

## VORONOI IMPLICIT INTERFACE METHOD FOR GEOMETRY EVOLUTION OF TWO MINERALS WITH APPLICATIONS IN REACTIVE POROUS MEDIA

P. FROLKOVIČ, N. GAJDOŠOVÁ\* AND S. GÄRTTNER, N. RAY†

**Abstract.** We present a numerical method to describe the precipitation and dissolution processes of two interacting mineral phases and one fluid phase in porous media. We use the Voronoi implicit interface method [9] to track an interface evolving in normal direction that can contain triple points. To represent the interface implicitly, one uses the  $\epsilon$  set of an evolving level set function that is given as the (unsigned) distance function initially or after reinitialization. To obtain the speed of evolution, we combine the Voronoi implicit interface method with a constant extrapolation of the normal speed which is prescribed only at the interface [1, 5].

**Key words.** distance function, advection in normal direction, three phases

**AMS subject classifications.** 35L50, 65M06, 65N06

**1. Introduction.** The tracking of solid-fluid interfaces is relevant for several applications such as melting/solidification and precipitation/dissolution processes. The latter ones are relevant in reactive transport modeling, particularly in complex geometric structures such as porous media. Here, the solid-liquid interface relates to the reactive surface which is an important driving force for possible alterations in the pore structure.

In level set methods [10, 6], the dynamic interface, here  $\Gamma = \Gamma(t)$ , is typically represented as the zero set of some level set function defined for each time  $t \geq 0$  on a fixed rectangular domain  $D \subset \mathbb{R}^2$ . To find such a level set function, e.g. at  $t = 0$ , one may first compute the distance function  $d \geq 0$  to the interface  $\Gamma$  that is typically obtained by solving the eikonal equation [10],

$$|\nabla d(x)| = 1, \quad x \in D, \quad d(x) = 0, \quad x \in \Gamma. \quad (1.1)$$

The interface  $\Gamma$  is the zero set of  $d$ , but the distance function  $d$  has no well-defined gradient at  $\Gamma$  that can cause significant difficulties in numerical methods when describing the position of evolving interface  $\Gamma(t)$ . Therefore, the so-called signed distance function is preferred in level set methods [10], if the interface  $\Gamma$  separates only two phases, and when the sign of  $\pm d$  identifies the phases. However, a single signed distance function cannot be used in scenarios where also triple points may be present, cf. Figure 1.1. Such a situation naturally occurs if, for instance, a system with two mineral phases and one liquid phase is considered. In this case, the (unsigned) distance function  $d$  may be more convenient to represent  $\Gamma$ . To bypass the difficulties arising from standard level set methods in such scenarios, the Voronoi Implicit Interface Method (VIIM) was proposed in [8, 9] that can track multiple evolving phases using the unsigned distance function. In what follows, we apply the VIIM method

---

\*Department of Mathematics and Descriptive Geometry, Slovak University of Technology, Bratislava, Slovakia ([peter.frolkovic@stuba.sk](mailto:peter.frolkovic@stuba.sk)). The work was supported by the grants VEGA 1/0709/19 and APVV-19-0460.

†Department of Mathematics, Friedrich-Alexander University Erlangen-Nürnberg, Erlangen, Germany.

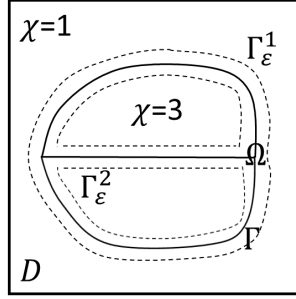


FIG. 1.1. An illustration of the fixed domain  $D$  with the interface  $\Gamma$  (the bold curve) containing two triple points. Two subsets  $\Gamma_\epsilon^1$  and  $\Gamma_\epsilon^2$  of the  $\epsilon$  level set  $\Gamma_\epsilon$  (the dashed curves), two values of the indicator function  $\chi$ , and the subdomain  $\Omega$  are illustrated.

[9] for the advection in normal direction for interfaces containing triple points when a variable speed of the evolution is prescribed only at the interface.

To distinguish the phases with such distance functions, we suppose that an indicator function  $\chi = \chi(x)$  is available which indicates for  $x \in D \setminus \Gamma$  to which phase the point belongs.

If instead of the zero level set of  $d$  one considers the  $\epsilon$  level set  $\Gamma^\epsilon$  defined by

$$x \in \Gamma_\epsilon \quad \Leftrightarrow \quad d(x) = \epsilon, \quad x \in D$$

for some well chosen small parameter  $\epsilon > 0$ , then the gradient  $\nabla d(x)$  is typically well-defined for almost all  $x$  in a neighborhood of  $\Gamma_\epsilon$ . Using the indicator function  $\chi$ , we obtain that the level set  $\Gamma_\epsilon$  consists of three parts  $\Gamma_\epsilon^p$ ,  $p = 1, 2, 3$ , that, for simplicity, we consider to be closed curves, see Figure 1.1 for an illustration. Moreover, for simplicity, we suppose that the level set  $\Gamma_\epsilon$  encloses a domain  $\Omega \subset D$  with  $\Gamma_\epsilon \equiv \partial\Omega$ , see again Figure 1.1 for an illustration.

