

IMMERSED INTERFACE METHOD FOR A LEVEL SET FORMULATION OF PROBLEMS WITH MOVING BOUNDARIES*

PETER FROLKOVIČ†

Abstract. The motivation of this study is a numerical solution of Laplace equation $-\Delta p = 0$ on a time dependent domain that contains a moving boundary evolving with a speed depending on the gradient ∇p . We use a level set formulation of this problem and propose an immersed interface method for the numerical solution of some partial differential equations solved in the level set formulation. This paper can be seen as an accompanying publication to [4] where an application to groundwater flow with dynamic water table is studied. Additionally to [4], technical details of the level set and immersed interface method are given and a new numerical experiment is provided.

Key words. level set method, immersed interface method, finite volume method

AMS subject classifications. 65M08, 65N08, 35R37

1. Introduction. Our representative model is the Laplace equation

$$-\Delta p(x, t) = 0, \quad x \in \Omega(t) \subset R^2, \quad (1.1)$$

where the dependency of p on t is only due to the time dependent domain $\Omega(t)$, $t \in [0, T]$. The domain is changing in time due to its moving boundary (or some part of it) denoted by $\Gamma(t)$, i.e. $\Gamma(t) \subset \partial\Omega(t)$. Moreover we suppose that the movement of $\Gamma(t)$ is driven by a vector function \vec{V} that depends on ∇p .

A popular numerical method to solve (1.1) for a fixed time t is the immersed interface method [11, 12]. The idea of the method is to immerse the boundary (or “interface”) $\Gamma(t)$ into some fixed domain D such that $\Omega(t) \subset D$. The problem (1.1) is then extended to D and the boundary conditions on $\Gamma(t)$ are treated without an explicit reconstruction of $\Gamma(t)$. We note that in many applications the problem defined on $\Omega(t)$ is coupled through some interface condition given on $\Gamma(t)$ with another problem defined on $D \setminus \Omega(t)$, see [11, 12].

To model the moving boundary $\Gamma(t)$ for $t \in [0, T]$ we use the level set formulation [14, 13]. This formulation is very natural when solving (1.1) with some immersed interface methods. The interface $\Gamma(t)$ in such methods is very often defined implicitly as the zero level set of some level set function. The time evolution of this function can be then obtained as a solution of some level set advection equation solved on D . The advection velocity in this equation must depend on \vec{V} .

To construct the advection equation for the level set function we use an approach presented in [1]. In this approach, two additional equations are defined in the level set formulation, the eikonal equation and the equation for so called extension velocity. The equations are accompanied with Dirichlet boundary conditions given on $\Gamma(t)$. To solve such equations with some standard level set methods, the boundary conditions must be reconstructed explicitly for grid nodes near the interface $\Gamma(t)$, see [1]. The idea presented in this paper is to avoid such reconstruction and to apply the immersed

*This work was supported by VEGA 1/1137/12 and APVV-0184-10.

†Department of Mathematics and Descriptive Geometry, Slovak University of Technology, Radlinského 11, 81368 Bratislava, Slovakia (peter.frolkovic@stuba.sk).

interface method for the numerical solution of eikonal equation and the extrapolation equation for the extension velocity.

2. Immersed interface level set formulation. Let D represent a fixed domain that encloses $\Omega(t)$ during the considered time interval, i.e. $\Omega(t) \subset D$, $t \in [0, T]$. The part of D outside of $\Omega(t)$ will be denoted by $\Omega^{out}(t) := D \setminus \Omega(t)$. If $\Gamma(t)$ is not a closed curve, the end points shall lie on ∂D .

The main principle of level set formulation is to describe $\Gamma(t)$ implicitly as a zero level set of some smooth (“level set”) function $\phi = \phi(x, t)$ together with a convenient sign property,

$$\phi(x, t) = 0, x \in D \quad \Leftrightarrow \quad x \in \Gamma(t), \quad (2.1)$$

$$\phi(x, t) < 0, x \in D \quad \Leftrightarrow \quad x \in \Omega(t). \quad (2.2)$$

The level set function offers several geometrical information about $\Gamma(t)$ and $\Omega(t)$ in a straightforward algebraic way. Particularly, the sign property (2.2) helps to identify easily if $x \in \Omega(t)$ or not. The normalized gradient (if it exists and is nonzero) of any level set function that fulfills (2.1) and (2.2) coincides at $\Gamma(t)$ with its unit outward normal vector \vec{N}_γ , i.e.

$$\vec{N}_\gamma = \frac{\nabla\phi(\gamma, t)}{|\nabla\phi(\gamma, t)|}, \quad \gamma \in \Gamma(t). \quad (2.3)$$

