

A simple thresholding-like numerical scheme for a Hele-Shaw-type problem

Md. Joni ALAM^{*1,2} and Norbert POŽÁR³

¹ Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa 920-1192, Japan

² Department of Mathematics, Comilla University, Cumilla-3506, Bangladesh

³ Faculty of Mathematics and Physics, Institute of Science and Engineering, Kanazawa University,
Kanazawa 920-1192, Japan

(Received May 2, 2025 and accepted in revised form July 3, 2025)

Abstract We introduce a new simple numerical scheme for a Hele-Shaw-type problem with a general non-homogeneous free boundary velocity coefficient. The scheme is formally derived as the large conductivity limit of the scheme introduced by [Berger-Brezis-Rogers, 1979] for the Stefan problem. Using numerical experiments, we compare the performance of the scheme to that of a state-of-the-art first-order level set method. We demonstrate a first order accuracy of the new scheme, and that the error in the free boundary position is comparable to that of the level set method, particularly in solutions with a complicated free boundary.

Keywords. Hele-Shaw problem, numerical scheme, level set method.

1 Introduction

We consider the following Hele-Shaw-type problem and study the performance of a new simple discrete scheme inspired by the scheme proposed by Berger, Brezis and Rogers [2] for Stefan-type problems. Let K represent a nonempty compact subset of \mathbb{R}^n with a smooth boundary ∂K , and let $\Omega_0 \subset \mathbb{R}^n$ be a bounded open set with a C^2 -smooth boundary, where $K \subset \Omega_0$. We are looking for the evolving family of sets $t \mapsto \Omega_t \subset \mathbb{R}^n$, with initial condition Ω_0 , whose outer normal velocity is given as

$$V(x, t) = g(x, t)|Dv|(x, t), \quad x \in \partial\Omega_t, t > 0, \quad (1.1a)$$

with given positive Lipschitz continuous $g = g(x, t)$, where $v(\cdot, t) : \mathbb{R}^n \rightarrow [0, \infty)$, $t \geq 0$, is the solution of the Laplace equation

$$\begin{cases} \Delta v(\cdot, t) = 0 & \text{in } \Omega_t \setminus K, \\ v(\cdot, t) = 1 & \text{on } \partial K, \\ v(\cdot, t) = 0 & \text{on } \partial\Omega_t. \end{cases} \quad (1.1b)$$

*Corresponding author. Email: jonialam@cou.ac.bd

The spatial gradient Dv on $\partial\Omega_t$ is understood as the limit from Ω_t . We note that since $\partial\Omega_t$ is the level set of v , we have $|Dv(x,t)| = \left| \frac{\partial v}{\partial \nu}(x,t) \right|$, where ν is the unit outer normal to Ω_t . See Figure 1 for an illustration. The well-posedness of viscosity solutions of (1.1), if K and Ω_0 are star-shaped, was established in [20].

As a model of the flow in the Hele-Shaw cell in two dimensions [6], Ω_t represents the region filled with fluid at time t , which is injected through K in such a way that the pressure v is kept constant 1 on ∂K . The surface of the fluid $\partial\Omega_t$ is assumed to have zero pressure, as any surface tension effects are neglected. The coefficient $1/g$ might be interpreted as the thickness of the gap in the Hele-Shaw cell that needs to be filled. More general boundary conditions can be assumed on ∂K , but we use the constant Dirichlet boundary condition for simplicity.

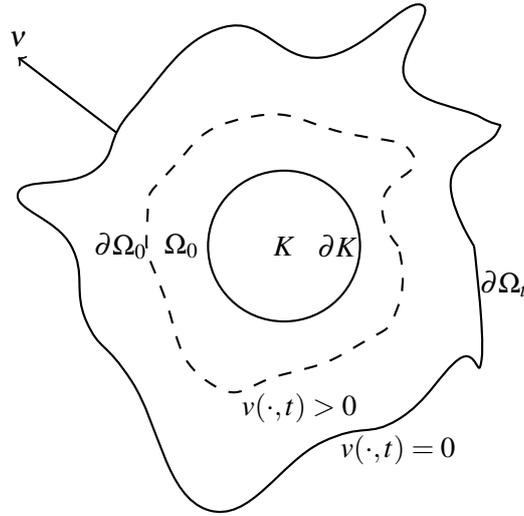


Figure 1: Hele-Shaw problem in a plane.

In this paper we introduce a simple discrete scheme for the above Hele-Shaw problem based on the scheme proposed by Berger, Brezis and Rogers [2] for Stefan-type problems (see also [15, 18]): Choosing a time step $\tau > 0$ and a regularization parameter $\gamma > 0$, we find the sequence of $w^k : \mathbb{R}^n \rightarrow \mathbb{R}$, $k = 0, 1, \dots$, which approximates $\Omega_{t_k} \approx \{w^k \geq 0\}$ at $t_k := k\tau$, as

$$w^k = \begin{cases} -\frac{1}{g(\cdot, 0)} \mathbf{1}_{\mathbb{R}^n \setminus \Omega_0}, & k = 0, \\ \min(w^{k-1}, 0) + \left(\gamma u^k - \frac{1}{g(\cdot, t_k)} + \frac{1}{g(\cdot, t_{k-1})} \right) \mathbf{1}_{\{w^{k-1} < 0\}}, & k \geq 1, \end{cases} \quad (1.2a)$$

where u^k solves

$$\begin{cases} \gamma \mathbf{1}_{\{w^{k-1} < 0\}} u^k - \tau \Delta u^k = \max(w^{k-1}, 0) & \text{in } \mathbb{R}^n \setminus K, \\ u^k = 1 & \text{on } \partial K, \\ u^k \rightarrow 0 & \text{as } |x| \rightarrow \infty. \end{cases} \quad (1.2b)$$

Here $\mathbf{1}_A(x) := 1$ for $x \in A$, and 0 otherwise. For the formal derivation, see Section 2. Our goal is to demonstrate its performance and numerically estimate the error, in particular in comparison to a basic first-order level set method scheme.

We refer to scheme (1.2) as the BBR scheme. Its main advantage is that it is particularly simple to implement, and automatically handles topological changes of the interface $\partial\Omega_t$. Unlike

other traditional numerical approaches to (1.1) like the level set method (see Section 3), the linear Poisson equation (1.2b) is posed on the fixed domain $\mathbb{R}^n \setminus K$ and there is no need to explicitly find $|Dv|$ or V . Furthermore, the original scheme in [2] for the Stefan problem is known to be stable for sufficiently small τ , and this appears to be also the case for (1.2).

The scheme (1.2) also conserves mass. Indeed, assuming that $g = g(x)$ is independent of t , noting that $w^{k-1} = \min(w^{k-1}, 0) + \max(w^{k-1}, 0)$ and integrating (1.2a) over $U \setminus K$ for some smooth open set $U \supset K$, we get

$$\begin{aligned} \int_{U \setminus K} w^k &= \int_{U \setminus K} w^{k-1} + \tau \int_{U \setminus K} \Delta u^k \\ &= \int_{U \setminus K} w^{k-1} - \tau \int_{\partial K} \frac{\partial u^k}{\partial \nu} dS + \tau \int_{\partial U} \frac{\partial u^k}{\partial \nu} dS, \end{aligned}$$

where ν is the outer unit normal to U and K . Since u^k is nonnegative by the maximum principle and decays exponentially away from $\{w^{k-1} \geq 0\}$, if U is sufficiently large we can neglect the integral over ∂U and we get the conservation law

$$\int_{U \setminus K} w^k \approx \int_{U \setminus K} w^{k-1} - \tau \int_{\partial K} \frac{\partial u^k}{\partial \nu} dS. \quad (1.3)$$

This is particularly useful if a Neumann boundary condition for v and hence u^k on ∂K is considered.

However, since w^k in general has a jump at $\partial\{w^{k-1} \geq 0\}$, it is not possible to determine the boundary position to a subgrid precision with an interpolation like in the case of the level set method, so we expect the scheme (1.2) to be less accurate by a constant factor compared to a first-order level set method. This is indeed what we observe, see Section 4.

Our main motivation for considering scheme (1.2) is the numerical study of the homogenization of the Hele-Shaw problem (1.1) when g is periodic both in x and t and cannot be decomposed into a product of functions of x and t . When g is rescaled as $g(x/\varepsilon, t/\varepsilon)$, in the homogenization limit $\varepsilon \rightarrow 0$ the solution converges to a Hele-Shaw-type problem with V depending only on Dv , see [20, 18] for details. However, there is no explicit formula for the homogenized velocity, and a numerical method is needed to approximate it. In this context, since the error of the approximation will be dominated by the averaging error, it is sufficient to consider a first-order accurate scheme like (1.2). Moreover, the mass conservation (1.3) is a desirable feature guaranteeing that mass is not lost even over large time scales when g is independent of t (and at least approximately when g depends on t). This should lead to a more robust estimate of the homogenized velocity.

In the previous work of one of the authors [18], the BBR scheme for the Stefan problem with large conductivity, approximating the Hele-Shaw problem, was used instead of (1.2) to estimate the homogenized velocity. However, even large finite conductivity introduces a noticeable systematic error to the estimate of the average velocity. In an upcoming paper, we apply the BBR scheme (1.2) to estimate the homogenization velocity.