As described in detail in [9], the interface  $\Gamma$  can then be approximated by the so-called Voronoi implicit interface  $\Gamma^V$  defined by the points  $x \in \Omega$  that are at the same distance to two parts of  $\Gamma^\epsilon$  and at a larger or equal distance to the third one, see details later. We note that the two interfaces,  $\Gamma$  and  $\Gamma^V$ , can differ slightly in general. Nevertheless, to simplify the explanation, we consider from now on that  $\Gamma^V$  approximates the interface we are interested in.

The main idea of the Voronoi implicit interface method is to track the evolution of the  $\epsilon$  level set  $\Gamma_\epsilon$  of  $d$  numerically. If necessary, the actual position of  $\Gamma^V$  is not taken directly from the zero level set of  $d$ , but it is reconstructed from  $\Gamma_\epsilon$  of  $d$  at any time point.

**2. Mathematical models.** As described in the previous section, we suppose that the position of the interface  $\Gamma(0)$  at  $t = 0$  is given implicitly by the unsigned distance function  $d$  from (1.1), and the phases are distinguished by the indicator function  $\chi$ . Next, at any time  $t \geq 0$  we suppose that three distinct interfaces  $\Gamma_\epsilon^p$ ,  $p = 1, 2, 3$  are given implicitly as the zero level set  $\Gamma_\epsilon$  of a function  $\phi$  such that  $\Gamma_\epsilon \equiv \partial\Omega$ , and  $\phi(x, t) > 0$  for  $x \in \Omega$ , and  $\phi(x, t) < 0$  for  $x \in D \setminus \bar{\Omega}$ . Note that at  $t = 0$  one can take  $\phi = \epsilon - d$ . We emphasize that it is sufficient if the function  $\chi$  is well-defined at the time  $t$  only in a small neighborhood of  $\Gamma_\epsilon$ .

In the next steps, we describe how to determine the position of the Voronoi implicit interface  $\Gamma^V$  for the set  $\Gamma_\epsilon$  from the given function  $\phi$  and how to track the

position of  $\Gamma^V$  in time. The method described here is a combination of VIIM in [9] coupled with the extrapolation of the variable speed prescribed only at the interface [1, 5].

**The first step** of the method is to find distance functions  $d^p = d^p(x)$ ,  $p = 1, 2, 3$  to the interfaces  $\Gamma_\epsilon^p$  by solving the eikonal equations

$$|\nabla d^p(x)| = 1, \quad x \in D, \quad d^p(\gamma) = 0, \quad \gamma \in \Gamma_\epsilon^p. \quad (2.1)$$

Note that the zero Dirichlet boundary conditions in (2.1) are defined on implicitly given boundaries  $\Gamma_\epsilon^p$ .

**The second step** in the method is to find (or define implicitly) the Voronoi implicit interface  $\Gamma^V$  for  $\Gamma_\epsilon$  such that

$$\Gamma^V = \{x \in \Omega : 0 < d^p(x) = d^q(x) \leq d^r(x)\}, \quad (2.2)$$

where  $p, q, r \in \{1, 2, 3\}$  and they are all distinct. A simple numerical algorithm is proposed later to solve this problem. Moreover, the values of the indicator function  $\chi(x)$  can now be extended for all  $x \in \Omega$  by

$$\chi(x) = p \Leftrightarrow d^p(x) < \min\{d^q(x), d^r(x)\},$$

and  $\chi(x) = 0$  otherwise.

**The third step** is to find (or “reconstruct”) the unsigned distance function  $d^V$  by solving the eikonal equation, i.e.

$$|\nabla d^V(x)| = 1, \quad x \in D, \quad d^V(\gamma) = 0, \quad \gamma \in \Gamma^V. \quad (2.3)$$

Note that  $\Gamma^V$  in (2.3) is defined only implicitly and it is supposed to contain triple points.

In **the fourth step**, one determines from the position (and eventually from some other properties) of the interface  $\Gamma^V$  a speed  $S = S(x)$  of the evolution of  $\Gamma^V$  in its normal direction at time  $t$ . As such speeds may be well-defined only for  $\gamma \in \Gamma^V$ , say  $S(\gamma) = S^V(\gamma)$ , we need to extrapolate it for  $x \in D$  in general. Such extrapolated function  $S = S(x)$  can be obtained by solving the linear stationary advection equation

$$\nabla d^V(x) \cdot \nabla S(x) = 0, \quad x \in D, \quad S(\gamma) = S^V(\gamma), \quad \gamma \in \Gamma^V. \quad (2.4)$$

(2.4) can be interpreted as a constant extrapolation along the normal vectors of  $\Gamma^V$  for the known values  $S^V$  (that are given only on the interface  $\Gamma^V$ ) to the values  $S$  defined in the whole domain  $D$  [2]. This definition has also some numerical advantages, see Remark 1 later.