A popular particular choice for the implicit representation of $\Gamma(t)$ is the so called signed distance function, say $\Phi = \Phi(x, t)$, that can be found as a weak solution of nonlinear eikonal equation, see e.g. [14, 13],

$$|\nabla\Phi| = 1, x \in D, \quad \Phi(x, t) = 0, x \in \Gamma(t). \quad (2.4)$$

Following [14], the weak solution of (2.4) is a continuous function on $D \times [0, T]$, and its gradient is defined almost everywhere in $D \times [0, T]$. The name of signed distance function Φ is justified by its property [14]

$$\Phi(x, t) = \begin{cases} \min_{\gamma \in \Gamma(t)} |x - \gamma|, & x \in \Omega^{out}(t) \\ - \min_{\gamma \in \Gamma(t)} |x - \gamma|, & x \in \Omega(t). \end{cases} \quad (2.5)$$

The time dependency of Φ is again only due to the variability of $\Gamma(t)$ in time. Clearly, $\Phi(x, t)$ describes $\Gamma(t)$ in an implicit way as required by (2.1) and (2.2).

The most important advantage of level set formulation is the possibility to use Eulerian type of numerical methods to track the movement of $\Gamma(t)$. Before explaining it we introduce a Lagrangian description that might be more related to a physical interpretation of this process.

Let the initial position $\Gamma(0)$ of interface be given. One can view the moving position $\Gamma(t)$ as a tracking of trajectories $X = X(P, t)$ for all points P located at $\Gamma(0)$, i.e.

$$\Gamma(t) = \{X(P, t); P \in \Gamma(0)\}, t \in [0, T]. \quad (2.6)$$

Consequently, for any $\gamma \in \Gamma(t)$ and $t \in [0, T]$ there exists a point $P \in \Gamma(0)$ such that $\gamma = X(P, t)$.

From this point of view, the Lagrangian description of moving interface is typically given by a parametric system of ordinary differential equations for unknown trajectories $X = X(P, t)$,

$$\partial_t X = \vec{V}(X, t), \quad t \in [0, T], \quad X(P, 0) = P, \quad (2.7)$$

where $\vec{V} = \vec{V}(\gamma, t)$ is some prescribed velocity defined at the interface for $\gamma \in \Gamma(t)$. The position $\Gamma(t)$ can be then determined by solving (2.7) for all $P \in \Gamma(0)$ and using (2.6).

The related Eulerian description for capturing the position $\Gamma(t)$ using a level set formulation can be now constructed. Before doing it, we replace the velocity \vec{V} by a vector field $S\vec{N}_\gamma$ to describe the movement of $\Gamma(t)$, where S is the component of \vec{V} projected on the normal direction \vec{N}_γ , i.e.

$$S(\gamma, t) = \vec{N}_\gamma \cdot \vec{V}(\gamma, t), \quad \gamma \in \Gamma(t). \quad (2.8)$$

By doing so, we neglect the movement of points $P \in \Gamma(t)$ along the interface $\Gamma(t)$. Consequently, by replacing \vec{V} in (2.7) with $S\vec{N}_\gamma$ one does not follow anymore the trajectories $X(P, t)$. Nevertheless, the movement of $\Gamma(t)$ as a whole curve does not change when moving it only in its normal direction with the speed S .

To finish the Eulerian description one needs some functions $s = s(x, t)$ and $\vec{N} = \vec{N}(x, t)$ that are defined for $(x, t) \in D \times [0, T]$ and that coincides with $S(\gamma, t)$ and \vec{N}_γ for $\gamma \in \Gamma(t)$.

If these functions are available, the level set function $\phi = \phi(x, t)$ in (2.1) is searched by solving the following advection equation

$$\partial_t \phi + s \vec{N} \cdot \nabla \phi = 0, \quad \phi(x, 0) = \phi^0(x). \quad (2.9)$$

The initial level set function $\phi^0 = \phi^0(x)$ in (2.9) must be a smooth function such that $\Gamma(0) = \{\phi^0(x) = 0, x \in D\}$ and the sign property (2.2) is fulfilled. For practical reasons we prefer the signed distance function, i.e. $\phi^0(x) = \Phi(x, 0)$. For a simple initial interface such function can be defined straightforwardly, see (2.5), in general the eikonal equation (2.4) to find $\Phi(x, 0)$ has to be solved.

Note that s and \vec{N} in (2.9) need not to be available directly in a particular application. Therefore these functions shall be determined by some appropriate extension of their known values for $\gamma \in \Gamma(t)$. We describe next how to obtain such globally defined functions s and \vec{N} , see also [1, 9, 4].