Let us point out that if the velocity coefficient g in (1.2a) can be written as $g(x, t) = X(x)T(t)$, then by a change of variables in t , the problem (1.1) can be reduced to the problem with $g(x, t) = g(x) = X(x)$. Furthermore, using the standard transform [1, 3] we define

$$u(x, t) = \int_0^t v(x, s) ds.$$

Surprisingly $u(\cdot, t)$ is a solution of an elliptic variational inequality that can be solved at each $t > 0$ completely independently from the previous states $t' < t$, see [19] for details. And since the free

boundary evolution is monotone, $\{v(\cdot, t) > 0\} = \{u(\cdot, t) > 0\}$. Many publications on the Hele-Shaw problem rely on such transformation and therefore they can be applied only in the special case $g(x, t) = X(x)T(t)$. However, for general $g = g(x, t)$ this transformation does not lead to a time-independent problem and hence in particular a time-stepping scheme like (1.2) is necessary.

A large number of numerical methods for various versions of the Hele-Shaw problem have been proposed, including level set methods, boundary integral methods, parametric interface methods, etc. We refer the reader to a review paper [14] for an extensive list of references. Here we give a few selected ones. For level set methods, introduced for the mean curvature flow by Osher and Sethian [16], see for example [7, 9]. The main advantage of the level set method is its ability to implicitly handle topological changes of the free boundary in any dimension. Higher order methods are available, and they are necessary if the free boundary velocity includes the curvature of the interface, representing the surface tension; see the review paper [4]. Unfortunately, the level set method does not automatically satisfy the mass conservation property. In particular, a special care needs to be taken during reinitialization of the level set function; see Section 3.5. For a recent example of a parametric boundary method that solves the Laplace equation using a method of fundamental solutions, see [22]. For a boundary integral method see [8]. However, the latter two approaches seem to be limited to two dimensions and handling topological changes might be challenging. Finally, for an application of the JKO scheme to the Hele-Shaw problem, see [11]. Although many different numerical methods have been used, to the best of our knowledge, the current paper is the first time the BBR scheme (1.2) has been proposed for the Hele-Shaw problem.

We conclude the introduction with the outline of the paper. In Section 2, we formally derive the BBR scheme (1.2) as the large conductivity limit of the scheme for the Stefan problem. A simple first-order level set method that we compare the scheme against is briefly described in Section 3, based on [4]. The settings and results of numerical experiments are presented in Section 4. We observe that the scheme (1.2) appears to be first order accurate, with a similar error as the level set method in solutions with complicated interface and topological changes.

Acknowledgments

The authors thank the anonymous referees for their careful reading of the manuscript and their helpful comments. Md. Joni Alam was supported by the Japanese Government Monbukagakusho (MEXT) Scholarship in the years 2022-2025. N. Pozar was partially supported by JSPS KAKENHI Kiban C Grant No. 23K03212.

2 BBR-Like Numerical Scheme for the Hele-Shaw Problem

In this section, we derive the proposed scheme (1.2) as a formal large conductivity limit of the Berger-Brezis-Rogers scheme [2] for the Stefan problem. In this limit the solutions of the Stefan problem converge to that of the Hele-Shaw problem.

The Stefan problem with free boundary velocity matching that of (1.1) and conductivity $1/\lambda$ is

$$\begin{cases} \lambda u_t - \Delta u = 0 & \text{in } \{u > 0\}, \\ V = g(x, t)|\nabla u| & \text{on } \partial\{u > 0\}, \\ u(\cdot, t) = 1 & \text{on } \partial K, \\ u(\cdot, 0) = u_0. \end{cases}$$

It is well-known that in the limit $\lambda \rightarrow 0$ the solutions of the Stefan problem converge to the solution of the Hele-Shaw problem. The Stefan problem can be rewritten in an enthalpy formulation,

$$\begin{cases} \lambda z_t - \Delta \beta(z) = - \left(\frac{\partial}{\partial t} \frac{1}{g} \right) \mathbf{1}_{\{z < 0\}} & \text{in } \mathbb{R}^n \setminus K, \\ \beta(z) = 1 & \text{on } \partial K, \\ z(\cdot, 0) = u_0 \mathbf{1}_{\{u_0 > 0\}} - \frac{1}{\lambda g(\cdot, 0)} \mathbf{1}_{\{u_0 \leq 0\}}. \end{cases} \quad (2.1)$$

Here, $\beta(z) := z_+ := \max(z, 0)$ is the positive part of z . See [18] for more details.

Given a timestep $\tau > 0$, the Berger-Brezis-Rogers [2] time discretization of (2.1) reads

$$z^k = z^{k-1} + \mu^{k-1} (u^k - \beta(z^{k-1})) - \left(\frac{1}{\lambda g(\cdot, t_k)} - \frac{1}{\lambda g(\cdot, t_{k-1})} \right) \mathbf{1}_{\{z^{k-1} < 0\}}, \quad (2.2a)$$

for $k = 1, 2, \dots$, with z^0 given by the initial condition in (2.1), where u^k solves

$$\begin{cases} \lambda \mu^{k-1} u^k - \tau \Delta u^k = \lambda \mu^{k-1} \beta(z^{k-1}) & \text{in } \mathbb{R}^n \setminus K, \\ u^k = 1 & \text{on } \partial K, \\ |u^k| \rightarrow 0 & \text{as } |x| \rightarrow \infty, \end{cases} \quad (2.2b)$$

and

$$\mu^{k-1}(x) := \frac{1}{\delta + \beta'(z^{k-1}(x))}. \quad (2.2c)$$

Here β' is the derivative of β that is defined as 1 at 0. The small regularization parameter $\delta > 0$ determines the width of the transition layer of u^k near $\partial\{z^{k-1} \geq 0\}$, where u^k decays exponentially to 0. Based on a simple one-dimensional calculation, see [18], the half-width is proportional to $\sqrt{\frac{\delta \tau}{\lambda}}$.

We fix $\tau > 0$ and consider the formal limit $\lambda \rightarrow 0$ for z^k and u^k in the scheme (2.2) of the Stefan problem. To keep the boundary layer thickness constant, we set $\delta = \lambda/\gamma$ for a fixed parameter $\gamma > 0$ so that $\delta \rightarrow 0$ as $\lambda \rightarrow 0$. Since z^k scales as λ^{-1} , we rewrite (2.2a) in terms of $z^k = \lambda^{-1} w^k$, multiply by λ and use the positive homogeneity of β , to get

$$w^k = w^{k-1} + \lambda \mu^{k-1} u^k - \mu^{k-1} \beta(w^{k-1}) - \left(\frac{1}{g(\cdot, t_k)} - \frac{1}{g(\cdot, t_{k-1})} \right) \mathbf{1}_{\{w^{k-1} < 0\}}.$$

Similarly, the right-hand side of (2.2b) becomes $\mu^{k-1} \beta(w^{k-1})$, and

$$\lambda \mu^{k-1}(x) = \frac{\lambda}{\delta + \beta'(w^{k-1}(x))}.$$

In the limit $\lambda \rightarrow 0$, at least formally assuming that w^{k-1} converges to a limit that we still denote by w^{k-1} , we obtain

$$\lambda \mu^{k-1}(x) \rightarrow \begin{cases} \gamma, & w^{k-1}(x) < 0 \\ 0, & w^{k-1}(x) \geq 0 \end{cases} = \gamma \mathbf{1}_{\{w^{k-1} < 0\}}(x).$$

Similarly, since $\delta = \lambda/\gamma \rightarrow 0$, we see that

$$\mu^{k-1} \beta(w^{k-1}) = \frac{\beta(w^{k-1})}{\delta + \beta'(w^{k-1})} = \frac{\beta(w^{k-1})}{\delta + 1} \rightarrow \beta(w^{k-1}) \quad \text{as } \lambda \rightarrow 0.$$

Noting that $w^{k-1} - \beta(w^{k-1}) = \min(w^{k-1}, 0)$, we formally recover the scheme (1.2) in the limit $\lambda \rightarrow 0$.

The proof of convergence of the time-discrete scheme (1.2) to the solution of the Hele-Shaw problem (1.1), and hence a justification of the above formal limit, is a subject of an ongoing research. In this paper we give a numerical verification of the convergence. To the best of our knowledge, the level set method for free boundary problems like the Hele-Shaw problem also does not have a known convergence proof, and only numerical verification similar to the current paper is in general available, see [4]. However, due to the simplicity of the proposed BBR scheme, and similarity to the scheme in [2], a convergence proof is feasible. In particular, the solution is monotone so comparison-type arguments should be available. One difficulty is that the generalized solution theory for the Hele-Shaw-type problem (1.1), due to the time dependence of $g = g(x, t)$, is rather restricted and relies on the theory of viscosity solutions.

2.1 Spatial discretization of the BBR scheme

At each time step, the PDE (1.2b) needs to be solved on a *fixed* unbounded domain $\mathbb{R}^n \setminus K$. However, the solution u^k decays exponentially away from $\{w^{k-1} \geq 0\}$. Indeed, in the set $\{w^{k-1} < 0\}$ the equation (1.2b) reduces to $\gamma u - \tau \Delta u = 0$. If u depends only on, say x_1 , we have $u = C \exp(-\sqrt{\gamma/\tau} x_1)$. By comparison with such one-dimensional barriers outside of a sufficiently large ball containing $\{w^{k-1} \geq 0\}$, we see that $u^k \approx 0$ sufficiently far from $\{w^{k-1} \geq 0\}$. The required distance is proportional to the half-width $\sqrt{\tau/\gamma}$. This allows us to solve the problem on a bounded domain with zero Dirichlet boundary condition on the outer boundary.

For simplicity, we discretize the BBR scheme (1.2) on a square domain $Q := (0, 1)^n$ with uniform grid $G := h\mathbb{Z}^n \cap \bar{Q}$ with $h = 1/N$ for given resolution $N \in \mathbb{N}$. The Dirichlet boundary condition $u^k = 0$ is imposed on ∂Q .