**The fifth step** consists of solving the advection equation for the motion in normal direction for some given time interval  $(t, t + \Delta t)$  with small  $\Delta t$  (see section 3 for proper choices),

$$\partial_t \phi + S|\nabla \phi| = 0, \quad x \in D. \quad (2.5)$$

Note that the given function  $\phi$  at time  $t$  in (2.5) can be replaced by  $\phi(t, x) = \epsilon - d^V(x)$  in general. This is called reinitialization, and it is quite often used in level set methods [6, 10].

Once the equation (2.5) is solved, we repeat the five steps for the time point  $t + \Delta t$ . If  $\Delta t$  is small enough, the indicator function  $\chi$  from the third step of our method is

well-defined in the neighborhood of the evolved interface  $\Gamma^V$  at the time  $t + \Delta t$  as required in the first step of the method. The numerical method described later will ensure a reasonable restriction on the choice of  $\Delta t$  to fulfill this requirement.

As discussed in [9], the method treats vanishing subdomains in a natural way, but emerging subdomains, if required, are to be initialized explicitly in the model.

REMARK 1. *We note that the usage of speed  $S$  in (2.5) obtained from (2.4) has the important property that the function  $\phi(x, t + \Delta t)$  is expected to be a distance function to the interface  $\Gamma^V$  at time  $t + \Delta t$ . Although such a property can be proven only for smooth interfaces [12, 1], several numerical experiments show a good approximation of this property in general [12, 1, 5].*

**3. Numerical discretization.** For simplicity of the notation, we consider the quadratic domain  $D = (0, L)^2$ . Let  $I > 0$  be a given number of partitions and  $h = L/I$ . We denote now the coordinates by two letters  $(x, y) \in D$ . We search for the values  $d_{ij}$  that approximate the exact (unsigned) distances  $d(x_i, y_j)$  with  $x_i = ih$  and  $y_j = jh$ ,  $i, j = 0, 1, \dots, I$  by solving numerically (2.1), when  $d \equiv d^p$  and  $\Gamma := \Gamma_\epsilon^p$ , or (2.3), when  $d \equiv d^V$  and  $\Gamma := \Gamma^V$ . We suppose that an approximation  $\phi_{ij}^n \approx \phi(x_{ij}, t^n)$  for some discrete time  $t^n$  is available.

In what follows we present self-contained details on the discretization methods to be used with the method described in Section 2. The main idea is to choose an appropriate approximation of the gradient  $\nabla d_{ij} \approx \nabla d(x_i, y_j)$  [7, 10] that will be used in (2.1), (2.3), or (2.4). It is obtained by a finite difference approximation

$$\nabla d_{ij} = (\partial_x d_{ij}, \partial_y d_{ij}) \approx \frac{1}{h} (\delta_x d_{ij}, \delta_y d_{ij}), \quad (3.1)$$

where

$$\delta_x d_{ij} := \text{sgn}(k)(d_{i+kj} - d_{ij}), \quad \delta_y d_{ij} := \text{sgn}(l)(d_{ij+l} - d_{ij}). \quad (3.2)$$

The indices  $k, l \in \{-1, 0, 1\}$  are determined from

$$k = \arg \min_{m \in \{-1, 0, 1\}} d_{i+mj}, \quad l = \arg \min_{m \in \{-1, 0, 1\}} d_{ij+m}. \quad (3.3)$$

We note that in cases when the interface intersects an edge between  $(x_i, y_j)$  and some of its neighbors, the values  $d_{ij}$  have to be initialized by taking the Dirichlet boundary condition into account, see Section 4.

When using (3.3) for the nodes lying on  $\partial D$ , one has to simply skip the unavailable indices  $m$  in (3.3). No particular preference is recommended when the definition of  $k$  or  $l$  is not unique. We note that the first-order accurate approximation  $\nabla d_{ij}$  can be extended to a second-order accurate form [11] quite straightforwardly.

To solve (2.1), it is rewritten in the form  $|\nabla d|^2 = 1$ . Consequently, the numerical scheme to find the approximative distance function can be written in the form

$$(d_{i+kj} - d_{ij})^2 + (d_{ij+l} - d_{ij})^2 = h^2. \quad (3.4)$$

We briefly discuss how to solve the system of quadratic equations (3.4) with the fast marching method in Section 4.4.