Firstly, we set

$$\vec{N}(x, t) = \nabla \Phi(x, t), \quad x \in D, \quad t \in [0, T]. \quad (2.10)$$

The signed distance function $\Phi(x, t)$ in (2.10) is obtained by solving the eikonal equation (2.4).

The gradient $\nabla \Phi(x, t)$, if it exists and is nonzero, is already normalized due to (2.4), it coincides with \vec{N}_γ at $\gamma \in \Gamma(t)$ and has some additional favorable properties. Namely, if $\Phi(x, t) \neq 0$ and $\gamma \in \Gamma(t)$ is the point in which the minimum in (2.5) is realized for x , then $\nabla \Phi$ is a constant vector for all points along the straight line connecting the points γ and x . Consequently, $\nabla \Phi$ (if it exists) can be seen as a constant prolongation of normal vectors \vec{N}_γ from $\gamma \in \Gamma(t)$ up to the point $x \in D$.

This property of $\nabla\Phi$ can be conveniently used also for an extrapolation of the speed $\mathcal{S}(\gamma, t)$ in the direction of normals to define $s(x, t)$ by solving the equation

$$\nabla\Phi \cdot \nabla s = 0, \quad x \in D, \quad s(\gamma, t) = S(\gamma, t), \quad \gamma \in \Gamma(t). \quad (2.11)$$

The solution s of (2.11) can be viewed as a constant prolongation of values $\mathcal{S}(\gamma, t)$ from $\gamma \in \Gamma(t)$ along the straight lines given by the direction of \vec{N}_γ up to points where $\nabla\Phi$ is still well defined. It is called the extension velocity in [1]. In general, $s(x, t)$ can be discontinuous at points $x \in D$ and $t \in [0, T]$ where $\nabla\Phi(x, t)$ does not exist.

The level set formulation of (1.1) now consists of solving (1.1) simultaneously with the equations (2.4), (2.11) and (2.9). We call it the immersed interface formulation because the moving boundary $\Gamma(t)$ (for which some boundary conditions in equations (1.1), (2.4) and (2.11) are defined) is given only implicitly as the zero level set of $\phi(x, t)$ from (2.9).

3. Finite volume discretization. We describe the flux-based level set method [5, 6] for the numerical solution of advection equations (2.4), (2.9) and (2.11). Before doing it, we introduce the advection equation of related general form,

$$\partial_\tau u + \vec{v} \cdot \nabla u = r, \quad u(x, 0) = u^0(x), \quad (3.1)$$

where $u = u(x, \tau)$ is the unknown function, τ is a (pseudo-) time variable, u^0 is a given initial function, $\vec{v} = \vec{v}(x, \nabla u)$ is a prescribed velocity function and r is a constant right hand side. Clearly, the eikonal equation (2.4), the extrapolation equation (2.11) and the advection equation (2.9) can be seen as particular forms of (3.1), see also [4] for some details.

Let the domain $D \subset \mathbb{R}^2$ be polygonal and triangulated using triangles T^e , $e = 1, 2, \dots, E$. The vertices (nodes) of the triangulation are denoted by x_i , $i = 1, 2, \dots, I$ and located in the corners of triangles. The time interval is divided into subintervals $0 = \tau^0 < \tau^1 < \dots$ and $\Delta\tau^m = \tau^{m+1} - \tau^m$. Our aim is to derive a fully explicit (in time) flux-based level set method to solve (3.1) to approximate $u_i^{m+1} \approx u(x_i, \tau^{m+1})$.

With the triangulation and its vertices we associate piecewise linear functions $\psi_i = \psi_i(x)$, $x \in D$ such that $\psi_i(x_j) = \delta_{ij}$ for $i, j = 1, 2, \dots, I$ and define

$$\tilde{u}^m(x) = \sum_{i=1}^I u_i^m \psi_i(x). \quad (3.2)$$

We denote by $\nabla^e \psi_i$ the constant gradient of $\psi_i(x)$ for $x \in T^e$ and analogously $\nabla^e u^m := \nabla \tilde{u}^m(x)$ for $x \in T^e$.

