_2)$  is the interface normal pointing in the direction of interface movement.  $\phi$  is the level set function introduced in section 2.2

problem is determined by the internal energy,

$$e = c_v \theta + LH(\phi), \quad (1)$$

where  $c_v$  denotes the specific heat capacity,  $\theta$  denotes the temperature distribution,  $L$  is a positive constant representing the latent heat of fusion and  $H(\phi)$  is a Heaviside function defined as

$$H(\phi) = \begin{cases} 0, & \text{if } \phi(\mathbf{x}, t) \leq 0 \text{ and } \mathbf{x} \in \Omega \setminus \Omega_t^l, \\ 1, & \text{if } \phi(\mathbf{x}, t) > 0 \text{ and } \mathbf{x} \in \Omega_t^l, \end{cases}$$

where  $\phi(\mathbf{x}, t)$  is the level set function introduced in section 2.2.

To derive the governing equation for each phase, we start from the conservation of energy, which is governed by

$$\frac{\partial(\rho e)}{\partial t} = \nabla \cdot (k \nabla \theta). \quad (2)$$

Substituting (1) into (2), we obtain,

$$\frac{\partial(\rho^s c_v^s \theta^s)}{\partial t} = \nabla \cdot (k^s \nabla \theta^s), \quad \text{in } \Omega_t^s, \quad (3)$$

$$\frac{\partial(\rho^l c_v^l \theta^l)}{\partial t} = \nabla \cdot (k^l \nabla \theta^l), \quad \text{in } \Omega_t^l, \quad (4)$$

where  $\rho$  is the density,  $c_v$  denotes the specific heat capacity,  $k$  is the thermal conductivity, and indices “ $s$ ”, “ $l$ ” denote the solid and liquid phase, respectively.

At the interface, the standard Rankine-Hugoniot jump condition of Eq. (2) is

$$V_n [\rho e]_\Gamma = - [k \nabla \theta \cdot \mathbf{n}]_\Gamma, \quad (5)$$

where  $V_n = \mathbf{V} \cdot \mathbf{n}$  is the normal velocity at the interface, and  $[f]_\Gamma$  denotes the jump of  $f$  across  $\Gamma$ , i.e.,  $[f]_\Gamma = (f^l)|_\Gamma - (f^s)|_\Gamma$ . By substituting (1) into (5), we obtain the jump condition at the interface with the assumption of  $\rho^s = \rho^l = \rho$  and  $c_v^s = c_v^l = c_v$ ,

$$\rho L V_n = -(k^l \nabla \theta^l - k^s \nabla \theta^s) \cdot \mathbf{n}. \quad (6)$$

Eq. (6) is commonly referred to as the *Stefan condition* that expresses the normal velocity of the moving interface as a function of the normal derivative of the temperature evaluated at both sides of the interface.

On  $\partial\Omega$ , Dirichlet boundary conditions are specified. At the interface, we set

$$\theta^s = \theta_\Gamma = \theta^l. \quad (7)$$

For a classical Stefan problem,  $\theta_\Gamma = \theta_m$  where  $\theta_m$  denotes the melting temperature. However, for a realistic modeling, such as dendritic solidification, it is important to take surface tension effects and interface curvature into account, which leads to the Gibbs-Thomson relation [27],  $\theta_\Gamma = -\epsilon_c \kappa - \epsilon_v V_n$ , where  $\kappa$  denotes the curvature at the front,  $\epsilon_c$  is the surface tension coefficient, and  $\epsilon_v$  is the molecular kinetic coefficient.

We refer the reader to [7, 50, 28] and the references therein for information on freezing models and numerical methods that take into account more detailed physics than we do in this article.

## 2.2 Level set function

We represent the time evolving interface by a level set function. The level set function  $\phi$  is a continuous function defined as follows

$$\begin{cases} \phi(\mathbf{x}, t) < 0, & \mathbf{x} \in \Omega_t^s, \\ \phi(\mathbf{x}, t) = 0, & \mathbf{x} \in \Gamma(t), \\ \phi(\mathbf{x}, t) > 0, & \mathbf{x} \in \Omega_t^l. \end{cases}$$

The evolution of the level set function is determined by the following equation,

$$\phi_t + \mathbf{W} \cdot \nabla \phi = 0, \quad (8)$$

where  $\mathbf{W}$  is a continuous extension of the normal velocity  $V_n$  off the interface. The outward unit normal pointing in the direction of interface movement is

denoted as  $\mathbf{n} = (n_1, n_2, \dots, n_d)$ , which can be determined by the level set function  $\phi$  as follows,

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \quad (9)$$

where  $\nabla$  is used for the spatial gradient operator, and defined as  $\nabla = (\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_d})$ .

The level function  $\phi$  is initialized as a signed distance function

$$\phi(\mathbf{x}, 0) = \begin{cases} -d_\Gamma, & \mathbf{x} \in \Omega_0^s, \\ 0, & \mathbf{x} \in \Gamma(0), \\ +d_\Gamma, & \mathbf{x} \in \Omega_0^l, \end{cases}$$

where  $d_\Gamma$  is the distance of a point  $\mathbf{x}$  to the interface  $\Gamma$ . Updating the level set function by Eq. (8), may change the level set function away from being a distance function. So it is necessary to perform a reinitialization procedure to keep the level set function as a distance function, which satisfies  $|\nabla\phi| = 1$ . (cf. [51])

### 3 Space-time discretization

In this section, we follow the notation of e.g. [45, 33] to introduce the definitions of the space-time domain, space-time slabs and space-time elements.

In a space-time discretization, we introduce a space-time domain  $\mathcal{E}$  by considering the domain  $\mathcal{E} = \Omega \times [t_0, T]$  in  $\mathbb{R}^{d+1}$ , where  $d$  denotes the spatial dimension. The coordinates of a point  $\bar{\mathbf{x}} \in \mathcal{E}$  are defined as  $\bar{\mathbf{x}} = (\mathbf{x}, x_{d+1})$  with the spatial variables  $\mathbf{x} = (x_1, x_2, \dots, x_d)$  and the time variable  $t = x_{d+1}$ . The space-time domain boundary  $\partial\mathcal{E}$  consists of the hyper-surfaces  $\Omega_0 := \{\bar{\mathbf{x}} \in \partial\mathcal{E} \mid x_{d+1} = t_0\}$ ,  $\Omega_T := \{\bar{\mathbf{x}} \in \partial\mathcal{E} \mid x_{d+1} = T\}$ , and  $\mathcal{Q} := \{\bar{\mathbf{x}} \in \partial\mathcal{E} \mid t_0 < x_{d+1} < T\}$ .

First, we partition the time interval  $[t_0, T]$  by the time levels  $0 = t_0 < t_1 < \dots < t_{E^{(t)}} = T$ . The space-time domain  $\mathcal{E}$  is then divided into  $E^{(t)}$  space-time slabs. The  $n$ -th space-time slab is denoted as  $\mathcal{E}^n = \mathcal{E} \cap I_n$ , where  $I_n = [t_n, t_{n+1}]$  is the  $n$ -th time interval with length  $\Delta t_n = t_{n+1} - t_n$ . Also, the space-time slab  $\mathcal{E}^n$  is bounded by  $\Omega_{t_n}$ ,  $\Omega_{t_{n+1}}$  and  $\mathcal{Q}^n = \partial\mathcal{E}^n \setminus (\Omega_{t_n} \cup \Omega_{t_{n+1}})$ .

Now, we describe the construction of the space-time elements in the space-time slab  $\mathcal{E}^n$ . Divide the spatial domain  $\Omega_{t_n}$  into  $E^{(\mathbf{x})}$  non-overlapping spatial elements  $K^n$  with a uniformed size  $h$  and similarly for the domain  $\Omega_{t_{n+1}}$  with  $E^{(\mathbf{x})}$  spatial elements  $K^{n+1}$ . A space-time element  $\mathcal{K}^n$  is then obtained by connecting  $K^n$  and  $K^{n+1}$  via linear interpolation. The element boundary  $\partial\mathcal{K}$  contains  $K^n$ ,  $K^{n+1}$  and  $\mathcal{Q}_{\mathcal{K}}^n = \partial\mathcal{K} \setminus (K^n \cup K^{n+1})$ . On  $\partial\mathcal{K}$ , the outward unit space-time normal is denoted by  $\bar{\mathbf{n}}_{\mathcal{K}} = (\mathbf{n}, n_{d+1})$ , where  $\mathbf{n} = (n_1, n_2, \dots, n_d)$  is the spatial part of the space-time normal while  $n_{d+1}$  is the time component.

The Stefan problem on a Cartesian domain  $\Omega$  involves two different phases, i.e.  $\Omega_t^s$  and  $\Omega_t^l$ . Thus, the space-time elements are classified into two categories: regular cells and cut-cells.

- *Regular cells.* A space-time element is referred to as a regular cell if it contains only one phase. We denote a regular cell as  $\mathcal{K}_j^{n,s}$  if the space-time element  $\mathcal{K}_j^n$  is in the solid phase. Similarly,  $\mathcal{K}_j^{n,l}$  denotes a regular cell which contains only liquid phase.
- *Cut-cells.* A cell is called a cut-cell if a regular cell intersects with the moving front. In particular, a cut-cell contains different phases, which are separated by the phase boundary. Here, we denote a cut-cell as  $\mathcal{K}_j^{n,sl}$  if the space-time element  $\mathcal{K}_j^n$  intersects with the front  $\Gamma$ . The subelements associated with the cut-cell  $\mathcal{K}_j^{n,sl}$  are denoted as  $\mathcal{K}_j^{n,s,\Gamma}$  and  $\mathcal{K}_j^{n,l,\Gamma}$ , such that  $\mathcal{K}_j^{n,sl} = \mathcal{K}_j^{n,s,\Gamma} \cup \mathcal{K}_j^{n,l,\Gamma}$ .

To generate a cut-cell in a space-time slab  $\mathcal{E}^n$ , one can partition the time interval  $[t_n, t_{n+1}]$  by a set of Gauss-Lobatto points  $(\tau_0, \tau_1, \dots, \tau_p)$ , and then determine the zero level set crossing values  $\Gamma(\tau_i)$  ( $i = 0, \dots, p$ ). By connecting  $\Gamma(\tau_i)$  using high order interpolation in time, we introduce a curve  $\mathcal{Q}_\Gamma^n$  cutting a regular cell into two subelements, i.e.  $\mathcal{K}_j^{n,s,\Gamma}$  and  $\mathcal{K}_j^{n,l,\Gamma}$ . (See Fig. 2).

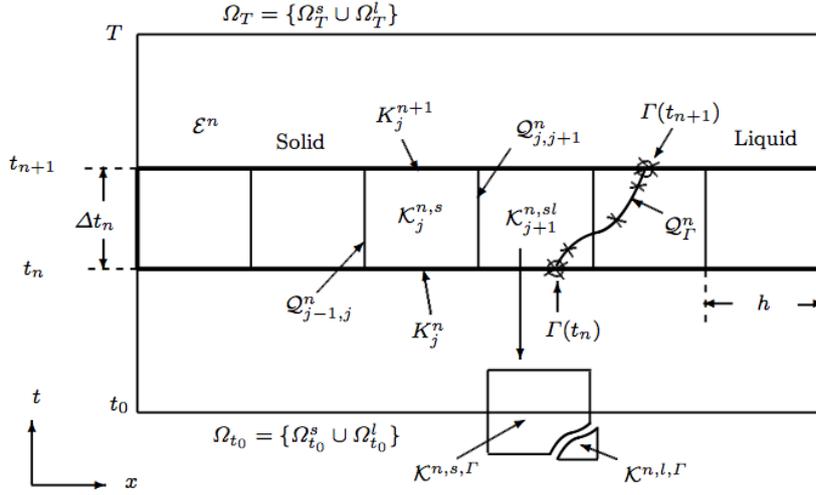
In the space-time slab  $\mathcal{E}^n$ , the tessellation of the solid (liquid) phase is defined as  $\mathcal{T}_h^{n,s}$  ( $\mathcal{T}_h^{n,l}$ ), which consists of all regular cells,  $\mathcal{K}^{n,s}$  ( $\mathcal{K}^{n,l}$ ), and subelements of cut-cells,  $\mathcal{K}^{n,s,\Gamma}$  ( $\mathcal{K}^{n,l,\Gamma}$ ). Then the tessellation of the space-time slab  $\mathcal{E}^n$  is defined as  $\mathcal{T}_h^n$ , such that  $\mathcal{T}_h^n = \mathcal{T}_h^{n,s} \cup \mathcal{T}_h^{n,l}$ . In addition, the tessellation of the space-time domain  $\mathcal{E}$  is denoted as  $\mathcal{T}_h = \cup_n \mathcal{T}_h^n$ . In a space-time domain  $\mathcal{E} \in \mathbb{R}^2$ , a sketch of a space-time slab  $\mathcal{E}^n$  with the moving interface  $\Gamma(t)$  is demonstrated in Fig. 2.

#### 4 Fixed frame of reference versus moving frame of reference

In a space-time slab  $\mathcal{E}^n$ , a cut-cell  $\mathcal{K}_j^{n,sl}$  is generated due to the moving front  $\mathcal{Q}_\Gamma^n$  intersecting or embedding within a space-time element. Some examples of a cut-cell are shown in Fig. 3. A subelement generated in a cut-cell could be a triangle, a quadrilateral or a pentagon. One way to handle non-quadrilateral subelements is to perform the element refinement rule, which is to divide pentagons into triangles. Thus, the computational grid in each space-time slab is a mixed mesh with both triangles and quadrilaterals. In this approach, the approximation of the curve  $\mathcal{Q}_\Gamma^n$  needs to be calculated at the beginning, then one can generate a mixed grid and update the temperature distributions.

An alternative approach to handle non-quadrilateral subelements is to define a mapping, which is to transform a time-varying front into a constant front. In one dimension, a computational grid with a constant front in a space-time slab is preferred over the one with a time-varying front due to a constant front leads to a rectangular space-time mesh. From Fig. 4, we see that the curve  $\mathcal{Q}_\Gamma^n$  is transformed to a straight line which is either aligned with an inter-element boundary (Fig. 4(b)) or cutting a regular cell into two subelements (Fig. 4(d)).

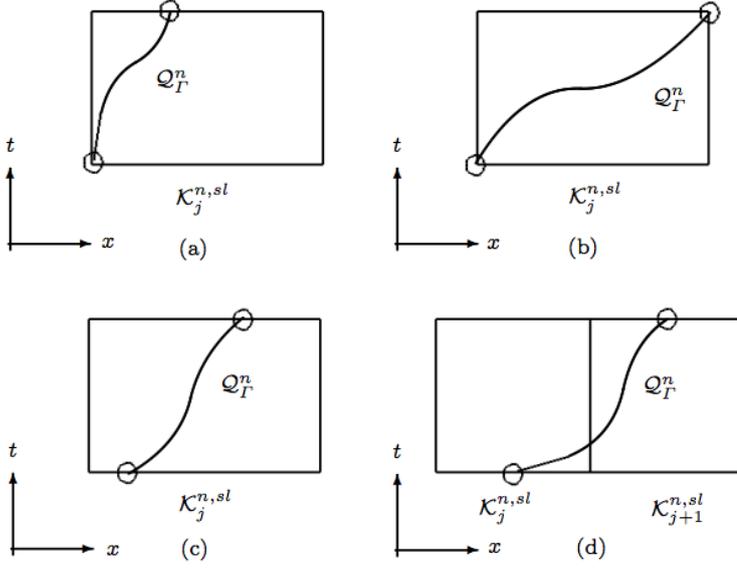
Now, we present the mapping technique that we employed in the space-time framework. First, we introduce two coordinate systems.