Once the numerical solution  $d_{ij}^V$ ,  $i, j \in \{0, 1, \dots, N\}$  of (2.3) is available, one can solve (2.4) numerically by using the first-order accurate upwind differences,

$$\text{sgn}(k)(d_{i+kj}^V - d_{ij}^V)(S_{i+kj} - S_{ij}) + \text{sgn}(l)(d_{ij+l}^V - d_{ij}^V)(S_{ij+l} - S_{ij}) = 0. \quad (3.5)$$

Note that  $k = l = 0$  for a grid point  $(x_i, y_j)$  can occur only if  $(x_i, y_j) \in \Gamma$ , therefore  $S_{ij}$  shall be then given by Dirichlet boundary conditions. How to solve the resulting linear system (3.5) in an efficient way is discussed in Section 4.4 later.

Finally, once the numerical values  $d_{ij}^V$  and  $S_{ij}$  are given, one can realize one time step for sufficiently small  $\Delta t$  in numerically solving the advection equation (2.5),

$$\phi_{ij}^{n+1} = \phi_{ij}^n - \Delta t S_{ij} |\nabla \phi_{ij}^n|, \quad (3.6)$$

where  $\nabla \phi_{ij}^n$  is obtained analogously to  $\nabla d_{ij}^V$  in (3.1) - (3.3). The time step  $\Delta t$  must be chosen such that the stability of the scheme (3.6) is attained. To derive a criterion for  $\Delta t$ , we rewrite (3.6) into the equivalent form,

$$\phi_{ij}^{n+1} = \phi_{ij}^n - \frac{\Delta t |S_{ij}|}{h} (C_{ij}(\phi_{ij}^n - \phi_{i+kj}^n) + D_{ij}(\phi_{ij}^n - \phi_{ij+l}^n)) \quad (3.7)$$

where

$$C_{ij} := \frac{|\phi_{ij}^n - \phi_{i+kj}^n|}{|\nabla \phi_{ij}^n|}, \quad \text{and} \quad D_{ij} := \frac{|\phi_{ij}^n - \phi_{ij+l}^n|}{|\nabla \phi_{ij}^n|}.$$

Clearly, the scheme (3.6) is stable if its right hand side represents a convex combination of  $\phi_{ij}^n$ ,  $\phi_{i+kj}^n$ , and  $\phi_{ij+l}^n$ , i.e.

$$\Delta t \left( \max_{i,j} |S_{ij}| (C_{ij} + D_{ij}) \right) \leq h. \quad (3.8)$$

As  $C_{ij} \geq 0$  and  $D_{ij} \geq 0$ , and  $C_{ij}^2 + D_{ij}^2 = 1$ , one obtains

$$1 \leq \max_{i,j} (C_{ij} + D_{ij}) \leq \sqrt{2}.$$

Consequently, to fulfill (3.8), it is sufficient to use

$$\Delta t \max_{i,j} |S_{ij}| \leq \frac{h}{\sqrt{2}}. \quad (3.9)$$

For both choices of  $\Delta t$ , (3.8) or (3.9), the value  $\phi_{ij}^{n+1}$  is bounded by the values  $\phi_{ij}^n, \phi_{i+kj}^n, \phi_{ij+l}^n$ . If, for instance, the function  $\phi$  at time  $t^n$  equals to the distance function  $d^V$ , see the discussion after (2.5) on the reinitialization, the values  $\phi_{ij}^{n+1}$  in (3.6) differs from  $\phi_{ij}^n$  at most by a value  $\pm h$ . This property insures that for a properly chosen value  $\epsilon$ , e.g.  $\epsilon = 2h$ , the  $\epsilon$  set of  $\phi$  remains in the same phase for  $t \in [t^n, t^{n+1}]$  as required in Step 5 of the method in Section 2.

**4. Initialization procedure.** We describe now how to define the values  $d_{ij}$  for the grid nodes  $(x_i, y_j)$  next to the interface  $\Gamma$ , for which the zero Dirichlet boundary conditions are prescribed. We do it first for the eikonal equation (2.1) following [1, 5], and then for (2.3) with a slight modification. Finally, the initialization procedure will be explained for the linear advection equation (2.4). Our main tool in deriving the following results is a simple assumption of linear interpolation along edges of the mesh for all involved numerical approximating functions.

We note that instead of defining (fixing) the values  $d_{ij}$  for the grid nodes next to the interface, one can obtain these values from (3.4) using an extrapolation procedure for the unavailable values  $d_{i+kj}$  or  $d_{ij+l}$  as described in [5]. The following description of the initialization procedure can be rather straightforwardly replaced with the extrapolation procedure in [5], and its implementation is foreseen for a future development as it can result in a more precise approximation, see [5] for details.