The so called vertex-centered finite volume discretization, see [7, 6] for details, is based on “finite volumes” V_i that are associated with the vertices x_i and the boundaries ∂V_i are made of line segments Γ_{ij}^e such that

$$\partial V_i = \bigcup_{e \in \Lambda_i} \bigcup_{j \in \Lambda_i^e} \Gamma_{ij}^e, \quad \Gamma_{ij}^e := \partial V_i \cap \partial V_j \cap T^e, \quad \Gamma_{i0}^e := \partial V_i \cap \partial D \cap T^e. \quad (3.3)$$

In (3.3), Λ_i is a set of indices e of all T^e of which x_i is a corner, and Λ_i^e is a set of indices j of all corners x_j of T^e except x_i and $0 \in \Lambda_i^e$ if $\Gamma_{ij}^e \subset \partial D$, for an illustration see Figure 3.1.

Let us suppose first that $\Omega(t^n) \equiv D$, the immersed interface formulation when $\Omega(t^n) \subset D$ will be given later.

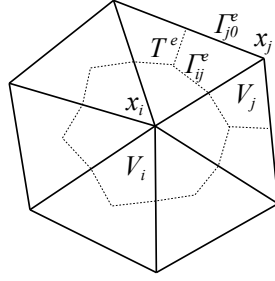


FIG. 3.1. Notation used for finite volume method.

To derive the numerical scheme we fix the velocity \vec{v} at time τ^m and use a shorter notation $\vec{v}(x) := \vec{v}(x, \nabla u(x, \tau^m))$ and $\vec{v}_{ij}^e \approx \vec{v}(x_{ij}^e)$. Following [6, 8], the flux-based level set method is defined by

$$u_i^{m+1} = u_i^m + \frac{\Delta\tau^m}{|V_i|} \left(r - \sum_{e \in \Lambda_i} \sum_{j \in \Lambda_i^e} |\Gamma_{ij}^e| \bar{n}_{ij}^e \cdot \vec{v}_{ij}^e (u_{ij}^{e,m+1/2} - u_i^{m+1/2}) \right). \quad (3.4)$$

In above, \bar{n}_{ij}^e is the unit normal vector w.r.t. Γ_{ij}^e pointing from V_i to V_j (if $j \neq 0$) or to outside if $j = 0$. Furthermore, $|V_i|$ is the area of V_i , $|\Gamma_{ij}^e|$ is the length of Γ_{ij}^e .

Furthermore,

$$u_i^{m+1/2} = u_i^m - \frac{\Delta\tau^m}{2} \vec{v}_i \cdot (\nabla u)_i^m, \quad (3.5)$$

$$u_{ij}^{e,m+1/2} = u_i^m + \left(x_{ij}^e - \frac{\Delta\tau^m}{2} \vec{v}_i \right) \cdot (\nabla u)_i^m, \quad \text{if } \bar{n}_{ij}^e \cdot \vec{v}_{ij}^e > 0, \quad (3.6)$$

where $\vec{v}_i \approx \vec{v}(x_i)$ and

$$(\nabla u)_i^m = \frac{1}{|V_i|} \sum_{e \in \Lambda_i} |T^e \cap V_i| \sum_{k \in \Lambda_i^e} (u_k^m - u_i^m) \nabla^e \psi_k. \quad (3.7)$$

A more detailed description of all derivations used to define (3.4) including supplementary information on the treatment of boundary conditions can be found in [6, 8, 3].

The numerical scheme (3.4) is fully explicit in time and enables us to compute directly the values u_i^{m+1} in sequence for $m = 0, 1, \dots$. It has a natural restriction on the choice of time step $\Delta\tau^m$ that can be formulated for the flux-based level set method [5, 6] by requiring

$$\Delta\tau^m \max_i \left\{ \frac{1}{|V_i|} \sum_{e \in \Lambda_i} \sum_{j \in \Lambda_i^e} \max\{0, -\bar{n}_{ij}^e \cdot \vec{v}_{ij}^e\} \right\} \leq 1. \quad (3.8)$$

We give now some details for particular forms of the advection equation (3.1). To solve the eikonal equation (2.4) in D , we search stationary solutions of two equations,

$$\partial_\tau u + |\nabla u| = 1, \quad (x, \tau) \in \Omega^{out}(t) \times (0, \infty), \quad u(\gamma, \tau) = 0, \quad \gamma \in \Gamma(t), \quad (3.9)$$

$$\partial_\tau u - |\nabla u| = -1, \quad (x, \tau) \in \Omega(t) \times (0, \infty), \quad u(\gamma, \tau) = 0, \quad \gamma \in \Gamma(t), \quad (3.10)$$

with the initial condition $u(x, 0) = \phi(x, t^n)$. The finite volume discretization can be written conveniently in the form

$$u_i^{m+1} = u_i^m + \Delta\tau^m \left(1 - (|\nabla u|)_i^{m+1/2} \right), \quad (3.11)$$

where

$$(|\nabla u|)_i^{m+1/2} = \sum_{e \in \Lambda_i} \sum_{j \in \Lambda_i^e} |\Gamma_{ij}^e| \bar{n}_{ij}^e \cdot \frac{\nabla^e \tilde{u}^m}{|\nabla^e \tilde{u}^m|} (u_{ij}^{e, m+1/2} - u_i^{m+1/2}). \quad (3.12)$$

