Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2016-0040

Vol. **21**, No. 3, pp. 679-691
March 2017

An Efficient Adaptive Rescaling Scheme for Computing Moving Interface Problems

Meng Zhao[1], Wenjun Ying[2], John Lowengrub[3] and Shuwang Li[1,*]

[1] *Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL 60616, USA.*
[2] *Department of Mathematics, Shanghai JiaoTong University, Shanghai 200240, China.*
[3] *Department of Mathematics, University of California at Irvine, Irvine, CA 92697, USA.*

Communicated by Ming-Chih Lai

Received 4 March 2016; Accepted (in revised version) 8 June 2016

Abstract. In this paper, we present an efficient rescaling scheme for computing the *long-time* dynamics of expanding interfaces. The idea is to design an adaptive time-space mapping such that in the new time scale, the interfaces evolves logarithmically fast at early growth stage and exponentially fast at later times. The new spatial scale guarantees the conservation of the area/volume enclosed by the interface. Compared with the original rescaling method in [J. Comput. Phys. 225(1) (2007) 554–567], this adaptive scheme dramatically improves the slow evolution at early times when the size of the interface is small. Our results show that the original three-week computation in [J. Comput. Phys. 225(1) (2007) 554–567] can be reproduced in about one day using the adaptive scheme. We then present the largest and most complicated Hele-Shaw simulation up to date.

AMS subject classifications: 45B05, 76D27, 35R37

Key words: Boundary integral, rescaling scheme, moving boundary problem, Hele-Shaw.

1 Introduction

Many interface problems involve curvature dependent boundary conditions. Examples include the Gibbs-Thomson condition in materials and the Laplace-Young condition in fluids. The interface dynamics is closely related to its local curvature. In these examples, the interface velocity is nonlinearly and nonlocally dependent on the local curvature. For an expanding interface, the curvature may become small and the motion of the interface

*Corresponding author. *Email addresses:* mzhao8@hawk.iit.edu (M. Zhao), wying@sjtu.edu.cn (W. Ying), lowengrb@math.uci.edu (J. Lowengrub), sli@math.iit.edu (S. Li)

 ©2017 Global-Science Press

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

slows down at later times. From a computational point of view, exploring the *long-time* dynamics of such interfaces is challenging to resolve accurately and efficiently.

Two decades ago, Hou *et al.* [15] investigated an air-oil interface problem in a Hele-Shaw cell, in which air is pumped into the center of a Hele-Shaw cell to produce a growing, nearly two-dimensional bubble surrounded by oil. They introduced the small scale decomposition (SSD) technique, which removes the numerical stiffness (due to surface tension and higher order curvature terms) and makes the computation using large time steps possible. In 1993, they were able to compute the dynamics of a complicated bubble until $R \approx 10$. In 2006, Fast *et al.* recomputed the original problem in [15] and pushed the size of the interface to $R \approx 32$ [8], which took 50 days to compute. The SSD idea has since been adapted to facilitate the numerical simulation of many types of interface problems, see [1,5,14,17,20,23]for a sample.

In this paper, we use the original air-oil problem in a Hele-Shaw cell [8,15] to benchmark the cost and efficiency of our adaptive rescaling scheme. In particular, the air bubble grows as a result of a constant influx of air [8,15], and the equivalent bubble radius evolves as $dR/dt \sim R^{-1}$, where R is the radius of a circle with the same area as the bubble. Consequently the velocity of the interface, dR/dt, decreases as R increases (the bubble grows). This, together with the numerical stiffness introduced by surface tension forces makes the simulation of the air-oil problem over *long-time* very expensive. In 2007, Li *et al.* [21] developed a rescaling scheme that substantially reduces the computation time. Their method is based on a time-space mapping scheme such that in the new frame the interface grows exponentially fast in time and the bubble maintains a constant area in space, while preserving the original physics. This scheme enables one to push the size of the interface to $R \approx 67$ in three weeks. In particular, it takes only six days to reproduce the results in [8], which originally took 50 days. This rescaling strategy has subsequently been used in numerous applications, e.g., see [2,9,30,31]. This rescaling scheme works best for large-size interfaces and at late times since the evolution in the original frame at these times is very slow. At early times when the bubble is small, however, we find that the rescaling scheme actually makes the evolution quite slow, and a significant portion of CPU time is used to compute the slow development of viscous fingers.

In this paper, we propose an adaptive time frame to speed up the motion of the interface at early growth stages while preserving the exponentially fast growth at later times as in [21]. The idea is to design a scaling function ρ_l such that in the new time frame, the interface evolves logarithmically fast when R is small, and transitions to exponential growth at later times. Specifically, we define a new time scaling function ρ_s that combines the logarithmic growth scaling function ρ_l and an exponential growth scaling function ρ_e. In addition, during growth the new spatial scale guarantees the conservation of the area/volume enclosed by the interface. Comparing with the original method in [21] which only uses the exponential scaling, this adaptive scheme dramatically accelerates the interface evolution at early times.

The new scaling function ρ_s helps reduce the CPU time significantly. On a Linux system with Xeon 2.53 GHz CPU, using the same resolution $N = 65{,}536$ points on the

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

interface and the time step $\Delta\bar{t}=1\times10^{-4}$ in the rescaled frame as [21], we can reproduce the *long-time* simulation of the bubble just in one day, which took three weeks in [21]. At early times, after mapping the same time step ($\Delta\bar{t}=1\times10^{-4}$) from the rescaled frame back to the original frame, we find that the equivalent time step from ρ_s frame is actually 20 times larger than that from the ρ_e frame. Using $N=131{,}072$ points along the interface and time step $\Delta\bar{t}=2.5\times10^{-5}$, we produce the largest and most complicated Hele-Shaw simulation in 5.8 days and reach $R=122.09$. The rescaling scheme proposed here states a general idea and is not limited to boundary integral methods, but can also be applied to Cartesian-grid based methods to improve efficiency [30].