The update of w^k in (1.2a) can be trivially performed at each node once the approximate solution u^k is known.

For the spatial discretization of equation (1.2b) we use the standard second-order central finite difference scheme. To handle the boundary of K , we use the extrapolation method discussed in Section 3.1, which yields a linear system with a symmetric matrix that can be solved efficiently by the CG method with a multigrid preconditioner. See Section 3.1 for details.

2.2 Choice of the regularization parameter γ .

In the following part, we give a motivation for a reasonable choice of γ , based on a 1D consideration. The behavior of the solution of the spatial discretization in $\{w^{k-1} < 0\}$ can be understood similarly to the above discussion of the original problem. In one dimension, the finite difference discretization of (1.2b) reads in $\{w^{k-1} < 0\}$ as

$$\gamma v_j - \frac{\tau}{h^2}(v_{j-1} - 2v_j + v_{j+1}) = 0.$$

Assuming $v_j \rightarrow 0$ as $j \rightarrow \infty$, the solution is of the form $v_j = C\theta^j$ with

$$\theta = 1 - c\sqrt{1 + \frac{c^2}{4}} + \frac{c^2}{2} \in (0, 1),$$

where $c = h\sqrt{\gamma/\tau}$.

In an upcoming paper, we perform an analysis of the behavior of the discretized problem in one dimension, deriving sharp bounds on the error in the position of the free boundary. Let us give a

brief summary of the 1D error estimate. Considering a simple 1D setting where the free boundary is a single point, and assuming that $w^k \equiv -z < 0$ in $\{w^k < 0\}$ and that $\frac{v_j^k - v_{j-1}^k}{h} = q$ in $\{w^k > 0\}$, we set $\eta = \frac{q\tau}{zh}$. Note that $V = \frac{q}{z}$ in this case. Then the error e_k of the position of the free boundary can be estimated as

$$\frac{\theta^{1-\lceil\eta\rceil}}{1-\theta} - \lceil\eta\rceil \leq \frac{e_k}{h} \leq \frac{\theta^{-\lfloor\eta\rfloor}}{1-\theta} - \lfloor\eta\rfloor, \quad \text{for } k \text{ large.} \quad (2.3)$$

Here $\lfloor\eta\rfloor$ and $\lceil\eta\rceil$ are respectively the floor and ceiling of η (the largest integer below resp. the smallest integer above η). See Figure 2 for a graph of the left- and right-hand sides.

Intuitively, $\eta = \frac{q\tau}{zh} = \frac{V\tau}{h}$ expresses how many nodes the free boundary of the exact solution moves per time step. In the BBR scheme, the boundary is advanced by updating w^k using γu^k in (1.2a), and therefore the decay of γu^k , and after spatial discretization of $\gamma w_j^k = C\theta^j$, determines how far from the free boundary this can occur. Therefore if $\eta < 1$, even small θ allows the update in the node neighboring the free boundary, but if $\eta \gg 1$ we need to take θ near 1 to update w^k sufficiently far away from the free boundary. Therefore the left-hand side of the error estimate (2.3) blows up near $\theta = 0$ if $\eta > 1$. Unfortunately, θ near 1 also causes a large error since a lot of energy is spread over a thick boundary layer, instead of advancing the interface.

The scheme has the smallest theoretical error if $\eta = \frac{qh}{z\tau} < 1$, which acts as a ‘‘stability’’ restriction. We therefore choose $\frac{\tau}{h}$ for given g so that $q \max(g) \frac{\tau}{h} < 1$ for typical values $q = |Du|$ at the free boundary. See Figure 2 for an illustration of the error behavior if $\frac{\tau}{h}$ is chosen too large. Unfortunately, it is not possible to control $|Du|$ near topological changes, as the gradient blows up when a bubble closes. Inspecting the error estimate (2.3) and Figure 2, it seems desirable to choose $\theta = 0.5$ since then the error lower bound does not worsen catastrophically when $\eta < 1$ is violated. Then we have

$$\gamma = \frac{\tau}{c^2 h^2} \quad \text{with} \quad c = \frac{\sqrt{\theta}}{1-\theta} = \sqrt{2}. \quad (2.4)$$

We provide an illustration of the error sensitivity on the choice of θ in two dimensions in Figure 2. We use the same parameters as in Section 4 with $g = g_1 \equiv 1$.

With $\theta = 0.5$, the value of the discrete solution away from $\{w^{k-1} \geq 0\}$ is expected to decay by a factor $\theta^{10} \approx 10^{-3}$ every 10 nodes. This justifies a zero Dirichlet boundary condition on the outer computational boundary ∂Q as long as the set $\{w^{k-1} \geq 0\}$ stays at a distance at least, for example, $20h$ from ∂Q . We enforce this in the numerical experiments in Section 4.

Let us point out that we choose a fixed θ . Therefore the discrete solution v_j does not converge to the exact solution of (1.2b) as $h, \tau \rightarrow 0$. Interestingly, the 1D error estimate (2.3) does not require the numerical solution v_j to approximate u^k as $h \rightarrow 0$ for the error to be $O(h)$. This appears to be also the case in two dimensions. Indeed, the accuracy of the approximation $\{w^{k-1} \geq 0\}$ of Ω_{t_k} does not seem to be impacted, as we demonstrate in the numerical experiments.

3 The Level Set Method

The level set method, introduced by Osher and Sethian [16], implicitly characterizes the family $\{\Omega_t\}_{t \geq 0}$ as the sublevel sets $\{x \in \mathbb{R}^n : \phi(\cdot, t) < 0\}$ of an auxiliary function $\phi = \phi(x, t)$, referred to as the level set function. The outer normal velocity of $\partial\Omega_t$ can be expressed using ϕ as

$$V(x, t) = -\frac{\phi_t}{|D\phi|}(x, t), \quad x \in \partial\Omega_t.$$

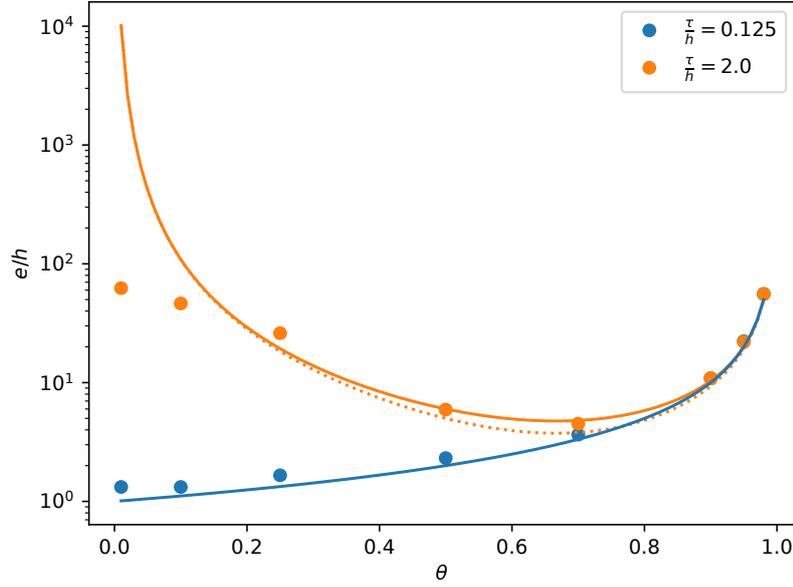


Figure 2: An illustration of the sensitivity of the free boundary error in $n = 2$ with Hele-Shaw coefficient $g = g_1 \equiv 1$ and $N = 256$, for various values of θ and two values $\frac{\tau}{h} = \frac{1}{8}$ and $\frac{\tau}{h} = 2$. The errors of a numerical solution (\bullet) are estimated in the setting explained in Section 4. The error upper bound (right-hand side in (2.3); solid line) with $\eta < 1$ and $\eta \in (2, 3)$ is also plotted. The lower bound (left-hand side in (2.3); dotted line) is plotted only with $\eta \in (2, 3)$.

The normal velocity V is then given by geometric evolution law like (1.1a). The key idea is to extend V given by (1.1a) from $\partial\Omega_t$ to the whole domain and then we can find ϕ as a solution of the equation

$$\phi_t + V|D\phi| = 0, \quad \text{on } \mathbb{R}^n \times (0, \infty). \quad (3.1)$$

The extension of V from $\partial\Omega_t$ can be for example done as the solution of the equation

$$\begin{cases} D\phi(\cdot, t) \cdot DV(\cdot, t) = 0, & \mathbb{R}^n \setminus \partial\Omega_t, \\ V(x, t) = g(x, t)|Dv|(x, t), & x \in \partial\Omega_t, \end{cases} \quad (3.2)$$

so that V is constant along the streamlines of $D\phi$. This choice preserves the slope of ϕ in the direction normal to the level set. For more details, see [17, 4].

We use a simple first-order discretization of the level set method, essentially following [4], on a space domain $Q = (0, 1)^n$ for simplicity. We believe that the only non-standard modification is the simplified initialization of the fast sweeping method in Section 3.5. Given a space resolution $N \in \mathbb{N}$, we define the spatial mesh step $h = 1/N$ and consider the uniform space grid $G := \bar{Q} \cap h\mathbb{Z}^n$ with $(N+1)^n$ nodes. This is the same setting as in Section 2.1 for the BBR scheme (1.2). The $(N-1)^d$ interior grid nodes are defined as $G^0 := Q \cap h\mathbb{Z}^n$. We also choose a time step $\tau > 0$ and define the time steps $t_k := k\tau$. We construct the sequence $\phi^k : G \rightarrow \mathbb{R}$, $k = 0, 1, \dots$, whose sublevel sets approximate Ω_{t_k} . At each time step $k = 0, 1, \dots$ we perform the following standard substeps:

- (1) Approximate the solution of the Laplace equation v in $\Omega_{t_k} \setminus K$.
- (2) Approximate $|Dv|$ near $\partial\Omega_{t_k}$ and find the normal velocity on $\partial\Omega_{t_k}$.