**Fig. 2** An Example of a space-time domain,  $\mathcal{E} = \Omega \times [t_0, T] \in \mathbb{R}^2$ , with the coordinates of a point  $\tilde{\mathbf{x}}$  denoted by  $\tilde{\mathbf{x}} = (x, x_2)$ , where  $x = x_1$  is the spatial variables and  $x_2$  denotes the time variable. The square with bold lines is an example of a space-time slab  $\mathcal{E}^n$ , such that  $\mathcal{E}^n = \mathcal{E} \cap I_n$ , where  $I_n = [t_n, t_{n+1}]$ . We assume that the solid phase is on the left hand side of the interface  $\mathcal{Q}_F^n$ , while the liquid is on the other side. The space-time element  $\mathcal{K}_j^{n,s}$  is referred to as a regular cell in the solid phase with boundary  $\partial\mathcal{K}_j^{n,s} = \{\mathcal{Q}_{\mathcal{K}_j^{n,s}}^n \cup K_j^n \cup K_j^{n+1}\}$ , where  $\mathcal{Q}_{\mathcal{K}_j^{n,s}}^n = \{\mathcal{Q}_{j-1,j}^n \cup \mathcal{Q}_{j,j+1}^n\}$ . However,  $\mathcal{K}_{j+1}^{n,sl}$  and  $\mathcal{K}_{j+2}^{n,sl}$  are examples of a cut-cell due to the time evolving interface,  $\mathcal{Q}_F^n$ , defined on a nonconforming computational grid. In this example, the curve  $\mathcal{Q}_F^n$  obtained by 4th order interpolation cuts  $\mathcal{K}_{j+1}^{n,sl}$  into two subelements, i.e. a pentagon  $\mathcal{K}_{j+1}^{n,s,\Gamma}$  and a triangle  $\mathcal{K}_{j+1}^{n,l,\Gamma}$ .

- *Fixed frame of reference.* The fixed frame of reference is the frame that we use to define the physical problem. The coordinates of the fixed frame of reference are denoted as  $(\mathbf{x}, t)$ , where  $\mathbf{x} = (x_1, x_2, \dots, x_d)$  is the spatial variables. (See Fig. 4(a, c)).
- *Moving frame of reference.* The moving frame of reference is the coordinate system that we use to define the transformed physical problem. In the moving frame of reference, the coordinates are defined as  $(\tilde{\mathbf{x}}, t)$ , where  $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_d)$  is the spatial variable. (See Fig. 4(b, d))

Next, we construct a mapping  $\tilde{\mathbf{x}} = \mathcal{M}^b(\mathbf{x}, t)$  between the moving frame of reference and the fixed frame of reference. In particular, the mapping will take a point on the time-varying front in the fixed frame of reference and transform it to a point on the stationary front relative to the moving frame of reference. For example,  $\Gamma(t_n)$  and  $\Gamma(t_{n+1})$  in Fig. 4(a or c) denote the zero level set at times  $t_n$  and  $t_{n+1}$ , respectively. The transformation  $\mathcal{M}^b(\mathbf{x}, t)$  transforms  $\Gamma(t_n)$  and  $\Gamma(t_{n+1})$  to  $\tilde{\Gamma}(t_n)$  and  $\tilde{\Gamma}(t_{n+1})$  in Fig. 4(b or d), such that  $\tilde{\Gamma}(t_n) = \Gamma(t_n)$  and  $\tilde{\Gamma}(t_{n+1}) = \Gamma(t_n)$ . Thus, in a space-time slab  $\mathcal{E}^n$ , our



**Fig. 3** Examples of a cut-cell in a space-time domain  $\mathcal{E} \in \mathbb{R}^2$ . The moving front is evolving from left to right in each case. The shape of the generated subelements could be triangle, quadrilateral or pentagon.

transformation  $\tilde{\mathbf{x}} = \mathcal{M}^b(\mathbf{x}, t)$  is determined by

$$\begin{cases} \mathcal{M}_t^b + (\nabla \mathcal{M}^b) \mathbf{W} = 0, \\ \mathcal{M}^b(\mathbf{x}, t_n) = \mathbf{x}, \end{cases} \quad (10)$$

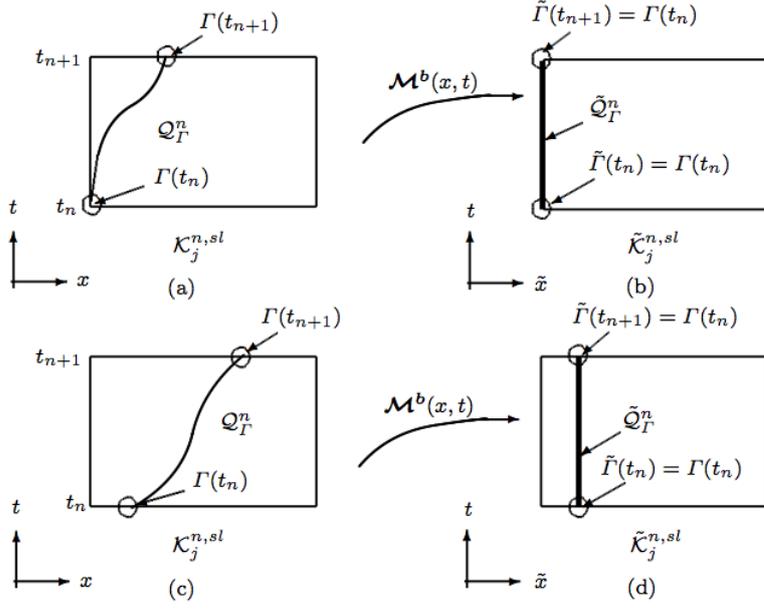
where  $\nabla$  denotes the gradient operator with respect to  $\mathbf{x}$ , and  $\mathbf{W}$  is the extension of the normal velocity  $V_n$  off the interface  $\Gamma$ , which is defined by

$$\mathbf{W}(\mathbf{x}, t) = V_n(\mathbf{x}^*, t), \quad (11)$$

where  $\mathbf{x}^*$  is the closest point on  $\Gamma(t)$  to the coordinates of a point  $(\mathbf{x}, t)$ . Since the zero level set of the *transformed* front is determined by the front at time  $t_n$ , we refer to this transformation as the *backward* mapping transformation. Note that matrix  $\nabla \mathcal{M}^b$  is the Jacobian of the transformation, which is defined by

$$\nabla \mathcal{M}^b = \nabla \tilde{\mathbf{x}} = \begin{pmatrix} \frac{\partial \tilde{x}_1}{\partial x_1} & \frac{\partial \tilde{x}_1}{\partial x_2} \\ \frac{\partial \tilde{x}_2}{\partial x_1} & \frac{\partial \tilde{x}_2}{\partial x_2} \end{pmatrix}. \quad (12)$$

After updating the temperature distribution in the moving frame of reference and calculating the updated interface curve  $\mathcal{Q}_\Gamma^n$ , we need to project the solutions back onto the fixed frame of reference. This projection  $\mathbf{x} = \mathcal{M}^f(\tilde{\mathbf{x}}, t)$



**Fig. 4** Examples of a mapping technique applied to two kinds of a cut-cell in a space-time domain  $\mathcal{E} \in \mathbb{R}^2$ . In each case, the curve  $\mathcal{Q}_I^n$  is mapped to a straight bold line  $\tilde{\mathcal{Q}}_I^n$ , which is either aligned with an inter-element boundary or cutting a regular cell into two subelements.

in a space-time slab  $\mathcal{E}^n$  is called the *forward* mapping transformation and is defined by

$$\begin{cases} \mathcal{M}^f_t - (\tilde{\nabla} \mathcal{M}^f) \mathbf{W} = 0, \\ \mathcal{M}^f(\tilde{\mathbf{x}}, t_n) = \tilde{\mathbf{x}}, \end{cases} \quad (13)$$

where  $\tilde{\nabla}$  denotes the gradient operator with respect to  $\tilde{\mathbf{x}}$ .

In one spatial dimension, we apply the method of characteristics to solve the transformations (10) and (13), and obtain,

– the backward transformation,

$$\tilde{x}(x, t) = x - \int_{t_n}^t W(s) ds, \quad (14)$$

– the forward transformation,

$$x(\tilde{x}, t) = \tilde{x} + \int_{t_n}^t W(s) ds. \quad (15)$$

Note that  $W$  is defined in Eq. (11).

#### 4.1 The transformed equations in the moving frame of reference

In this section, we describe a general approach to transform the heat equations in the fixed frame of reference, Eqs. (3) and (4), into the moving frame of reference. In the moving frame of reference, the coordinates of a point is denoted as  $(\tilde{\mathbf{x}}, t)$ . As the mapping procedure is identical in both phases, we consider only the solid phase, Eq. (3), and drop the index “s”,

$$\frac{\partial(\rho c_v \theta)}{\partial t} = \nabla \cdot (k \nabla \theta). \quad (16)$$

Given a mapping  $\tilde{\mathbf{x}} = \mathcal{M}^b(\mathbf{x}, t)$ , We find the effect of the mapping on Eq. (16) by use of the chain rule. First, the time derivative of  $\theta$  transforms as,

$$\frac{\partial(\rho c_v \theta)}{\partial t} = \frac{\partial(\rho c_v \tilde{\theta})}{\partial t} + \frac{\partial \tilde{\mathbf{x}}}{\partial t} \cdot \tilde{\nabla}(\rho c_v \tilde{\theta}), \quad (17)$$

where  $\tilde{\theta}$  denotes the temperature distribution in the moving frame of reference. For the gradient operator  $\nabla \theta \in \mathbb{R}^2$ , we have

$$\nabla \theta = \begin{pmatrix} \frac{\partial \theta}{\partial x_1} \\ \frac{\partial \theta}{\partial x_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial \tilde{x}_1}{\partial x_1} & \frac{\partial \tilde{x}_2}{\partial x_1} \\ \frac{\partial \tilde{x}_1}{\partial x_2} & \frac{\partial \tilde{x}_2}{\partial x_2} \end{pmatrix} \begin{pmatrix} \frac{\partial \tilde{\theta}}{\partial \tilde{x}_1} \\ \frac{\partial \tilde{\theta}}{\partial \tilde{x}_2} \end{pmatrix} = (\nabla \tilde{\mathbf{x}})^\top \tilde{\nabla} \tilde{\theta}, \quad (18)$$

where superscript “ $\top$ ” denotes the matrix transpose, and  $\nabla \tilde{\mathbf{x}}$  is the Jacobian of the transformation, which is defined in Eq. (12). Similarly, we have

$$\tilde{\nabla} \tilde{\theta} = (\tilde{\nabla} \mathbf{x})^\top \nabla \theta. \quad (19)$$

It follows that

$$\nabla \tilde{\mathbf{x}} = (\tilde{\nabla} \mathbf{x})^{-1}. \quad (20)$$

From (18), (19) and (20), it then follows that

$$\nabla \theta = ((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta}. \quad (21)$$

The transformation for the divergence operator,  $\nabla \cdot \mathbf{F} \in \mathbb{R}^2$ , is defined as

$$\begin{aligned} \nabla \cdot \mathbf{F} &= \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2}, \\ &= \left( \frac{\partial \tilde{F}_1}{\partial \tilde{x}_1} \frac{\partial \tilde{x}_1}{\partial x_1} + \frac{\partial \tilde{F}_1}{\partial \tilde{x}_2} \frac{\partial \tilde{x}_2}{\partial x_1} \right) + \left( \frac{\partial \tilde{F}_2}{\partial \tilde{x}_1} \frac{\partial \tilde{x}_1}{\partial x_2} + \frac{\partial \tilde{F}_2}{\partial \tilde{x}_2} \frac{\partial \tilde{x}_2}{\partial x_2} \right), \\ &= \tilde{\nabla} \tilde{\mathbf{F}} : (\nabla \tilde{\mathbf{x}})^\top, \\ &= \tilde{\nabla} \tilde{\mathbf{F}} : ((\tilde{\nabla} \mathbf{x})^{-1})^\top, \end{aligned} \quad (22)$$

where

$$\tilde{\nabla} \tilde{\mathbf{F}} = \begin{pmatrix} \tilde{\nabla} \tilde{F}_1 \\ \tilde{\nabla} \tilde{F}_2 \end{pmatrix} = \begin{pmatrix} \frac{\partial \tilde{F}_1}{\partial \tilde{x}_1} & \frac{\partial \tilde{F}_1}{\partial \tilde{x}_2} \\ \frac{\partial \tilde{F}_2}{\partial \tilde{x}_1} & \frac{\partial \tilde{F}_2}{\partial \tilde{x}_2} \end{pmatrix}, \quad (23)$$

and the symbol “ $\cdot$ ” is defined as  $\mathbf{A} : \mathbf{B} = \sum_j \sum_i a_{i,j} b_{i,j}$ , where  $\mathbf{A} = (a_{i,j})_{d \times d}$  and  $\mathbf{B} = (b_{i,j})_{d \times d}$  are matrices. Hence, by applying the transformations (17), (21) and (22), the heat equation (16) can be rewritten as

$$\frac{\partial(\rho c_v \tilde{\theta})}{\partial t} + \frac{\partial \tilde{\mathbf{x}}}{\partial t} \cdot \tilde{\nabla}(\rho c_v \tilde{\theta}) = \tilde{\nabla}(k((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta}) : ((\tilde{\nabla} \mathbf{x})^{-1})^\top. \quad (24)$$

From the definition of the backward mapping in Eq. (10), we have

$$\frac{\partial \tilde{\mathbf{x}}}{\partial t} = -(\tilde{\nabla} \mathbf{x})^{-1} \mathbf{W}, \quad (25)$$

where  $\mathbf{W}$  is defined in Eq. (11). The transformed heat equation is then written as

$$\frac{\partial(\rho c_v \tilde{\theta})}{\partial t} - ((\tilde{\nabla} \mathbf{x})^{-1} \mathbf{W}) \cdot \tilde{\nabla}(\rho c_v \tilde{\theta}) = \tilde{\nabla}(k((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta}) : ((\tilde{\nabla} \mathbf{x})^{-1})^\top. \quad (26)$$

Using vector identities, the second term and the third term in Eq. (26) are written as,

$$((\tilde{\nabla} \mathbf{x})^{-1} \mathbf{W}) \cdot \tilde{\nabla}(\rho c_v \tilde{\theta}) = \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1} \mathbf{W} \rho c_v \tilde{\theta}) - \rho c_v \tilde{\theta} \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1} \mathbf{W}), \quad (27)$$

$$\begin{aligned} \tilde{\nabla}(k((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta}) : ((\tilde{\nabla} \mathbf{x})^{-1})^\top &= \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1} k((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta}) \\ &\quad - (k((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta}) \cdot \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1})^\top, \end{aligned} \quad (28)$$

where  $\tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1})^\top$  is defined as

$$\tilde{\nabla} \cdot ((\tilde{\nabla} \tilde{\mathbf{x}})^\top) = \begin{pmatrix} \tilde{\nabla} \cdot \left( \frac{\partial \tilde{\mathbf{x}}}{\partial x_1} \right) \\ \tilde{\nabla} \cdot \left( \frac{\partial \tilde{\mathbf{x}}}{\partial x_2} \right) \end{pmatrix}. \quad (29)$$

Thus, the transformed heat equation in the moving frame of reference is written as

$$\frac{\partial(\rho c_v \tilde{\theta})}{\partial t} - \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1} \mathbf{W} \rho c_v \tilde{\theta}) = \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1} k((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta}) + N.C.T., \quad (30)$$

where  $N.C.T.$  denotes the sum of the non conservative terms, and is defined as follows

$$N.C.T. = \rho c_v \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1}) \mathbf{W} - (k((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta}) \cdot \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1})^\top. \quad (31)$$

Now, we reformulate the transformed heat equation (30) in the space-time framework. In the space-time domain, we introduce a vector function  $\mathbf{B} \in \mathbb{R}^{d+1}$  and a matrix  $\mathbf{A} \in \mathbb{R}^{(d+1) \times (d+1)}$ , as

$$\mathbf{B} = (-(\tilde{\nabla} \mathbf{x})^{-1} \mathbf{W}, 1), \quad \mathbf{A} = \begin{pmatrix} k((\tilde{\nabla} \mathbf{x})^{-1})^\top \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix}, \quad (32)$$

where  $\mathbf{0} \in \mathbb{R}^{d \times 1}$ . Then the transformed heat equation (30) is transformed into a space-time formulation as,

$$\nabla \cdot (\rho c_v \mathbf{B} \tilde{\theta} - (\tilde{\nabla} \mathbf{x})^{-1} \mathbf{A} \nabla \tilde{\theta}) = N.C.T., \quad (33)$$

where  $\nabla = (\frac{\partial}{\partial \tilde{x}_1}, \dots, \frac{\partial}{\partial \tilde{x}_d}, \frac{\partial}{\partial \tilde{x}_{d+1}})^\top$  denotes the gradient operator in the space-time domain. Note that the term  $\mathbf{A} \nabla \tilde{\theta}$  is a critical component when calculating the normal velocity at the interface. More detail is discussed in section 5.4.