This paper is organized as follows. In Section 2, we review the governing equations for a bubble in a Hele-Shaw cell. In Section 3, we present the rescaling idea. In Section 4, we discuss the numerical results. We give conclusions in Section 5.

2 Review of the Hele-Shaw problem

Though simple in terms of formulation (as shown below), the Hele-Shaw problem exhibits the interesting and challenging features of a general moving boundary problem – sensitive nonlinearity, non-locality and a strong dependence on the geometric properties of the interface. The Hele-Shaw problem has been used as a model system for studying interface dynamics and pattern formation [3,4,6,7,12,13,15,18,19,22,24,25,27–29]. Consider a radial Hele-Shaw cell and let $\Gamma(t)$ be the moving interface separating a less viscous fluid domain E_1 from a viscous fluid domain E_2. See Fig. 1 for a schematic diagram for an air-oil interface system.

Differential Form. For an air-oil interface system, the pressure of the interior bubble is constant in space, so we only need to solve the exterior problem. We assume the exterior fluid obeys Darcy's Law,

$$\mathbf{u}=-M\nabla P \quad \text{for} \quad \mathbf{x}\in E_2, \tag{2.1}$$

where $\mathbf{u}$ is the velocity of the exterior fluid, P is the corresponding pressure, and $M=\frac{h^2}{12\mu}$ is the mobility. The parameters h is the width of the gap between the two parallel plates of the Hele-Shaw cell and μ is the viscosity of the exterior fluid. For incompressible fluids, we have $\nabla\cdot\mathbf{u}=0$. Thus, the pressure of the fluids satisfies

$$\nabla^2 P=0 \quad \text{for} \quad \mathbf{x}\in E_2. \tag{2.2}$$

Across the interface, the fluid normal velocity is continuous and the pressure has a jump given by the Laplace-Young condition

$$[P]_t=\tau\kappa \quad \text{for} \quad \mathbf{x}\in\Gamma(t), \tag{2.3}$$

where τ is the surface tension and κ is the curvature. To complete the problem, the air is injected at flow rate $J(t)$,

$$\int_{\Sigma_0}\frac{\partial P}{\partial\mathbf{n}}ds=J(t), \tag{2.4}$$

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

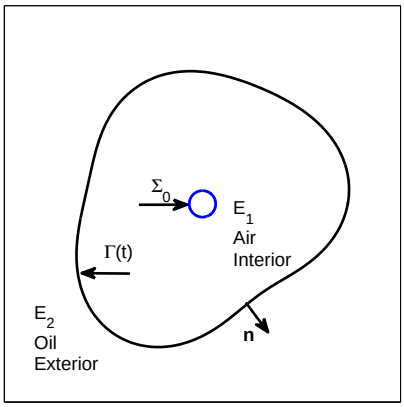


Figure 1: A schematic diagram for an air-oil interface system, where the less viscous fluid is pumped into the system through the region marked by the blue circle.

where s is the arclength, Σ_0 is a small circle centered at origin and $\mathbf{n}$ is the outward normal. The motion of the interface is given by $\frac{d\mathbf{x}}{dt}=\mathbf{u}$ for $\mathbf{x}\in\Gamma(t)$.

Integral Form. From potential theory, the pressure of the exterior viscous fluid can be written as

$$P(\mathbf{x})=\frac{1}{2\pi}\int_{\Gamma(t)}\mu(\mathbf{x}')\left(\frac{\partial \ln|\mathbf{x}-\mathbf{x}'|}{\partial \mathbf{n}(\mathbf{x}')}+1\right)ds(\mathbf{x}')+J\ln|\mathbf{x}|, \tag{2.5}$$

where $\mu(\mathbf{x})$ is a dipole density on the moving interface $\Gamma(t)$. Taking the limit $\mathbf{x}\to\mathbf{x}'\in\Gamma(t)$ and using Eq. (2.3), we obtain a Fredholm integral equation of the second kind

$$\mu(\mathbf{x})-\frac{1}{\pi}\int_{\Gamma(t)}\mu(\mathbf{x}')\left(\frac{\partial \ln|\mathbf{x}-\mathbf{x}'|}{\partial \mathbf{n}(\mathbf{x}')}+1\right)ds(\mathbf{x}')-2J\ln|\mathbf{x}|=-2\tau\kappa \tag{2.6}$$

with

$$\int_{\Gamma(t)}\mu(\mathbf{x})ds(\mathbf{x})=0. \tag{2.7}$$

Eq. (2.6) is well-conditioned and $\mu(\mathbf{x})$ can be solved efficiently using an iterative method, e.g. GMRES [26]. Once we get $\mu(\mathbf{x})$, the normal velocity V can be computed via the Dirichlet-Neumann mapping [10],

$$V(t)=\frac{1}{2\pi}\int_{\Gamma(t)}\mu_{s'}\frac{(\mathbf{x}'-\mathbf{x})^{\perp}\cdot\mathbf{n}}{|\mathbf{x}'-\mathbf{x}|^2}ds'+J\frac{\mathbf{x}\cdot\mathbf{n}}{|\mathbf{x}|^2}, \tag{2.8}$$

where $\mathbf{x}^{\perp}=(y,-x)$ and the subscript denotes derivatives with respect to arclength s. Note that in Eq. (2.8), the normal velocity decreases as the bubble size $|\mathbf{x}|$ gets large.

