Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 212 (2006) 1–5
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Short note

Moore's law and the Saffman–Taylor instability

Petri Fast [a,*], Michael J. Shelley [b]

[a] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 7000 East Avenue, L-365, Livermore, CA 94551, United States
[b] Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, United States

Received 28 December 2004; received in revised form 18 May 2005; accepted 29 June 2005
Available online 19 August 2005

Ten years ago Hou, Lowengrub and Shelley [4] published a state-of-the-art boundary integral simulation of a classical viscous fingering problem, the Saffman–Taylor instability [12]. In terms of complexity and level of detail, those computations [4] are still among the most ramified and accurately computed interfacial instability patterns that have appeared in the literature. Since 1994, the computational power of a standard workstation has increased a hundredfold as predicted by Moore's law [7]. The purpose of this Note is to consider Moore's law and its consequences in computational science, and in particular, its impact on studying the Saffman–Taylor instability. We illustrate Moore's law and fast algorithms in action by presenting the worlds largest viscous fingering simulation to date.

Viscous fingering is one of the fundamental interfacial instabilities in fluid dynamics: Perturbations to an expanding circular air bubble displacing a viscous fluid in a thin gap flow device become unstable, resulting in intricate densely branched interfacial patterns. The viscous fingering problem is governed by the three-dimensional incompressible Navier–Stokes equations with moving free boundaries at the fluid/air interface. Without simplification, this problem is computationally intractable and will remain so for the foreseeable future. The simulations in this Note build upon decades of advances in mathematical modeling and numerical methods: (i) The fluid dynamics in a thin gap is reduced from the three-dimensional Navier–Stokes equations by asymptotic analysis to a Darcy's law [12]

$$\mathbf{u} = -\frac{b^2}{12\mu}\nabla p, \quad \nabla \cdot \mathbf{u} = 0 \tag{1}$$

for a two-dimensional velocity and pressure field (b is the gap thickness and μ is the viscosity); (ii) The three-dimensional boundary conditions at the free-surface are approximated at Γ to leading order [9] by a two-dimensional Young–Laplace boundary condition $[p] = -\gamma\kappa$ relating the pressure, the surface tension γ, and curvature κ, and by the kinematic condition $\partial\mathbf{x}/\partial t = \mathbf{u}$; (iii) The pressure satisfies a Laplace equation

* Corresponding author. Tel.: +925 424 2649; fax: +925 424 2477.
E-mail address: fast1@llnl.gov (P. Fast).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.06.022

which is written in a boundary integral formulation [2]; (iv) The integral equation is solved iteratively using GMRES [11] without forming the matrix explicitly; (v) The integral operator [2] is evaluated at each iteration using the fast multipole method (FMM) [3] in O(N) operations with N discretization points along Γ; (vi) The small-scale decomposition (SSD) [4] removes a severe restriction on the size of a time-step that arises from the Young–Laplace boundary condition.