**4.1. The eikonal equation (2.1).** Let  $\phi_{ij}^n > 0$  and  $\Gamma = \Gamma_\epsilon^p$ , i.e.  $(x_i, y_j) \in \Omega$ . We say that the grid node  $(x_i, y_j)$  is *next to the interface  $\Gamma$  in  $x$  direction* if at least one of the following two inequalities is valid:

$$\phi_{ij}^n \phi_{i-1j}^n < 0 \text{ or } \phi_{ij}^n \phi_{i+1j}^n \leq 0. \quad (4.1)$$

Analogously, we say the node  $(x_i, y_j)$  is *next to the interface  $\Gamma$  in  $y$  direction* if at least one of following inequalities is true:

$$\phi_{ij}^n \phi_{ij-1}^n < 0 \text{ or } \phi_{ij}^n \phi_{ij+1}^n < 0. \quad (4.2)$$

Using (4.1) and (4.2), we approximate positions of the points on  $\Gamma$  that intersect edges of the grid. Let  $(x_i, y_j)$  be, e.g., a grid node next to the interface in  $x$  direction. We search for a coordinate  $x_{i-\alpha_j}$  or  $x_{i+\alpha_j}$  that will be identified with some  $\alpha_{i-1j} \in (0, 1)$  or  $\alpha_{i+1j} \in (0, 1)$ , respectively, such that

$$x_{i\pm\alpha_j} = \alpha_{i\pm 1j} x_{i\pm 1} + (1 - \alpha_{i\pm 1j}) x_i. \quad (4.3)$$

Analogously, the values  $\alpha_{ij\pm 1} \in (0, 1)$  and the point  $y_{ij\pm\alpha}$  shall be found. The *interface points*  $(x_{i\pm\alpha}, y_j)$  and  $(x_i, y_{j\pm\alpha})$  approximate the intersection points of  $\Gamma$  with the edges of the grid.

The values  $\alpha_{i\pm 1j}$  are determined from the linear interpolation of  $\phi_{ij}^n$  and  $\phi_{i\pm 1j}^n$  by requiring that  $0 = \alpha_{i\pm 1j} \phi_{i\pm 1j}^n + (1 - \alpha_{i\pm 1j}) \phi_{ij}^n$  and analogously for  $\alpha_{ij\pm 1}$ . Clearly,

$$\alpha_{i\pm 1j} = \frac{\phi_{ij}^n}{\phi_{ij}^n - \phi_{i\pm 1j}^n}, \quad \text{and} \quad \alpha_{ij\pm 1} = \frac{\phi_{ij}^n}{\phi_{ij}^n - \phi_{ij\pm 1}^n}. \quad (4.4)$$

Note that the values  $\alpha_{i\pm 1j}$  and  $\alpha_{ij\pm 1}$  are specific for each grid node  $(x_i, y_j)$  that we do not emphasize in their notation. Furthermore, some care shall be taken in general for very small values of  $\phi_{ij}^n > 0$  when one might prefer to set  $\phi_{ij}^n = 0$ .

Following [1, 5], one can now define the values  $d_{ij}$  for the nodes next to the interface explicitly. Firstly, if there is only one interface point next to  $(x_i, y_j)$ , we approximate [1]

$$d_{ij} = \alpha_{i\pm 1j} h \text{ or } d_{ij} = \alpha_{ij\pm 1} h. \quad (4.5)$$

If there is one interface point in  $x$  direction and one interface point in  $y$  direction, we define

$$d_{ij} = \frac{\alpha_{i\pm 1j} \alpha_{ij\pm 1}}{\sqrt{\alpha_{i\pm 1j}^2 + \alpha_{ij\pm 1}^2}} h, \quad (4.6)$$

where the signs in  $\pm$  must be chosen accordingly. The definitions (4.5-4.6) must be modified in cases, when there are two interface points next to  $(x_i, y_j)$  in  $x$  or  $y$  direction by replacing the corresponding  $\alpha_{i\pm 1j}$  or  $\alpha_{ij\pm 1}$  in (4.5-4.6) by the minimum of the two values, e.g. for (4.5) we modify it by

$$d_{ij} = \min\{\alpha_{i-1j}, \alpha_{i+1j}\} h \text{ or } d_{ij} = \min\{\alpha_{ij-1}, \alpha_{ij+1}\} h,$$

and analogously for (4.6).

**4.2. The eikonal equation (2.3).** Let  $d \equiv d^V$  and  $\Gamma := \Gamma^V$ . We suppose that the distance functions  $d^p$  in (2.1) are obtained by the numerical method described in Section 4.1. Consequently, we now define for all combinations of three different values of  $p, q, r \in \{1, 2, 3\}$  that  $\chi_{ij} = p$  if  $d_{ij}^p < \min\{d_{ij}^q, d_{ij}^r\}$  and  $\chi_{ij} = 0$  otherwise. Note that if  $\chi_{ij} = 0$  then  $x_{ij} \in \Gamma$ , and therefore  $d_{ij}^V = 0$ .