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

3 Adaptive rescaling scheme

Following [21], we introduce a new rescaled space and time frame $(\bar{\mathbf{x}},\bar{t})$ such that

$$\mathbf{x}=\bar{R}(\bar{t})\bar{\mathbf{x}}(\bar{t},\alpha), \tag{3.1}$$

$$\bar{t}=\int_0^t \frac{1}{\rho(t')}dt', \tag{3.2}$$

where $\bar{R}(\bar{t})=R(t(\bar{t}))$ is the space scaling factor representing the size of the interface, and $\bar{\mathbf{x}}$ is the position vector of the scaled interface with parametrization α. The time scale function $\rho(t)$ maps the original time t to the new time $\bar{t}$. In general, $\rho(t(\bar{t}))=\bar{\rho}(\bar{t})$ can be chosen arbitrarily to make the interface evolve in the new frame at any speed. The rescaled normal velocity $\bar{V}$ satisfies

$$\bar{V}(\bar{t})=\frac{\bar{\rho}}{\bar{R}}V(t(\bar{t}))-\frac{\bar{\mathbf{x}}\cdot\mathbf{n}}{\bar{R}}\frac{d\bar{R}}{d\bar{t}}, \tag{3.3}$$

where V is the original normal velocity and we have suppressed the spatial dependence. In the rescaled frame, we require the area enclosed by the interface remains constant, $\bar{A}(\bar{t})=\bar{A}(0)$. That is the integration of the normal velocity along the interface in the scaled frame vanishes, $\int_{\bar{\Gamma}(\bar{t})}\bar{V}d\bar{s}=0$. As a consequence,

$$\frac{d\bar{R}}{d\bar{t}}=\frac{\pi\bar{\rho}\bar{J}}{\bar{A}(0)\bar{R}}, \tag{3.4}$$

where $\bar{J}(\bar{t})=J(t(\bar{t}))$. In [21], the authors chose ρ such that $\bar{R}=\exp(\bar{t})$ in the rescaled frame. At early times when $\bar{R}$ is small, however, the exponential scaling makes evolution actually quite slow, and a significant portion of CPU time is used to compute the slow development of viscous fingers. Here, we choose $\bar{R}$ to be a logarithmic function $\bar{R}=\frac{\log(a+b\bar{t})}{\log(a)}$ for some constants $a>1$ and $b>0$, which gives

$$\rho_l=\frac{b}{(a+b\bar{t})\log(a)}\frac{\bar{A}(0)}{\pi\bar{J}}\bar{R}, \tag{3.5}$$

$$t=\frac{\bar{A}(0)}{2\log^2(a)\pi\bar{J}}(\log^2(a+b\bar{t})-\log^2(a)). \tag{3.6}$$

The normal velocity in the logarithmic frame is given by

$$\bar{V}_l=\frac{b\bar{A}(0)}{(a+b\bar{t})\log(a)\pi\bar{J}\bar{R}}\left(\frac{1}{2\pi\bar{R}}\int_\Gamma\bar{\mu}_{\bar{s}'}\frac{(\bar{\mathbf{x}}'-\bar{\mathbf{x}})^\perp\cdot\bar{\mathbf{n}}(\bar{\mathbf{x}})}{|\bar{\mathbf{x}}'-\bar{\mathbf{x}}|^2}d\bar{s}'+\bar{J}\frac{\bar{\mathbf{x}}\cdot\bar{\mathbf{n}}}{|\bar{\mathbf{x}}|^2}\right)$$
$$-\frac{b}{\log(a)(a+b\bar{t})\bar{R}}\bar{\mathbf{x}}\cdot\bar{\mathbf{n}}. \tag{3.7}$$

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

The disadvantage of using ρ_l for the entire evolution is that it slows down the motion of the interface when $\bar{R}$ gets large, see Fig. 2 for a comparison. One intuitive remedy is to switch the logarithmic time scale back to the exponential scale at some time $\bar{t}=\bar{t}_0$ with $\bar{R}=\bar{R}_0$, which can be chosen arbitrarily. That is when $\bar{R}>\bar{R}_0$, we use

$$\rho_e = c\bar{R}^2, \tag{3.8}$$

$$t = \frac{R_0^2\bar{A}(0)}{2\pi\bar{J}}(\exp(2c(\bar{t}-\bar{t}_0))-1)+\frac{\bar{A}(0)}{2\pi\bar{J}}(\bar{R}_0^2-1), \quad \bar{t}\geq\bar{t}_0, \tag{3.9}$$

where $c=\frac{b\bar{A}(0)}{(a+b\bar{t}_0)\log(a)\pi\bar{J}\bar{R}_0}$ and $\bar{R}_0=\frac{\log(a+b\bar{t}_0)}{\log(a)}$ is the space scaling factor at $\bar{t}_0$. The normal velocity in the exponential frame is computed as,

$$\bar{V}_e = c\left(\frac{1}{2\pi\bar{R}}\int_{\bar{\Gamma}}\bar{\mu}_{\bar{s}'}\frac{(\bar{\mathbf{x}}'-\bar{\mathbf{x}})^{\perp}\cdot\bar{\mathbf{n}}(\bar{\mathbf{x}})}{|\bar{\mathbf{x}}'-\bar{\mathbf{x}}|^2}d\bar{s}'+\bar{J}\frac{\bar{\mathbf{x}}\cdot\bar{\mathbf{n}}}{|\bar{\mathbf{x}}|^2}\right)-\frac{\pi\bar{J}c}{\bar{A}(0)}\bar{\mathbf{x}}\cdot\bar{\mathbf{n}}. \tag{3.10}$$

Note the time scaling function ρ is only continuous but not smooth at the switch point. This may cause numerical inconsistency if a multistep method (e.g. Adams-Bashforth) is used to update the interface, as the time-step $\Delta\bar{t}$ in these two frames is not equivalent. A simple fix is to use the forward Euler method to compute the first step when the switch occurs. Then the Adams-Bashforth method can be implemented thereafter. However, this approach can increase the overall numerical error.