In one spatial dimension, we have  $\tilde{\nabla} \mathbf{W} = 0$ . According to the transformations (14) and (15), we obtain

$$\nabla \tilde{\mathbf{x}} = (\tilde{\nabla} \mathbf{x})^{-1} = 1. \quad (34)$$

Using the vector identities, the term  $N.C.T.$  equals zero due to

$$\tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1})^\top = 0, \quad (35)$$

$$\tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1} \mathbf{W}) = \tilde{\nabla} \mathbf{W} : ((\tilde{\nabla} \mathbf{x})^{-1})^\top + \mathbf{W} \cdot \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1})^\top = 0. \quad (36)$$

The transformed heat equation is then written in the conservative form as

$$\frac{\partial(\rho c_v \tilde{\theta})}{\partial t} - \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1} \mathbf{W} \rho c_v \tilde{\theta}) = \tilde{\nabla} \cdot ((\tilde{\nabla} \mathbf{x})^{-1} k((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta}), \quad (37)$$

Hence, the space-time formulation of Eq. (37) is written as,

$$\nabla \cdot (\rho c_v \mathbf{B} \tilde{\theta} - (\tilde{\nabla} \mathbf{x})^{-1} \mathbf{A} \nabla \tilde{\theta}) = 0. \quad (38)$$

*Remark 1* In higher dimensions, it will be necessary to reduce the spatial and temporal order as well as reduce  $\Delta t_n$  if the condition number of  $\tilde{\nabla} \mathbf{x}$  exceeds a threshold. One can also use the condition number of  $\tilde{\nabla} \mathbf{x}$  as an indicator for adaptive mesh refinement.

*Remark 2* In one spatial dimension, the velocity extension,  $W$ , is a function of time only so that (13) reduces to a linear differential equation; in this case we can use (15) to find the forward mapping. In multiple dimensions, (13) would have to be solved by the method of characteristics where a Picard iteration technique (e.g. [32,33]) is applied in order to find the arrival points of the characteristics. At each iteration, we solve

$$\frac{\partial}{\partial t} (\mathbf{x}^{(k+1)}(\tilde{\mathbf{x}}, t)) = \mathbf{W}(\mathbf{x}^{(k)}(\tilde{\mathbf{x}}, t), t), \quad (39)$$

with the initial guess  $\mathbf{x}^{(0)}(\tilde{\mathbf{x}}, t) = \tilde{\mathbf{x}}$ , which is the initial condition of Eq. (13).

## 5 Space-time DG spectral element method

In this section, we describe our space-time DG spectral element method for solving the Stefan problem.

### 5.1 Outline of the space-time DG spectral element method

We present an outline of our space-time DG spectral element method for solving the Stefan problem in Algorithm 1. Note that the transformed heat equation (37) in the moving frame of reference is a nonlinear equation since  $\mathbf{W}$  in (11) and  $\mathbf{B}$  in (32) are functions of the interfacial temperature gradients evaluated from both sides of the interface. We use the following expression to indicate the nonlinear property,

$$\mathbf{W} = \mathbf{W}(\tilde{\nabla}\tilde{\theta}), \quad \mathbf{B} = \mathbf{B}(\tilde{\nabla}\tilde{\theta}). \quad (40)$$

To solve such a nonlinear equation, we use a Picard iteration scheme [32,33] for which at each Picard iteration the linear advection-diffusion equation has to be solved in the space-time framework.

---

**Algorithm 1** the space-time discontinuous Galerkin spectral element method for solving the Stefan problem in a space-time slab  $\mathcal{E}^n$ .

---

- 1: Given  $\theta^{n,s}$ ,  $\theta^{n,l}$  and  $\phi^n$  at time  $t_n$  in the space-time slab  $\mathcal{E}^n$ .
- 2: Compute the normal velocity  $V_n^n$  at the interface from Eq. (6). Details are given in section 5.4.
- 3: Define a space-time slab  $\tilde{\mathcal{E}}^n$  in the moving frame of reference by introducing the backward transformation  $\tilde{\mathbf{x}} = \mathcal{M}^b(\mathbf{x}, t)$  (Eq. (10)) and forward transformation  $\mathbf{x} = \mathcal{M}^f(\tilde{\mathbf{x}}, t)$  (Eq. (13)). Here, the temperature values in the solid and liquid phases are denoted as  $\tilde{\theta}^{n,s}$  and  $\tilde{\theta}^{n,l}$ , respectively.
- 4: In  $\tilde{\mathcal{E}}^n$ , apply a Picard iteration scheme [32,33] to solve (38) in each phase, and let the initial guess of the normal velocity to be  $V_n^{j,(k)} = V_n^n$ , where  $k = 0$  and  $j = 1, \dots, p^{(t)}$ . Note that  $V_n^{p^{(t)},(k)} = V_n^{n+1,(k)}$ . In each phase, we solve

$$\nabla \cdot (\rho c_v \mathbf{B}(\tilde{\nabla}\tilde{\theta}^{(k)})\tilde{\theta}^{(k+1)} - (\tilde{\nabla}\tilde{\mathbf{x}})^{-1} \mathbf{A} \nabla \tilde{\theta}^{(k+1)}) = 0. \quad (41)$$

- 5: **while** ( $\|V_n^{n+1,(k+1)} - V_n^{n+1,(k)}\| > tol$ ) **do**
    - i Use the DG spectral element method to solve the linear equation (41) in each phase. The discretization of (41) will be discussed in section 5.3.
    - ii Applying the forward transformation  $\mathcal{M}^{f,(k)}(\tilde{\mathbf{x}}, t)$  defined in (13), we obtain the level set function  $\phi^{j,(k)}$  at time  $t_j$ , where  $j = 1, \dots, p^{(t)}$ .
    - iii The backward transformation  $\mathcal{M}^{b,(k)}(\mathbf{x}, t)$  is applied to obtain  $\theta^{j,s,(k+1)}$  and  $\theta^{j,l,(k+1)}$ , where  $j = 1, \dots, p^{(t)}$ .
    - iv Compute  $V_n^{j,(k+1)}$  ( $j = 1, \dots, p^{(t)}$ ),  $\mathcal{M}^{b,(k+1)}(\mathbf{x}, t)$  and  $\mathcal{M}^{f,(k+1)}(\tilde{\mathbf{x}}, t)$ .
  - 6: **end while**
  - 7: Update the level set function,  $\phi^j$  ( $j = 1, \dots, p^{(t)}$ ), using the forward transformation.
  - 8: **return**  $\theta^{n+1,s}$ ,  $\theta^{n+1,l}$ ,  $V_n^{n+1}$  and  $\phi^{n+1}$ .
-

## 5.2 Function spaces and notations

First, we introduce a transformation  $\tilde{\mathbf{x}} = \mathbf{X}(\xi, \eta)$  which defines a mapping that connects a one spatial dimension space-time element  $\tilde{\mathcal{K}}$  in the moving frame of reference and a reference square  $\mathcal{R} = [-1, 1] \times [-1, 1]$ , where  $\tilde{\mathbf{x}} = (\tilde{x}, t) \in \mathbb{R}^2$ . In the space-time domain  $\tilde{\mathcal{E}} \in \mathbb{R}^2$ , each space-time element in the moving frame of reference is a square. So the transformation  $(\tilde{x}, t) = \mathbf{X}(\xi, \eta)$  [43] is linear in each coordinate direction and is written as

$$\mathbf{X}(\xi, \eta) = \frac{1}{4} \{ \mathbf{x}^1(1-\xi)(1-\eta) + \mathbf{x}^2(1+\xi)(1-\eta) + \mathbf{x}^3(1+\xi)(1+\eta) + \mathbf{x}^4(1-\xi)(1+\eta) \}, \quad (42)$$

where  $\{\mathbf{x}^j\}_{j=1,\dots,4}$  are four corners of a space-time element  $\tilde{\mathcal{K}}$ , which are numbered counter clockwise.

Let's define the transformation between the space-time element  $\tilde{\mathcal{K}}_j^n$  and the reference square  $\mathcal{R}$  as  $(\tilde{x}, t)|_{\tilde{\mathcal{K}}_j^n} = \mathbf{X}_{\tilde{\mathcal{K}}_j^n}(\xi, \eta)$ , such that  $\mathbf{X}_{\tilde{\mathcal{K}}_j^n} = (X_\xi^{n,j}, Y_\eta^{n,j})$ . So the gradient operator  $\nabla$  is transformed to

$$\nabla f = \frac{1}{\mathcal{J}^{n,j}} \mathbf{C}^{n,j} \nabla^r f, \quad (43)$$

where  $\mathcal{J}^{n,j} = X_\xi^{n,j} Y_\eta^{n,j} - X_\eta^{n,j} Y_\xi^{n,j}$ , the matrix  $\mathbf{C}^{n,j}$  is defined as

$$\mathbf{C}^{n,j} = \begin{pmatrix} Y_\eta^{n,j} & -Y_\xi^{n,j} \\ -X_\eta^{n,j} & X_\xi^{n,j} \end{pmatrix}, \quad (44)$$

and  $\nabla^r$  denotes the gradient operator defined on the reference square  $\mathcal{R}$ . In addition, the divergence operator  $\nabla \cdot$  is transformed to

$$\nabla \cdot \mathbf{F} = \frac{1}{\mathcal{J}^{n,j}} \nabla^r \cdot ((\mathbf{C}^{n,j})^\top \mathbf{F}). \quad (45)$$

Next, Let's introduce two function spaces,  $V_h$  and  $\Sigma_h$ , associated with the tessellation  $\tilde{\mathcal{T}}_h$  defined in the moving frame of reference,

$$V_h = \{ \nu \in L^2(\tilde{\mathcal{E}}) : \nu|_{\tilde{\mathcal{K}}} \circ \mathbf{X}_{\tilde{\mathcal{K}}} \in \mathcal{P}^p(\mathcal{R}), \forall \tilde{\mathcal{K}} \in \tilde{\mathcal{T}}_h \}, \quad (46)$$

$$\Sigma_h = \{ \boldsymbol{\tau} \in (L^2(\tilde{\mathcal{E}}))^{d+1} : \boldsymbol{\tau}|_{\tilde{\mathcal{K}}} \circ \mathbf{X}_{\tilde{\mathcal{K}}} \in (\mathcal{P}^p(\mathcal{R}))^{d+1}, \forall \tilde{\mathcal{K}} \in \tilde{\mathcal{T}}_h \}, \quad (47)$$

where  $\mathcal{P}^p(\mathcal{R})$  is the set of all polynomials of degree at most  $p = (p^{(x)}, p^{(t)})$  on  $\mathcal{R}$ , with  $p^{(x)}$  in the spatial direction and  $p^{(t)}$  in the time direction.

Spectral element methods with non-periodic boundaries use orthogonal polynomial approximations of the solution [41, 43]. In the space-time domain  $\tilde{\mathcal{E}} \in \mathbb{R}^2$ , we introduce a tensor product basis,  $\ell_i(\xi) \ell_j(\eta) \in \mathcal{P}^p(\mathcal{R})$ , such that

$$\ell_i(\xi) = \prod_{\substack{i=0 \\ i \neq k}}^{p^{(x)}} \frac{\xi - \xi_i}{\xi_k - \xi_i}, \quad \ell_j(\eta) = \prod_{\substack{j=0 \\ j \neq k}}^{p^{(t)}} \frac{\eta - \eta_j}{\eta_k - \eta_j}, \quad (48)$$

where  $\{\xi_i\}_{i=0,\dots,p(x)}$  and  $\{\eta_j\}_{j=0,\dots,p(t)}$  are the sets of Legendre Gauss-Lobatto points. Then the exact solution  $\tilde{\sigma}$  and  $\tilde{\theta}$  are approximated by  $\tilde{\sigma}_h \in \Sigma_h$  and  $\tilde{\theta}_h \in V_h$ , which are written in nodal Lagrange form,

$$\tilde{\sigma} \sim \tilde{\sigma}_h = \sum_{j=0}^{p(t)} \sum_{i=0}^{p(x)} \tilde{\sigma}_{i,j} \ell_i(\xi) \ell_j(\eta), \quad (49)$$

$$\tilde{\theta} \sim \tilde{\theta}_h = \sum_{j=0}^{p(t)} \sum_{i=0}^{p(x)} \tilde{\theta}_{i,j} \ell_i(\xi) \ell_j(\eta). \quad (50)$$

In addition, we define the trace of  $\nu \in V_h$  on  $\tilde{\mathcal{K}}$  as

$$\nu^- = \lim_{\epsilon \rightarrow 0} \nu(\tilde{\mathbf{x}} - \epsilon \bar{\mathbf{n}}_{\tilde{\mathcal{K}}}) \quad (51)$$

where  $\tilde{\mathbf{x}} = (\tilde{x}, t) \in \partial \tilde{\mathcal{K}}$  and  $\bar{\mathbf{n}}_{\tilde{\mathcal{K}}}$  is outward unit normal of  $\tilde{\mathcal{K}}$ . The trace of  $\tau \in \Sigma_h$  is defined similarly.