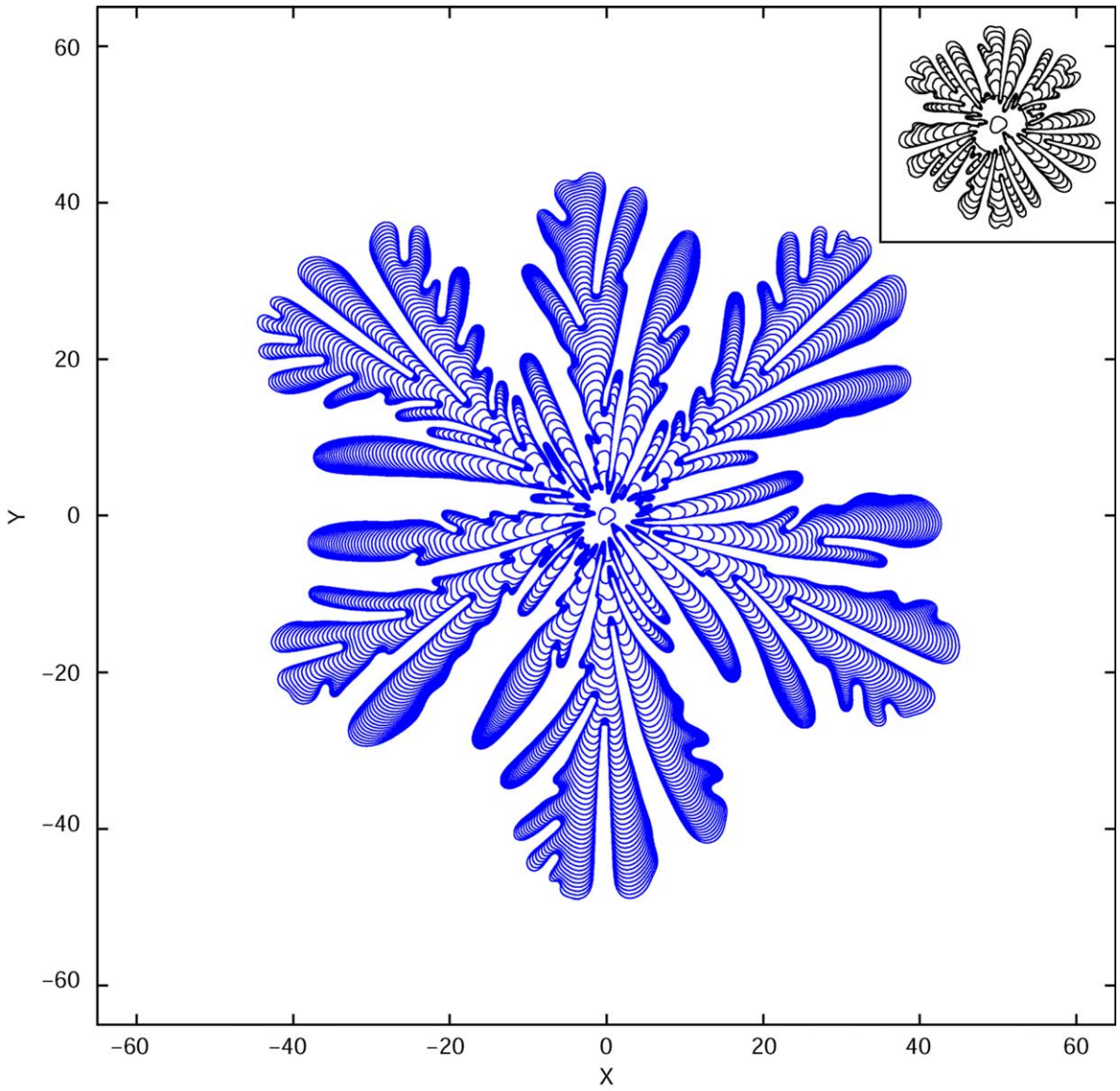

Fig. 1. The world's largest viscous fingering simulation at $T = 500$. The computation completes in 50 days on a modern desktop PC. The resolution has been increased as necessary through $N = 4096$, $N = 8192$, $N = 16{,}384$, $N = 32{,}768$ to resolve the expanding interface during the 500,000 time steps taken during the simulation. Inset: The reference viscous fingering computation published in 1994 by Hou et al. [4]. The computation required 50 days in 1994 to reach $T = 45$ in 45,000 timesteps using at most $N = 8192$ grid points along the interface (shown to scale).

It is the combination of mathematical modeling, items (i) and (ii), and advanced numerical algorithm development, items (iii) and (vi), that has made possible the world's largest viscous fingering simulation to date (Fig. 1). The use of the FFT makes the algorithm formally $O(N \log N)$ per time-step in complexity, though the real cost is currently dominated by the FMM and the solution of the integral equation. Boundary integral algorithms for viscous fingering simulation are relatively mature and their computational complexity has not improved since 1994 (see, however, the recent review [5]).

We have repeated the viscous fingering simulation from 1994 [4] on a workstation from 2004. The computations were performed on a single processor of a 2.2 GHz Pentium 4 desktop PC running Linux. The initial data is a slightly perturbed circular interface $(x(\alpha), y(\alpha)) = R_0 \cdot (1 + 0.1(\cos 3\alpha + \sin 2\alpha))(\cos\alpha, \sin\alpha)$, as in [4]. The modified capillary number [1] is $Ca = 12\mu U R_0^2/b^2\gamma = 1000$ (U is a the initial velocity of the interface). The computational time for reproducing [4, Fig. 1], also shown as the inset in Fig. 1, has decreased to 14 h which is roughly 1% of the 50 days required in 1994. This performance gain can be explained by Moore's law [7], which predicts that computational power doubles every 18 months. In other words, we can expect computing power to increase roughly 100-fold in 10 years.