Alternatively, we can simply define a new scaling time function ρ_s that smoothly combines the logarithmic scaling ρ_l and the exponential scaling ρ_e. The advantage is that ρ_s is smooth at the switch point, while a drawback is that we do not have an analytical formula for the radius $\bar{R}$. That is the radius $\bar{R}$ now has to be calculated numerically using an ODE solver. Specifically, ρ_s is given by

$$\rho_s = \frac{b}{\log(a)}\frac{\bar{A}(0)}{\pi\bar{J}}\bar{R}^2\Bigg(\underbrace{\frac{1}{a^{\bar{R}}\bar{R}}}_{\text{logarithmic part}}+\underbrace{\frac{1}{a^{\bar{R}_0}\bar{R}_0}}_{\text{exponential part}}\Bigg). \tag{3.11}$$

At early times, ρ_s is dominated by the logarithmic term in Eq. (3.11); at later times it is dominated by the exponential term. Though the switch point $\bar{R}_0$ can be chosen arbitrarily, here we choose $\bar{R}_0$ such that at the switch point, $\rho_l(\bar{R}_0)=\rho_e(\bar{R}_0)$. Fig. 2 shows an example of the logarithmic scaling function $\rho_l=20.3007\bar{R}\frac{1}{1.0515^{\bar{R}}}$, the exponential scaling function $\rho_e=20.3007\bar{R}^2\frac{1}{1.0515^{11}\times 11}$, and the combined smooth time scaling function $\rho_s=20.3007\bar{R}^2(\frac{1}{1.0515^{\bar{R}}\bar{R}}+\frac{1}{1.0515^{11}\times 11})$, where we set the switch point $\bar{R}_0=11$ with $\bar{t}_0=0.6797$ and parameters $a=1.0515$, $b=1.01$, $\bar{J}=1$, and $\bar{A}(0)=3.173$ for an initial shape $r(\alpha,0)=1+0.1\times(\sin(2\alpha)+\cos(3\alpha))$. Note that the speed of the evolution depends on the choice of a and b in ρ_l. Although making b large and a close to 1 can lead to a fast evolution in the rescaled frame, there is still a time step constraint to be considered in practice. After testing different values of a and b, we find that $a=1.0515$ and $b=1.01$ yield good stability and efficiency, and work nicely for the time step $\Delta\bar{t}=1\times10^{-4}$.

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

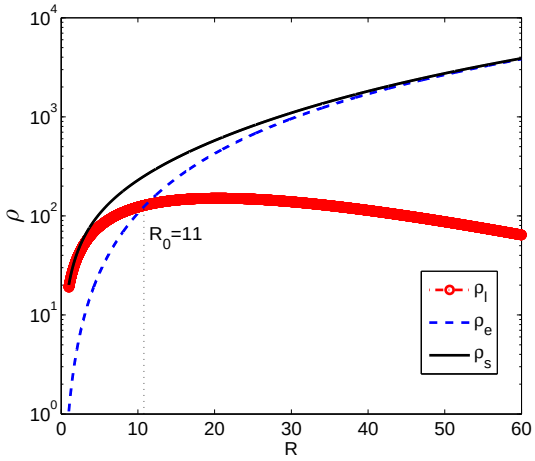


Figure 2: Relation between the time scaling function ρ and the radius R. The function ρ_l represents the logarithmic time scaling function, ρ_e represents the exponential time scaling function, and ρ_s represents the combined time scaling function in Eq. (3.11).

Setting $\bar{\mu}=\mu\bar{R}$, we rewrite Eqs. (2.6) and (2.7) in the rescaled frame as

$$\bar{\mu}(\bar{\mathbf{x}})-\frac{1}{\pi}\int_{\bar{\Gamma}(\bar{t})}\bar{\mu}(\bar{\mathbf{x}}')\left[\frac{\partial\ln|\bar{\mathbf{x}}-\bar{\mathbf{x}}'|}{\partial\mathbf{n}(\bar{\mathbf{x}}')}+\bar{R}(\bar{t})\right]d\bar{s}(\bar{\mathbf{x}}')=-2\tau\bar{\kappa}+2\bar{R}\bar{J}(\ln(\bar{R}(\bar{t}))+\ln|\bar{\mathbf{x}}|), \tag{3.12}$$

and

$$\int_{\bar{\Gamma}(\bar{t})}\bar{\mu}(\bar{\mathbf{x}})d\bar{s}(\bar{\mathbf{x}})=0. \tag{3.13}$$

Accordingly, using ρ_s in Eq. (3.11) as the time scaling function, we compute the normal velocity in the new frame as

$$\bar{V}_s=\frac{b\bar{A}(0)}{\log(a)\pi\bar{J}}\left(\frac{1}{a^{\bar{R}}\bar{R}}+\frac{1}{a^{R_0}R_0}\right)\left(\frac{1}{2\pi\bar{R}}\int_{\bar{\Gamma}}\bar{\mu}_{\bar{s}'}\frac{(\bar{\mathbf{x}}'-\bar{\mathbf{x}})^{\perp}\cdot\bar{\mathbf{n}}(\bar{\mathbf{x}})}{|\bar{\mathbf{x}}'-\bar{\mathbf{x}}|^2}d\bar{s}'+\bar{J}\frac{\bar{\mathbf{x}}\cdot\bar{\mathbf{n}}}{|\bar{\mathbf{x}}|^2}-\frac{\pi\bar{J}}{\bar{A}(0)}\bar{\mathbf{x}}\cdot\bar{\mathbf{n}}\right), \tag{3.14}$$

where $\bar{\mathbf{x}}^{\perp}=(\bar{y},-\bar{x})$.