One has to compute (3.9) for $m = 0, 1, \dots, M-1$ where M is chosen sufficiently large. When done, one can set $\tilde{\Phi}^n(x) = \tilde{u}^M(x)$ where $\tilde{\Phi}^n(x) \approx \Phi(x, t^n)$. Analogously, one proceed with the complementary equation (3.10). When done, the function $\tilde{\Phi}^n(x)$ is defined for $x \in D$ analogously to (3.2).

Afterward, the extrapolation equation (2.11) in $\Omega^{out}(t)$ can be solved by searching for a stationary solution of the equation

$$\partial_\tau u + \frac{\nabla \Phi(x, t^n)}{|\nabla \Phi(x, t^n)|} \cdot \nabla u = 0, \quad u(\gamma, \tau) = \mathcal{S}(\gamma, t), \quad \gamma \in \Gamma(t) \quad (3.13)$$

for $(x, \tau) \in \Omega^{out}(t) \times (0, \infty)$ with some initial condition for $u(x, 0)$, e.g. $u(x, 0) \equiv 0$. The finite volume discretization for (3.13) takes then the form

$$u_i^{m+1} = u_i^m + \Delta\tau^m \sum_{e \in \Lambda_i} \sum_{j \in \Lambda_i^e} |\Gamma_{ij}^e| \bar{n}_{ij}^e \cdot \frac{\nabla^e \tilde{\Phi}^n}{|\nabla^e \tilde{\Phi}^n|} (u_{ij}^{e, m+1/2} - u_i^{m+1/2}). \quad (3.14)$$

Again, one has to use (3.14) for $m = 0, 1, 2, \dots, M-1$ with sufficiently large M . When ready, one can set $\tilde{s}^n(x) = \tilde{u}^M(x)$ where $\tilde{s}^n(x) \approx s(x, t^n)$.

Similarly to (3.10), one has to treat the complementary problem of (3.13) in $\Omega(t^n)$. At the end the function $\tilde{s}^n(x)$ is defined for $x \in D$.

When $\tilde{\Phi}^n(x)$ and $\tilde{s}^n(x)$ are available, one can apply (3.4) to solve the advection equation (2.9) in the following form,

$$\phi_i^{n+1} = \phi_i^n - \frac{\Delta\tau^n}{|V_i|} \sum_{e \in \Lambda_i} \sum_{j \in \Lambda_i^e} |\Gamma_{ij}^e| \tilde{s}^n(x_{ij}^e) \bar{n}_{ij}^e \cdot \frac{\nabla^e \tilde{\Phi}^n}{|\nabla^e \tilde{\Phi}^n|} (\phi_{ij}^{e, n+1/2} - \phi_i^{n+1/2}). \quad (3.15)$$

In the next section we discuss the immersed interface method when solving (3.1) on implicitly defined domain $\Omega^{out}(t)$, respectively on $\Omega(t)$.

4. Immersed interface formulation. Next we extend the described finite volume discretization for the case of $\Omega(t^n) \subset D$ using the immersed interface method [11, 12, 9, 3, 4].

Let the approximation $\tilde{\phi}^n(x) \approx \phi(x, t^n)$, given analogously to (3.2), define implicitly the approximative boundary $\Gamma^n \approx \Gamma(t^n)$ and the domain $\Omega^n \approx \Omega(t^n)$. We describe now the necessary modifications or extensions of (3.4) when solved on Ω^n . Analogous treatment shall be done when solving it on $D \setminus \Omega^n$.

We suppose that $u(\gamma, t)$ is given by Dirichlet boundary conditions, i.e. for a given function u_D one has

$$u(\gamma, \tau) = u_D(\gamma, t^n), \quad \gamma \in \Gamma(t^n), \quad \tau \geq 0. \quad (4.1)$$

Because the boundary $\Gamma(t^n) \subset \partial\Omega(t^n)$ is given only implicitly, the boundary conditions (4.1) will be treated by immersed interface method. We do not describe boundary conditions on the rest of boundary $\partial\Omega(t^n) \cap \partial D$ that can be treated by standard numerical techniques.