- (3) Extend the normal velocity to the domain Q using (3.2).
- (4) Update $\phi^k \rightarrow \phi^{k+1}$ using (3.1).
- (5) Adjust ϕ^{k+1} if necessary.

Let us briefly explain the above steps in more detail.

3.1 Discretization of the Laplace equation

The Laplace equation (1.1b) for v needs to be solved in a moving domain, given approximately by ϕ^k , so that the gradient at the boundary can be approximated with $O(h)$ accuracy. There are many approaches for this in the literature, but we use the second order discretization proposed in [5]. $O(h^2)$ convergence of this scheme in the maximum norm is established in [24]. This method relies on the standard central difference 5-point stencil for the Laplacian on a uniform grid, with a linear extrapolation through the boundary into “ghost” nodes that are outside of the domain. This can be done along each axis independently and therefore can be easily implemented in any dimension.

Let us briefly explain the scheme that results from the extrapolation procedure and its accuracy. Recall that for given $h = 1/N$, we consider the uniform grid $G := \bar{Q} \cap h\mathbb{Z}^n$. Suppose that we have a discrete level set function $\phi : G \rightarrow \mathbb{R}$. The points in $G_\phi := G \cap \{\phi < 0\}$ are the interior nodes. We index them by indices in an index set I^0 , so that $G_\phi = \{x_i : i \in I^0\}$. For simplicity, we assume that $G_\phi \cap \partial Q = \emptyset$ so that no interior nodes are on the boundary of the computational domain. We call $\mathcal{D} := \{e_k, -e_k : k = 1, \dots, n\}$ the set of $2n$ grid directions. For any direction $\xi \in \mathcal{D}$ and interior node $i \in I^0$, if the neighbor $y := x_i + h\xi$ does not belong to G_ϕ , that is, $\phi(y) \geq 0$, we add a boundary node at position $x_i + th\xi$ where t is the root of the linear interpolation of ϕ between $\phi(x_i)$ and $\phi(y)$. Specifically,

$$(1-t)\phi(x_i) + t\phi(y) = 0.$$

This gives

$$t = \frac{\phi(x_i)}{\phi(x_i) - \phi(y)}. \quad (3.3)$$

In this way, we add a set of boundary nodes that we index by an index set ∂I , with $I^0 \cap \partial I = \emptyset$. See Figure 3 for an illustration. The values at the boundary nodes are given by the boundary condition, that is, as 0 on the free boundary and 1 on ∂K .

For any node indexed by $i \in I^0$ and direction $\xi \in \mathcal{D}$, we denote the index of the closest node from x_i in the direction ξ as $j(i, \xi) \in I := I^0 \cup \partial I$. Given a grid function $v : I \rightarrow \mathbb{R}$, then the discretization of $-\Delta$ at node $i \in I^0$ is given by

$$-\Delta u(x_i) \approx (Lv)_i := \frac{1}{h} \sum_{\xi \in \mathcal{D}} \frac{v_i - v_{j(i, \xi)}}{|x_i - x_{j(i, \xi)}|}. \quad (3.4)$$

Yoon-Min [24] proved that, in dimension $n = 2$, if u is the solution of

$$\begin{cases} -\Delta u = f, & \text{in } \{\phi < 0\}, \\ u = g, & \text{on } \partial\{\phi < 0\}, \end{cases}$$

under the assumption that $\partial\{\phi < 0\}$, f and g are smooth, and v is the solution of the linear system

$$\begin{cases} (Lv)_i = f(x_i), & i \in I^0, \\ v_i = g(x_i), & i \in \partial I, \end{cases}$$

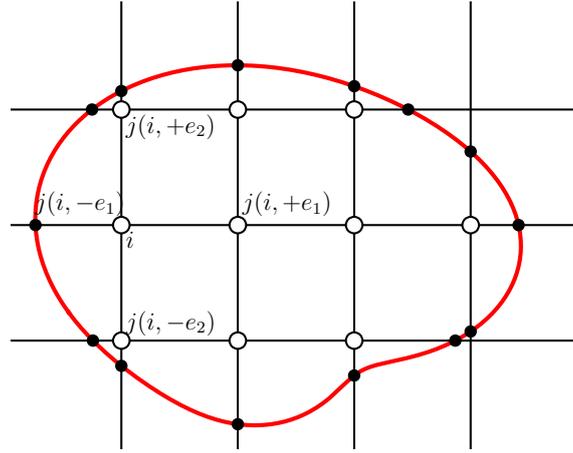


Figure 3: Grid nodes: The location x_i of interior nodes $i \in I^0$ indicated by empty circles \circ , and interface nodes $i \in \partial I$ indicated by filled circles \bullet . For each interior node $x_i, i \in I^0$ and direction $\xi \in \mathcal{D} = \{\pm e_1, \pm e_2\}$, its closest neighbor $x_{j(i, \xi)}$ is given by the index $j(i, \xi) \in I = I^0 \cup \partial I$.

with the boundary node position assumed to be exact (i.e., no interpolation), then

$$\max_{i \in I^0} |u(x_i) - v_i| = O(h^2).$$

Let us give a formal argument on why we can expect $O(h^2)$ accuracy in our case; see also [5, Sec. 3.1] for discussion. Since we find the boundary nodes by a linear interpolation (3.3) along direction $\xi \in \mathcal{D}$, we obtain a $Ch^2/|D\phi \cdot \xi|$ accurate position of the boundary nodes. Furthermore, u can be extended to a smooth function near $\partial\{\phi < 0\}$, and this translates to $Ch^2|Du \cdot \xi|/|D\phi \cdot \xi|$ error for the boundary data of v . Since Du and $D\phi$ are parallel at the free boundary, we deduce that the boundary data has $Ch^2|Du|/|D\phi|$ error. By the maximum principle for the discrete scheme (note that the scheme is monotone), the resulting error is still $O(h^2)$ in the maximum norm, as long as $D\phi$ is nondegenerate at the free boundary.

The resulting linear system is symmetric and therefore can be solved using a CG method. We use a simple multigrid preconditioner to speed up the convergence. The linear systems on the lower resolution grids are constructed by the same approach based on a sub-sampled level set function. The use of the multigrid method as a preconditioner of the CG method rather than a direct solver is inspired by [12]. The convergence of the multigrid solver often slows down drastically on domains with complicated topology, and the CG method recovers the original convergence.

Let us explain the used multigrid solver. Given a level set function $\phi : G \rightarrow \mathbb{R}$, we apply it to the resulting linear system

$$a_i v_i - (Lv)_i = f_i, \quad i \in I^0, \quad (3.5)$$

where Lv is defined in (3.4) and $a_i, i \in I^0$ is a given vector. For the level set method we set $a_i \equiv 0$, while for the BBR scheme we use $a_i = \frac{\gamma}{\tau} \mathbf{1}_{\{w_i^{k-1} < 0\}}$.

The level set function ϕ for the level set method is taken to be

$$\phi := \max(\phi^k, \phi_K),$$

where ϕ^k is the level set function of the free boundary at time step k , and ϕ_K is the signed distance function of K such that $\{\phi_K < 0\} = \mathbb{R}^2 \setminus K$.

For the BBR method, we formally take ϕ to be

$$\phi := \max(\phi_Q, \phi_K),$$

where ϕ_Q is the signed distance function of Q with $\{\phi_Q < 0\} = Q$. Therefore the nodes on ∂Q are boundary nodes, with Dirichlet boundary condition 0.

The multigrid solver is implemented on the uniform grid G on \bar{Q} . We have grids $G_h := \bar{Q} \cap h\mathbb{Z}^2$ and $G_{2h} := \bar{Q} \cap 2h\mathbb{Z}^2$, etc., up to the coarsest grid $G_{1/2}$.

We first define the standard transfer operators: restriction and prolongation. Let $v^h : G_h \rightarrow \mathbb{R}$ and $v^{2h} : G_{2h} \rightarrow \mathbb{R}$. The restriction operator $I_h^{2h} : (G_{2h} \rightarrow \mathbb{R}) \rightarrow (G_h \rightarrow \mathbb{R})$ is defined at $x = 2hp \in G_{2h}$, $p \in \mathbb{Z}^2$, as

$$I_h^{2h} v^{2h}(x) := \frac{1}{4} v^h(x) + \frac{1}{8} \sum_{\xi \in \mathcal{D}} v^h(x + h\xi) + \frac{1}{16} \sum_{\xi = (\pm 1, \pm 1)} v^h(x + h\xi).$$

Note that the last term sums over the 4 diagonal directions.