We present new computations of much more ramified viscous fingering patterns using the methods from 1994 [4]. Fig. 1 shows a simulation with same initial data and parameters as in the inset of Fig. 1 (from [4]). Our simulations gain one order of magnitude increase in the length and size of the simulations in comparison to [4] also using 50 days of wall time. The complexity of the pattern evolution process has outstripped the exponential growth in computing power predicted by Moore's law. Nevertheless, long time simulations such as the one presented in Fig. 1 may reveal a new asymptotic scaling regime in Fig. 2, where the interface length could be related to the bubble area by a power-law relation [13].

In recent years, many computational methods have been developed for simulating interfacial instabilities: (see e.g., Fast and Shelley [1] and the references therein). Finite difference, finite volume or finite element based methods solve equations in the bulk, not just on the boundary as in [4], thereby requiring significantly more computational effort. However, the results obtained by such methods are close in appearance to those obtained in 1994 with a boundary integral method. This is due to many advances in algorithms, but it is also in large part due to the continued increase in the computational power of standard workstations, as predicted by Gordon Moore in 1965.

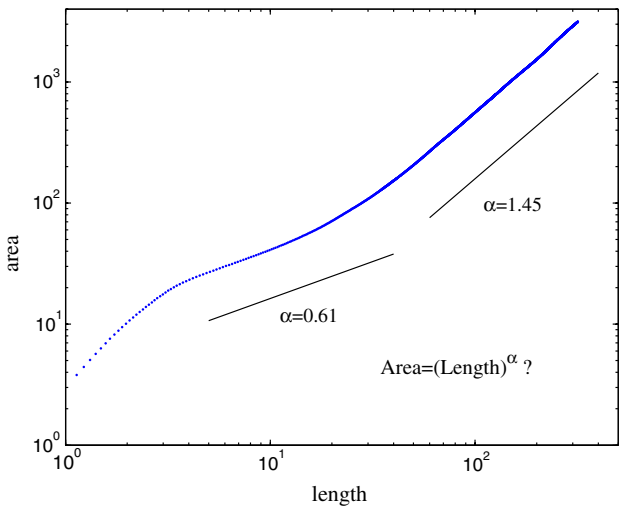

Fig. 2. The area of the growing bubble vs. the length of the interface. Based on this limited sample, the large time asymptotic behavior of the area seems to scale with the interfacial length as $A \sim L^{\alpha}$ with $\alpha = 1.45$ for a bubble driven by a *constant massflux*.

Table 1
The speed of supercomputers has increased by more than a 100-fold between June 1994 and November 2004

Date	Rank	Machine	R_{max}	N_{procs}	R_{max}/N_{procs}
November 2004	1	IBM/LLNL BG/L	70,720	32,768	2.158
June 2004	2	Intel Itanium 1.4 GHz	19,940	40,96	4.868
June 2004	5	Intel Xeon P4 3.06 GHz	9819	2500	3.928
June 2004	12	Intel Xeon P4 2.2 GHz	7634	2340	3.313
November 1994	1	Fujitsu Numerical Wind Tunnel	170	140	1.23
June 1994	1	Intel Paragon XP/S140	143.4	3680	0.039

R_{max} is the maximum number of floating point operations in GFlop/s, N_{procs} is the number of processors, and R_{max}/N_{procs} estimates the floating point performance of a single processor. Excerpt from the Top 500 supercomputer list[6] that measures performance using the LINPACK benchmark.

Table 1 from the Top 500 supercomputer list [6] demonstrates that Moore's law has been accurate in the past 10 years. Indeed, the single processor performance increased from 39 million floating point operations per second (MFlop/s) in June 1994 on an Intel Paragon to at least 3928 MFlop/s in June 2004 on a commodity Intel 3.06 GHz Pentium based cluster node. The fastest machine in November 1994 was a Fujitsu vector processor based architecture that breaks the trend with its performance of 1230 MFlop/s per processor, but it also represents an unlikely technology for present day workstations. Examining the Top 500 list closer shows that the broad trends are well explained by Moore's law, with a few notable outliers such as the Fujitsu Numerical Windtunnel in 1994, and the IBM/LLNL Blue Gene/Light which is currently the world's fastest supercomputer. Moore's law is expected to hold for at least another 10 years [8], which should have implications in algorithms design and computational science.