Firstly, if $\phi_i^n > 0$ then $x_i \notin \Omega^n$ and the discrete equations (3.4) are not considered. If $\phi_i^n = 0$, the value of u is set directly by $u_i^m = u_D(x_i, t^n)$ for $x_i \in \Gamma^n$.

It remains to specify (3.4) for indices i such that $\phi_i^n < 0$. In fact, these discrete equations are well defined if $\phi_i^n < 0$ and $\phi_j^n \leq 0$ for $j \in \Lambda_i^e$ and $e \in \Lambda_i$, meaning that x_i is far away from Γ^n . On the other hand, the equations (3.4) can not be computed directly if $\phi_i^n < 0$ and there exists a finite element T^e , $e \in \Lambda_i$ and a vertex x_j , $j \in \Lambda_i^e$ such that $\phi_j^n > 0$. In such a case, the finite element T^e is intersected by Γ^n and the discrete variable u_j^m is not available, therefore one has to add a description how (3.4) shall be computed.

If $\phi_i^n < 0$ and $\phi_j^n > 0$ and $j \in \Lambda_i^e$, there exists a point γ_{ij}^n lying on the edge connecting x_i and x_j such that $\tilde{\phi}^n(\gamma_{ij}^n) = 0$. From linear interpolation of ϕ_i^n and ϕ_j^n on this edge we obtain

$$\gamma_{ij}^n = \frac{\alpha_{ij}^n - 1}{\alpha_{ij}^n} x_i + \frac{1}{\alpha_{ij}^n} x_j, \quad (4.2)$$

$$\alpha_{ij}^n = \frac{\phi_i^n}{\phi_i^n - \phi_j^n}. \quad (4.3)$$

Computing α_{ij}^n according to (4.3) one can extrapolate the missing value u_j^m using the linear interpolation $\tilde{u}^m(x)$ of the known values u_i^m and $u_D(\gamma_{ij}^n)$,

$$u_j^m = \frac{\alpha_{ij}^n - 1}{\alpha_{ij}^n} u_i^m + \frac{1}{\alpha_{ij}^n} u_D(\gamma_{ij}^n). \quad (4.4)$$

In theory, $\alpha_{ij}^n \neq 0$, so (4.2) is well defined. In practice, for a chosen triangulation an appropriate parameter $0 < \epsilon \ll 1$ shall be defined and if $-\epsilon < \phi_i^n < 0$ then one sets directly $\gamma_{ij}^n = x_i$.

The idea of immersed interface method is to use (4.4) for each i -th discrete equation (3.4) when necessary. Note that no geometric reconstruction of Γ^n is required in this method.

Analogous immersed interface method can be used to solve the Laplace equation (1.1) that is published elsewhere [4]. Despite the analogy, there is one important difference when applying the immersed interface method for elliptic equation (1.1) and for hyperbolic problem (3.1). In the latter case, we use the discretization that is explicit in time and that has the CFL type of restriction (3.8) for the choice of time step.

Considering the simplest possible one-dimensional form of (3.1) with a constant velocity v and Dirichlet boundary conditions, one can easily show that the resulting scheme (3.4), when using the immersed interface formulation, requires a time step that must be proportional to the distance between γ_{ij}^n and x_i . As this distance can be arbitrary small in our applications, the related restriction on the time step would make the unmodified immersed interface method extremely inefficient.

Fortunately, one can introduce a simple extension [3] of the immersed interface method to remove such time step restriction for (3.4) when applying it to (3.9) and

(3.13) that we describe next. In fact, similarly motivated extensions are used in [2, 5] to remove the CFL type of restriction for finite volume discretization of some advection equations.

In our extension of immersed interface method we modify the i -th discrete equation (3.11) for all vertices x_i near the interface Γ^m , i.e. when there exists e and j such that $i \in \Lambda^e$, $j \in \Lambda_i^e$ and $\phi_i^m < 0$ and $\phi_j^m > 0$. To do so, we consider (3.11) with $r \equiv 0$, when it takes the form

$$u_i^{m+1} = u_i^m - \Delta\tau^m (|\nabla u|_i)^{m+1/2}. \quad (4.5)$$

The corresponding differential equation for $x \in \Omega^{out}(t^n)$ and $\tau \geq \tau^m$ with Dirichlet boundary conditions $u(\gamma, \tau^m) = 0$ at $\gamma \in \Gamma(t^n)$ and initial condition $u(x, \tau^m) > 0$ for $x \in \Omega^{out}(t^n)$, see (2.2), can be written as

$$\partial_t u + |\nabla u| = 0. \quad (4.6)$$

Without going into too much details, see [3, 8, 14] for more information, the scheme (4.5) can be used for the approximation of “first arrival time”, say the (pseudo-) time τ_i at which the interface, when moving from the initial position Γ^n at $\tau = \tau^m$ with the speed 1 in its normal direction, will cross the point x_i and so $u(x_i, \tau^m + \tau_i) = 0$.