The prolongation $I_{2h}^h : (G_h \rightarrow \mathbb{R}) \rightarrow (G_{2h} \rightarrow \mathbb{R})$ is defined as $I_{2h}^h := 4(I_h^{2h})^*$, that is, it is a scaled adjoint of the restriction. Setting $x = 2ph$, $p \in \mathbb{Z}^2$, it reads as

$$\begin{aligned} I_{2h}^h v^{2h}(x) &:= v^{2h}(x), \\ I_{2h}^h v^{2h}(x + he_1) &:= \frac{1}{2} (v^{2h}(x) + v^{2h}(x + 2he_1)), \\ I_{2h}^h v^{2h}(x + he_2) &:= \frac{1}{2} (v^{2h}(x) + v^{2h}(x + 2he_2)), \\ I_{2h}^h v^{2h}(x + h(e_1 + e_2)) &:= \frac{1}{4} \left(v^{2h}(x) + v^{2h}(x + 2he_1) \right. \\ &\quad \left. + v^{2h}(x + 2he_2) + v^{2h}(x + 2h(e_1 + e_2)) \right), \end{aligned}$$

whenever the node on the left-hand side is in G_h . It is the bilinear interpolation of the values on G_{2h} . It is crucial that the prolongation and restriction operators are (scaled) adjoints of each other for the multigrid V-cycle to be an admissible preconditioner for the CG method, see [23, 12].

With the transfer operators defined, we can introduce the matrices A^h, A^{2h}, \dots , for the individual grids. We are given $\phi^h(x) = \phi(x)$, $x \in G = G^h$ and

$$a^h(x) = \begin{cases} a_i, & x = x_i \text{ for some } i \in I^0, \\ 0, & \text{otherwise.} \end{cases}$$

Note that this is trivial in the case of the level set method as $a_i \equiv 0$. It is also natural for the BBR scheme as $a_i = 0$ at nodes near K .

We define $\phi^{2h} = \phi^h$ on G^{2h} , $\phi^{4h} = \phi^{2h}$ on G^{4h} , etc., which is a simple subsampling of ϕ . On the other hand, $a^{2h} = I_h^{2h} a^h$, $a^{4h} = I_{2h}^{4h} a^{2h}$, etc., using the restriction operator to add some smoothing to the coefficients. This appears to improve the convergence of the CG method.

With $a^{2^k h}$, $\phi^{2^k h}$ defined on each $G^{2^k h}$ grid, we construct $A^{2^k h}$ and $b^{2^k h}$ as the matrix and right-hand side of the corresponding linear system (3.5) (with index sets $I^{2^k h}$, etc.)

We are finally ready to state the multigrid V-cycle:

- (a) k times iterate the smoother for $A^h v^h = b^h$ starting with $v^{h,(0)}$, obtaining $v^{h,(k)}$.
- (b) Find the residual $r^h = b^h - A^h v^{h,(k)}$.
- (c) Restrict the residual $b^{2h} = I_h^{2h} r^h$.
- (d) Solve $A^{2h} e^{2h} = b^{2h}$ recursively, getting an approximate solution \tilde{e}^{2h} .

- (e) Prolong the approximation error $\tilde{e}^h = I_{2h}^h \tilde{e}^{2h}$, but set $\tilde{e}^h = 0$ on $G^h \cap \{\phi^h \geq 0\}$.
- (f) Correct the approximation $\tilde{v}^h = v^{h,(k)} + \tilde{e}^h$.
- (g) k times iterate the smoother for $A^h v^h = b^h$ starting with \tilde{v}^h , obtaining $\tilde{v}^{h,(k)}$.

At step (d) to obtain \tilde{e}^{2h} , we apply the V-cycle once, with initial guess the zero vector. On the lowest resolution grid with a nontrivial linear system we use a few iterations of the Jacobi method to approximate the exact solution. We use the damped Jacobi method as the smoother with damping parameter $\omega = 2/3$, applied $k = 4$ times.

In Figure 4 we present a convergence test of the multigrid-preconditioned CG method for a numerical setup matching Section 4 with coefficient g_5 (4.5). The CG method decays the residual by about the same factor at each iteration. In general, the convergence is slightly faster for the level set method. However, in the case of the level set the factor worsens when the topology of the free boundary becomes more complex. The convergence, in particular for the BBR method, can be sped up by using the V-cycle twice at step (d) to obtain a better estimate \tilde{e}^{2h} , for about 2 times increased computational cost of the CG iteration. But the improvement of the residual decay is not large enough to justify the additional computational cost in the tests we performed.

3.2 Approximation of the normal velocity near the boundary

Based on the discrete solution of the Laplace equation, we approximate its gradient Dv^k at the nodes inside the computational domain as described in [5, Sec. 3.3.1]. The normal velocity at these nodes is then approximated as

$$V^k(x) := g(x, t_k) |Dv^k|(x)$$

at grid points $x \in G$ near the boundary, i.e., such that $\phi^k(x) < 0$ while at least one of its stencil neighbors $x \pm he_i$, $1 \leq i \leq n$, satisfies $\phi^k \geq 0$. This provides an $O(h)$ approximation of the normal velocity.

3.3 Extension of the normal velocity

The values of V^k are extended from the nodes near the boundary to the whole domain as an approximate solution to (3.2). We use the upwind discretization and the fast marching method as described in [17, Sec 8.3].

3.4 Update of the level set function

We discretize the level set equation (3.1) in time using a forward Euler method and in space using a Godunov discretization of the Hamiltonian, see for example [4].

Defining the finite differences at the interior nodes,

$$D_i^\pm(x) = \frac{\phi^k(x \pm he_i) - \phi^k(x)}{h}, \quad x \in G^0, i = 1, \dots, n,$$

the Godunov Hamiltonian $H_G(\phi^k)$ that approximates the Hamiltonian $V|D\phi|$ is

$$H_G(\phi^k) = \begin{cases} V^k \left(\sum_{i=1}^n \max(D_i^+, D_i^-, 0)^2 \right)^{\frac{1}{2}}, & V^k \leq 0, \\ V^k \left(\sum_{i=1}^n \min(D_i^+, D_i^-, 0)^2 \right)^{\frac{1}{2}}, & V^k > 0. \end{cases}$$

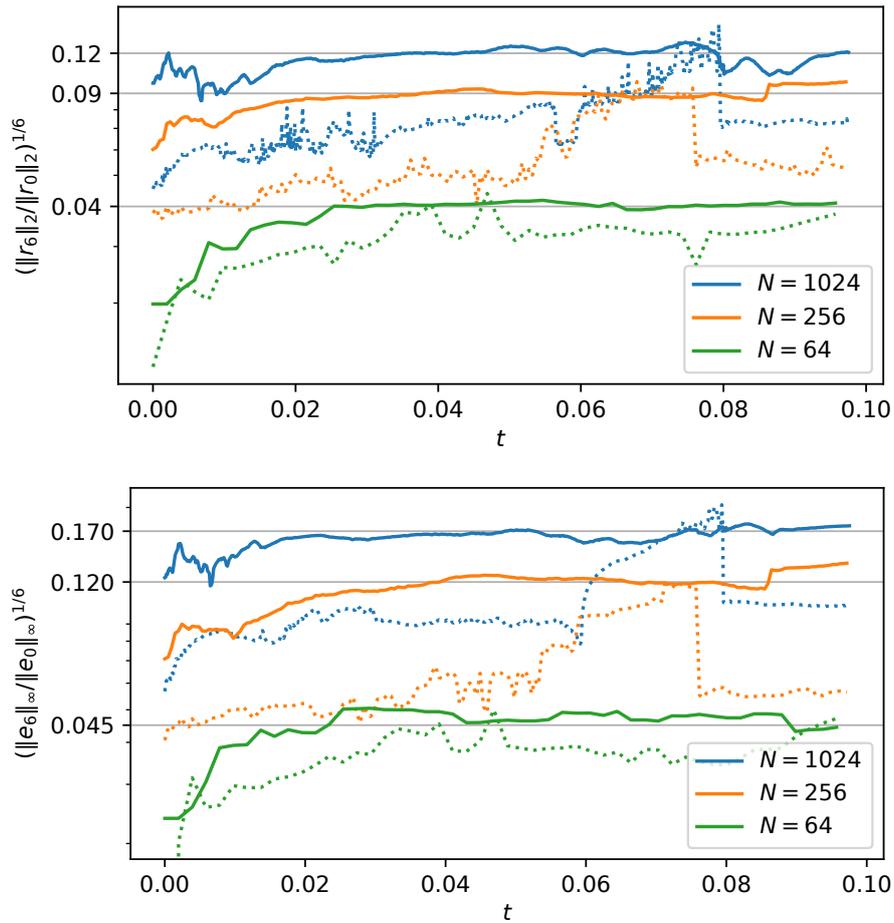


Figure 4: The mean residual decay in the ℓ^2 -norm (top) and the mean error decay of $e_k = v^k - v$ in the max norm (bottom), with v the exact solution of the linear system, of the first 6 iterations of the preconditioned CG solver as a function of time t for 3 grid resolutions: BBR scheme (solid) and level set method (dotted). We use Hele-Shaw coefficient g_5 (4.5) so that a topological change of the free boundary occurs around $t \in [0.06, 0.08]$, see Figure 5.