Due to the discontinuous approximation spaces, the temperature and the temperature gradient approximations are double valued across element boundaries. Thus, we introduce the average  $\{\{\cdot\}\}$  and jump  $\llbracket \cdot \rrbracket$  operators. Considering two adjacent elements  $\tilde{\mathcal{K}}_j^n$  and  $\tilde{\mathcal{K}}_{j+1}^n$  such that  $\tilde{\mathcal{Q}}_{j,j+1}^n = \tilde{\mathcal{K}}_j^n \cap \tilde{\mathcal{K}}_{j+1}^n$ , let  $\bar{\mathbf{n}}_{\tilde{\mathcal{K}}_j^n}$  and  $\bar{\mathbf{n}}_{\tilde{\mathcal{K}}_{j+1}^n}$  denote the corresponding outward unit normal of  $\tilde{\mathcal{K}}_j^n$  and  $\tilde{\mathcal{K}}_{j+1}^n$  on  $\tilde{\mathcal{Q}}_{j,j+1}^n$ . Then the average  $\{\{\cdot\}\}$  and jump  $\llbracket \cdot \rrbracket$  operators are defined as,

$$\{\{\nu\}\} = (\nu_{\tilde{\mathcal{K}}_j^n}^- + \nu_{\tilde{\mathcal{K}}_{j+1}^n}^-)/2, \quad \{\{\tau\}\} = (\tau_{\tilde{\mathcal{K}}_j^n}^- + \tau_{\tilde{\mathcal{K}}_{j+1}^n}^-)/2, \quad (52)$$

$$\llbracket \nu \rrbracket = \nu_{\tilde{\mathcal{K}}_j^n}^- \bar{\mathbf{n}}_{\tilde{\mathcal{K}}_j^n} + \nu_{\tilde{\mathcal{K}}_{j+1}^n}^- \bar{\mathbf{n}}_{\tilde{\mathcal{K}}_{j+1}^n}, \quad \llbracket \tau \rrbracket = \tau_{\tilde{\mathcal{K}}_j^n}^- \cdot \bar{\mathbf{n}}_{\tilde{\mathcal{K}}_j^n} + \tau_{\tilde{\mathcal{K}}_{j+1}^n}^- \cdot \bar{\mathbf{n}}_{\tilde{\mathcal{K}}_{j+1}^n}. \quad (53)$$

### 5.3 Discretization of DG spectral element method

To derive the space-time DG weak formulation for Eq. (38), we follow the same approach presented in [38]. By introducing an auxiliary variable  $\tilde{\sigma}$ , we rewrite Eq. (38) into a first order system

$$\tilde{\sigma} = A \nabla \tilde{\theta}, \quad (54)$$

$$\nabla \cdot (\rho c_v \mathbf{B} \tilde{\theta} - (\tilde{\nabla} \mathbf{x})^{-1} \tilde{\sigma}) = 0. \quad (55)$$

In the moving frame of reference, a tessellation of a space-time slab  $\tilde{\mathcal{E}}^n$  is defined as  $\tilde{\mathcal{T}}_h^n$  such that  $\tilde{\mathcal{T}}_h^n = \tilde{\mathcal{T}}_h^{n,s} \cup \tilde{\mathcal{T}}_h^{n,l}$ . As the discretization procedure is identical in both phases, we consider only the solid phase with  $\tilde{\mathcal{T}}_h^{n,s}$ . Recall that a space-time element  $\tilde{\mathcal{K}}^s$  in  $\tilde{\mathcal{T}}_h^{n,s}$  is either  $\tilde{\mathcal{K}}_j^{n,s}$  or  $\tilde{\mathcal{K}}_j^{n,s,\Gamma}$ , where  $\tilde{\mathcal{K}}_j^{n,s,\Gamma}$  denotes a subelement of the cut-cell  $\tilde{\mathcal{K}}_j^{n,sl}$ .

Multiply (54) with a test function  $\boldsymbol{\tau} \in \boldsymbol{\Sigma}_h$ , substitute  $\tilde{\boldsymbol{\sigma}}$  and  $\tilde{\boldsymbol{\theta}}$  with the approximations  $\tilde{\boldsymbol{\sigma}}_h \in \boldsymbol{\Sigma}_h$  and  $\tilde{\boldsymbol{\theta}}_h \in V_h$ , and integrate over a space-time element  $\tilde{\mathcal{K}}^s \in \mathcal{T}_h^{n,s}$ , then we obtain

$$\int_{\tilde{\mathcal{K}}^s} \tilde{\boldsymbol{\sigma}}_h \cdot \boldsymbol{\tau} d\tilde{\mathcal{K}}^s = \int_{\tilde{\mathcal{K}}^s} \mathbf{A} \nabla \tilde{\boldsymbol{\theta}}_h \cdot \boldsymbol{\tau} d\tilde{\mathcal{K}}^s. \quad (56)$$

According to the transformation  $(\tilde{x}, t) = \mathbf{X}(\xi, \eta)$  and Eq. (43), we transform the integrals over  $\tilde{\mathcal{K}}^s$  to the integrals over the reference square  $\mathcal{R}$  and obtain

$$\int_{\mathcal{R}} \tilde{\boldsymbol{\sigma}}_h \cdot \boldsymbol{\tau} \mathcal{J}^{n,j} d\mathcal{R} = \int_{\mathcal{R}} \frac{1}{\mathcal{J}^{n,j}} (\mathbf{A} \mathbf{C}^{n,j} \nabla^r \tilde{\boldsymbol{\theta}}_h \cdot \boldsymbol{\tau}) \mathcal{J}^{n,j} d\mathcal{R}. \quad (57)$$

After integration by parts twice, we have the following formulation,

$$\int_{\mathcal{R}} \tilde{\boldsymbol{\sigma}}_h \cdot \boldsymbol{\tau} \mathcal{J}^{n,j} d\mathcal{R} = \int_{\mathcal{R}} \mathbf{A} \mathbf{C}^{n,j} \nabla^r \tilde{\boldsymbol{\theta}}_h \cdot \boldsymbol{\tau} d\mathcal{R} + \int_{\partial \mathcal{R}} \mathbf{A} \mathbf{C}^{n,j} (\hat{\boldsymbol{\theta}}_h^{\text{D}} - \tilde{\boldsymbol{\theta}}_h^-) \mathbf{n}^r \cdot \boldsymbol{\tau}^- d\partial \mathcal{R}, \quad (58)$$

where  $\mathbf{n}^r$  denotes the outward unit normal vector of the reference square  $\mathcal{R}$  and  $\hat{\boldsymbol{\theta}}_h^{\text{D}}$  denotes the numerical flux, which should be carefully defined to ensure stability of the method.

Similarly, the weak formulation of (55) for a space-time element  $\tilde{\mathcal{K}}^s \in \tilde{\mathcal{T}}_h^{n,s}$  is obtained by first multiplying (55) with a test function  $\nu \in V_h$ , and substituting  $\tilde{\boldsymbol{\sigma}}$  and  $\tilde{\boldsymbol{\theta}}$  with the approximations  $\tilde{\boldsymbol{\sigma}}_h \in \boldsymbol{\Sigma}_h$  and  $\tilde{\boldsymbol{\theta}}_h \in V_h$ . Then the transformation  $(\tilde{x}, t) = \mathbf{X}(\xi, \eta)$  is applied on a space-time element  $\tilde{\mathcal{K}}^s$  using Eq. (45) in order to obtain

$$\int_{\mathcal{R}} \frac{1}{\mathcal{J}^{n,j}} \{ \nabla^r \cdot \left( (\mathbf{C}^{n,j})^\top (\rho c_v \mathbf{B} \tilde{\boldsymbol{\theta}}_h - (\tilde{\nabla} \mathbf{x})^{-1} \tilde{\boldsymbol{\sigma}}_h) \right) \nu \} \mathcal{J}^{n,j} d\mathcal{R} = 0. \quad (59)$$

After performing integration by parts once, we have

$$\begin{aligned} & - \int_{\mathcal{R}} \left( \rho c_v (\mathbf{C}^{n,j})^\top \mathbf{B} \tilde{\boldsymbol{\theta}}_h - (\mathbf{C}^{n,j})^\top (\tilde{\nabla} \mathbf{x})^{-1} \tilde{\boldsymbol{\sigma}}_h \right) \cdot \nabla^r \nu d\mathcal{R} + \\ & \int_{\partial \mathcal{R}} \rho c_v \left( (\mathbf{C}^{n,j})^\top \mathbf{B} \hat{\boldsymbol{\theta}}_h^{\text{A}} - (\mathbf{C}^{n,j})^\top (\tilde{\nabla} \mathbf{x})^{-1} \hat{\boldsymbol{\sigma}}_h^{\text{D}} \right) \cdot \mathbf{n}^r \nu^- d\partial \mathcal{R} = 0, \end{aligned} \quad (60)$$

where  $\hat{\boldsymbol{\theta}}_h^{\text{A}}$  and  $\hat{\boldsymbol{\sigma}}_h^{\text{D}}$  are the numerical fluxes. Note that the only coupling between elements in Eq. (58) and Eq. (60) occur when we compute the element boundary flux values.

Next, we find appropriate choices for the numerical fluxes. We separate the numerical fluxes into an advection flux  $\hat{\boldsymbol{\theta}}_h^{\text{A}}$  and the diffusive fluxes  $(\hat{\boldsymbol{\theta}}_h^{\text{D}}, \hat{\boldsymbol{\sigma}}_h^{\text{D}})$ . For the advection flux, we apply the upwind flux, as described in [45].

$$(\mathbf{C}^{n,j})^\top \mathbf{B} \hat{\boldsymbol{\theta}}_h^{\text{A}} = \{ \{ (\mathbf{C}^{n,j})^\top \mathbf{B} \tilde{\boldsymbol{\theta}}_h \} \} + \frac{1}{2} |(\mathbf{C}^{n,j})^\top \mathbf{B} \cdot \mathbf{n}^r| [\tilde{\boldsymbol{\theta}}_h]. \quad (61)$$

Now, we consider the diffusive fluxes  $(\hat{\boldsymbol{\theta}}_h^{\text{D}}, \hat{\boldsymbol{\sigma}}_h^{\text{D}})$ . In [52], a mathematical framework is proposed for analyzing different DG approaches for elliptic problems.

Kirby and Karniadakis [53] presented a study of three different formulations selecting from [52] for diffusion problems, which are listed in Table 1. In Table 1, we drop the symbol “ $\sim$ ” and the index “D” in order to be more concise.

| Method            | Fluxes  | stencil in 1D | $h$ -convergence |
|-------------------|---|---------------|------------------|
| Bassi-Rebay [54]  | $\hat{\theta}_h = \{\{\theta_h\}\};$<br>$\hat{\sigma}_h = \{\{\sigma_h\}\}$   | five-element  | suboptimal       |
| LDG [38]          | $\hat{\theta}_h = \{\{\theta_h\}\} - \beta \cdot \llbracket \theta_h \rrbracket;$<br>$\hat{\sigma}_h = \{\{\sigma_h\}\} + \beta \cdot \llbracket \sigma_h \rrbracket - \alpha_j \llbracket \theta_h \rrbracket$ | three-element | optimal          |
| Baumann-Oden [55] | $\hat{\theta}_h = \{\{\theta_h\}\};$<br>$\hat{\sigma}_h = \{\{\nabla \theta_h\}\}$  | three-element | suboptimal       |

**Table 1** DG methods for diffusion problems and the corresponding numerical fluxes.  $\beta$  is called the auxiliary parameter and defined as  $\beta \cdot \mathbf{n} = \text{sign}(\mathbf{v} \cdot \mathbf{n})/2$  [39], where  $\mathbf{v} = (1, 1)^\top$ .  $\alpha_j \geq 0$  is referred to as the stabilization parameter

Considering the compactness of the stencil and the  $h$ -convergence rate, we choose the LDG [38] fluxes listed in Table 1 with the stabilization parameter  $\alpha_j = 0$ . To complete the derivations of the weak formulation for (54) and (55), we perform summation over all elements.

$$\begin{aligned}
& \sum_n \sum_j \int_{\mathcal{R}} \tilde{\sigma}_h \cdot \tau \mathcal{J}^{n,j} d\mathcal{R} = \sum_n \sum_j \int_{\mathcal{R}} \mathbf{A} \mathbf{C}^{n,j} \nabla^r \tilde{\theta}_h \cdot \tau d\mathcal{R} + \\
& \quad \sum_n \sum_j \int_{\partial \mathcal{R}} \mathbf{A} \mathbf{C}^{n,j} (\hat{\theta}_h^{\text{D}} - \tilde{\theta}_h^-) \mathbf{n}^r \cdot \tau^- d\partial \mathcal{R}, \quad (62) \\
& - \sum_n \sum_j \int_{\mathcal{R}} \left( \rho c_v (\mathbf{C}^{n,j})^\top \mathbf{B} \tilde{\theta}_h - (\mathbf{C}^{n,j})^\top (\tilde{\nabla} \mathbf{x})^{-1} \tilde{\sigma}_h \right) \cdot \nabla^r \nu d\mathcal{R} + \\
& \quad \sum_n \sum_j \int_{\partial \mathcal{R}} \left( \rho c_v (\mathbf{C}^{n,j})^\top \mathbf{B} \hat{\theta}_h^{\text{A}} - (\mathbf{C}^{n,j})^\top (\tilde{\nabla} \mathbf{x})^{-1} \hat{\sigma}_h^{\text{D}} \right) \cdot \mathbf{n}^r \nu^- d\partial \mathcal{R} = 0. \quad (63)
\end{aligned}$$

Note that special care is needed when imposing the boundary conditions in Eq. (62) through  $\hat{\theta}_h^{\text{D}}$ . The reason is that the boundary conditions must be the exact values imposed in the moving frame of reference, which are obtained by applying the backward transformation on the boundary conditions imposed in the fixed frame of reference. Since the backward transformation (10) depends on the interface velocity, we have

$$(\hat{\theta}_h^{\text{D}})|_{(\partial \tilde{\mathcal{E}}^{n,s})} = (\hat{\theta}_h^{\text{D}}(\tilde{\nabla} \tilde{\theta}_h))|_{(\partial \tilde{\mathcal{E}}^{n,s})}. \quad (64)$$

We proceed by approximating the integrals in Eq. (62) and Eq.(63) using Gaussian quadrature. In order to avoid the computation of an inverse matrix

in Eq. (62), we choose the quadrature nodes to be the same as the interpolation nodes, i.e. Legendre Gauss-Lobatto nodes. By choosing the same nodes, the mass matrix in each element becomes an identity matrix. Thus, we have

$$\int_{\mathcal{R}} \tilde{\boldsymbol{\sigma}}_h \cdot \boldsymbol{\tau} \mathcal{J}^{n,j} d\mathcal{R} \approx \mathcal{J}^{n,j} (\tilde{\boldsymbol{\sigma}}_{i,j} \cdot \boldsymbol{\tau}_{i,j}) \omega_i \omega_j, \quad (65)$$

$$i = 0, \dots, p^{(x)},$$

$$j = 0, \dots, p^{(t)},$$

where  $\{\omega_i\}_{i=0,\dots,p^{(x)}}$  and  $\{\omega_j\}_{j=0,\dots,p^{(t)}}$  are the integration weights in the  $\xi$  direction and in the  $\eta$  direction, respectively. In a space-time slab  $\tilde{\mathcal{E}}^n \in \mathbb{R}^2$ ,  $\mathcal{J}^{n,j}$  in each element is a positive constant, such that  $\mathcal{J}^{n,j} = \frac{\Delta t^n}{2} \frac{h}{2}$ .

From Table 1, we notice that the numerical flux  $\hat{\theta}_h^D$  only depends on the variable  $\tilde{\theta}_h$ . So we can eliminate the auxiliary variable  $\tilde{\boldsymbol{\sigma}}_h$  from the formulation (63). Solve for  $\tilde{\boldsymbol{\sigma}}_h$  in (62) and then substitute  $\tilde{\boldsymbol{\sigma}}_h$  and  $\nu$  with the resulting expression and  $\ell_i(\xi)\ell_j(\eta)$  into (63). We then obtain a linear system at  $k$ -th Picard iteration as ,

$$\mathbf{G}(\tilde{\theta}_h^{(k)}) \tilde{\theta}_h^{(k+1)} = \mathbf{f}(\tilde{\theta}_h^{(k)}) \quad (66)$$

where  $\mathbf{G}(\tilde{\theta}_h^{(k)})$  denotes the matrix of the linear system due to (41), and the vector  $\mathbf{f}(\tilde{\theta}_h^{(k)})$  obtained by (64) is the Dirichlet boundary conditions imposed on the boundary of  $\tilde{\mathcal{T}}_h^{n,s} \cap (\partial \tilde{\mathcal{E}}^n \setminus \Omega_{t_{n+1}} \cup \hat{\mathcal{Q}}_I^n)$ . Note that there is no boundary condition imposed on  $\Omega_{t_{n+1}}$ .