Note that we derive the following considerations only for the vertices x_i near the interface Γ^n . Clearly, $u_i^m > 0$. Consequently, if $(|\nabla u|_i)^{m+1/2} > 0$ in (4.5) then for some time step, say $\Delta\tau_i^m$, one must obtain $u_i^{m+1} = 0$ and

$$\Delta\tau_i^m = \frac{1}{(|\nabla u|_i)^{m+1/2}} u_i^m. \quad (4.7)$$

For the unmodified scheme (3.11), one has the restriction $\Delta\tau^m \leq \Delta\tau_i^m$, because otherwise $u_i^{m+1} < 0$ in (4.5) that is unphysical.

Now it is simple to modify the scheme (4.5) to include the case $\Delta\tau^m > \Delta\tau_i^m$ by simply splitting $\Delta\tau^m = \Delta\tau_i^m + (\Delta\tau^m - \Delta\tau_i^m)$ and considering that $u(x_i, \tau) = 0$ for $\tau \geq \tau^m + \Delta\tau_i^m$. Therefore, when applying this idea to (3.11), the scheme is modified to the form

$$u_i^{m+1} = u_i^m + \min\{\Delta\tau^m, \Delta\tau_i^m\} \left(1 - (|\nabla u|_i)^{m+1/2}\right). \quad (4.8)$$

Analogous treatment for the vertices x_i near the interface as in (4.8) is also valid for the complementary eikonal equation (3.10) and an equivalent replacement of the time step $\Delta\tau^m$ as in (4.8) must be done also for the discrete scheme (3.15).

5. Numerical experiments. We present an example taken from [10] that is motivated by the computations of moving water table for a dam. The Laplace equation for $p(x, y)$ is solved in a non-dimensional fashion on a quadrangle $D = [0, 6] \times [0, 5]$. The value $p = 0$ is prescribed at the top and the right side of D . At the bottom side of D one has $p(x, 0) = 0$ for $x \in [3, 6]$ and $\partial_y p(x, 0) = -1$ for $x \in (0, 3)$. Finally at the left side of D the values $p(0, y) = 6 - y$ are prescribed, see also Figure 5.1.

For this initial setting, i.e. $t = 0$ and $\Omega(0) \equiv D$, the approximative pressure p from (1.1) is computed with standard finite volume method from the Section 3. The mesh has 33×33 grid nodes and 2048 triangles. The numerical results are presented in Figure 5.1. In this Figure, the corresponding groundwater flow velocity is presented that is described in an analytical way by $\vec{V} = -\nabla p + (0, -1)$.

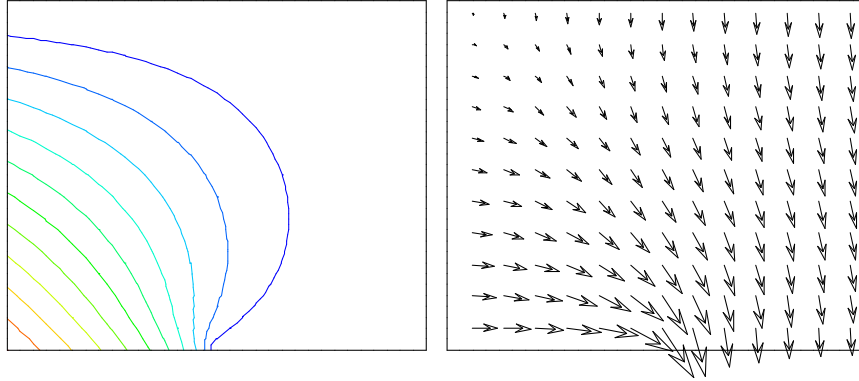


FIG. 5.1. The left picture shows the computed pressure at $t = 0$ (ten contour lines from 0.5 up to 5.5) and the right picture the corresponding vector field $-\nabla p + (0, -1)$.

Having \vec{V} , we continue the computations by considering the top side of D to be the moving boundary with the speed \vec{V} as described in this paper. The evolution of moving boundary up to $T = 12$ where (practically) a stationary state is reached, can be found in Figure 5.2.