The forward Euler method for the level set equation then reads

$$\phi^{k+1} = \phi^k - \tau H_G(\phi^k) \quad \text{on } G^0.$$

3.5 Adjusting the level set function

In this section we write $\phi = \phi^k : G \rightarrow \mathbb{R}$ for fixed k . In the above discretization it is important that $D\phi$ does not degenerate to 0 or develop a singularity near the zero level set of ϕ . This is one motivation for the constant extension of V along the streamlines of $D\phi$ using (3.2). However, as the zero level set evolves and possibly undergoes topological changes, the gradient of the level set function degenerates and needs to be artificially adjusted. Most commonly, ϕ is reinitialized as an approximation of the signed distance function of the zero level set of ϕ , see (A.2). Often, an iterative scheme is used, [4, Sec. 3]. Unfortunately, this iteration scheme requires many (~ 100) iterations and needs to be implemented carefully not to move the interface [21]. However, since $O(h^2)$ accuracy is sufficient near the zero level set in our case, we perform the reinitialization using the fast sweeping method [25]. This approach requires only 2^n sweeps through the domain

in the 2^n diagonal directions in a Gauss-Seidel fashion to produce an $O(h^2)$ accurate result near the zero level set. However, the fast-sweeping method requires an $O(h^2)$ accurate initialization at the nodes directly neighboring $\{\phi = 0\}$. These nodes are initialized by a simple second order accurate scheme: we define the grid cells $C_q = q + [0, h]^n$, $q \in h\mathbb{Z}^n$. Let $D(q)$, $q \in h\mathbb{Z}^n$ denote the second order accurate finite difference approximation of the gradient of ϕ at the center of the grid cell C_q , that uses the corner nodes of the cell C_q . For a given node $p \in G^0$ we define the set of neighboring cells that intersect $\{\phi = 0\}$, i.e.,

$$I(p) := \{q \in h\mathbb{Z}^n : p \in C_q \text{ and } \phi \text{ changes sign on } C_q \cap G\}.$$

We define the initialized value of the distance function d as

$$d_0(p) := \inf_{q \in I(p)} \frac{|\phi(p)|}{\|D(q)\|},$$

and set it to $+\infty$ if $I(p) = \emptyset$, that is, when p is not “near” the level set $\{\phi = 0\}$. We believe that this initialization is new. After this initialization, 2^n sweeps of the fast sweeping method are performed, and finally the sign is set to match that of ϕ at each grid node.

4 Numerical results and Discussion

In this section, we present numerical experiments with our proposed BBR-like scheme (1.2) in $n = 2$ spatial dimensions, and compare its performance to the level set method described in Section 3. For simplicity, we consider the spatial computational domain $Q = (0, 1)^2$ with grid $G = h\mathbb{Z}^2 \cap \bar{Q}$ given resolution $N \in \mathbb{N}$ so that $h := 1/N$, as discussed in Section 2.1 and Section 3. To allow easy implementation of the multigrid method, N is always a power of 2. We set $K = \{|x - (0.5, 0.5)| \leq \rho\}$ and $\Omega_0 = \{|x - (0.5, 0.5)| < r_0\}$ with $\rho = 0.05$ and $r_0 = 0.1$.

The time step must be chosen sufficiently small $\tau \sim h/\max g$ for both methods to be stable. We set $\tau = h/8$ for simplicity (see also Section 2.2), and compute the approximate solution at time T_i for given g_i below, $i = 1, \dots, 5$:

T_1	T_2	T_3	T_4	T_5
100×2^{-9}	100×2^{-9}	80×2^{-9}	50×2^{-9}	200×2^{-9}

The values T_i are chosen so that the free boundary stays sufficiently away from ∂Q to justify the Dirichlet boundary condition on ∂Q for the BBR scheme; see the discussion in Section 2.2.

Finally, for the BBR scheme (1.2) we choose γ according to (2.4).

For numerical tests, we consider $g(x, t) = g_i(10x, 10t)$ where g_i is one of the following functions:

$$g_1(x, t) = 1, \tag{4.1}$$

$$g_2(x, t) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 + t, \tag{4.2}$$

$$g_3(x, t) = 0.5 + \cos^2(\pi t), \tag{4.3}$$

$$g_4(x, t) = \sin(2\pi(x_1 + t)) + 2, \tag{4.4}$$

$$g_5(x, t) = \frac{1}{1 + 25 \sin^6(\pi x_1) \sin^6(\pi x_2) (1 + \cos^2(\pi t))}. \tag{4.5}$$

The functions g_1 , g_2 and g_3 are radially symmetric with respect to $(0.5, 0.5)$ and therefore we can compare the numerical solution directly against the “exact” radially symmetric solution in

Section A.1. More precisely, we find the numerical solution $r(t)$ of the ODE (A.1) accurately using Runge-Kutta 4th order method (with time step $\tau = 10^{-4}$). The signed distance function of the radial solution is given exactly as

$$\text{sd}_{\Omega_t}(x) = |x - (0.5, 0.5)| - r(t). \quad (4.6)$$

The function g_4 is not a radially symmetric function but the free boundary is still smooth without topological changes. It is interesting for the study of homogenization as its homogenization limit exhibits strong velocity pinning in x_1 direction, see [18]. The function g_5 , due to its “high contrast”, induces topological changes in the free boundary during evolution, see Figure 5.

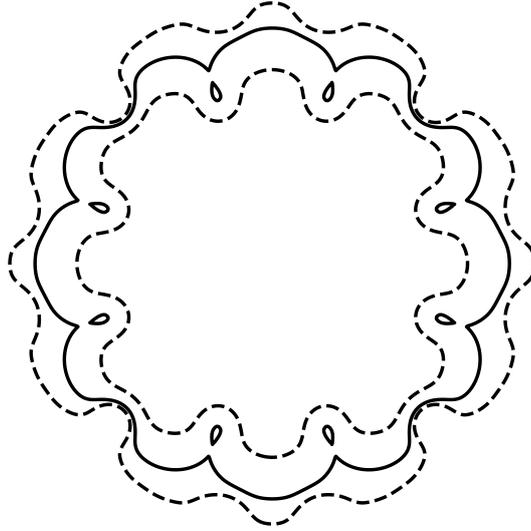


Figure 5: Free boundary evolution at three different time steps illustrating the topological change induced by the function g_5 defined in (4.5). The time step drawn by the solid contour developed small “bubbles” that close up after a short time.

For each test case g_1, \dots, g_5 and each numerical method, we estimate the error $e(N)$ based on the Hausdorff distance of the zero level set of the solution for resolutions $N = 2^6, \dots, 2^{10}$. The estimated order of convergence (EOC) is given by the formula

$$EOC(N) = \log_2 e(N/2) - \log_2 e(N).$$

Depending on g , we perform two kinds of numerical error estimates:

- (a) For g_1, g_2 and g_3 we have a radial solution available so we find the approximate Hausdorff distance of the numerical free boundary of both the BBR scheme and the level set method against the radial solution, using the formula (A.4). The approximate signed distance functions from Section A.3 are compared against the exact signed distance (4.6). See Table 1 and Figure 6.
- (b) For g_4 and g_5 , the exact solution is not available, so we find the approximate Hausdorff distance of the BBR scheme and the level set method against the numerical solution using the BBR scheme (1.2) with higher resolution $N_{\text{high}} = 2^{14}$. See Table 2 and Figure 7. For this resolution, the uniform grid implementation is too slow, so we use an optimized implementation on an adaptive quadtree mesh with high resolution only near the interface, via the

discretization of the Laplacian described in [13]. The solution with this method has essentially the same free boundary position, see the caption of Figure 7. Near the free boundary, the nodes of the quadtree mesh are at the same location as the nodes of the uniform high resolution grid (with resolution N_{high}). The signed distance is computed as explained in Section A.3, on the quadtree mesh. Since the quadtree mesh is uniform near the free boundary, this produces the same result as if it was computed on a uniform grid. Then the result is copied onto the high resolution uniform grid. On the other hand, the signed distance of the low resolution result is first computed on the original low resolution grid using Section A.3, and then bilinearly interpolation onto the high resolution grid (we use a repeated application of the prolongation operator of the multigrid method). The Hausdorff distance is estimated using (A.4) on the high resolution grid from the two resulting approximate signed distance functions.

Comparison of the computational cost

The results of this section have been obtained using a simple Python reference implementation of both methods. We provide the code at <https://github.com/pozar-lab/hele-shaw-bbr>. The priority of this implementation is its correctness and it has not been optimized for performance. The BBR method takes about twice the computational time of the level set method. However, we believe that with proper optimization, the BBR method can potentially be faster. Due to its simplicity, the matrix for discretization of the Laplacian is constructed only once and then each time step requires only

- a linear system solve for the approximation of u^k , and
- a trivial update of w^k .

The proposed CG method with multigrid preconditioner only relies on sparse matrix products, both of these steps are trivially parallelizable. This parallelization is not explored in the Python code. Furthermore, the stopping condition of the CG method should be tuned more precisely. Based on Figure 4, only 4–6 iterations should provide sufficiently accurate solution for mesh resolutions $N \leq 4096$. However, it should be noted that the BBR method requires a band of $10h$ – $20h$ nodes outside of the free boundary to resolve the transition layer (1.2b), in contrast to the level set method, increasing the number of unknowns.

On the other hand, the CG method in the level set method case converges slightly faster, saving on 2–3 iterations if tuned properly, but additionally requires at each time step an update of the discrete Laplacian matrix as the free boundary advances (Section 3.1), a velocity extension (Section 3.3) and a periodic reinitializations (Section 3.5). The latter two rely on the fast marching or fast sweeping methods, both of these are more challenging to parallelize; see [4] for references.

Conclusion.