Lastly, we examine the structure of the matrix  $\mathbf{G}(\tilde{\theta}_h^{(k)})$  in each phase. Here, we take the matrix in the liquid phase as an example, which is denoted as  $\mathbf{G}^l(\tilde{\theta}_h^{(k)})$ . The matrix  $\mathbf{G}^l(\tilde{\theta}_h^{(k)})$  depends on the polynomial order in space,  $p^{(x)}$ , the polynomial order in time,  $p^{(t)}$ , and the number of element in the liquid phase,  $E^{(x),l}$ . For example, we take  $p^{(x)} = 2$ ,  $p^{(t)} = 1$  and  $E^{(x),l} = 2$ . The order of the matrix  $\mathbf{G}^l(\tilde{\theta}_h^{(k)})$  is  $(12 \times 12)$ , i.e.  $(p^{(x)} + 1) \times (p^{(t)} + 1) \times E^{(x),l} = 12$ , and the structure of the matrix,  $\mathbf{G}^l(\tilde{\theta}_h^{(k)})$ , is shown as follows

$$\mathbf{G}^l(\tilde{\theta}_h^{(k)})_{(12 \times 12)} = \begin{pmatrix} * & * & * & * & 0 & 0 & * & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & * & 0 & * & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & * & * & 0 & 0 & 0 & 0 & 0 \\ * & 0 & 0 & * & * & * & 0 & 0 & 0 & * & 0 & 0 \\ 0 & * & 0 & * & * & * & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & * & * & * & * & 0 & 0 & 0 & * & 0 & 0 \\ * & * & * & 0 & 0 & 0 & * & * & * & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & * & * & * & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & * & * & * & 0 & 0 & * \\ 0 & 0 & 0 & * & * & * & * & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & * & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & * & * & * & * \end{pmatrix}$$

Note that  $\mathbf{G}^l(\tilde{\theta}_h^{(k)})$  is a sparse non symmetric matrix.

#### 5.4 Calculating the interface velocity

In the Stefan problem, the phase boundary is evolving in time with the velocity defined in Eq. (6). The accuracy of the interface velocity directly affects the accuracy of the interface position. The interface velocity is a function of the temperature gradient evaluated at both sides of the interface. The order of accuracy for approximating the interface velocity must be commensurate with the order for approximating the solutions of the heat equations, otherwise the overall order of accuracy will reduce to the minimum of the velocity or temperature order. For example, in [28,30], the authors applied finite difference methods to solve the Stefan problem, and the loss of one order of overall accuracy was reported due to the discretization error in calculating the interface velocity.

In order to obtain overall spectral accuracy, the interface velocity must be computed by a spectrally accurate method. We propose two options for computing the interface velocity. We find that each option leads to a spectrally accurate method for computing the interface velocity.

Since the velocity is determined by the temperature gradient evaluated at the interface in both phases, one option is to approximate the interface normal velocity using the following expression

$$V_n = \frac{1}{\rho L} (\tilde{\sigma}^s - \tilde{\sigma}^l) \cdot \tilde{\mathbf{n}}, \quad (67)$$

where  $\tilde{\mathbf{n}} = (\tilde{\mathbf{n}}, 0)$  is the unit normal at the interface pointing in the direction of interface movement in a space-time slab  $\tilde{\mathcal{E}}^n$ , and  $\tilde{\mathbf{n}} = (\tilde{n}_1, \dots, \tilde{n}_d)$  is the spatial part of  $\tilde{\mathbf{n}}$  in the moving frame of reference. Note that  $\tilde{\sigma}$  is the auxiliary variable introduced in (54). Now, we show that Eq. (67) is equivalent to Eq. (6) defined in the fixed frame of reference by,

$$\begin{aligned} \tilde{\sigma} \cdot \tilde{\mathbf{n}} &= \mathbf{A} \nabla \tilde{\theta} \cdot \tilde{\mathbf{n}}, \\ &= k ((\tilde{\nabla} \mathbf{x})^{-1})^\top \tilde{\nabla} \tilde{\theta} \cdot \tilde{\mathbf{n}}, \\ &= k \nabla \theta \cdot \mathbf{n}, \end{aligned} \quad (68)$$

where  $\mathbf{n}$  is the unit normal at the interface pointing in the direction of interface movement in the fixed frame of reference. We refer to Eq. (67) as the weak form method.

Another option is to apply the normal probe method in the fixed frame of reference. In a space-time slab  $\mathcal{E}^n \in \mathbb{R}^2$ , the interface curve  $\mathcal{Q}_I^n$  is updated by the forward transformation  $\mathbf{x} = \mathcal{M}^f(\tilde{\mathbf{x}}, t)$ , while the temperature values  $\theta(x, t)$  in the fixed frame of reference are obtained by the backward transformation  $\tilde{\mathbf{x}} = \mathcal{M}^b(\mathbf{x}, t)$ .

Note that the transformations (14) and (15) are discretized by a  $(p^{(t)} + 1)$ -point Gaussian quadrature rule, where  $p^{(t)}$  is the polynomial order in the time direction. In order to obtain the temperature values  $\theta(x, t)$  in the fixed frame of reference, we first find the departure points (i.e.  $\tilde{x}$ ) of the Legendre Gauss-Lobatto nodes (i.e.  $x$ ) in the fixed frame of reference by the backward

transformation (14), then the temperature values  $\theta(x, t)$  are obtained via a  $p^{(x)}$ -th order accurate interpolation, such that

$$\theta(x, t) = \mathcal{I}_m(\tilde{\theta}(\tilde{x}, t)), \quad (69)$$

where  $\mathcal{I}_m$  is the interpolation operator due to the mapping, and  $p^{(x)}$  is the polynomial order in the spatial direction. According to the zero level set at time  $t^j$  ( $j = 1, \dots, p^{(t)}$ ), we construct a fictitious element in each phase with the size  $h$  that is the length of a regular space-time element in the spatial direction, i.e. Fig. 5 is an example of the normal probe method applied at time  $t^{p^{(t)}}$ . In each fictitious element, the temperature values,  $\theta^{fic}(x, t)$ , at its Gauss-Lobatto nodes are obtained by

$$\theta^{fic}(x, t) = \mathcal{I}_n(\theta(x, t)), \quad (70)$$

where  $\mathcal{I}_n$  is the interpolation operator due to the normal probe method. The normal interface velocity is then computed by,

$$V_n = \frac{1}{\rho L} \left\{ \nabla \left( \mathcal{I}_n(\mathcal{I}_m(\tilde{\theta}^{n+1,s})) \right) - \nabla \left( \mathcal{I}_n(\mathcal{I}_m(\tilde{\theta}^{n+1,l})) \right) \right\} \cdot \mathbf{n}, \quad (71)$$

where  $\nabla$  is the gradient operator with respect to the spatial variable  $\mathbf{x} = (x_1, \dots, x_d)$  in the fixed frame of reference, and  $\mathbf{n} = (n_1, \dots, n_d)$  is the unit normal at the interface pointing in the direction of the interface movement. The approximation of  $\nabla \left( \mathcal{I}_n(\mathcal{I}_m(\tilde{\theta}^{n+1,s})) \right)$  is written as

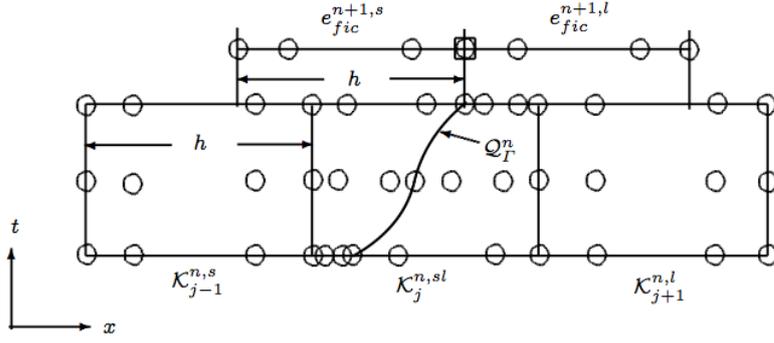
$$\nabla \left( \mathcal{I}_n(\mathcal{I}_m(\tilde{\theta}^{n+1,s})) \right) \approx \sum_i \left( \mathcal{I}_n(\mathcal{I}_m(\tilde{\theta}_h^{n+1,s})) \right)_i D_{\Gamma^{n+1},i}, \quad (72)$$

where  $D_{\Gamma^{n+1},i}$  is the derivative matrix evaluated at the interface  $\Gamma(t_{n+1})$ , i.e.  $D_{\Gamma^{n+1},i} = \ell'_i(\Gamma(t_{n+1}))$  in one spatial dimension. The aim of using fictitious elements is to avoid computing the temperature gradient in a small cell, which can eliminate the instability caused by computing the temperature gradient directly (i.e. spectral collocation discretization) in a small cell. The description of the normal probe method is illustrated in Fig. 5.

## 6 Numerical Experiments

In this section, we test the space-time DG spectral element method for solving the Stefan problem on two benchmark problems in one spatial dimension. The numerical results are compared with the exact solutions in order to examine the convergence rate both with respect to space and time. The time step is restricted by the following CFL criterion

$$\Delta t = C \frac{h}{V_n}, \quad (73)$$



**Fig. 5** Description of the normal probe method applied in a space-time slab  $\mathcal{E}^n \in \mathbb{R}^2$  in the fixed frame of reference. The polynomial order,  $p = (p^{(x)}, p^{(t)})$ , in each space-time element is chosen to be (3, 2). We assume that the solid phase is on the left hand side of the interface  $\mathcal{Q}_T^n$  while the liquid is on the other side. The fictitious element in the solid phase at time  $t^{n+1}$  is denoted as  $e_{fic}^{n+1,s}$ , and  $e_{fic}^{n+1,l}$  is the fictitious element for the liquid phase at time  $t_{n+1}$ . The length of both fictitious elements is  $h$ , which is the same as the length of a regular space-time element, i.e. the length of  $\mathcal{K}_{j-1}^{n,s}$ . The number of Gauss-Lobatto nodes in each fictitious element is  $p + 1$ , where  $p = p^{(x)}$ . The temperature gradient is evaluated at the node with symbol “□” in each fictitious element.

where  $C$  is a constant and set to be less than 1.0, and  $h$  is the length of a regular space-time element. This CFL condition (73) makes sure that the interface does not move across two regular space-time elements in one space-time slab.

Note that  $\mathbf{G}(\tilde{\theta}^k)$  in Eq. (66) is a non symmetric matrix due to the discretization of the advection term in the transformed heat equation. In our tests, we apply a direct solver from the LAPACK library in order to invert the resulting linear system.

In the last space-time slab  $\mathcal{E}^{E^{(t)}}-1$ , the errors are measured in the  $l_\infty$  norms at time  $t_{E^{(t)}}$ , i.e.  $t_{E^{(t)}} = T$ , where  $T$  is the final computational time.

$$\|Err_\theta\|_\infty = \max_{i=0,\dots,p^{(x)}} \|\theta_i^{E^{(t)}} - (\theta_h)_i^{E^{(t)}}\|, \quad (74)$$

where  $\theta_i^{E^{(t)}}$  denotes the exact temperature solution evaluated at node  $x_i$ , and  $(\theta_h)_i^{E^{(t)}}$  is the approximation temperature evaluated at the same node.

### 6.1 Test 1-constant velocity

Consider a computational domain  $\Omega = [-0.5, 1.0]$  with an exact solution

$$\theta(x, t) = \begin{cases} -1 + e^{-V_n(x-V_nt)}, & x > V_nt, \\ 0, & x \leq V_nt, \end{cases} \quad (75)$$

where the normal velocity  $V_n$  is a constant, and we take  $V_n = 1.0$ . The interface is governed by

$$\Gamma(t) = V_n t. \quad (76)$$

In this test, the ice phase is on the left and motionless, existing at the melting temperature, i.e.  $\theta^s = \theta_m = 0$ , while the liquid phase is on the right and initialized as a supercooled liquid. The interface between the ice and liquid phase is initialized at the origin,  $\Gamma(0)$ , and moves to the right with the normal velocity  $V_n$  defined in Eq. (6). We take  $\rho^l = \rho^s = 1.0$ ,  $k^l = k^s = 1.0$ ,  $c_v^l = c_v^s = 1.0$  and  $L = 1.0$ . Dirichlet boundary conditions for temperature distributions are enforced on the domain's boundary  $\partial\Omega$  using the exact solution. At the interface, we choose  $\theta_\Gamma = \theta_m$ .

First, we demonstrate the spectral accuracy of the approximation by plotting the maximum errors in the temperature and the interface velocity as a function of polynomial order,  $p = (p^{(x)}, p^{(t)})$ , respectively. Fig. 6 applies the weak form method (67) to compute the interface velocity, while Fig. 7 uses the normal probe method (71). Both figures show exponential convergence of approximate temperature to the exact solution, as well as the interface velocity. The simulation is computed over the time  $t = 0$  to  $t = 0.5$  with the time step  $\Delta t = 3.84 \times 10^{-2}$ . The polynomial orders in space and in time are chosen to be the same, i.e.  $p^{(x)} = p^{(t)}$ . The computational domain is divided into  $E^x = 5$ ,  $E^{(x)} = 10$  and  $E^{(x)} = 20$ . Different slopes are observed in both Fig. 6 and Fig. 7.

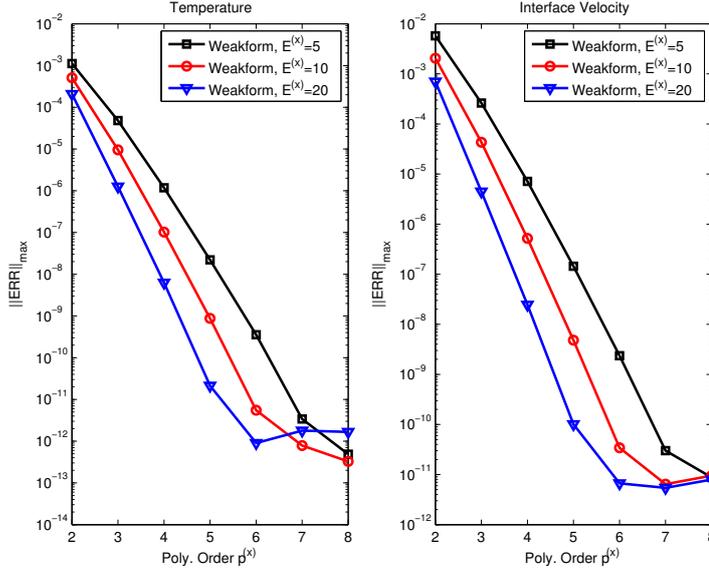
Two options for approximate the interface velocity, i.e. the weak form method (67) and the normal probe method (71) are compared in Fig. 8. Through the comparison of the temperature (left) and the interface velocity (right), we see that both options exhibit spectral accuracy.