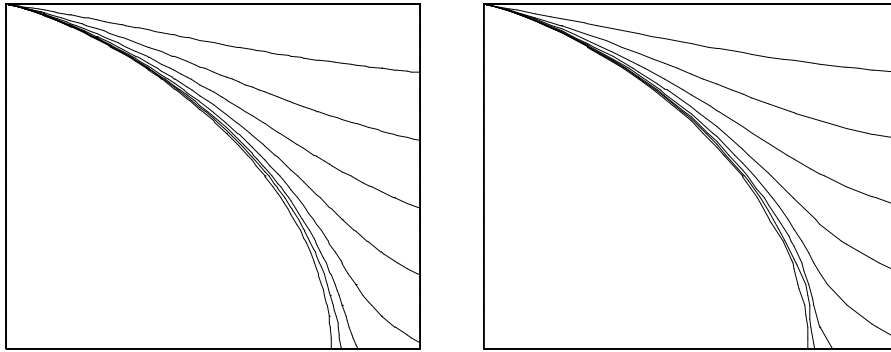


FIG. 5.2. The evolution of moving boundary for $t = 1, 2, \dots, 7$ and for $t = 12$. The left picture shows the evolution for the finer grid, the right one for the coarser grid to illustrate a good visual grid convergence of the numerical solution.

In Figure 5.2, the results are presented not only for the grid of 33×33 nodes, but also for a coarser grid (17×17 nodes) to illustrate a numerical grid convergence for the problem. The time step Δt for the coarser grid is 0.05, for the finer one has $\Delta t = 0.025$. The computed pressure and the corresponding vector field for the stationary situation can be seen in Figure 5.3.

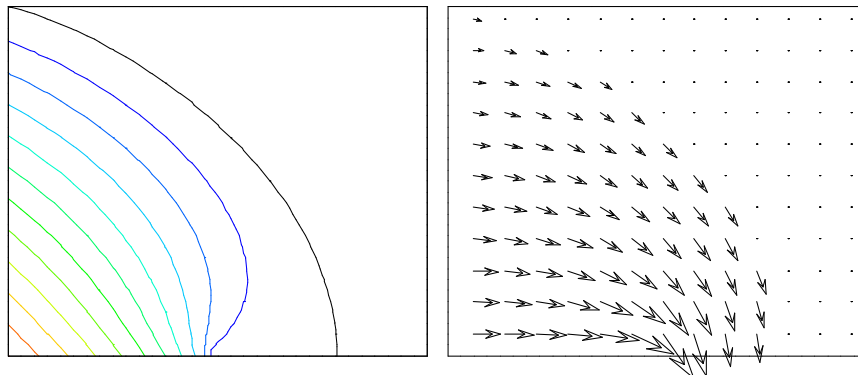


FIG. 5.3. The computed pressure at $t = 0.95$ together with the position of moving boundary (the left picture) and the corresponding vector field $\vec{V} = -\nabla p + (0, -1)$ (the right picture).

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] P. Frolkovič. Flux-based method of characteristics for contaminant transport in flowing groundwater. *Comput. Visual. Sci.*, 5(2):73–83, 2002.
- [3] P. Frolkovič. Flux-based level set method for extrapolation along characteristics using immersed interface formulation. In P. Struk, editor, *Magia*, pages 15–26, 2010.
- [4] P. Frolkovič. Application of level set method for groundwater flow with moving boundary. *Adv. Water Resour.*, accepted, 2012.
- [5] P. Frolkovič and K. Mikula. Flux-based level set method: A finite volume method for evolving interfaces. *Appl. Numer. Math.*, 57(4):436–454, 2007.
- [6] P. Frolkovič and K. Mikula. High-resolution flux-based level set method. *SIAM J. Sci. Comp.*, 29(2):579–597, 2007.
- [7] P. Frolkovič and H. De Schepper. Numerical modelling of convection dominated transport coupled with density driven flow in porous media. *Adv. Water Resour.*, 24(1):63–72, 2001.
- [8] P. Frolkovič and C. Wehner. Flux-based level set method on rectangular grids and computations of first arrival time functions. *Comput. Visual. Sci.*, 12(5):297–306, 2009.
- [9] P. Frolkovič and P. Zacharovská. Numerical Modeling of Dynamic Groundwater Table Using Level Set Formulation. In *XVIII International Conference on Water Resources, CIMNE, Barcelona*, 2010.
- [10] S.J. Lacy and J.H. Prevost. Flow through porous media: A procedure for locating the free surface. *Int. J. Num. Anal. Methods Geomech.*, 11:585–601, 1987.
- [11] R.J. Leveque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31(4):1019–1044, 1994.
- [12] Z. Li and K. Ito. *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*. SIAM, 2006.
- [13] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.
- [14] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.