We have discussed two aspects of Moore's law and its impact on computational science in general, and on studying the Saffman–Taylor instability in particular. On one hand, a computer that is one hundred times faster than computers were ten years ago allows us to repeat a computation from 1994 in only one percent of the time the simulation took originally. The speed-up is entirely due to Moore's law and is independent of the algorithms used, and of their complexity.

On the other hand, a hundredfold increase in computing power allows us to simulate a physical problem in greater detail using the same amount of wall time as 10 years ago. The algorithmic complexity will determine to a large degree how much more is achieved by an increase in processing power. Fast algorithms that scale as $O(N)$ or $O(N \log N)$ in the number of unknowns allow significantly larger simulations, whereas simulations using "slow" algorithms with, e.g., $O(N^2)$ or $O(N^3)$ complexity will gain very little even from a 100-fold increase in computational resources.

We note that in 1984, ten years before Hou et al. [4], computers were 100 times slower than in 1994, and 10,000 times slower than they are today. If the *original solver* from 1994 [4] were to be run in 1984 on a commodity PC (using perhaps the Intel 286), one might reach in 50 days of wall time the very early stages of the pattern formation process (around $T = 1,\ldots,4$ in Fig. 1), where the viscous fingers are first beginning to split. However, three key pieces of the algorithm were not available in 1984: (1) The fast multipole method [3,10], published in 1985–1987, allows the efficient evaluation of the boundary-integral operator in an iterative scheme; (2) GMRES [11], published in 1986, converges rapidly for integral equation formulations; and (3) The small-scale decomposition [4], published in 1994, allows time steps that are 10^3 larger than in prior work. A viscous fingering simulation without FMM, GMRES, and SSD would have to use iterative methods with direct summation or direct factorization methods to solve the boundary-integral formulation, and would need at least 45 million computationally expensive timesteps and years of computing time just to reproduce the original results [4] even on today's computers. Fast computers help but it is the fast algorithms that make a big difference in our ability to simulate physical processes.

Acknowledgements

We thank Katarzyna Bernacki and Kyle Chand for discussions. M.J.S was supported in part by DOE Grant DE-FG02-88ER25053. A part of this work was performed under the auspices of the US Department of Energy by University of California Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

References

[1] P. Fast, M.J. Shelley, A moving overset grid method for interface dynamics applied to non-Newtonian Hele–Shaw flow, J. Comput. Phys. 195 (2004) 117–142.
[2] A. Greenbaum, L. Greengard, G.B. McFadden, Laplace's equation and the Dirichlet–Neumann map in multiply connected domains, J. Comput. Phys. 105 (1993) 267.
[3] L. Greengard, V.I. Rokhlin, A fast algorithm for particle summations, J. Comput. Phys. 73 (1987) 325–348.
[4] T. Hou, J. Lowengrub, M.J. Shelley, Removing the stiffness from interfacial flow with surface-tension, J. Comput. Phys. 114 (1994) 312.
[5] T. Hou, J. Lowengrub, M.J. Shelley, Boundary integral methods for multicomponent fluids and multiphase materials, J. Comput. Phys. 169 (2001) 302.
[6] H. Meuer, E. Strohmaeier, J. Dongarra, H.D. Simon, Top 500 Supercomputer Sites (November) (2004) (Available from: <http://www.spstab1.org>).
[7] G.E. Moore, Cramming more components onto integrated circuits, Electronics 38 (8) (1965).
[8] G.E. Moore, No exponential is forever... but we can delay "forever", in: International Solid State Circuits Conference (ISSCC), February 10, 2003 (Available from: <ftp://download.intel.com/research/silicon/Gordon_Moore_ISSCC_021003.pdf>).
[9] C.-W. Park, G.M. Homsy, Two-phase displacement in Hele–Shaw cells: theory, J. Fluid Mech. 139 (1984) 291–308.
[10] V.I. Rokhlin, Rapid solution of integral equations of classical potential theory, J. Comput. Phys. 60 (1985) 187–207.
[11] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[12] P.G. Saffman, G.I. Taylor, The penetration of a fluid into a porous medium or a Hele–Shaw cell containing a more viscous fluid, Proc. R. Soc. London A 245 (1958) 312.
[13] E. Sharon, M.G. Moore, W.D. McCormick, H.L. Swinney, Coarsening of fractal viscous fingering patterns, Phys. Rev. Lett. 91 (2003) 205504.