Note that the DG spectral element method is applied on the transformed heat equations in both phases in the moving frame of reference (Algorithm 1). If we transform the exact solution (75) into the moving frame of reference by the backward transformation (14), we have

$$\tilde{\theta}(\tilde{x}, t) = \begin{cases} -1 + e^{-V_n(\tilde{x})}, & \tilde{x} > 0, \\ 0, & \tilde{x} \leq 0. \end{cases} \quad (77)$$

So the convergence rate does not depend on the polynomial order  $p^{(t)}$  in this test problem, which is shown in Figs. (9) and (10). We choose three values of the polynomial order in time, i.e.  $p^{(t)} = p^{(x)}$ ,  $p^{(t)} = 2$  and  $p^{(t)} = 10$ . All cases lead to the same error behavior, which is consistent with the exact solution (77) in the moving frame of reference. Fig. (9) applies the weak form method (67) to calculate the interface velocity, while Fig. (10) applies the normal probe method (71).

In Table 2 and Table 3, we list the maximum errors of the temperature, interface position and interface velocity in the last space-time slab with different values of the polynomial order. The computational domain is divided into

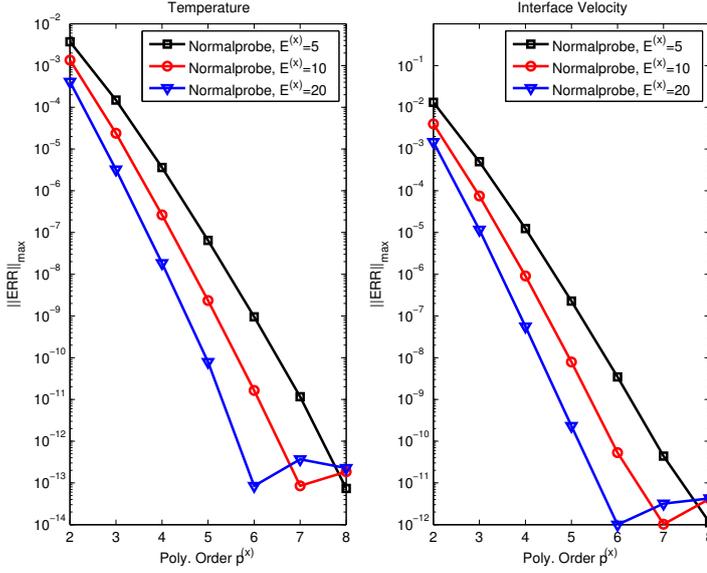


**Fig. 6** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the liquid phase. The polynomial order in time direction,  $p^{(t)}$ , is chosen to be the same as  $p^{(x)}$ . The interface velocity is computed by the weak form method (67). The comparison is made among  $E^{(x)} = 5$ ,  $E^{(x)} = 10$  and  $E^{(x)} = 20$ . For  $E^{(x)} = 20$ , the interface moves across 7 inter-element boundaries, while the interface moves across 3 inter-element boundaries for  $E^{(x)} = 10$  and 2 inter-element boundaries for  $E^{(x)} = 5$ . The simulation is computed over the time  $t = 0$  to  $t = 0.5$  with the time step  $\Delta t = 3.84 \times 10^{-2}$

$E^{(x)} = 5$  elements in the spatial direction, and the time step is chosen to be  $\Delta t = 3.84 \times 10^{-2}$ . We choose the tolerance of the Picard iteration to be  $10^{-13}$  (Algorithm 1). The errors in Table 2 are obtained using the weak form velocity algorithm (67), while Table 3 uses the normal probe velocity algorithm (71).

| $p = (p^{(x)}, p^{(t)})$ | $\ Err_{\theta}\ _{\infty}$ | $\ Err_{\Gamma}\ _{\infty}$ | $\ Err_{V_n}\ _{\infty}$ | iter |
|--------------------------|-----------------------------|-----------------------------|--------------------------|------|
| (2,1)                    | 1.24E-003                   | 1.75E-003                   | 6.12E-003                | 11   |
| (3,1)                    | 5.16E-005                   | 6.40E-005                   | 2.68E-004                | 11   |
| (4,1)                    | 1.28E-006                   | 1.47E-006                   | 7.20E-006                | 9    |
| (5,1)                    | 2.40E-008                   | 2.63E-008                   | 1.44E-007                | 8    |
| (6,1)                    | 3.77E-010                   | 4.02E-010                   | 2.31E-009                | 5    |

**Table 2** Errors of the temperature ( $\|Err_{\theta}\|_{\infty}$ ), interface position ( $\|Err_{\Gamma}\|_{\infty}$ ) and interface velocity ( $\|Err_{V_n}\|_{\infty}$ ) in the last space-time slab. The interface velocity is computed by applying the weak form method (67). The number of the Picard iteration is listed in the last column with  $tol = 10^{-13}$



**Fig. 7** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the liquid phase. The polynomial order in time direction,  $p^{(t)}$ , is chosen to be the same as  $p^{(x)}$ . The interface velocity is computed by the normal probe method (71). The comparison is made among  $E^{(x)} = 5$ ,  $E^{(x)} = 10$  and  $E^{(x)} = 20$ . For  $E^{(x)} = 20$ , the interface moves across 7 inter-element boundaries, while the interface moves across 3 inter-element boundaries for  $E^{(x)} = 10$  and 2 inter-element boundaries for  $E^{(x)} = 5$ . The simulation is computed over the time  $t = 0$  to  $t = 0.5$  with the time step  $\Delta t = 3.84 \times 10^{-2}$

| $p = (p^{(x)}, p^{(t)})$ | $\ Err_{\theta}\ _{\infty}$ | $\ Err_{\Gamma}\ _{\infty}$ | $\ Err_{V_n}\ _{\infty}$ | iter |
|--------------------------|-----------------------------|-----------------------------|--------------------------|------|
| (2,1)                    | 3.89E-003                   | 5.29E-003                   | 1.34E-002                | 12   |
| (3,1)                    | 1.55E-004                   | 1.85E-004                   | 5.08E-004                | 12   |
| (4,1)                    | 3.73E-006                   | 4.16E-006                   | 1.25E-005                | 10   |
| (5,1)                    | 6.80E-008                   | 7.32E-008                   | 2.29E-007                | 6    |
| (6,1)                    | 9.97E-010                   | 1.05E-009                   | 3.47E-009                | 5    |

**Table 3** Errors of the temperature ( $\|Err_{\theta}\|_{\infty}$ ), interface position ( $\|Err_{\Gamma}\|_{\infty}$ ) and interface velocity ( $\|Err_{V_n}\|_{\infty}$ ) in the last space-time slab. The interface velocity is computed by applying the normal probe method (71). The number of the Picard iteration is listed in the last column with  $tol = 10^{-13}$

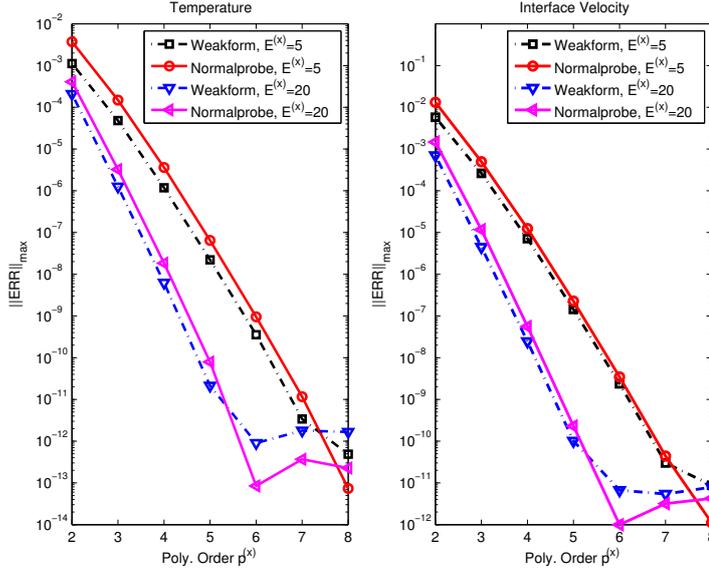
## 6.2 Test 2 - variable velocity

Consider a computational domain  $\Omega = [0, l]$  with an exact solution [4, 56]

$$\theta = \theta_{wall} + \frac{\theta_m - \theta_{wall}}{\text{erf}(\lambda)} \text{erf}\left(\frac{x}{2\sqrt{\lambda t}}\right), \quad (78)$$

$$\Gamma(t) = 2\lambda\sqrt{\alpha t}, \quad (79)$$

where  $\alpha = k^s / (\rho^s c_v^s)$ ,  $\theta_{wall}$  is the Dirichlet boundary condition at  $x = 0$ ,  $\text{erf}(\cdot)$  is the error function and  $\lambda$  is a solution to the transcendental equation.



**Fig. 8** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the liquid phase. The polynomial order in time direction,  $p^{(t)}$ , is chosen to be the same as  $p^{(x)}$ . The comparison is made between the weak form method (67) and the normal probe method (71). The simulation is computed over the time  $t = 0$  to  $t = 0.5$  with the time step  $\Delta t = 3.84 \times 10^{-2}$

$$\lambda \exp(\lambda^2) \operatorname{erf}(\lambda) = \frac{c_v^s (\theta_{wall} - \theta_m)}{L \sqrt{\pi}}. \quad (80)$$

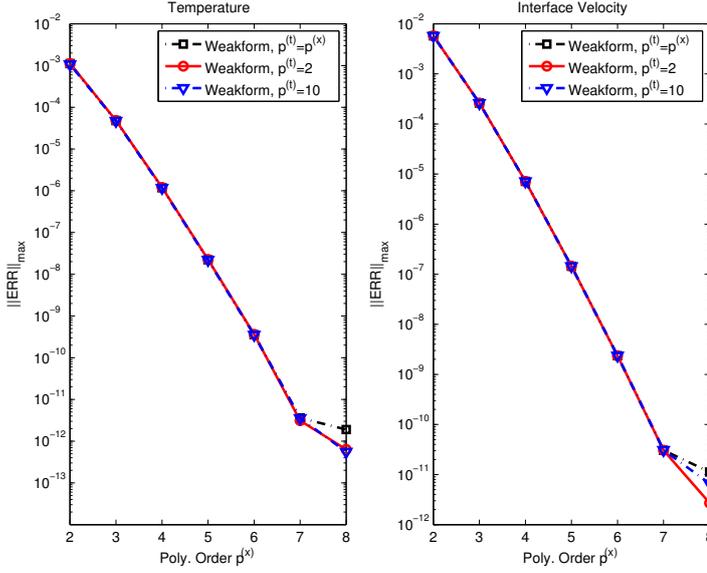
In this test, the ice phase is on the left, while the liquid phase is on the right and motionless, existing at the melting temperature, i.e.  $\theta^l = \theta_m$ . The interface between the ice and liquid phase is initialized at  $\Gamma(0) = 0.1$  and moves to the right with the normal velocity  $V_n$  defined in Eq. (6). Dirichlet boundary conditions for temperature distributions are enforced on the domain's boundary  $\partial\Omega$  using the exact solution.

We set  $\rho^l = \rho^s$  and  $c_v^l = c_v^s$ . Now, we derive the following non-dimensional form,

$$\frac{\partial \theta^s}{\partial \tau} = \frac{\partial^2 \theta^s}{\partial \zeta^2}, \quad 0 < \zeta < \zeta_\Gamma \quad (81)$$

$$\frac{\partial \theta^l}{\partial \tau} = K^{ls} \frac{\partial^2 \theta^l}{\partial \zeta^2}, \quad \zeta_\Gamma < \zeta < 1 \quad (82)$$

$$V_n = St \left( \frac{\partial \theta^s}{\partial \zeta} - K^{sl} \frac{\partial \theta^l}{\partial \zeta} \right), \quad \zeta = \zeta_\Gamma, \quad (83)$$



**Fig. 9** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the liquid phase. The simulation is computed over the time  $t = 0$  to  $t = 0.5$ . The computational domain is divided into  $E^{(x)} = 5$  in spatial direction, with  $E^{(x),l} = 4$  in the liquid in the beginning and  $E^{(x),l} = 2$  in the liquid at the end. The interface velocity is computed by the weak form method (67). The time step is chosen to be  $\Delta t = 3.84 \times 10^{-2}$

With Dirichlet boundary conditions

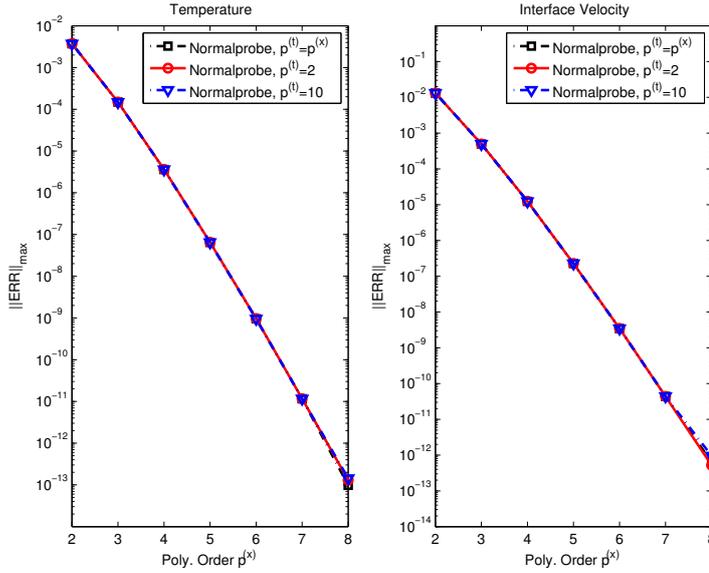
$$\Theta(0, \tau) = 1.0, \quad \zeta = 0, \quad (84)$$

$$\Theta(\zeta_\Gamma, \tau) = 0.0 \quad \zeta = \zeta_\Gamma. \quad (85)$$

Here  $\zeta = x/l$ ,  $\tau = \alpha(t/l^2)$ ,  $\Theta = (\theta - \theta_m)/\Delta\theta$ ,  $\zeta_\Gamma = \Gamma/l$ ,  $K^{ls} = k^l/k^s$  and  $St = (c_v \Delta\theta)/L$ , where  $l$  is the length of the domain,  $\Delta\theta = \theta_{wall} - \theta_m$  and  $\Gamma$  is the phase boundary position. Note that  $St$  is called the Stefan number.

First, we demonstrate the spectral accuracy of the approximation by plotting the maximum errors in the temperature and the interface velocity as a function of polynomial order,  $p = (p^{(x)}, p^{(t)})$ , respectively. Fig. 11 applies the weak form method (67) to compute the interface velocity, while Fig. 12 uses the normal probe method (71). The computational domain is divided into  $E^x = 6$ ,  $E^{(x)} = 12$  and  $E^{(x)} = 24$ . The Stefan number  $St$  is set to be 0.02. The interface starts at  $x = 0.1$  and moves to  $x = 0.3$ . Fig. 11 and Fig. 12 are shown the exponential convergence in space by fixing the polynomial order in time,  $p^{(t)}$ , to be 10 so that the temporal errors are negligible.

Two options for approximate the interface velocity, i.e. the weak form method (67) and the normal probe method (71) are compared in Fig. 13. In Fig. 13, the left part shows the exponential convergence of the temperature, while the exponential convergence of the interface velocity is observed from the right part. We see that both options exhibit spectral accuracy.