To evolve the interface numerically, we first solve Eq. (3.4) for $\bar{R}(\bar{t})$ using the 2nd order accurate Adams-Bashforth method. Then we discretize Eqs. (3.12) and (3.13) in space. Following [15, 16], we evaluate these integrals using the fast multipole method [11], and we solve for rescaled dipole density $\bar{\mu}$ using GMRES [26]. Because Eq. (3.12) is well-conditioned, no preconditioner is needed. Once the solution $\bar{\mu}$ is obtained, we evaluate Eq. (3.14) for the rescaled normal velocity $\bar{V}$ via the Dirichlet-Neumann map [10]. Finally we evolve the interface in the scaled frame using a second order accurate non-stiff updating scheme in time and the equal arclength parameterization [15, 16].

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

4 Results

4.1 Convergence test

We test the convergence of our scheme using the initial shape $r(\alpha,0)=1+0.1\times(\sin(2\alpha)+\cos(3\alpha))$, which is the same as that used in [15, 21]. The air is injected into the viscous oil at a constant flow rate $J=1$ and the surface tension is $\tau=0.001$. First we study the temporal accuracy using $N=8192$ mesh points along the interface. The time steps are set as $\Delta\bar{t}=2\times10^{-4}$, 1×10^{-4}, 5×10^{-5}, 2.5×10^{-5}, and 1.25×10^{-5}. The numerical error is measured by $Error=|\bar{A}_{\bar{t}}-\bar{A}_0|$, where $\bar{A}_{\bar{t}}$ is the area enclosed by the interface in the scaled frame at time $\bar{t}$, and $\bar{A}_0$ is the initial area. Fig. 3(a) shows the base 10 logarithm of the temporal error plotted versus the scaling factor $R(t)=\bar{R}(\bar{t})$. The morphologies of the interfaces are shown as insets. When the time step is reduced by half, the numerical error is decreased by 0.6 in distance indicating the convergent rate in time is almost 2.

Next we study the accuracy in space. We compare the shape of the interface using $N=8192$, 16384, 32768, 65536 and the time step $\Delta\bar{t}=1E-4$ which is small enough for our study. The error is again measured as $Error=|\bar{A}_{\bar{t}}-\bar{A}_0|$. The results are summarized in Fig. 3(b). The detailed morphologies are shown as insets.The final radii are $\bar{R}(\bar{t})=10.55$ for $N=8192$; $\bar{R}(\bar{t})=20.21$ for $N=16{,}384$; $\bar{R}(\bar{t})=38.90$ for $N=32{,}768$; and $\bar{R}(\bar{t})=65.52$ for $N=65{,}536$. The morphologies at the same radius are identical. To run longer, more mesh points are needed to resolve the complicated interface.

As an additional test, we compute the maximum point-wise error by comparing with a highly resolved reference calculation. For the time convergence study, we set $N=8192$ and $\Delta\bar{t}=1.25\times10^{-5}$ as the reference calculation (subscript notation c). We define relative error using L_∞ norm: $Error(\bar{t})_{L_\infty}=\frac{|r(\alpha,\bar{t})-r_c(\alpha,\bar{t})|_{L_\infty}}{|r_c(\alpha,\bar{t})|_{L_\infty}}$, where $r(\alpha,\bar{t})=\sqrt{x^2(\alpha,\bar{t})+y^2(\alpha,\bar{t})}$, $r_c(\alpha,\bar{t})=\sqrt{x_c^2(\alpha,\bar{t})+y_c^2(\alpha,\bar{t})}$, and $(x(\alpha,\bar{t}),y(\alpha,\bar{t}))$ is the interface point for calculations using time steps $\Delta\bar{t}=2\times10^{-4}$,$1\times10^{-4}$,$0.5\times10^{-4}$ and 0.25×10^{-4}. Note that $(x_c(\alpha,\bar{t}),y_c(\alpha,\bar{t})$ is the corresponding point in the reference calculation. As shown in Fig. 3(c), the distance between neighboring curves is about 0.6, indicating the scheme is 2nd order accurate in time. For the space convergence study, we use $N=65536$, $\Delta\bar{t}=1\times10^{-4}$ as the reference calculation. Similar to the temporal study, we compute point-wise errors and plot the their evolutions in Fig. 3(d). It is evident that the scheme is convergent. At early times ($R<2$), we observe spectral accuracy as expected. As $R(t)$ increases, the temporal error dominates and the spatial accuracy is a bit lower than the spectral accuracy at long times.