Based on the numerical experiments, we see that both methods appear to have $O(h)$ convergence. When the free boundary is smooth, as in the radially symmetric cases g_1 , g_2 , g_3 , the level set method appears more accurate by a constant factor as it can track the free boundary to a subgrid precision via a linear interpolation, which is not possible with the BBR-like method. However, this advantage seems to diminish or completely disappear in a solution with a more complicated free boundary, especially when topological changes occur (g_5 in Figure 5), where both methods have about the same error in the free boundary position. Since our proposed BBR-like method

	N	Error (BBR)	EOC (BBR)	Error (LS)	EOC (LS)
g_1	64	3.63×10^{-2}		6.53×10^{-3}	
	128	1.80×10^{-2}	1.01	3.22×10^{-3}	1.02
	256	9.04×10^{-3}	0.99	1.59×10^{-3}	1.02
	512	4.72×10^{-3}	0.94	7.94×10^{-4}	1.00
	1024	2.32×10^{-3}	1.02	3.98×10^{-4}	1.00
g_2	64	3.38×10^{-2}		4.67×10^{-3}	
	128	1.65×10^{-2}	1.03	2.36×10^{-3}	0.98
	256	8.49×10^{-3}	0.96	1.17×10^{-3}	1.02
	512	4.12×10^{-3}	1.04	5.87×10^{-4}	0.99
	1024	2.12×10^{-3}	0.96	2.95×10^{-4}	0.99
g_3	64	4.23×10^{-2}		7.39×10^{-3}	
	128	2.17×10^{-2}	0.96	3.72×10^{-3}	0.99
	256	1.00×10^{-2}	1.12	1.90×10^{-3}	0.97
	512	5.07×10^{-3}	0.98	9.54×10^{-4}	0.99
	1024	2.51×10^{-3}	1.01	4.80×10^{-4}	0.99

Table 1: Errors and estimated order of convergence for the BBR scheme and the level set (LS) method compared to the exact solution in the radially symmetric case for g_1 , g_2 , and g_3 .

	N	Error (BBR)	EOC (BBR)	Error (LS)	EOC (LS)
g_4	64	3.86×10^{-2}		2.14×10^{-2}	
	128	1.98×10^{-2}	0.96	8.86×10^{-3}	1.27
	256	9.93×10^{-3}	1.00	3.95×10^{-3}	1.16
	512	5.37×10^{-3}	0.89	1.91×10^{-3}	1.05
	1024	2.65×10^{-3}	1.02	9.61×10^{-4}	0.99
g_5	64	8.87×10^{-2}		9.21×10^{-2}	
	128	8.45×10^{-2}	0.07	2.50×10^{-2}	1.88
	256	8.69×10^{-3}	3.28	1.04×10^{-2}	1.26
	512	4.09×10^{-3}	1.09	4.59×10^{-3}	1.18
	1024	2.37×10^{-3}	0.79	2.22×10^{-3}	1.05

Table 2: Errors and estimated order of convergence for the BBR scheme and the level set (LS) method compared to a high resolution solution ($N_{\text{high}} = 2^{14}$ using BBR scheme) for g_4 and g_5 .

(1.2) is simpler and significantly easier to implement, and potentially faster to compute than the level set method, it appears to be a competitive alternative to the level set method when higher order accurate method is not needed, as in the numerical homogenization of the Hele-Shaw problem (1.1). It trivially generalizes to higher dimensions. However, to make this method practical in 3D, an adaptive mesh refinement seems necessary. We chose the discretization and multigrid-preconditioned CG method so that they can be easily adapted to quadtree and octree meshes. In an upcoming paper on the homogenization of the Hele-Shaw-type problem in three dimensions we use a 3D implementation with an octree mesh refinement. We do not pursue the 3D case in this paper as the main complexity comes from the mesh refinement aspect and it would overshadow the otherwise simple scheme.

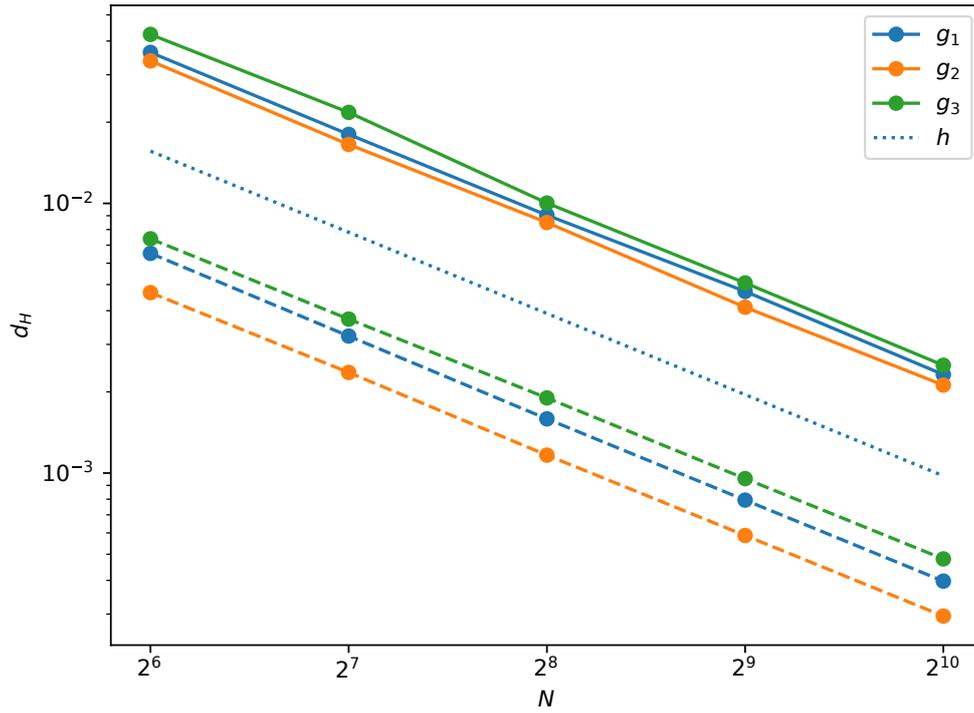


Figure 6: Log-log plot of the errors in Table 1 for g_1 , g_2 and g_3 , each indicated by the color with solid line (BBR scheme) and dashed line (level set method).

A Radial solutions and error estimation

A.1 Radially symmetric solutions of the Hele-Shaw problem

We consider the Hele-Shaw problem (1.1) with radially symmetric data, that is, we assume that $K = \{|x| \leq \rho\}$ and $\Omega_0 = \{|x| < r_0\}$ for some $0 < \rho < r_0$, and that $g(x, t) = f(|x|, t)$ for some positive Lipschitz continuous f . By uniqueness of (1.1) in star-shaped setting, see [20], the solution will also be radially symmetric $\Omega_t = \{|x| < r(t)\}$ for some $r(t)$ that can be determined by an ODE.

Indeed, let us for simplicity assume $n = 2$, as $n \geq 3$ is analogous. The unique solution of (1.1b) is

$$v(x, t) = \frac{\log r(t) - \log |x|}{\log r(t) - \log \rho}, \quad \text{in } \overline{\Omega}_t \setminus \text{int} K = \{\rho \leq |x| \leq r(t)\}.$$

Therefore for $x \in \partial\Omega_t$ we have

$$r'(t) = V(x, t) = g(x, t) |Dv(x, t)| = \frac{f(r(t), t)}{r(t)} \log \left(\frac{r(t)}{\rho} \right), \quad t > 0, \quad (\text{A.1})$$

with $r(0) = r_0$.

A.2 Hausdorff distance via the signed distance function

To quantify the error of the numerical approximation of Ω_t , we use an approximation of the Hausdorff distance of the boundary of two sets. Let $A, B \subset \mathbb{R}^n$ and let sd_A and sd_B be their respective

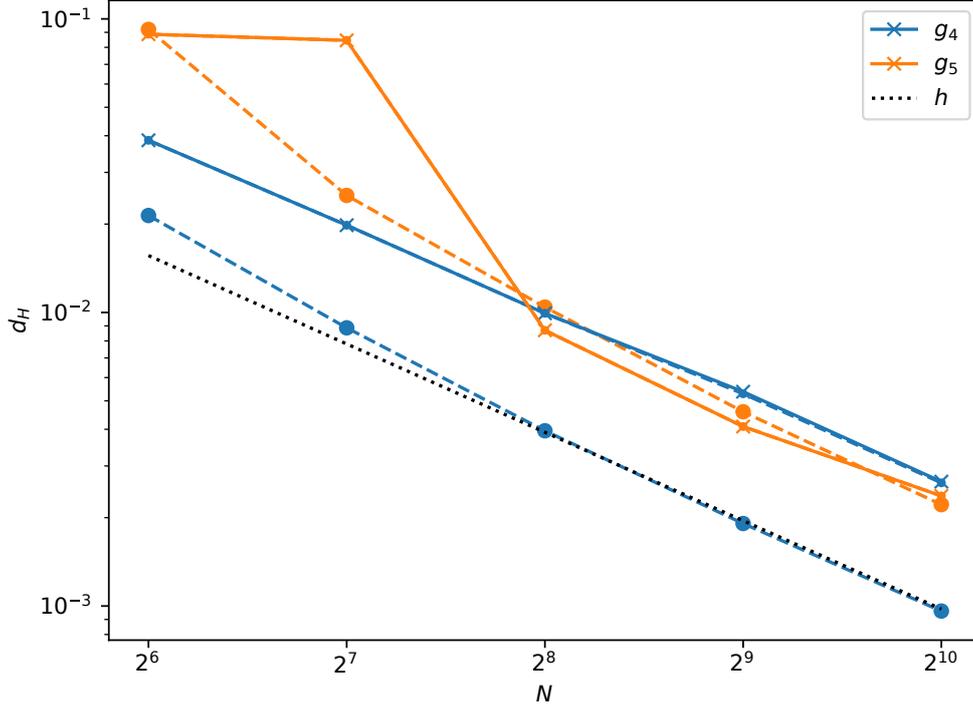


Figure 7: Log-log plot of the errors in Table 2 for g_4 and g_5 , each indicated by the color with solid line (BBR scheme) and dashed line (level set method). For the solid line (BBR scheme), we plot errors for both the reference uniform grid implementation (marker \times) and the optimized quadtree mesh implementation (used also for the high-resolution solution; marker \cdot), verifying that both implementations yield essentially the same position of the free boundary.

signed distance functions, that is, for example

$$\text{sd}_A(x) := \begin{cases} \text{dist}(x, A), & x \in A^c, \\ -\text{dist}(x, A^c), & x \in A. \end{cases} \quad (\text{A.2})$$

Note that $\text{sd}_A(x) = 0$ on ∂A .