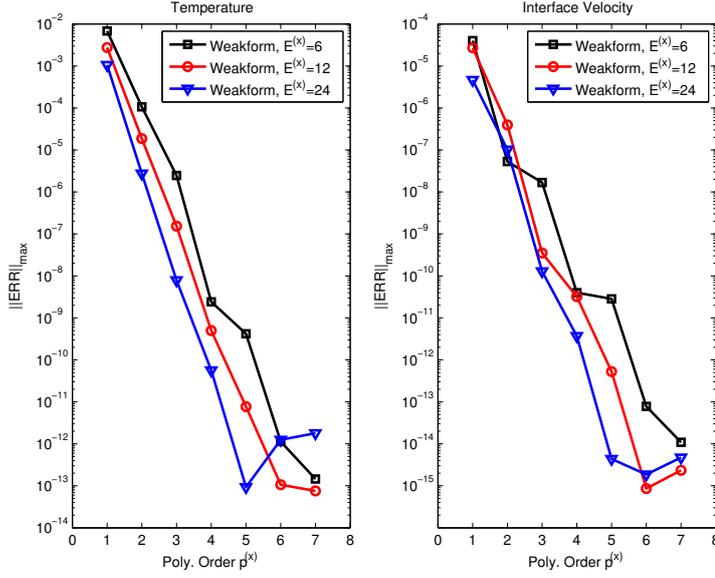


**Fig. 10** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the liquid phase. The simulation is computed over the time  $t = 0$  to  $t = 0.5$ . The computational domain is divided into  $E^{(x)} = 5$  in spatial direction, with  $E^{(x),l} = 4$  in the liquid in the beginning and  $E^{(x),l} = 2$  in the liquid at the end. The interface velocity is computed by the normal probe method (71). The time step is chosen to be  $\Delta t = 3.84 \times 10^{-2}$

Fig. 14 applies the weak form method (67) to compute the interface velocity, while Fig. 15 uses the normal probe method (71). The number of the space-time slab is denoted by  $E^{(t)}$ , which is chosen to  $E^t = 41$ ,  $E^{(t)} = 81$  and  $E^{(t)} = 162$ . The interface starts at  $x = 0.1$  and moves to  $x = 0.3$ . Fig. 14 and Fig. 15 are shown the exponential convergence in time by fixing the polynomial order in space,  $p^{(x)}$ , to be 10 so that the spatial errors are negligible.

In Table 4 and Table 5, we show the maximum errors of the temperature, interface position and interface velocity in the last space-time slab. The computational domain is divided into  $E^{(x)} = 6$  elements in the spatial direction, and the time step is chosen to be  $\Delta t = 4.9 \times 10^{-2}$ . We choose the tolerance for the Picard iteration to be  $10^{-15}$  (Algorithm 1). The errors in Table 4 are obtained using the weak form method (67) for computing the interface velocity, while Table 5 uses the normal probe method (71).

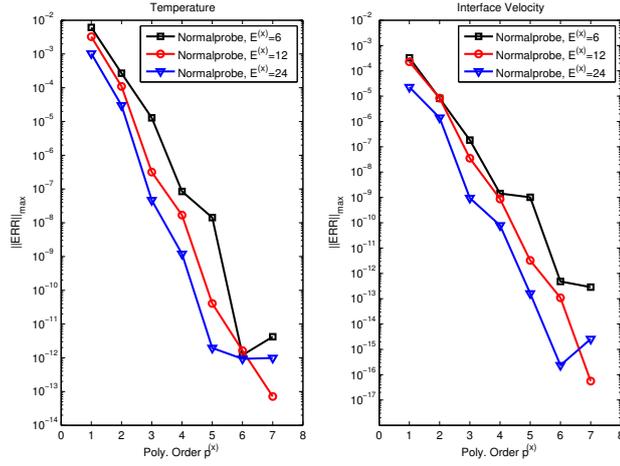
Note that the solution to the Stefan problem is quasi-steady, whereas for the Stefan number  $St$  large, the interface moves relatively rapidly [8]. We vary the Stefan number, i.e.  $St = 0.001$ ,  $St = 0.02$ ,  $St = 0.05$  and  $St = 0.08$ . Fig. 16 applies the weak form method (67) to compute the interface velocity, while Fig. 17 uses the normal probe method (71). The interface starts at  $x = 0.1$  and moves to  $x = 0.4$ . Fig. 16 and Fig. 17 are shown the exponential convergence for different values of  $St$ .



**Fig. 11** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the ice phase. The interface velocity is computed by the weak form method (67). The comparison is made between  $E^{(x)} = 6$ ,  $E^{(x)} = 12$  and  $E^{(x)} = 24$ . The interface starts at  $x = 0.1$  and moves to  $x = 0.3$ . The polynomial order in time direction,  $p^{(t)}$ , is chosen to be 10 so that the temporal errors are negligible. The time step is chosen to be  $\Delta t = 4.9 \times 10^{-2}$

## 7 Conclusions

A space-time DG spectral element method for solving the Stefan problem has been presented in one spatial dimension. Transformation techniques are introduced to deal with the space-time curved, prior unknown, subelements in a cut-cell by mapping the fixed frame of reference to the moving frame of reference. In the moving frame of reference, we have a rectangular mesh in each space-time slab. So in one space-time slab, a DG spectral element method is applied to solve the transformed heat equations, written in the space-time formulation, in both phases. We presented two options for computing the interface velocity, the weak form method (67) and the normal probe method (71). Numerical experiments on standard 1D benchmark problems demonstrate the spectral convergence in both space and time for the temperature and the interface velocity. Both options for calculating the interface velocity exhibit the spectral convergence in both space and time. By comparing the weak form method (67) with the normal probe method (71), we show that the errors of the temperature or the interface velocity computed by applying the weak form method is less than that computed by using the normal probe method. In addition, the number of the Picard iterations for the weak form method is less than that for the normal probe method. We have analyzed our new method for different values of the Stefan number,  $St$ , and we have found that



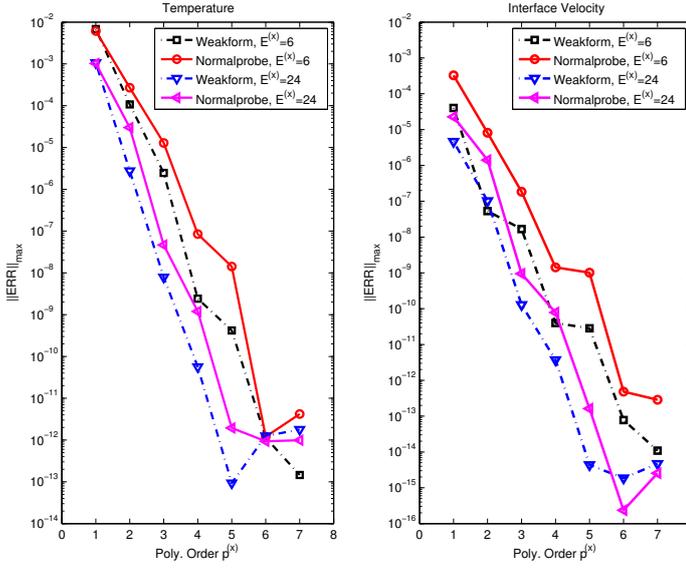
**Fig. 12** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the ice phase. The interface velocity is computed by the normal probe method (71). The comparison is made between  $E^{(x)} = 6$ ,  $E^{(x)} = 12$  and  $E^{(x)} = 24$ . The interface starts at  $x = 0.1$  and moves to  $x = 0.3$ . The polynomial order in time direction,  $p^{(t)}$ , is chosen to be 10 so that the temporal errors are negligible. The time step is chosen to be  $\Delta t = 4.9 \times 10^{-2}$

| $p = (p^{(x)}, p^{(t)})$ | $\ Err_{\theta}\ _{\infty}$ | $\ Err_{\Gamma}\ _{\infty}$ | $\ Err_{V_n}\ _{\infty}$ | iter |
|--------------------------|-----------------------------|-----------------------------|--------------------------|------|
| (3,3)                    | 3.86E-006                   | 8.44E-007                   | 1.87E-007                | 7    |
| (3,10)                   | 2.46E-006                   | 7.67E-008                   | 1.67E-008                | 7    |
| (4,4)                    | 3.82E-009                   | 9.46E-010                   | 2.11E-010                | 7    |
| (4,10)                   | 2.41E-009                   | 1.74E-010                   | 3.98E-011                | 7    |
| (5,5)                    | 1.48E-009                   | 4.72E-010                   | 1.05E-010                | 7    |
| (5,10)                   | 4.18E-010                   | 1.28E-010                   | 2.84E-011                | 7    |
| (6,6)                    | 3.07E-012                   | 9.52E-013                   | 2.12E-013                | 7    |
| (6,10)                   | 1.14E-012                   | 3.47E-013                   | 7.79E-014                | 7    |
| (7,7)                    | 3.19E-013                   | 9.91E-014                   | 2.19E-014                | 7    |
| (7,10)                   | 1.46E-013                   | 4.55E-014                   | 1.09E-014                | 7    |

**Table 4** Errors of the temperature ( $\|Err_{\theta}\|_{\infty}$ ), interface position ( $\|Err_{\Gamma}\|_{\infty}$ ) and interface velocity ( $\|Err_{V_n}\|_{\infty}$ ) in the last space-time slab. The interface velocity is computed by applying the weak form method (67). The computational domain is divided into  $E^{(x)} = 6$  in the spatial direction. The interface starts at  $x = 0.1$  and moves to  $x = 0.3$ . The time step is chosen to be  $\Delta t = 4.9 \times 10^{-2}$ . The number of the Picard iteration is listed in the last column with  $tol = 10^{-15}$

the space-time spectral accuracy property of our method is not sensitive to the values of  $St$ .

As is characteristic of DG spectral element methods, only element boundary flux information is communicated between elements. This property allows one to adjust the polynomial order used in a given element; e.g. the subelements in a cut-cell can have an order that is distinct from its neighbors in order to capture thin thermal layers.

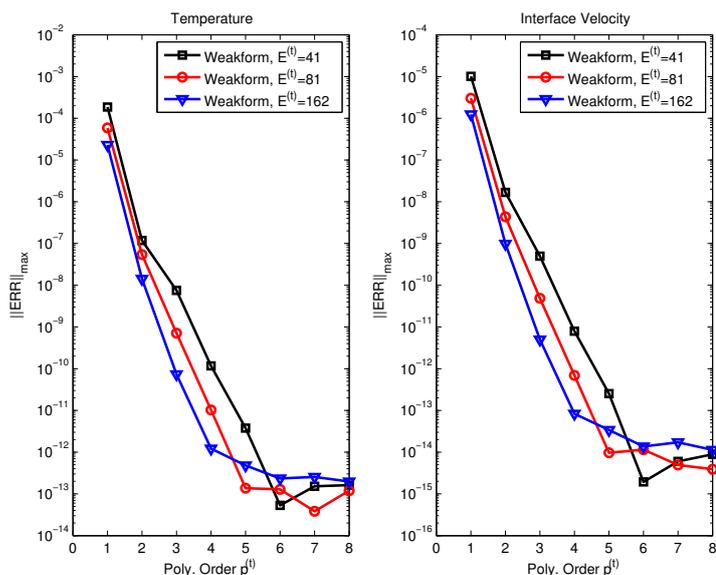


**Fig. 13** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the ice phase. The comparison is made between the weak form method (67) and the normal probe method (71). The interface starts at  $x = 0.1$  and moves to  $x = 0.3$ . The polynomial order in time direction,  $p^{(t)}$ , is chosen to be 10 so that the temporal errors are negligible. The time step is chosen to be  $\Delta t = 4.9 \times 10^{-2}$ . The Stefan number  $St$  is set to be 0.02

Our method is not restricted to one spatial dimension, and its extension to multiple space dimensions is the subject of future activity. In multiple space dimensions, a high-order accurate numerical quadrature needs to be developed for computing the integrals over curved and implicitly defined surfaces and volumes, i.e. the phase interface, in the weak formulation. The numerical quadrature in two or three spatial dimensions can be accomplished by performing the element refinement rule, which is to divide elements into triangles in 2D or tetrahedral element shapes in 3D. However, the methods proposed in [57,58] are more efficient for problems with complex geometries, because there is no need to subdivide elements. In higher spatial dimensions, the level set equation (8) needs to be solved with spectral accuracy, which can be done by the method proposed in [15]. Future work will also explore an alternative direct solver methodology [59] for the sparse linear systems (66) that are obtained in each Picard iteration.

## Acknowledgments

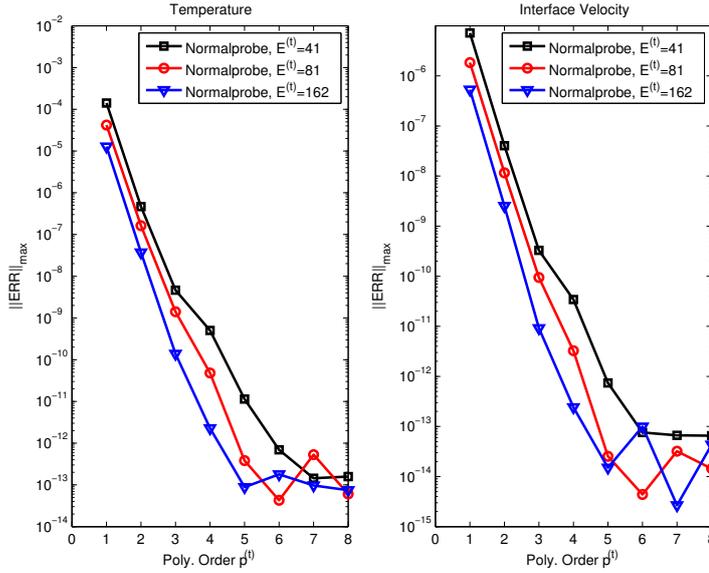
This work and the authors were supported in part by the National Science Foundation under contract DMS 1418983.