Let  $\chi_{ij} \neq 0$ . We say that the grid node  $(x_i, y_j)$  is *next to the interface  $\Gamma$  in  $x$  direction* if at least one of the following statement is valid:

$$\chi_{ij} \neq \chi_{i-1j} \neq 0 \text{ or } \chi_{ij} \neq \chi_{i+1j} \neq 0. \quad (4.7)$$

Analogously, we say the node  $(x_i, y_j)$  is *next to the interface  $\Gamma$  in  $y$  direction* if at least one of the following inequalities is true:

$$\chi_{ij} \neq \chi_{ij-1} \neq 0 \text{ or } \chi_{ij} \neq \chi_{ij+1} \neq 0. \quad (4.8)$$

Let  $(x_i, y_j)$  be a point next to  $\Gamma$  in  $x$  direction. Let  $p = \chi_{ij}$  and  $q = \chi_{i\pm 1j}$ , i.e.,  $d_{ij}^p < d_{ij}^q$  and  $d_{i\pm 1j}^p > d_{i\pm 1j}^q$ . If we now denote in (4.4), formally, that  $\phi_{ij}^n := d_{ij}^p - d_{ij}^q$  and  $\phi_{i\pm 1j}^n := d_{i\pm 1j}^p - d_{i\pm 1j}^q$ , then the point  $x_{i\pm\alpha j}$  is given by (4.3) with the coefficient  $\alpha_{i\pm 1j}$  given by (4.4). Special care shall be given to the case when  $|d_{ij}^p - d_{ij}^q|$  is very small, when one shall prefer to redefine  $\chi_{ij} = 0$ . We proceed analogously to define the points  $(x_i, y_{j\pm\alpha})$  and the coefficients  $\alpha_{ij\pm 1}$ . Consequently, the *interface points*  $(x_{i\pm\alpha}, y_j)$  and  $(x_i, y_{j\pm\alpha})$  approximate the intersection points of  $\Gamma$  with the edges of the grid.

Analogously to the previous section, we can initialize the values of  $d_{ij}$  for the points next to the interface. The definitions (4.5) and (4.6) remain unchanged, if  $(x_i, y_j)$  and its neighbors lie only in two phases, i.e.  $\chi_{i\pm 1j} = \chi_{ij\pm 1} \neq \chi_{ij}$ . The presence of a triple point is indicated, if the vertices  $(x_i, y_j)$ ,  $(x_{i\pm 1j}, y_j)$ , and  $(x_i, y_{j\pm 1})$  of a triangle lie in three different phases, i.e.  $\chi_{ij} \neq \chi_{i\pm 1j}$ ,  $\chi_{ij} \neq \chi_{ij\pm 1}$ , and  $\chi_{i\pm 1j} \neq \chi_{ij\pm 1}$ . In this case we define

$$d_{ij} = \min\{\alpha_{i\pm 1j}, \alpha_{ij\pm 1}\}h. \quad (4.9)$$

Finally, if some approximation of the position of a triple point is required, for instance to compute the length of an interface or the area of an enclosed domain, one might take any reasonably approximated position inside of the triangular element, where the triple point is indicated.

**4.3. The advection equation (2.4).** The previous idea from section 4.2 can be used also for the advection equation, where the main difference to (2.3) is that the Dirichlet boundary conditions in (2.4) are non-homogeneous. Having the positions of interface points  $(x_{i\pm\alpha}, y_j)$  or  $(x_i, y_{j\pm\alpha})$  from (4.3) at hand, we use the Dirichlet boundary conditions in (2.4) to define the values  $S_{i\pm\alpha j} = S^V(x_{i\pm\alpha}, y_j)$  or  $S_{ij\pm\alpha} = S^V(x_i, y_{j\pm\alpha})$ , respectively. Now depending on which formula from (4.5), (4.6), or (4.9) was used to define  $d_{ij}$ , we define analogously

$$\text{if (4.5), then } S_{ij} = S_{i\pm\alpha j} \text{ or } S_{ij} = S_{ij\pm\alpha}, \quad (4.10)$$

$$\text{if (4.6), then } S_{ij} = \frac{\alpha_{i\pm 1j}^2 S_{i\pm\alpha j} + \alpha_{ij\pm 1}^2 S_{ij\pm\alpha}}{\alpha_{i\pm\alpha j}^2 + \alpha_{ij\pm 1}^2}, \quad (4.11)$$

$$\text{if (4.9) then } S_{ij} = S_{i\pm\alpha j} \text{ for } \alpha_{i\pm 1j} \leq \alpha_{ij\pm 1} \text{ and } S_{ij} = S_{ij\pm\alpha} \text{ otherwise.} \quad (4.12)$$

The extrapolated values  $S_{i,j}$  shall be used in the scheme (3.5) for the nodes next to the interface  $\Gamma^V$ .