The Hausdorff distance between sets ∂A and ∂B is

$$d_H(\partial A, \partial B) = \max \left(\sup_{x \in \partial A} \text{dist}(x, \partial B), \sup_{x \in \partial B} \text{dist}(x, \partial A) \right),$$

which can be written using the signed distance functions as

$$d_H(\partial A, \partial B) = \sup \{ |\text{sd}_A(x) - \text{sd}_B(x)| \mid \min(|\text{sd}_A(x)|, |\text{sd}_B(x)|) = 0 \}. \quad (\text{A.3})$$

For the numerical solution, we only know an approximation of the signed distance functions at the grid nodes in the grid $G \subset \mathbb{R}^n$, so we approximate the Hausdorff distance by considering the signed distance difference on a $\delta = 2h > 0$ neighborhood proportional to the grid step h , that is,

$$d_H(\partial A, \partial B) \approx \max \{ |\tilde{\text{sd}}_A(x) - \tilde{\text{sd}}_B(x)| \mid x \in G, \min(|\tilde{\text{sd}}_A(x)|, |\tilde{\text{sd}}_B(x)|) \leq \delta \}, \quad (\text{A.4})$$

where $\tilde{\text{sd}}_A$ and $\tilde{\text{sd}}_B$ are approximate signed distance functions on the grid G . Note that if the same δ is added in (A.3), the right-hand side is an upper bound on $d_H(\partial A, \partial B)$. See [10] for related results and estimates on approximating the Hausdorff distance of two sets using their distance functions.

A.3 Approximate signed distance function

Here we briefly explain how we approximate the signed distance functions used in (A.4) for the error estimates in Section 4.

For the level set method, we approximate the signed distance function using the fast sweeping method discussed in Section 3.5. This provides a discrete approximation of the signed distance function to $\{\phi^k = 0\}$ that is $O(h^2)$ accurate in a Ch -neighborhood of $\{\phi^k = 0\}$. We denote this approximation by $\text{sd}_{\text{LS}}(\phi^k)$.

For the BBR method (1.2), since w^k has a jump at $\partial\{w^k \geq 0\}$, it is not possible to determine the position of the interface with better than the grid resolution precision. We therefore incorporate this error into the signed distance function approximation in the following way. Let us denote the discrete approximation of w^k on the grid G as \tilde{w}^k . We use the fast sweeping method [25] to compute approximate discrete distance functions \tilde{d}^+ and \tilde{d}^- to the sets $\{x \in G \mid \tilde{w}^k(x) \geq 0\}$ and $\{x \in G \mid \tilde{w}^k(x) < 0\}$, respectively, by initializing their value to zero in the respective sets and running the fast sweeping method. The approximate signed distance function to $\{w^k \geq 0\}$ is then set to be

$$\text{sd}_{\text{BBR}}(\tilde{w}^k)(x) := \tilde{d}^+(x) - \tilde{d}^-(x), \quad x \in G.$$

References

- [1] C. Baiocchi. *Su un problema di frontiera libera connesso a questioni di idraulica*. *Annali di Matematica Pura ed Applicata (IV)*, **92**, 107-127, 1972. <https://doi.org/10.1007/BF02417940>.
- [2] A. E. Berger, H. Brézis, and J. C. W. Rogers. *A numerical method for solving the problem $u_t - \Delta f(u) = 0$* . *RAIRO Anal. Numér.*, **13**(4), 297-312, 1979. DOI: <https://doi.org/10.1051/m2an/1979130402971>
- [3] G. Duvaut. *Résolution d'un problème de Stefan (fusion d'un bloc de glace à zéro degré)*. *Comptes Rendus de l'Académie des Sciences de Paris, Série A-B*, **276**, A1461-A1463, 1973.
- [4] F. Gibou, R. Fedkiw, and S. Osher. *A review of level-set methods and some recent applications*. *Journal of Computational Physics*, **353**, 82-109, 2018. DOI: <https://doi.org/10.1016/j.jcp.2017.10.006>.
- [5] F. Gibou, R. P. Fedkiw, L.-T. Cheng, and M. Kang. *A second-order-accurate symmetric discretization of the Poisson equation on irregular domains*. *Journal of Computational Physics*, **176**(1), 205-227, 2002. DOI: <https://doi.org/10.1006/jcph.2001.6977>.
- [6] H. S. Hele-Shaw. *The flow of water*. DOI: *Nature*, **58**, 34-36, 1898.
- [7] T. Y. Hou, Z. Li, S. Osher, and H. Zhao, *A Hybrid Method for Moving Interface Problems with Application to the Hele-Shaw Flow*. *Journal of Computational Physics*, **134**(2), 236-252, 1997. <https://doi.org/10.1006/jcph.1997.5689>.
- [8] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley, *Removing the Stiffness from Interfacial Flows with Surface Tension*. *Journal of Computational Physics*, **114**(2), 312-338, 1994. <https://doi.org/10.1006/jcph.1994.1170>.
- [9] M. Kimura, and H. Notsu. *A level set method using the signed distance function*. *Japan Journal of Industrial and Applied Mathematics*, **19**, 415-446, 2002. DOI: <https://doi.org/10.1007/BF03167487>.
- [10] D. Kraft. *Computing the Hausdorff Distance of Two Sets from Their Distance Functions*. *International Journal of Computational Geometry and Applications*, **30**(1), 19-49, 2020. DOI: <https://doi.org/10.1142/S0218195920500028>.
- [11] I. Lavi, N. Meunier, and O. Pantz. *Implicit like time discretization for the one-phase Hele-Shaw problem with surface tension*. *ESAIM: Mathematical Modelling and Numerical Analysis (ESAIM: M2AN)*, **59**(1), 419-448, 2025. <https://doi.org/10.1051/m2an/2024005>.
- [12] A. McAdams, E. Sifakis, and J. Teran. *A Parallel Multigrid Poisson Solver for Fluids Simulation on Large Grids*. In: *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, edited by M. Otaduy and Z. Popovic, 2010.
- [13] C. Min, and F. Gibou. *A second order accurate level set method on non-graded adaptive Cartesian grids*. *Journal of Computational Physics*, **225**(1), 300-321, 2007. DOI: <https://doi.org/10.1016/j.jcp.2006.11.034>.

- [14] L. C. Morrow, T. J. Moroney, M. C. Dallaston, and S. W. McCue. *A review of one-phase Hele-Shaw flows and a level-set method for nonstandard configurations*. *The ANZIAM Journal*, **63**(3), 269-307, 2021. DOI: <https://doi.org/10.1017/S144618112100033X>.
- [15] H. Murakawa. *A linear scheme to approximate nonlinear cross-diffusion systems*. *ESAIM: Mathematical Modelling and Numerical Analysis*, **45**(6), 1141-1161, 2011. DOI: <https://doi.org/10.1051/m2an/2011010>.
- [16] S. Osher, and J. A. Sethian. *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*. *Journal of Computational Physics*, **79**(1), 12-49, 1988. DOI: [https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2).
- [17] S. Osher, and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, New York, NY, 2003.
- [18] I. Palupi, and N. Požár. *An efficient numerical method for estimating the average free boundary velocity in an inhomogeneous Hele-Shaw problem*. *Sci. Rep. Kanazawa Univ.*, **62**, 69-86, 2018. DOI: <https://doi.org/10.48550/arXiv.1808.03044>.
- [19] N. Požár. *Long-time behavior of a Hele-Shaw type problem in random media*. *Interfaces and Free Boundaries*, **13**(3), 373-395, 2011. DOI: <https://doi.org/10.4171/IFB/263>.
- [20] N. Požár. *Homogenization of the Hele-Shaw problem in periodic spatiotemporal media*. *Archive for Rational Mechanics and Analysis*, **217**, 155-230, 2015. DOI: <https://doi.org/10.1007/s00205-014-0831-0>.
- [21] G. Russo and P. Smereka. *A Remark on Computing Distance Functions*. *Journal of Computational Physics*, **163**(1), 51-67, 2000. <https://doi.org/10.1006/jcph.2000.6553>.
- [22] K. Sakakibara, Y. Shimoji, and S. Yazaki. *A simple numerical method for Hele-Shaw type problems by the method of fundamental solutions*. *Japan Journal of Industrial and Applied Mathematics*, **39**, 869-887, 2022. <https://doi.org/10.1007/s13160-022-00530-1>.
- [23] O. Tatebe. *The multigrid preconditioned conjugate gradient method*. In: *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, NASA Conference Publication 3224, 1993, pp. 621-634.
- [24] G. Yoon and C. Min. *Analyses on the finite difference method by Gibou et al. for Poisson equation*. *Journal of Computational Physics*, **280**, 184-194, 2015. <https://doi.org/10.1016/j.jcp.2014.09.009>.
- [25] H. Zhao. *A fast sweeping method for Eikonal equations*. *Mathematics of Computation*, **74**(250), 603-627, 2004. DOI: <https://doi.org/10.1090/S0025-5718-04-01678-3>.