4.2 Efficiency

To benchmark the efficiency of our adaptive rescaling scheme, we consider the same interface problem as in Fig. 3, i.e. the initial interface $r(\alpha,0)=1+0.1\times(\sin(2\alpha)+\cos(3\alpha))$, and set the mesh points $N=65{,}536$ along the interface and the time step $\Delta\bar{t}=1\times10^{-4}$ [15, 21]. On a Linux system with Xeon 2.53 GHz CPU, we compute the dynamics of the interface using the exponential time scaling function ρ_e and the combined time scaling

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

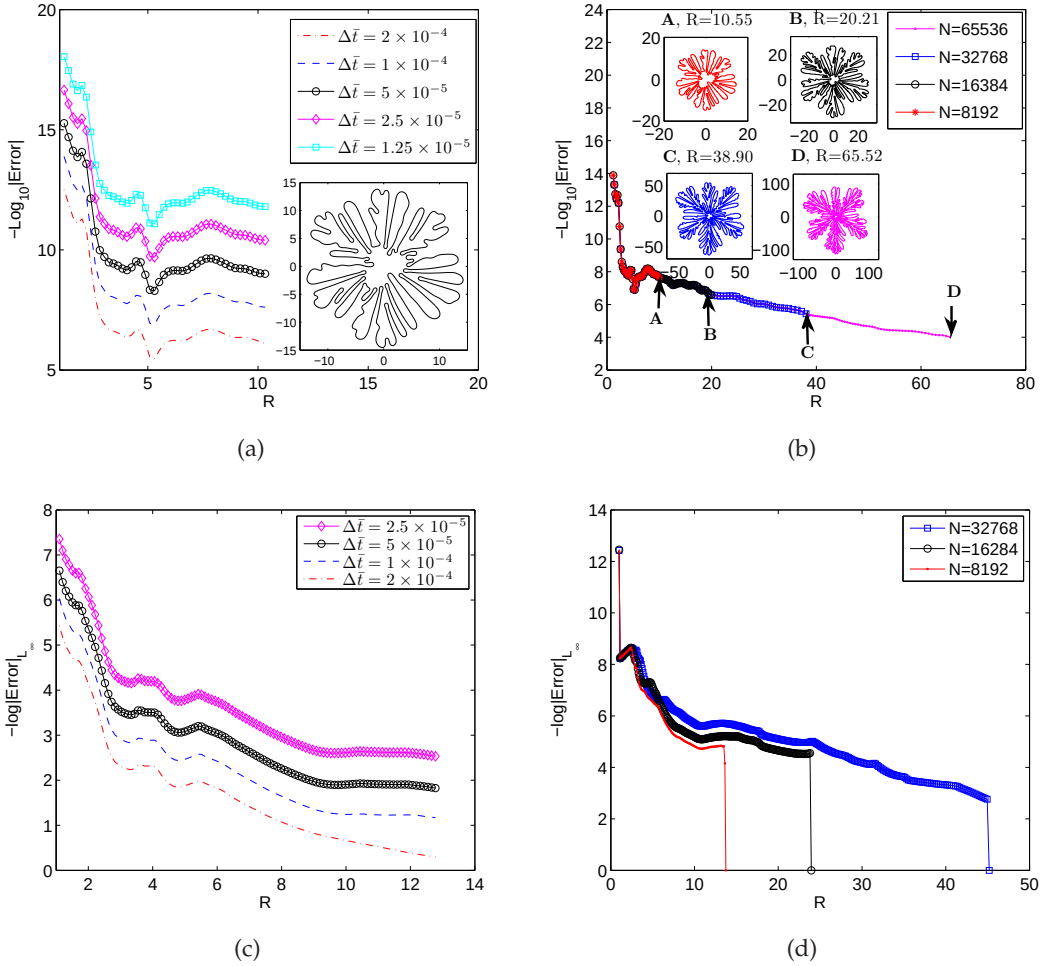
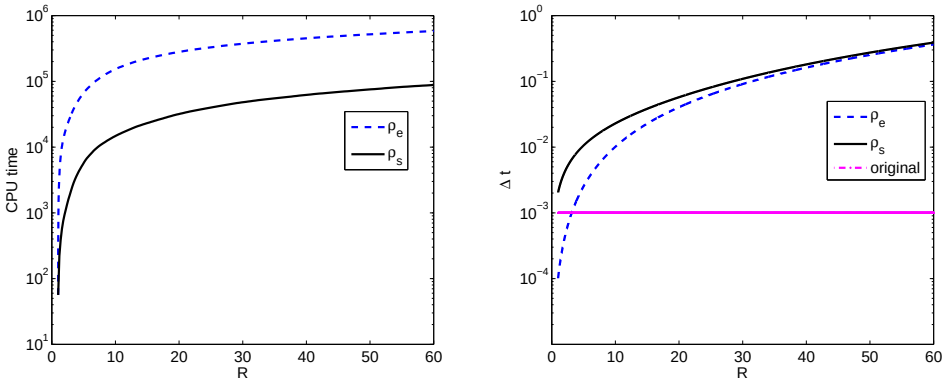


Figure 3: (a) shows the scheme is convergent in time (absolute error measured using area $A(t)$). (b) shows the scheme is convergent in space (absolute error measured using $A(t)$). (c) shows the scheme is convergent in time using the point-wise maximum relative error measure. (d) shows the scheme is convergent in space using the point-wise maximum relative error measure.

function ρ_s, respectively. In Fig. 4(a), we plot the CPU time (seconds) as a function of the size of the interface. It is evident that the performance of ρ_s surpasses ρ_e, and in particular, it takes only one day to reproduce the largest Hele-Shaw bubble simulation in [21], which originally took three weeks (now takes a week on the same Xeon cluster). The reason for such high efficiency is demonstrated in Fig. 4(b), where we map the time step $\Delta\bar{t}$ back to the original time frame. Note that at early growth stages, the equivalent time step in the current time frame for ρ_s is about 20 times larger than that for the exponential scaling function ρ_e, while they are almost identical at later times, as expected.