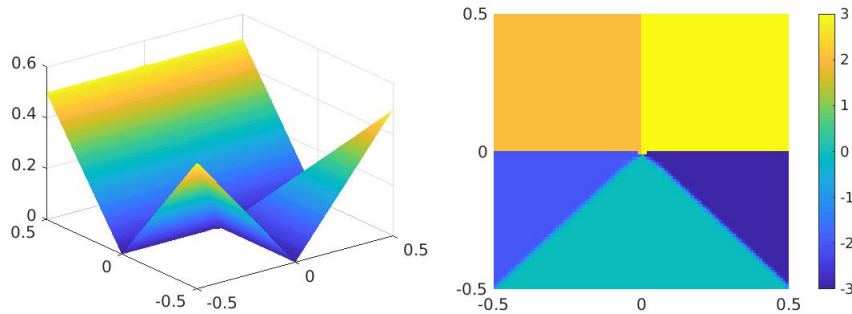


FIG. 5.1. Distance function from the Voronoi interface  $d^V$  (left) and velocity extension  $S$  (right).

**4.4. Numerical solution of algebraic system (3.4).** Before introducing an algorithm to solve the algebraic system of quadratic equations (3.4), we comment briefly the possible forms of these equations. We note that using the previous sections, the values  $d_{ij}$  for the nodes next to the interface are initialized.

In a generic case when  $k \neq 0$  and  $l \neq 0$  in (3.3), the equation (3.4) has three unknowns  $d_{ij}$ ,  $d_{i+kj}$ , and  $d_{ij+l}$ . If either  $k = 0$  or  $l = 0$ , the equation (3.4) has only two unknowns. Note that one cannot have  $k = l = 0$  in (3.4) as this can happen only when Dirichlet boundary conditions are used for  $d_{ij}$ , and when the algebraic equation (3.4) is not used.

To solve such a system of quadratic algebraic equations, the fast marching method [10] is used with no modifications. This iterative method suggests consecutive steps in solving (3.4) in such a way that only one scalar quadratic equation with single unknown  $d_{ij}$  is to be solved in each step. The solution is then obtained in a finite number of iterations, for details see [10, 1, 5].

**5. Numerical experiments.** We now illustrate the capability of our method introduced in the previous sections with the following two scenarios: First, we consider a configuration with two mineral phases and one fluid phase forming initially a T-shaped geometry with triple point. The example is chosen to clearly demonstrate the behavior of the model and the method to solve it. The corresponding distance functions to the connected components of  $\Gamma_\epsilon$ , i.e.  $d_1$ ,  $d_2$ ,  $d_3$ , are depicted in Figure 5.2. We now assume that the interface 1 moves with speed one and the interface 2 with speed two, while the third interface is immobile, i.e. the speed equals zero. These locally defined velocities must be extended to the whole domain as discussed above and illustrated in Figure 5.1 (right). Figure 5.3 depicts the comparison between the initial and final interface configuration of the simulation.

We applied our method with the parameter  $\epsilon = 2h$  being chosen as the doubled mesh spacing on a regular 100x100 grid. The time stepping follows the stability constraints on the maximal time step size as discussed in Section 3. All simulations are performed using MATLAB. The fast marching solver is built upon an implementation by Jens Oberlander.

As a second and more sophisticated illustration, we apply our method to the example motivated in the introduction. Here, we again consider two minerals and one fluid phase - but this time arranged in a circle being composed of two semicircles for mineral D (right) and P (left), cf. Figure 5.4 left. In this research we focus on the



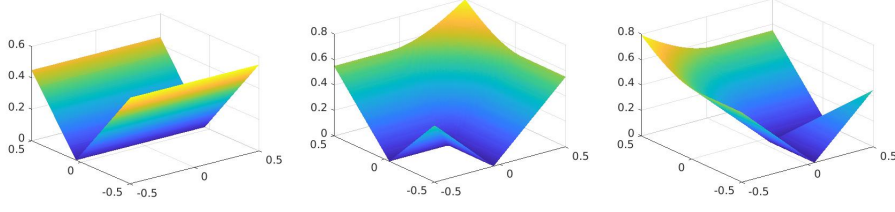
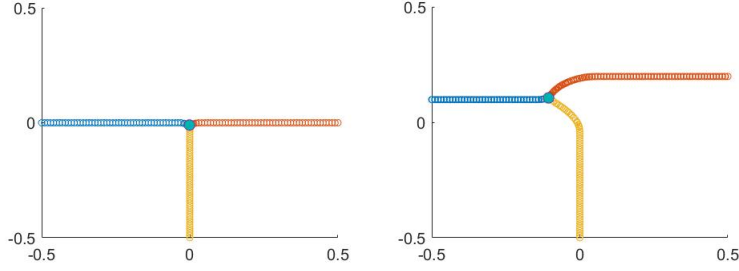

 FIG. 5.2. Distance functions  $d^1$ ,  $d^2$ ,  $d^3$ .


FIG. 5.3. Comparison between the initial interface position and its final position. The location of the tripel point is highlighted in green.

following reactive system, cf. [3]



This system includes three mobile species and two mineral species which dissolve or precipitate respectively. First, species B is produced while D is dissolved (reaction 5.1), followed by its chemical conversion into the precipitating mineral P (reaction 5.2).