**Fig. 14** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(t)}$  in the ice phase. The interface velocity is computed by the weak form method (67). The interface starts at  $x = 0.1$  and moves to  $x = 0.3$ . The comparison is made among different numbers of the space-time slab, i.e.  $E^{(t)} = 41$ ,  $E^{(t)} = 81$  and  $E^{(t)} = 162$ . The corresponding time step is  $\Delta t = 4.9 \times 10^{-2}$ ,  $\Delta t = 2.5 \times 10^{-2}$  and  $\Delta t = 1.2 \times 10^{-2}$ . The polynomial order in spatial direction,  $p^{(x)}$ , is chosen to be 10 so that the spatial errors are negligible. The computational domain is divided into  $E^x = 5$  in the spatial direction. The Stefan number  $St$  is set to be 0.02

## References

1. Šarler, B.: Stefan's work on solid-liquid phase changes. *Engineering Analysis with Boundary Elements* **16**(2), 83–92 (1995)
2. Li, B., Da-Wen, S.: Novel methods for rapid freezing and thawing of foods - a review. *Journal of Food Engineering* **54**(3), 175–182 (2002)
3. Peskin, C.S.: Numerical analysis of blood flow in the heart. *J. Comput. Phys.* **25**(3), 220–252 (1977)
4. Alexiades, V., Solomon, A.D.: *Mathematical Modelling of Melting and Freezing Processes*. Hemisphere Publishing Corporation, Washington (1981)
5. Sethian, J.A., Strain, J.: Crystal growth and dendritic solidification. *J. Comput. Phys.* **98**(2), 231–253 (1992)
6. Farid, M.: The moving boundary problems from melting and freezing to drying and frying of food. *Chemical Engineering and Processing: Process Intensification* **41**(1), 1–10 (2002)
7. Gupta, S.C.: *The Classical Stefan Problem: Basic Concepts, Modelling and Analysis*. North-Holland Series in Applied Mathematics and Mechanics, vol. 45. Elsevier Science B.V., Amsterdam (2003)
8. Rønquist, E.M., Patera, A.T.: A Legendre spectral element method for the Stefan problem. *Int. J. Numer. Meth. Eng.* **24**(12), 2273–2299 (1987)
9. Juric, D., Tryggvason, G.: A front-tracking method for dendritic solidification. *J. Comput. Phys.* **123**(1), 127–148 (1996)
10. Al-Rawahi, N., Tryggvason, G.: Numerical simulation of dendritic solidification with convection: Two-dimensional geometry. *J. Comput. Phys.* **180**(2), 471–496 (2002)



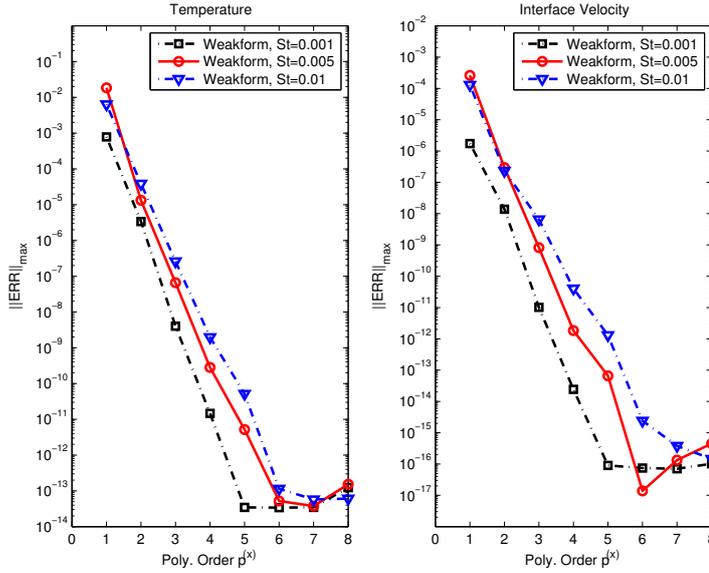
**Fig. 15** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(t)}$  in the ice phase. The interface velocity is computed by the normal probe method (71). The interface starts at  $x = 0.1$  and moves to  $x = 0.3$ . The comparison is made among different numbers of the space-time slab, i.e.  $E^{(t)} = 41$ ,  $E^{(t)} = 81$  and  $E^{(t)} = 162$ . The corresponding time step is  $\Delta t = 4.9 \times 10^{-2}$ ,  $\Delta t = 2.5 \times 10^{-2}$  and  $\Delta t = 1.2 \times 10^{-2}$ . The polynomial order in spatial direction,  $p^{(x)}$ , is chosen to be 10 so that the spatial errors are negligible. The computational domain is divided into  $E^x = 5$  in the spatial direction. The Stefan number  $St$  is set to be 0.02

11. Al-Rawahi, N., Tryggvason, G.: Numerical simulation of dendritic solidification with convection: Three-dimensional flow. *J. Comput. Phys.* **194**(2), 677–696 (2004)
12. Karma, A., Rappel, W.-J.: Quantitative phase-field modeling of dendritic growth in two and three dimensions. *Phys. Rev. E* **57**, 4323–4349 (1998)
13. George, W.L., Warren, J.A.: A parallel 3D dendritic growth simulator using the phase-field method. *J. Comput. Phys.* **177**(2), 264–283 (2002)
14. Osher, S., Fedkiw, R.: *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences, vol. 153. Springer, New York, N.Y. (2003)
15. Sussman, M., Hussaini, M.Y.: A discontinuous spectral element method for the level set equation. *J. Sci. Comput.* **19**(1-3), 479–500 (2003). Special issue in honor of the sixtieth birthday of Stanley Osher
16. Javierre, E., Vuik, C., Vermolen, F.J., van der Zwaag, S.: A comparison of numerical models for one-dimensional Stefan problems. *J. Comput. Appl. Math.* **192**(2), 445–459 (2006)
17. Li, Z.: Immersed interface methods for moving interface problems. *Numer. Algorithms* **14**(4), 269–293 (1997)
18. Li, Z., Lai, M.-C.: The immersed interface method for the Navier-Stokes equations with singular forces. *J. Comput. Phys.* **171**(2), 822–842 (2001)
19. Fedkiw, R.P., Aslam, T., Merriman, B., Osher, S.: A Non-oscillatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method). *J. Comput. Phys.* **152**(2), 457–492 (1999)
20. Chessa, J., Smolinski, P., Belytschko, T.: The extended finite element method (XFEM) for solidification problems. *Int. J. Number. Meth. Eng.* **53**(8), 1959–1977 (2002)

| $p = (p^{(x)}, p^{(t)})$ | $\ Err_\theta\ _\infty$ | $\ Err_\Gamma\ _\infty$ | $\ Err_{V_n}\ _\infty$ | iter |
|--------------------------|-------------------------|-------------------------|------------------------|------|
| (3,3)                    | 1.36E-005               | 4.65E-006               | 2.34E-007              | 7    |
| (3,10)                   | 1.29E-005               | 4.42E-006               | 1.84E-007              | 8    |
| (4,4)                    | 8.27E-008               | 2.67E-008               | 1.26E-009              | 8    |
| (4,10)                   | 8.50E-008               | 2.75E-008               | 1.43E-009              | 8    |
| (5,5)                    | 1.44E-008               | 4.58E-009               | 1.02E-009              | 7    |
| (5,10)                   | 1.42E-008               | 4.52E-009               | 1.01E-009              | 7    |
| (6,6)                    | 1.43E-012               | 1.62E-014               | 4.03E-013              | 10   |
| (6,10)                   | 1.20E-012               | 3.86E-013               | 4.81E-013              | 8    |
| (7,7)                    | 4.08E-012               | 1.27E-012               | 2.80E-013              | 10   |
| (7,10)                   | 4.20E-012               | 1.31E-012               | 2.87E-013              | 9    |

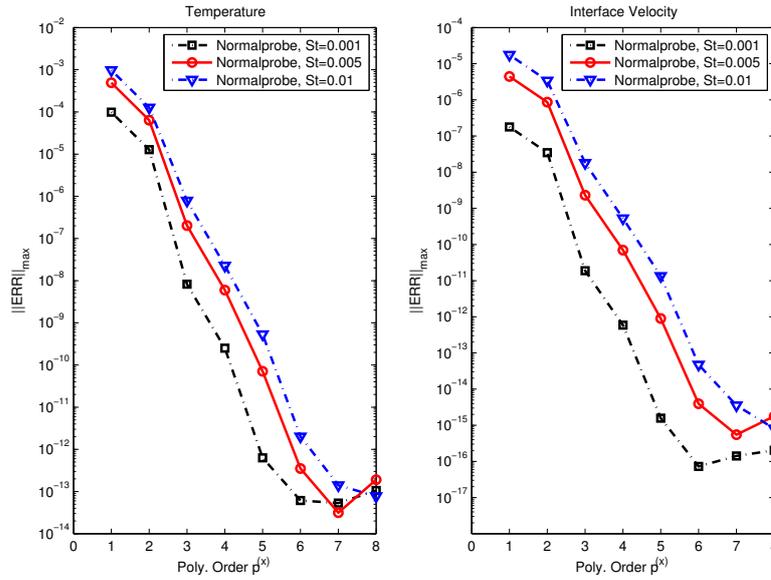
**Table 5** Errors of the temperature ( $\|Err_\theta\|_\infty$ ), interface position ( $\|Err_\Gamma\|_\infty$ ) and interface velocity ( $\|Err_{V_n}\|_\infty$ ) in the last space-time slab. The interface velocity is computed by applying the normal probe method (71). The computational domain is divided into  $E^{(x)} = 6$  in the spatial direction. The interface starts at  $x = 0.1$  and moves to  $x = 0.3$ . The time step is chosen to be  $\Delta t = 4.9 \times 10^{-2}$ . The number of the Picard iteration is listed in the last column with  $tol = 10^{-15}$

21. Bedrossian, J., von Brecht, J.H., Zhu, S., Sifakis, E., Teran, J.M.: A second order virtual node method for elliptic problems with interfaces and irregular domains. *J. Comput. Phys.* **229**(18), 6405–6426 (2010)
22. Assêncio, D.C., Teran, J.M.: A second order virtual node algorithm for Stokes flow problems with interfacial forces, discontinuous material properties and irregular domains. *J. Comput. Phys.* **250**, 77–105 (2013)
23. Sussman, M., Smith, K.M., Hussaini, M.Y., Ohta, M., Zhi-Wei, R.: A sharp interface method for incompressible two-phase flows. *J. Comput. Phys.* **221**(2), 469–505 (2007)
24. Jemison, M., Loch, E., Sussman, M., Shashkov, M., Arienti, M., Ohta, M., Wang, Y.: A coupled level set-moment of fluid method for incompressible two-phase flows. *J. Sci. Comput.* **54**(2-3), 454–491 (2013)
25. Jemison, M., Sussman, M., Arienti, M.: Compressible, multiphase semi-implicit method with moment of fluid interface representation. *J. Comput. Phys.* **279**, 182–217 (2014)
26. Vahab, M., Miller, G.: A front-tracking shock-capturing method for two gases. *Communications in Applied Mathematics and Computational Science* (2015). (in press)
27. Chen, S., Merriman, B., Osher, S., Smereka, P.: A simple level set method for solving Stefan problems. *J. Comput. Phys.* **135**(1), 8–29 (1997)
28. Gibou, F., Fedkiw, R.P., Cheng, L.-T., Kang, M.: A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys.* **176**(1), 205–227 (2002)
29. Gibou, F., Fedkiw, R., Caffisch, R., Osher, S.: A level set approach for the numerical simulation of dendritic growth. *J. Sci. Comput.* **19**(1-3), 183–199 (2003)
30. Gibou, F., Fedkiw, R.: A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. *J. Comput. Phys.* **202**(2), 577–601 (2005)
31. Aslam, T.D.: A partial differential equation approach to multidimensional extrapolation. *J. Comput. Phys.* **193**(1), 349–355 (2004)
32. Benzi, M., Olshanskii, M.A.: An augmented Lagrangian-based approach to the Oseen problem. *SIAM J. Sci. Comput.* **28**(6), 2095–2113 (2006)
33. Rhebergen, S., Cockburn, B.: A space-time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains. *J. Comput. Phys.* **231**(11), 4185–4204 (2012)
34. Berger, M.J., Olinger, J.: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **53**(3), 484–512 (1984)
35. Berger, M.J., Colella, P.: Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* **82**(1), 64–84 (1989)



**Fig. 16** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the ice phase. The interface velocity is computed by the weak form method (67). The polynomial order in time direction,  $p^{(t)}$ , is chosen to be the same as  $p^{(x)}$ . The interface starts at  $x = 0.1$  and moves to  $x = 0.4$ . The comparison is made among different values of  $St$ , i.e.  $St = 0.001$ ,  $St = 0.005$ ,  $St = 0.01$  and  $St = 0.05$ . The computational domain is divided into  $E^x = 6$  in the spatial direction. The time step is chosen to be  $\Delta t = 0.9 \times 10^{-3}$

36. Sussman, M., Almgren, A.S., Bell, J.B., Colella, P., Howell, L.H., Welcome, M.L.: An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.* **148**(1), 81–124 (1999)
37. Chen, H., Min, C., Gibou, F.: A numerical scheme for the Stefan problem on adaptive cartesian grids with supralinear convergence rate. *J. Comput. Phys.* **228**(16), 5803–5818 (2009)
38. Cockburn, B., Shu, C.-W.: The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.* **35**(6), 2440–2463 (1998)
39. Cockburn, B., Kanschat, G., Perugia, I., Schötzau, D.: Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids. *SIAM J. Numer. Anal.* **39**(1), 264–285 (2001)
40. Cockburn, B.B., Karniadakis, G., Shu, C.-W. (eds.): *Discontinuous Galerkin Methods : Theory, Computation, and Applications*. Lecture notes in computational science and engineering. Springer, Berlin, New York (2000)
41. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: *Spectral Methods: Fundamentals in Single Domains*. Berlin, Springer (2006)
42. Hesthaven, J.S., Warburton, T.: *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Texts in Applied Mathematics, vol. 54. Springer, New York (2008)
43. Kopriva, D.A.: *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Scientific Computation. Springer, Berlin (2009)
44. Karniadakis, G.E., Sherwin, S.J.: *Spectral/hp Element Methods for CFD*, 2nd edn. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2013)



**Fig. 17** Errors in the temperature (left) and the interface velocity (right) as a function of polynomial order  $p^{(x)}$  in the ice phase. The interface velocity is computed by the normal probe method (71). The polynomial order in time direction,  $p^{(t)}$ , is chosen to be the same as  $p^{(x)}$ . The interface starts at  $x = 0.1$  and moves to  $x = 0.4$ . The comparison is made among different values of  $St$ , i.e.  $St = 0.001$ ,  $St = 0.02$ ,  $St = 0.05$  and  $St = 0.08$ . The computational domain is divided into  $E^x = 6$  in the spatial direction. The time step is chosen to be  $\Delta t = 0.9 \times 10^{-3}$

45. Sudirham, J.J., van der Vegt, J.J.W., van Damme, R.M.J.: Space-time discontinuous Galerkin method for advection-diffusion problems on time-dependent domains. *Appl. Numer. Math.* **56**(12), 1491–1518 (2006)
46. Klaij, C.M., van der Vegt, J.J.W., van der Ven, H.: Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations. *J. Comput. Phys.* **217**(2), 589–611 (2006)
47. van der Vegt, J.J.W., Sudirham, J.J.: A space-time discontinuous Galerkin method for the time-dependent Oseen equations. *Appl. Numer. Math.* **58**(12), 1892–1917 (2008)
48. Sollie, W.E.H., Bokhove, O., van der Vegt, J.J.W.: Space-time discontinuous Galerkin finite element method for two-fluid flows. *J. Comput. Phys.* **230**(3), 789–817 (2011)
49. Rhebergen, S., Cockburn, B., van der Vegt, J.J.W.: A space-time discontinuous Galerkin method for the incompressible Navier-Stokes equations. *J. Comput. Phys.* **233**, 339–358 (2013)
50. Criscione, A., Kintea, D., Tuković, Ž., Jakirlić, S., Roisman, I.V., Tropea, C.: Crystalization of supercooled water: A level-set-based modeling of the dendrite tip velocity. *Int. J. Heat Mass Transfer* **66**(0), 830–837 (2013)
51. Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* **114**(1), 146–159 (1994)
52. Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D.: Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.* **39**(5), 1749–1779 (2001)
53. Kirby, R.M., Karniadakis, G.E.: Selecting the numerical flux in discontinuous Galerkin methods for diffusion problems. *J. Sci. Comput.* **22/23**, 385–411 (2005)

54. Bassi, F., Rebay, S.: A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.* **131**(2), 267–279 (1997)
55. Baumann, C.E., Oden, J.T.: A discontinuous *hp* finite element method for convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.* **175**(3-4), 311–341 (1999)
56. Welch, S.W.J., Wilson, J.: A volume of fluid based method for fluid flows with phase change. *J. Comput. Phys.* **160**(2), 662–682 (2000)
57. Gassner, G.J., Lörcher, F., Munz, C.-D., Hesthaven, J.S.: Polymorphic nodal elements and their application in discontinuous Galerkin methods. *J. Comput. Phys.* **228**(5), 1573–1590 (2009)
58. Saye, R.I.: High-order methods for computing distances to implicitly defined surfaces. *Commun. Appl. Math. Comput. Sci.* **9**(1), 107–141 (2014)
59. Martinsson, P.G.: A direct solver for variable coefficient elliptic PDEs discretized via a composite spectral collocation method. *J. Comput. Phys.* **242**, 460–479 (2013)