The horizontal line indicates the time step $\Delta t = 1\times10^{-3}$ used in [8, 15]. For the ex-

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press



(a) (b)

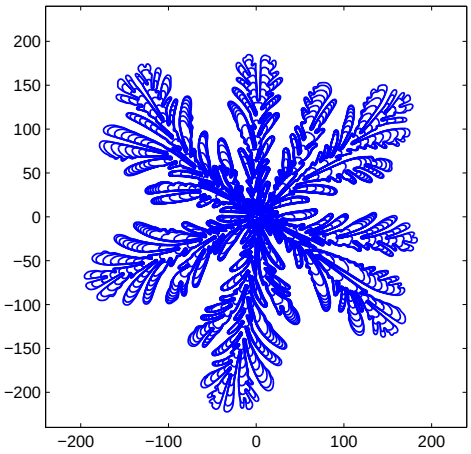
Figure 4: (a) shows the CPU time as a function of interface size for the different scaling factors ρ_e and ρ_s, as labeled. (b) shows the corresponding time step in the original time frame.

ponential time scale ρ_e used in [21], at early times when $R<3.146$, we observe that the equivalent time step is actually smaller than those used in [8,15].

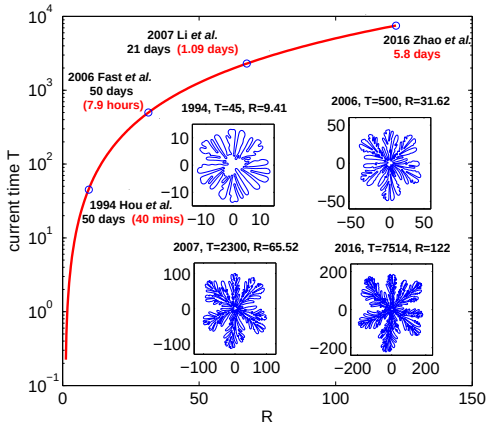
Next, we use $N=131{,}072$ and the time step $\Delta\bar{t}=2.5\times10^{-5}$ to explore the dynamics further. The computation takes 5.8 days to reach $R=122.09$ at the final rescaled time $\bar{T}=2.43$ (corresponding final time in the original frame $T=7514$). In Fig. 5(a), we show the largest and most complicated Hele-Shaw simulation up to date. Comparing with the results in [21], the unceasingly development and splitting of viscous fingers leads to a much more ramified branching pattern. In Fig. 5(b), we plot the final time T in the current frame for previous studies in [8,15,21]. We also plot the final morphologies as insets, and indicates the CPU times needed (in red brackets) to reproduce the previous results using the adaptive rescaling scheme. It is evident that the adaptive scheme enables one to efficiently and accurately simulate the *very long-time* dynamics of moving interfaces.

In Fig. 5(c), we shows the relation between the area $A(t)$ and the arclength $L(t)$ of the interface using $A(t)\propto L(t)^{\gamma}$. Both area and arclength are measured in the current time frame. For a circular geometry, $\gamma=2$. Smaller γ indicates a more complicated geometry, i.e. the arclength $L(t)$ grows faster the area $A(t)$. It is evident that the evolution experiences several transitions connected approximately by a few linear segments (with sample morphologies shown as insets). At early times, the interface is small and compact with γ=1.98. Later, the fast development of new fingers by tip-splitting events leads to a much more rapid increase in interface length and the scaling $\gamma=0.61$. Then the interface tends to develop long fingers rather than produce more fingers, indicating slower increment in arclength than the increment in area. As a consequence, γ is increased gradually from 0.61 to 1.45. At $L=398$, γ drops to 1.39, and slightly picks up to 1.42 after $L=794$. For the last three stages, the γs are very close indicating the evolution may have achieved a subtle balance between developing new fingers and prolonging the existing fingers. We will explore this further in a forthcoming paper.

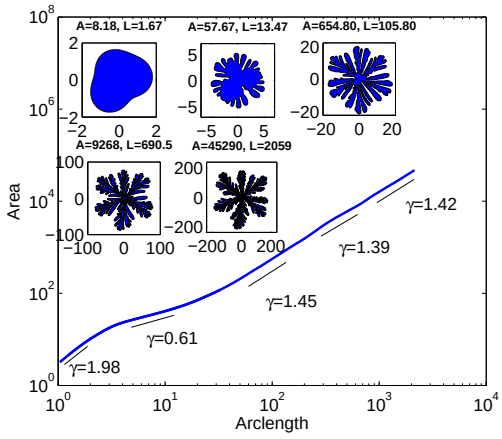
https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press



(a)



(b)



(c)

Figure 5: (a) shows the largest and most complicated Hele-Shaw bubble simulation up to date. (b) shows the current time for the classic work in [8,15,21]. (c) shows the measurement of area $A(t)$ and arclength $L(t)$ using $A(t) \propto L(t)^{\gamma}$.

5 Conclusion

In this paper, we re-examined the efficiency of the rescaling scheme proposed earlier by Li *et al.* [21] and introduced an adaptive rescaling scheme to speed up the slow evolution at early times that arises from the earlier rescaling scheme of Li *et al.* [21]. The adaptive scaling function combines the advantages of a logarithmic time scale function that gives fast growth at early times, and an exponential time scale function that yields fast growth at later times. The new adaptive scheme enables one to reproduce the results in [21]

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

about 1 day, which originally costs 21 days. We then presented the largest and most complicated Hele-Shaw bubble simulation up to date.

Though we focus our study on a radial Hele-Shaw problem with the boundary integral method, the rescaling scheme is actually a general idea that can applied to a set of moving boundary problems using other numerical methods, including kernel-free Cartesian mesh-based methods [30]. The rescaling scheme is especially valuable for Cartesian mesh-based methods because the computational domain typically does not need to be changed in the rescaled frame since in this frame, the rescaled interface has a fixed enclosed area.