The chemical reactions are represented by the following system of ordinary differential equations denoting the vector of (mobile) species' concentrations as  $\vec{c} = (c_A, c_B, c_C)^T$

$$\frac{d}{dt} \vec{c} = \begin{pmatrix} \sigma_D \left( \frac{c_B}{c_A} - 1 \right) (\rho_D - c_A) \\ -\sigma_P (c_B c_C - 1) (\rho_P - c_B) - \sigma_D \left( \frac{c_B}{c_A} - 1 \right) (\rho_D - c_B) \\ -\sigma_P (c_B c_C - 1) (\rho_P - c_C), \end{pmatrix} \quad (5.3)$$

where  $\rho_{(\cdot)}$  represents the mineral density and  $\sigma_{(\cdot)}$  denotes the length of the respective interface to the fluid (reactive surface). In this example, the values  $\rho_D = 20$  and  $\rho_P = 4$  are chosen. The normal velocities  $S_D$  and  $S_P$  along the fluid-solid interfaces of  $D$  and  $P$  are accordingly given by

$$S_D = \frac{c_B}{c_A} - 1, \quad S_P = c_B c_C - 1, \quad (5.4)$$

respectively. The initial conditions are chosen as  $\vec{c}_0 = (2, 1, 1)^T$  such that the reaction (5.2) is at chemical equilibrium. At the same time, the abundance of species  $A$  triggers reaction (5.1).

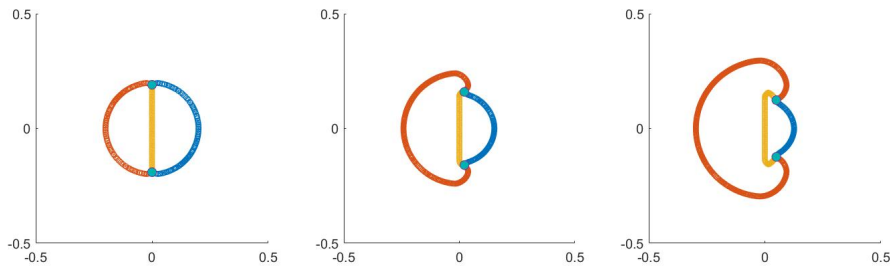


FIG. 5.4. Position of the interfaces and the two triple points at different times: 0s, 0.2s, 2s. The simulation is performed on a  $200 \times 200$  grid.

The respective mineral phases dissolve or precipitate according to the reaction network (5.1, 5.2), which leads to dynamic interface velocities according to (5.4). This is opposed to the situation of the first example, in which the interface velocity was prescribed. Our method allows to accurately capture the alteration of the geometry as well as the position of the triple points. A time series of its evolution is depicted in Figure 5.4.

As such reactions inherently lead to structural changes of the medium, one might wish to capture the geometry accurately since flow and transport paths are altered during its evolution. This is crucial for reactive transport applications in porous media such as  $\text{CO}_2$  storage, see e.g. [4].

#### REFERENCES

- [1] D. Adalsteinsson and J. Sethian. The fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148:2–22, 1999.
- [2] T. D. Aslam. A partial differential equation approach to multidimensional extrapolation. *J. Comput. Phys.*, 193:349–355, 2003.
- [3] L. Chen, Q. Kang, B. Carey, W.-Q. Tao. Pore-scale study of diffusion–reaction processes involving dissolution and precipitation using the lattice Boltzmann method. *International Journal of Heat and Mass Transfer*, 75: 483-496 (2014).
- [4] J. Gao, H. Xing, Z. Tian, J. K. Pearce, M. Sedek, S. D. Golding, V. Rudolph. Reactive transport in porous media for  $\text{CO}_2$  sequestration: Pore scale modeling using the lattice Boltzmann method. *Computers & Geosciences*, 98, 9-20, 2017.
- [5] P. Frolkovič, K. Mikula, and J. Urbán. Distance function and extension in normal direction for implicitly defined interfaces. *Discrete and continuous dynamical systems-series S*, 8(5), 871-880.
- [6] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.
- [7] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Num. Anal.*, 29:867–884, 1992.
- [8] R.I. Saye and J.A. Sethian. The Voronoi Implicit Interface Method for computing multiphase physics. *Proc. Natl. Acad. Sci. U. S. A.*, 108:19498–19503, 2011.
- [9] R.I. Saye and J.A. Sethian. Analysis and applications of the Voronoi Implicit Interface Method. *J. Comp. Phys.*, 231:6051–6085, 2012.
- [10] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [11] J.A. Sethian. Fast Marching Methods. *SIAM Review*, 41:199–235, 1999.
- [12] H. Zhao, T. Chan, B. Merriman, and S. Osher. A Variational Level Set Approach to Multiphase Motion. *J. Comput. Phys.*, 127:179–195, 1996.