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)

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