Acknowledgments

SL, and JL would like to thank the support from National Science Foundation through grants DMS-1217277 (SL) and DMS-1217273 (JL) for partial support. SL also acknowledges the partial support through grant ECCS-1307625. Some computations in this work were performed on computers acquired using NSF grant (SCREMS) DMS-0923111.

References

[1] D. M. Ambrose, M. Siegel, and S. Tlupova. A small-scale decomposition for 3d boundary integral computations with surface tension. *J. Comp. Phys.*, 247:168–191, 2013.
[2] A. Barua, S. Li, H. Feng, X. Li, and J. Lowengrub. An efficient rescaling algorithm for simulating the evolution of multiple elastically stressed precipitates. *Commun. Comput. Phys.*, 14(04):940–959, 2013.
[3] E. Ben-Jacob and P. Garik. The formation of patterns in non-equilibrium growth. *Nature*, 343:523–530, 1990.
[4] G. D. Carvalho, H. Gadêlha, and J. A. Miranda. Stationary patterns in centrifugally driven interfacial elastic fingering. *Phys. Rev. E*, 90(6):063009, 2014.
[5] Y.-C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *J. Comp. Phys*, 124(2):449–464, 1996.
[6] W. Dai and M. Shelley. A numerical study of the effect of surface tension and noise on an expanding Hele-Shaw bubble. *Phys. Fluids A*, 5:2131–2146, 1993.
[7] L. dos Reis and J. A. Miranda. Controlling fingering instabilities in nonflat Hele-Shaw geometries. *Phys. Rev. E*, 84(6):066313, 2011.
[8] P. Fast and M. J. Shelley. Moore's law and the Saffman-Taylor instability. *J. Comput. Phys.*, 212(1):1–5, 2006.
[9] H. Feng, A. Barua, S. Li, and X. Li. A parallel adaptive treecode algorithm for evolution of elastically stressed solids. *Commun. Comput. Phys.*, 15:365–387, 2014.
[10] A. Greenbaum, L. Greengard, and G. B. McFadden. Laplace's equation and the Dirichlet-Nuemann map in multiply connected domains. *J. Comput. Phys.*, 105(2):267–278, 1993.
[11] L. Greengard and V. I. Rokhlin. A fast algorithm for particles summations. *J. Comp. Phys.*, 73:325–348, 1987.

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press

[12] A. He, J. S. Lowengrub, and A. Belmonte. Modeling an elastic fingering instability in a reactive Hele-Shaw flow. *SIAM J. Appl. Math*, 72(3):842–856, 2012.
[13] H. S. Hele-Shaw. Flow of water. *Nature*, 58(1509):520, 1898.
[14] T. Y. Hou, Z. Li, S. Osher, and H. Zhao. A hybrid method for moving interface problems with application to the Hele-Shaw flow. *J. Comp. Phys*, 134(2):236–252, 1997.
[15] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Removing the stiffness from interfacial flows with surface tension. *J. Comput. Phys.*, 114(2):312–338, 1994.
[16] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Boundary integral methods for multicomponent fluids and multiphase materials. *J. Comput. Phys.*, 169(2):302–362, 2001.
[17] L. Ju, J. Zhang, L. Zhu, and Q. Du. Fast explicit integration factor methods for semilinear parabolic equations. *J. Sci. Comput.*, 62(2):431–455, 2015.
[18] J. S. Langer. Instabilities and pattern formation in crystal growth. *Reviews of Modern Physics*, 52:1–28, 1980.
[19] J. S. Langer. Dendrites, viscous fingers, and the theory of pattern formation. *Science*, 243(4895):1150–1156, 1989.
[20] R. J. LeVeque, and Z. Li. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.*, 18(3):709–735, 1997.
[21] S. Li, J. Lowengrub, and P. Leo. A rescaling scheme with application to the long-time simulation of viscous fingering in a Hele-Shaw cell. *J. Comput. Phys.*, 225(1):554–567, 2007.
[22] S. Li, J. S. Lowengrub, J. Fontana, and P. Palffy-Muhoray. Control of viscous fingering patterns in a radial Hele–Shaw cell. *Phys. Rev. Lett.*, 102:174501, 2009.
[23] X. Liu and Q. Nie. Compact integration factor methods for complex domains and adaptive mesh refinement. *J. Comp. Phys.*, 229(16):5692–5706, 2010.
[24] L. Paterson. Radial fingering in a Hele-Shaw cell. *Journal of Fluid Mechanics*, 113:513–529, 1981.
[25] O. Praud and H. Swinney. Fractal dimension and unscreened angles measured for radial viscous fingering. *Phys. Rev. E*, 72(1):011406, 2005.
[26] Y. Saad and M. H. Schultz. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.
[27] P. G. Saffman and G. Taylor. The penetration of a fluid into a porous medium or a Hele-Shaw cell containing a more viscous fluid. *Proc. R. Soc. London Ser. A*, 245(1242):312–329, 1958.
[28] M. J. Shelley, F.-R. Tian, and K. Wlodarski. Hele-Shaw flow and pattern formation in a time-dependent gap. *Nonlinearity*, 10:1471–1495, 1997.
[29] F. R. Tian and Q. Nie. Singularities in Hele–Shaw flows. *SIAM J. Appl. Math.*, 58(1):34–54, 1998.
[30] W. Ying, S. Li, and X. Li. A kernel free boundary integral method for moving interface problems. *in preparation*.
[31] M. Zhao, A. Belmonte, S. Li, X. Li, and J. Lowengrub. Nonlinear simulations of elastic fingering in a Hele-Shaw cell. *J. Comput. Appl. Math.*, page in press., 2015.

https://doi.org/10.4208/cicp.OA-2016-0040 Published online by Cambridge University Press